INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Mixtures of Deep Neural Experts for Automated Speech Scoring

Sara Papi[1], *Edmondo Trentin*[1], *Roberto Gretter*[2], *Marco Matassoni*[2], *Daniele Falavigna*[2]

[1]DIISM - Università di Siena, Italy
[2]Fondazione Bruno Kessler (FBK), Italy

sara.papi@student.unisi.it, trentin@dii.unisi.it, (gretter,matasso,falavi)@fbk.eu

Abstract

The paper copes with the task of automatic assessment of second language proficiency from the language learners' spoken responses to test prompts. The task has significant relevance to the field of computer assisted language learning. The approach presented in the paper relies on two separate modules: (1) an automatic speech recognition system that yields text transcripts of the spoken interactions involved, and (2) a multiple classifier system based on deep learners that ranks the transcripts into proficiency classes. Different deep neural network architectures (both feed-forward and recurrent) are specialized over diverse representations of the texts in terms of: a reference grammar, the outcome of probabilistic language models, several word embeddings, and two bag-of-word models. Combination of the individual classifiers is realized either via a probabilistic pseudo-joint model, or via a neural mixture of experts. Using the data of the third Spoken CALL Shared Task challenge, the highest values to date were obtained in terms of three popular evaluation metrics.

Index Terms: computer aided language learning, L2 proficiency, speech recognition, deep learning, mixture of experts.

1. Introduction

The problem of automatic proficiency scoring in second language learning (L2) [1] has been largely investigated in the past in the framework of computer assisted language learning (CALL) [2]. Approaches have been proposed for two input modalities: written and spoken. In both cases, specific competencies of the human learners are processed by suitable proficiency classifiers. The goal is to measure L2 proficiency relying on some ground truth provided by human experts. To this aim, the paper proposes and investigates the use of models for proficiency classification on a public data set released for the third Spoken CALL Shared Task [3] challenge. This took place in 2019 (hereafter "2019 challenge"), and addressed the automatic scoring of sentences uttered by Swiss German teenagers learning English in their second and third year.

Most of the approaches used by participants in the 2017 and 2018 editions of the challenge [4, 5] rely on hand-crafted features, extracted from both audio signals and automatic transcriptions of utterances, fed to a traditional classifier (e.g., based on logistic regression). These approaches, used also in most commercial systems (see [6] for a review), exhibited good performance on the task of the challenge. In both 2018 and 2019 challenges [5, 3] some approaches based on word embeddings [7] were investigated, as well.

In the work reported in [8], sentence similarities among ASR transcripts and the corresponding responses contained in a reference "grammar" (i.e. a non-exhaustive, prompt-specific list of appropriate responses, provided by the organizers of the challenge) are used as features for a scoring system based on a neural network (NN). The performances obtained using several in-domain and out-of-domain word embeddings (namely, Word2Vec and doc2vec [9]) are compared in [8], as well. An alternate approach based on word embeddings was presented in [10] for the 2019 challenge. The scoring system proposed in [10] relies on a NN fed with 918-dimensional word vectors. Each vector is formed by concatenating the outputs of the Bidirectional Encoder Representations from Transformers (BERT) [11] and a NN-based language model (LM) [12] trained on the data sets delivered for the challenge. Also, alternative word embeddings (i.e. word2vec, doc2vec, and ELMO [13]) were evaluated experimentally in [10].

In the present paper, building on the aforementioned experiences, we propose an approach that properly combines the outputs of several scoring systems, including the system winner of the 2019 challenge [14]. A speech recognizer is applied first, so as to obtain transcripts of the noisy responses uttered by the students. Feedforward and recurrent deep neural networks (DNN) are then used to model different representations of the automatic transcriptions. For each of these representations, a DNN is trained (hence, specialized) over a corresponding set of features, namely: (i) the scores yielded by a reference grammar, (ii) the likelihoods estimated by a number of probabilistic LMs, (iii) sequences of word embeddings of different type, and (iv) two variants of the bag-of-words representation. Besides applying individually each of these DNN-based "experts", multiple classifier systems are presented that combine all of them into a higher-level, more robust classifier capable of exploiting the specific capabilities of the individual experts. The combination is accomplished by either applying a pseudo-joint probability criterion over the individual DNN estimates [15], or by means of a mixture of DNNs [16]. In so doing the highest values to date are obtained in terms of the evaluation metric used officially for the challenge, as well as of other popular metrics.

The combination of multiple DNNs for speech scoring was proposed in [17], where two DNNs are employed to encode the lexical and the acoustic information contained in a spoken utterance, respectively. The lexical DNN encodes an automatically recognized input sentence relying on a pre-trained model (namely "GloVe", described in [18]), while the acoustic DNN encodes a corresponding sequence of word-level acoustic features. A linear regression model is used to combine the scores provided by the two DNNs. Different DNN architectures were evaluated empirically in [17]. The best results were obtained via a bidirectional LSTM (long-short term memory) [19] NN, together with an attention mechanism.

Mixtures of experts (in the form of shallow NNs) have long been investigated and successfully applied to a number of tasks in the machine learning community [16]. Applications of mixtures of experts include acoustic modeling [20], language modeling and machine translation [21], and other natural language processing tasks [22]. Since the outbreak of deep learning [21],

http://dx.doi.org/10.21437/Interspeech.2020-1055

also mixtures of DNNs have been receiving increasing attention. We follow in the footsteps of [23], insofar that a hard mixture of DNNs is built from independent experts that are individually specialized on expert-specific features, and efficiently trained in parallel. A probabilistic gating function can be applied that realizes a pseudo-joint likelihood of the independent DNNs (provided that a proper probabilistic interpretation of the DNNs outputs is given). More generally, a higher-level gating network can be trained a posteriori to assign individual credit to the pre-trained experts.

2. Task and systems description

The third Spoken CALL Shared Task [3] is composed by *(prompt, response)* pairs where prompts are written questions in German, while responses are speech recordings of spoken utterances given in English by native German-speaking Swiss teenagers. Each response was tagged by human raters with two Boolean labels denoting the correctness of the responses in terms of *language* and *meaning*, respectively. The task consists in classifying utterances as *accepted* or *rejected*: a response shall be accepted if both its *language* and its *meaning* are labeled as "correct". A reference grammar is made available by the organizers of the challenge, in the form of a list of correct written responses to each given prompt. For the 2019 challenge, the two training sets of the past editions [4, 5] are merged to form the training set (hereinafter called *TrainingSet*, 11919 utterances), the test set of the first challenge plays the role of development set (*DevSet*, 995 utterances), and the test set of the second edition is adopted as evaluation set (*EvalSet*, 1000 utterances). *TrainingSet* was used to train the acoustic models and the LMs. *DevSet* and *EvalSet* were used to tune hyper-parameters of the whole system and for applying model selection, respectively. Finally, the test data of the 2019 challenge (*TestSet*, 1000 utterances) were used for testing the performance of the resulting system.

The acoustic model, improved over [24], was trained using a popular Kaldi recipe [25] that relies on a time-delay NN optimized using the lattice-free maximum mutual information approach, i.e. with a sequence-level objective function. The acoustic model was trained on an extended dataset that, in addition to *TrainingSet*, embraced (i) the subset of PF-STAR [26] comprising the recordings of read English speech from German children (about 3.5h), and (ii) the ISLE corpus [27], consisting of 11484 utterances recorded by intermediate-level German learners of English (about 18h). As for the LM, the 3-gram stochastic LM provided by the organizers was adopted (details in [3]), resulting in a 7.5% word error rate on *EvalSet*.

2.1. FBK baseline system

The winner of the 2019 challenge used the following sets of features: *standard*, 4 features counting the number of words, of content words, the number and percentage of out-of-vocabulary (OOV) words; *reference*, 5 features computed using the reference grammar and the edit error (see [14]); *LMs*, features computed using some LMs. For each LM, 5 features related to log-probability, OOVs and number of back-offs were computed. A maximum of 12 LMs were defined, 1-grams to 4-grams, computed on 3 data sets: *Generic*, around 3 million words from English TED talks; *TrainRejRec*, ASR outputs bounded by labels *_start_* and *_end_*, corresponding to the incorrect utterances of *TrainingSet*; *TrainAccRec*, the same but corresponding to the correct utterances of *TrainingSet*.

Several feed-forward NNs (FFNNs) were used to perform classification; then, majority voting was applied to the most promising (on *DevSet*) classification outputs to contrast the high variability of the results observed on *DevSet*.

2.2. Stand-alone DNNs and textual features used

Sections 2.2.1 and 2.2.2 present the stand-alone DNNs used in the present paper, along with the corresponding features extracted from the prompts and the transcripts of the responses. Unless otherwise stated, henceforth a categorical cross-entropy loss function is used for training the networks.

2.2.1. LSTM on word embeddings (Word2Vec, BERT)

The LSTM model is considered first, and trained over sequences of 300-dimensional real-valued Word2Vec [7] word embeddings (hereafter we write LSTM-W2V to refer to this approach). The Word2Vec embeddings for any given *(prompt, response)* pair are then concatenated in order to form a single sequence of vectors, corresponding to the words in the prompt followed by the words in the response.

An ad hoc version of the loss function to train the LSTM-W2V is devised, as well, so as to account for the mismatch between the usual training criterion and the evaluation metric used in the 2019 challenge, that is the D_{full} [5]. The following criterion is proposed:

$$L(y,\hat{y}) = \begin{cases} \lambda MSE(y,\hat{y}), & \text{if } y \text{ is correct in language and} \\ & \hat{y} \text{ is incorrect in meaning} \\ MSE(y,\hat{y}), & \text{otherwise} \end{cases}$$

where MSE denotes the mean squared error loss, whose value is multiplied by λ (where $\lambda > 1$) whenever the meaning of the current response is incorrect while the system accepts it (thus, penalizing the *gross false accept* errors). We write LSTM-W2V-L to refer to the LSTM trained over the modified loss. Within the experimental framework reported in Section 3, a grid-search model selection procedure singled out an empirically suitable value of $\lambda = 3$.

Another variant of LSTM-W2V is achieved by interposing an end-of-sentence marker amidst the sequence of embeddings representing the prompt and the sequence representing the corresponding response. Roughly speaking, the marker is expected to provide the network with explicit information on when exactly to switch its internal state from prompt-processing to response-processing, possibly easing the LSTM learning and classification tasks. Henceforth, the approach is referred to as LSTM-W2V-M. In practice, the present variant turns out to yield the best performance when realized by means of an additional component to the embedding vectors (which become 301-dimensional), where the additional component is permanently set to zero except for the aforementioned marker, where it is set to one (other components in the marker are set to zero).

Finally, we replaced Word2Vec with BERT embeddings [11] (768-dimensional real-valued vectors) applying the same concatenation procedure of the word encodings in the prompt and in the response, respectively, separated by a marker (to this end, an additional binary component was added to the BERT vectors, as well, resulting in a 769-dimensional feature space). The present variant is abbreviated as LSTM-BERT.

Note that the use we propose of LSTMs with sequences of Word2Vec or BERT vectors is different from the use made in [8]. The latter relies on Word2Vec and doc2vec embeddings in order to compute individual sentence similarities between the responses and the reference grammar for the challenge. Such similarities are then used to train FFNN-based classifiers. In like manner, our approach differs entirely from that proposed in [10]. In fact, the latter averages over the sequence of embeddings in the prompt and in the corresponding response, obtaining a single static vector, which is concatenated with another vector drawn from a probabilistic LM. This flat, fixed-dimensional representation is eventually fed into a shallow FFNN-based classifier [10].

2.2.2. Deep FFNN on Bag-Of-Words and TF-IDF

A deep FFNN was applied to two completely different encodings of the transcripts, obtained by extracting either bag-of-words (BOW) or term frequency-inverse document frequency (TF-IDF) [28] vectors from the prompt and the response, respectively. The BOW model considered herein is a plain counter of the occurrences of individual words in the text. Both BOW and TF-IDF representations rely on a 1020 word vocabulary (embracing both German prompts and English responses in the dataset). The 2040-dimensional input vector fed to the FFNN was formed by concatenation of the encodings corresponding to the current prompt and response.

2.3. Combining multiple DNNs

A first probabilistic strategy for combining the different DNNs is readily achieved in terms of a pseudo-joint probability model as follows [15]. Henceforth, we write $\omega_1, \ldots, \omega_c$ to represent the c different classes involved in the classification task. Let ψ_1 and ψ_2 be the continuous-valued random quantities yielded by two distinct functions (or, regression models) of the outcome $\mathbf{x}$ of a given random phenomenon (e.g., the transcript of the response to a prompt). We refer to ψ_1 and ψ_2 as the "models" (it is straightforward to extend the following discussion to an arbitrary number of models). Different representations of $\mathbf{x}$ are given in terms of model-specific random vectors of features (or, sequences of feature vectors), say $\mathbf{x}^{(1)}$ for ψ_1 and $\mathbf{x}^{(2)}$ for ψ_2, respectively. Assuming that the models are independent of each other, for any state of nature ω_i ($i = 1, \ldots, c$) we can write:

$$\begin{aligned} P(\omega_i \mid \psi_1, \psi_2) &= \frac{p(\psi_1, \psi_2 \mid \omega_i) P(\omega_i)}{p(\psi_1, \psi_2)} \\ &= \frac{p(\psi_1 \mid \omega_i) p(\psi_2 \mid \omega_i) P(\omega_i)}{p(\psi_1) p(\psi_2)} \\ &= \frac{P(\omega_i \mid \psi_1) P(\omega_i \mid \psi_2)}{P(\omega_i)} \end{aligned} \quad (1)$$

where Bayes theorem was used in the third step of the calculations. Equation (1) allows the computation of $P(\omega_i \mid \psi_1, \psi_2)$ is terms of a pseudo-joint probability (the product of quantities at the numerator) normalized by the class-prior. The rationale behind the use of the expression "pseudo" lies in the fact that, in general, the models are actually not independent of each other under real-world circumstances. Nonetheless, equation (1) can still be applied, for practical intents and purposes, in a naive-Bayes fashion. A discriminant function $g_i(.)$ can thus be defined for each class ω_i by revolving around the usual maximum-a-posteriori probability given the models, i.e. $\max_i P(\omega_i \mid \psi_1, \psi_2)$. In the light of equation (1) such a discriminant function turns out to be defined as a normalized pseudo-joint probability in the form $g_i(\mathbf{x}) = P(\omega_i \mid \psi_1(\mathbf{x}^{(1)})) P(\omega_i \mid \psi_2(\mathbf{x}^{(2)})) / P(\omega_i)$, and the resulting decision rule assigns $\mathbf{x}$ to class ω_i if $g_i(\mathbf{x}) \geq g_j(\mathbf{x})$ for each $j \neq i$, as usual. Hereafter we assume that $\psi_1(.)$ and $\psi_2(.)$ are the functions computed by two distinct stand-alone, feature-specific DNNs to be combined, and we let $P(\omega_i \mid \psi_j(\mathbf{x}^{(j)})) \approx \psi_j(\mathbf{x}^{(j)})$ in compliance with the formal probabilistic interpretation of the DNN outputs as estimates of the class-posterior probabilities [29]. It is seen that equation (1) can be readily extended to any number k of models $\psi_1, \ldots, \psi_k$ with model-specific representations $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}$ of $\mathbf{x}$, allowing for a pseudo-joint combination of an arbitrary number of feature-specific DNNs.

The second technique for combining the stand-alone DNNs described in the previous sections relies on a hard mixture of experts [23]. While in [23] the mixture is hard insofar that the individual experts are trained independently over a crisp partitioning into expert-specific clusters of the (shared) feature space, herein the overall set of available features is partitioned into homogeneous, non-overlapping subsets of specialized features, and each DNN expert takes (independently) responsibility for the corresponding feature-specific representation of the *(prompt, response)* pairs of the whole dataset. In so doing, the stand-alone DNNs presented in the previous sections can be used as the (pre-trained) experts. The different DNNs in the mixture are then combined using a gating network, as follows. Assuming that k experts $E_1, E_2, \ldots, E_k$ are involved in the mixture, let $\mathbf{x}^{(j)}$ represent the specific feature vector (or, sequence of feature vectors) for the generic j-th neural expert E_j (i.e. $\mathbf{x}^{(j)}$ may represent the BERT-based embeddings of current *(prompt, response)* pair, or the corresponding BOW-based representation, etc.). Let $\mathbf{y}_j = \varphi_j(\mathbf{x}^{(j)})$ be the function computed by E_j over $\mathbf{x}^{(j)}$, and let $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$ be the vector embracing all the experts outputs. The gating network is trained to compute a mapping between its input vector $\mathbf{y}$ and a k-dimensional credit vector $(\alpha_1(\mathbf{y}), \ldots, \alpha_k(\mathbf{y}))$ such that the overall output $\mathbf{z}$ of the mixture is defined as $\mathbf{z} = \sum_{j=1}^{k} \alpha_j(\mathbf{y}) \varphi_j(\mathbf{x}^{(j)})$, where $\alpha_j(\mathbf{y}) \in (0,1)$ is the credit assigned by the gating network to the j-th expert, for $j = 1, \ldots, k$. In so doing, no arbitrary prior choices are imposed on the overall multiple-classifier combination criterion: in fact, the machine learns from the available examples how to assign credit to the individual experts of the mixture. The gating network is trained over the modified criterion function $L(\cdot, \cdot)$ presented in Section 2.2.1. It is seen that defining the mixture this way allows for combining both FFNN experts and recurrent LSTM experts within the same framework. The proposed mixture generalizes the notion of linear regression model over the regressors $\mathbf{y}_1, \ldots, \mathbf{y}_k$, insofar that the regression parameters $\alpha_1(\mathbf{y}), \ldots, \alpha_k(\mathbf{y})$ are themselves parametric nonlinear functions of the regressors themselves.

3. Experiments and results

The *TrainingSet*, *DevSet* and *EvalSet* were first merged and then, applying a uniform random sampling, partitioned into two subsets: the training set (80% of the data) and the validation set (20% of the data). These subsets were used, respectively, to train the DNNs and to accomplish model selection [30], respectively. The selected models were eventually evaluated in terms of different metrics on *TestSet*. Both the popular *Adam* and *RMSprop* optimizers were applied for training the DNNs. A grid-search model selection procedure was used [31], which ended up prescribing a learning rate equal to 0.001 for all models except for the MIX2-3FbkB-F1 variant (see below), where a learning rate of 0.0001 was selected. As for the number of training epochs, we relied on the early stopping strategy based on the validation loss. Early stopping resulted in an overall number of epochs in the range $[30 \div 50]$, depending on the DNN and on the features used. Table 1 reports the architectures (number of layers and number of neurons per layer, in the order) of the stand-alone DNNs as determined via model selection.

Table 1: *DNNs: no. of layers and no. of neurons per layer.*

Neural Network	Layers	Neurons per layer
LSTM-W2V	3	300-300-4
LSTM-W2V-L	3	300-175-4
LSTM-W2V-M	6	301-75-25-25-20-4
FFN-BOW-WC	7	2040-150-150-150-170-15-4
FFN-BOW-TFIDF	7	2040-210-130-150-170-15-4
LSTM-BERT	6	769-100-100-20-20-4

The metrics used are the D_{full} (the official metric of the 2019 challenge [5]), the accuracy (percentage of correct classifications), and the F-measure ($F1$). The test results in terms of D_{full} obtained from the selected stand-alone DNNs are shown in the second column of Table 2, where they are compared with the baseline performance yielded by the 2019 challenge-winning system (FbkB) [14]. It is seen that the proposed networks (albeit competitive w.r.t. the other participants in the Challenge [3]) did not improve the state-of-the-art. Cautions are required in assessing the present relative comparison, insofar that the FbkB is a multiple-classifier system, exploiting several different feature sets as well as a number of different NN architectures at once. The best performing stand-alone DNN turns out to be the LSTM-W2V-M, shown in boldface in Table 2. It is observed that both the accuracy and the D_{full} of the LSTM-W2V-M are close to the corresponding metrics yielded by the FbkB. All in all, the use of alternate textual feature turns out to be viable and capable of performances that are in line with the baseline, although the latter could not be improved. The latter consideration suggests that exploiting the reference grammar and the probabilistic LMs remains rewarding in facing the task. Nonetheless, all the proposed stand-alone approaches improved significantly in terms of D_{full} over the best system presented in [10] (that ranked 3rd in the 2019 challenge), in spite of the latter exploiting also an underlying LM (beside the word embeddings). Furthermore, three of the proposed DNNs (LSTM-W2V-M, FFN-BOW-WC, LSTM-BERT) improved the D_{full} over the best system presented in [8] (that ranked 2nd in the 2019 challenge), in spite of the latter exploiting the reference grammar besides Word2Vec. Noticeably, LSTM-W2V-M yielded a 8.79% relative D_{full} improvement over [8].

Table 2: *Stand-alone DNNs: results on* TestSet.

Model	D_{full}	Accuracy (%)	F1
LSTM-W2V	5.65	86.1	0.88
LSTM-W2V-L	4.92	86.1	0.88
LSTM-W2V-M	**6.19**	**87.3**	**0.88**
FFN-BOW-WC	5.97	86.0	0.88
FFN-BOW-TFIDF	5.59	85.3	0.88
LSTM-BERT	6.04	85.8	0.90
FbkB	6.34	87.5	0.92
2nd-best [8]	5.61	n/a	0.91
3rd-best [10]	5.43	n/a	0.91

The subsequent experimental round revolved around the multiple DNNs systems, investigating whether different features/models could effectively combine and complement each other. In the following we will write PJ-1FbkB-Dfull to represent the pseudo-joint (PJ) combination between the best network in the FbkB multiple-classifier system and the best DNN (among those evaluated in Table 2) in terms of D_{full}, and PJ3-3FbkB-F1 to denote the pseudo-joint combination of the 3-best FbkB networks and the 3-best proposed DNNs in terms of F-measure. Model selection was carried out over a number of combinations of subsets of the 1-best/3-best/6-best Challenge-winning network(s) reviewed in Section 2.1, and the 1st/2nd/3rd best network(s) proposed in Sections 2.2.1 and 2.2.2. All available evaluation metrics were considered in the model selection process. In short, it turned out that the best results yielded by the pseudo-joint combination technique relied on the 1-best DNN proposed in Sections 2.2.1 and 2.2.2 and the 1-best FbkB network according to the D_{full} (PJ-1FbkB-Dfull), as well as on the 3-best models proposed in Sections 2.2.1 and 2.2.2 and the 3-best FbkB networks according to the F-measure (PJ3-3FbkB-F1). As for the mixtures of experts, it was observed that the top-notch performances were achieved by the 3-best FbkB networks according to the F-measure, combined with the 1-best (MIX-3FbkB-F1) or the 2-best (MIX2-3FbkB-F) DNNs presented in Sections 2.2.1 and 2.2.2, respectively. The gating DNN selected for the MIX-3FbkB-F1 model was a 7-layers architecture with 4, 175, 75, 90, 50, 75, and 4 neurons per layer. The gating DNN selected for the MIX2-3FbkB-F1 model had 7 layers, as well, having 5, 512, 126, 256, 126, 126, and 4 neurons per layer.

Table 3: *Mixtures of DNNs: results on* TestSet.

Model	D_{full}	Accuracy (%)	F1
PJ-1FbkB-Dfull	6.82	87.9	0.89
PJ3-3FbkB-F1	7.00	87.1	0.89
MIX-3FbkB-F1	7.56	89.1	0.92
MIX2-3FbkB-F1	**8.08**	**88.7**	**0.92**
FbkB	6.34	87.5	0.92

Results are reported in Table 3. It is seen that the mixtures of experts yielded the highest D_{full}, Accuracy, and F-Measure on the data of the 2019 challenge to date. In particular, the MIX2-3FbkB-F1 resulted in a 27.44% relative D_{full} increase over FbkB, and in a 9.60% relative error rate reduction w.r.t. FbkB. The combination based on the pseudo-joint approach proved effective, as well. In fact, the PJ3-3FbkB-F1 yielded a significant improvement over FbkB in terms of D_{full}, while PJ-1FbkB-Dfull improved the baseline in terms of Accuracy. The outcome of the experiments pinpoints the fact that the different features used and the corresponding stand-alone DNNs do actually model diverse information on the linguistic phenomena at hand, fruitfully complementing the grammar-specific knowledge exploited by the FbkB models.

4. Conclusions

In their stand-alone versions, the DNNs proved competitive w.r.t. the state-of-the-art, improving over previous attempts to use NNs on word embeddings for the 2019 challenge. Both the techniques proposed for combining multiple DNNs achieved significant improvements over the winner of the 2019 challenge, yielding the highest values of the metrics for the task to date. Results confirm that different text representations and different DNNs may actually capture diverse facets of the linguistic phenomena at hand, complementing each other effectively. In future we aim to address more complex tasks in speech scoring (e.g., [32, 33]), and to extend the multiple classifiers so as to consider also DNNs experts trained on acoustic features.

5. References

[1] K. Zechner, D. Higgins, X. Xi, and D. Williamson, “Automatic scoring of non-native spontaneous speech in tests of spoken English,” *Speech Communication*, vol. 51, no. 10, pp. 883–895, 2009.

[2] N. Garrett, “Computer-assisted language learning trends and issues revisited: Integrating innovation,” *The Modern Language Journal*, no. 92, pp. 719–740, 2009.

[3] C. Baur, A. Caines, C. Chua, J. Gerlach, M. Qian, M. Rayner, M. Russell, H. Strik, and X. Wei, “Overview of the 2019 spoken call shared task,” in *Proc. of SLATE*, Graz, Austria, 2019, pp. 1–5.

[4] C. Baur, C. Chua, J. Gerlach, M. Rayner, M. Russell, H. Strik, and X. Wei, “Overview of the 2017 spoken call shared task,” in *Proc. of SLATE*, Stockolm, Sweden, 2017, pp. 71–78.

[5] C. Baur, A. Caines, C. Chua, J. Gerlach, M. Qian, M. Rayner, M. Russell, H. Strik, and X. Wei, “Overview of the 2018 spoken call shared task,” in *Proc. of Interspeech*, Hyderabad, India, 2018, pp. 2354–2358.

[6] K. Zechner and K. Evanini, *Automated Speaking Assessment: Using Language Technologies to Score Spontaneous Speech*. Princeton (NJ): Educational Testing Service, 2019.

[7] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[8] M. Qian, P. Jancovic, and M. Russel, “The university of birmingham 2019 spoken call shared task systems: Exploring the importance of word order in text processing,” in *Proc. of SlaTe*, Graz, Austria, 2019, pp. 11–15.

[9] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proc. of International Conference on Machine Learning*, Beijing, China, 2014.

[10] V. Sokhatskyi, O. Zvyeryeva, I. Karaulov, and D. Tkanov, “Embedding-based system for the text part of call v3 shared task,” in *Proc. SLATE*, Graz, Austria, 2019, pp. 16–19.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. of NAACL-HLT*, Minneapolis, USA, 2019.

[12] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[13] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *Proc. of NAACL-HLT*, vol. 1, 2018.

[14] R. Gretter, M. Matassoni, and D. Falavigna, “The FBK system for the 2019 spoken call shared task,” in *Proc. of SLATE*, Graz, Austria, 2019, pp. 6–10.

[15] E. Trentin, L. Lusnig, and F. Cavalli, “Comparison of combined probabilistic connectionist models in a forensic application,” in *Partially Supervised Learning - 1st IAPR TC3 Workshop, Revised Selected Papers.*, 2011, pp. 128–137.

[16] S. Masoudnia and R. Ebrahimpour, “Mixture of experts: A literature survey,” *Artificial Intelligence Review*, vol. 42, 08 2014.

[17] L. Chen, J. Tao, S. Ghaffarzadegan, and Y. Qian, “End-to-end neural network based automated speech scoring,” in *Proc. of ICASSP*, Calgary, Canada, 2018, pp. 6234–6238.

[18] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” *Proc. of EMNLP*, pp. 1532–1543, 2014.

[19] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] A. Jain, V. Singh, and S. Rath, “A multi-accent acoustic model using mixture of experts for speech recognition,” in *Proc. of INTERSPEECH 2019*. isca, 09 2019, pp. 779–783.

[21] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” in *Proc. of* 5^{th} *ICLR*, 2017.

[22] P. Le, M. Dymetman, and J. Renders, “LSTM-based mixture-of-experts for knowledge-aware dialogues,” in *Proc. of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, 2016, pp. 94–99.

[23] S. Gross, M. Ranzato, and A. Szlam, “Hard mixtures of experts for large scale weakly supervised vision,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 5085–5093.

[24] M. Matassoni, R. Gretter, D. Falavigna, and D. Giuliani, “Non-native children speech recognition through transfer learning,” in *Proc. of ICASSP*, Calgary, Canada, 2018, pp. 6229–6233.

[25] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, “Purely sequence-trained neural networks for ASR based on lattice-free MMI,” in *Proc. of Interspeech*, 2016, pp. 2751–2755.

[26] A. Batliner, M. Blomberg, S. D’Arcy, D. Elenius, D. Giuliani, M. Gerosa, C. Hacker, M. Russell, S. Steidl, and M. Wong, “The PF-STAR children’s speech corpus,” in *Proc. of Eurospeech*, 2005, pp. 2761–2764.

[27] W. Menzel, E. Atwell, P. Bonaventura, D. Herron, P. Howarth, R. Morton, and C. Souter, “The ISLE corpus of non-native spoken English,” in *Proc. of LREC*, 2000, pp. 957–964.

[28] Y. Goldberg, *Neural Network Methods for Natural Language Processing*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2017.

[29] E. Trentin and A. Freno, “Probabilistic interpretation of neural networks for the classification of vectors, sequences and graphs,” in *Innovations in Neural Information Paradigms and Applications*, M. Bianchini et al., Ed. Springer, 2009, pp. 155–182.

[30] U. Anders and O. Korn, “Model selection in neural networks,” *Neural Networks*, vol. 12, no. 2, pp. 309 – 323, 1999.

[31] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, 2012.

[32] R. Gretter, M. Matassoni, K. Allgaier, S. Tchistiakova, and D. Falavigna, “Automatic assessment of spoken language proficiency of non-native children,” in *Proc. of ICASSP*, 2019.

[33] R. Gretter, M. Matassoni, S. Bannò, and D. Falavigna, “Tlt-school: a corpus of non native children speech,” in *Proc. of LREC*, 2020.