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@di.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],
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 DNN 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-likelihood, OOVs and number of back-offs were computed. 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.

#### 2.2. 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 vector-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 $D_{f \text{set}}$ [5]. The following criterion is proposed:

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

where MSE denotes the mean squared error loss, whose value is multiplied by $\lambda$ (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 two distinct functions (or, regression models) of the outcome and \( \psi \) the first probabilistic strategy for combining the different DNNs corresponding to the current prompt and response. FFNN was formed by concatenation of the encodings computed by \( \omega \) and \( \psi \) 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 \( x^{(j)} \) represent the specific feature vector \( j \)-th neural expert \( E_j \) (i.e. \( x^{(j)} \) may represent the BERT-based embeddings of the prompt response) pair, or the corresponding BOW-based representation, etc.). Let \( y_j = \psi_j(x^{(j)}) \) be the function computed by \( E_j \) over \( x^{(j)} \), and let \( y = (y_1, \ldots, y_k) \) be the vector embracing all the experts outputs. The gating network is trained to compute a mapping between its input vector \( y \) and a \( k \)-dimensional credit vector \( (\alpha_1(y), \ldots, \alpha_k(y)) \) such that the overall output \( z \) of the mixture is defined as \( z = \sum_{j=1}^{k} \alpha_j(y) \psi_j(x^{(j)}) \), where \( \alpha_j(y) \in (0, 1) \) 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 \( y_1, \ldots, y_k \), insofar that the regression parameters \( \alpha_1(y), \ldots, \alpha_k(y) \) 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 MXX2-3PbR-F1 variant (see below), where a learning rate of 0.0001 was selected. As for the number of train-
ing 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 ÷ 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>
<thead>
<tr>
<th>Neural Network</th>
<th>Layers</th>
<th>Neurons per layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-W2V</td>
<td>3</td>
<td>300-300-4</td>
</tr>
<tr>
<td>LSTM-W2V-L</td>
<td>3</td>
<td>300-175-4</td>
</tr>
<tr>
<td>LSTM-W2V-M</td>
<td>6</td>
<td>301-75-25-25-20-4</td>
</tr>
<tr>
<td>FFN-BOW-WC</td>
<td>7</td>
<td>2040-150-150-150-170-15-4</td>
</tr>
<tr>
<td>FFN-BOW-TFIDF</td>
<td>7</td>
<td>2040-210-130-150-170-15-4</td>
</tr>
<tr>
<td>LSTM-BERT</td>
<td>6</td>
<td>769-100-100-20-20-4</td>
</tr>
</tbody>
</table>

The metrics used are the $D_{t_{full}}$ (the official metric of the 2019 challenge [5]), the accuracy (percentage of correct classifications), and the F-measure ($F_1$). The test results in terms of $D_{t_{full}}$ 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 as 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 $D_{t_{full}}$ 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 $D_{t_{full}}$ 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 $D_{t_{full}}$ 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 $D_{t_{full}}$ improvement over [8].

<table>
<thead>
<tr>
<th>Model</th>
<th>$D_{t_{full}}$</th>
<th>Accuracy (%)</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-W2V</td>
<td>5.65</td>
<td>86.1</td>
<td>0.88</td>
</tr>
<tr>
<td>LSTM-W2V-L</td>
<td>4.92</td>
<td>86.1</td>
<td>0.88</td>
</tr>
<tr>
<td>LSTM-W2V-M</td>
<td><strong>6.19</strong></td>
<td><strong>87.3</strong></td>
<td><strong>0.88</strong></td>
</tr>
<tr>
<td>FFN-BOW-WC</td>
<td>5.97</td>
<td>86.0</td>
<td>0.88</td>
</tr>
<tr>
<td>FFN-BOW-TFIDF</td>
<td>5.59</td>
<td>85.3</td>
<td>0.88</td>
</tr>
<tr>
<td>LSTM-BERT</td>
<td>6.04</td>
<td>85.8</td>
<td>0.90</td>
</tr>
<tr>
<td>FbkB</td>
<td>6.34</td>
<td>87.5</td>
<td>0.92</td>
</tr>
</tbody>
</table>

The subsequent experimental round revolved around the multiple DNN 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 $D_{t_{full}}$, 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 $D_{t_{full}}$ (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, 50, 75, and 4 neurons per layer. The gating DNN selected for the MIX2-3FbkB-F1 model had 7 layers, as well, having 5, 126, 256, 126, 126, and 4 neurons per layer.

<table>
<thead>
<tr>
<th>Model (TestSet)</th>
<th>$D_{t_{full}}$</th>
<th>Accuracy (%)</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PJ-1FbkB-Dfull</td>
<td>6.82</td>
<td>87.9</td>
<td>0.89</td>
</tr>
<tr>
<td>PJ3-3FbkB-F1</td>
<td>7.00</td>
<td>87.1</td>
<td>0.89</td>
</tr>
<tr>
<td>MIX-3FbkB-F1</td>
<td>7.56</td>
<td>89.1</td>
<td>0.92</td>
</tr>
<tr>
<td>MIX2-3FbkB-F1</td>
<td><strong>8.08</strong></td>
<td><strong>88.7</strong></td>
<td><strong>0.92</strong></td>
</tr>
<tr>
<td>FbkB</td>
<td>6.34</td>
<td>87.5</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Results are reported in Table 3. It is seen that the mixtures of experts yielded the highest $D_{t_{full}}$, 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 $D_{t_{full}}$ 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 $D_{t_{full}}$, 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


