TDNN: A Two-stage Deep Neural Network for Prompt-independent Automated Essay Scoring

Cancan Jin[1] **Ben He**[1,3] **Kai Hui**[2] **Le Sun**[3,4]

[1]School of Computer & Control Engineering,
University of Chinese Academy of Sciences, Beijing, China
[2] SAP SE, Berlin, Germany
[3] Institute of Software, Chinese Academy of Sciences, Beijing, China
[4] Beijing Advanced Innovation Center for Language Resources, Beijing, China
jincancan15@mails.ucas.ac.cn, benhe@ucas.ac.cn
kai.hui@sap.com, sunle@iscas.ac.cn

Abstract

Existing automated essay scoring (AES) models rely on rated essays for the target prompt as training data. Despite their successes in prompt-dependent AES, how to effectively predict essay ratings under a prompt-independent setting remains a challenge, where the rated essays for the target prompt are not available. To close this gap, a two-stage deep neural network (TDNN) is proposed. In particular, in the first stage, using the rated essays for non-target prompts as the training data, a shallow model is learned to select essays with an extreme quality for the target prompt, serving as pseudo training data; in the second stage, an end-to-end hybrid deep model is proposed to learn a prompt-dependent rating model consuming the pseudo training data from the first step. Evaluation of the proposed TDNN on the standard ASAP dataset demonstrates a promising improvement for the prompt-independent AES task.

1 Introduction

Automated essay scoring (AES) utilizes natural language processing and machine learning techniques to automatically rate essays written for a target prompt (Dikli, 2006). Currently, the AES systems have been widely used in large-scale English writing tests, e.g. Graduate Record Examination (GRE), to reduce the human efforts in the writing assessments (Attali and Burstein, 2006).

Existing AES approaches are prompt-dependent, where, given a target prompt, rated essays for this particular prompt are required for training (Dikli, 2006; Williamson, 2009; Foltz et al., 1999). While the established models are effective (Chen and He, 2013; Taghipour and Ng, 2016; Alikaniotis et al., 2016; Cummins et al., 2016; Dong et al., 2017), we argue that the models for prompt-independent AES are also desirable to allow for better feasibility and flexibility of AES systems especially when the rated essays for a target prompt are difficult to obtain or even unaccessible. For example, in a writing test within a small class, students are asked to write essays for a target prompt without any rated examples, where the prompt-dependent methods are unlikely to provide effective AES due to the lack of training data. Prompt-independent AES, however, has drawn little attention in the literature, where there only exists unrated essays written for the target prompt, as well as the rated essays for several non-target prompts.

We argue that it is not straightforward, if possible, to apply the established prompt-dependent AES methods for the mentioned prompt-independent scenario. On one hand, essays for different prompts may differ a lot in the uses of vocabulary, the structure, and the grammatic characteristics; on the other hand, however, established prompt-dependent AES models are designed to learn from these prompt-specific features, including the on/off-topic degree, the *tf*-*idf* weights of topical terms (Attali and Burstein, 2006; Dikli, 2006), and the n-gram features extracted from word semantic embeddings (Dong and Zhang, 2016; Alikaniotis et al., 2016). Consequently, the prompt-dependent models can hardly learn generalized rules from rated essays for non-target prompts, and are not suitable for the prompt-independent AES.

Being aware of this difficulty, to this end, a two-stage deep neural network, coined as *TDNN*, is proposed to tackle the prompt-independent AES problem. In particular, to mitigate the lack of the prompt-dependent labeled data, at the first stage,

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1088–1097
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

a shallow model is trained on a number of rated essays for several non-target prompts; given a target prompt and a set of essays to rate, the trained model is employed to generate pseudo training data by selecting essays with the extreme quality. At the second stage, a novel end-to-end hybrid deep neural network learns prompt-dependent features from these selected training data, by considering semantic, part-of-speech, and syntactic features.

The contributions in this paper are threefold: 1) a two-stage learning framework is proposed to bridge the gap between the target and non-target prompts, by only consuming rated essays for non-target prompts as training data; 2) a novel deep model is proposed to learn from pseudo labels by considering semantic, part-of-speech, and syntactic features; and most importantly, 3) to the best of our knowledge, the proposed TDNN is actually the first approach dedicated to addressing the prompt-independent AES. Evaluation on the standard ASAP dataset demonstrates the effectiveness of the proposed method.

The rest of this paper is organized as follows. In Section 2, we describe our novel TDNN model, including the two-stage framework and the proposed deep model. Following that, we describe the setup of our empirical study in Section 3, thereafter present the results and provide analyzes in Section 4. Section 5 recaps existing literature and put our work in context, before drawing final conclusions in Section 6.

2 Two-stage Deep Neural Network for AES

In this section, the proposed two-stage deep neural network (TDNN) for prompt-independent AES is described. To accurately rate an essay, on one hand, we need to consider its pertinence to the given prompt; on the other hand, the organization, the analyzes, as well as the uses of the vocabulary are all crucial for the assessment. Henceforth, both prompt-dependent and -independent factors should be considered, but the latter ones actually do not require prompt-dependent training data. Accordingly, in the proposed framework, a supervised ranking model is first trained to learn from prompt-independent data, hoping to roughly assess essays without considering the prompt; subsequently, given the test dataset, namely, a set of essays for a target prompt, a subset of essays are selected as positive and negative training data based on the prediction of the trained model from the first stage; ultimately, a novel deep model is proposed to learn both prompt-dependent and -independent factors on this selected subset. As indicated in Figure 1, the proposed framework includes two stages.

2.1 Overview

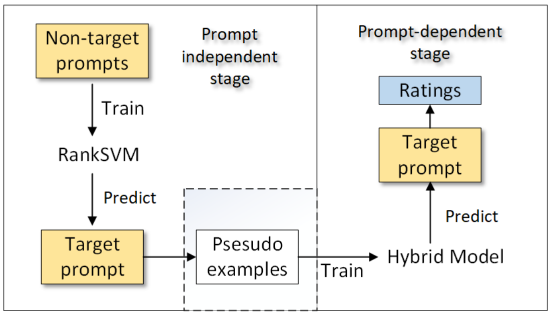


Figure 1: The architecture of the TDNN framework for prompt-independent AES.

Prompt-independent stage. Only the prompt-independent factors are considered to train a shallow model, aiming to recognize the essays with the extreme quality in the test dataset, where the rated essays for non-target prompts are used for training. Intuitively, one could recognize essays with the highest and the lowest scores correctly by solely examining their quality of writing, e.g., the number of typos, without even understanding them, and the prompt-independent features such as the number of grammatic and spelling errors should be sufficient to fulfill this screening procedure. Accordingly, a supervised model trained solely on prompt-independent features is employed to identify the essays with the highest and lowest scores in a given set of essays for the target prompt, which are used as the positive and negative training data in the follow-up prompt-dependent learning phase.

Prompt-dependent stage. Intuitively, most essays are with a quality in between the extremes, requiring a good understanding of their meaning to make an accurate assessment, e.g., whether the examples from the essay are convincing or whether the analyzes are insightful, making the consideration of prompt-dependent features crucial. To achieve that, a model is trained to learn from the comparison between essays with the highest and lowest scores for the target prompt according to the predictions from the first step. Akin to the settings in transductive transfer learning (Pan and

Yang, 2010), given essays for a particular prompt, quite a few confident essays at two extremes are selected and are used to train another model for a fine-grained content-based prompt-dependent assessment. To enable this, a powerful deep model is proposed to consider the content of the essays from different perspectives using semantic, part-of-speech (POS) and syntactic network. After being trained with the selected essays, the deep model is expected to memorize the properties of a good essay in response to the target prompt, thereafter accurately assessing all essays for it. In Section 2.2, building blocks for the selection of the training data and the proposed deep model are described in details.

2.2 Building Blocks

Select confident essays as training data. The identification of the extremes is relatively simple, where a RankSVM (Joachims, 2002) is trained on essays for different non-target prompts, avoiding the risks of over-fitting some particular prompts. A set of established prompt-independent features are employed, which are listed in Table 2. Given a prompt and a set of essays for evaluation, to begin with, the trained RankSVM is used to assign prediction scores to individual prompt-essay pairs, which are uniformly transformed into a 10-point scale. Thereafter, the essays with predicted scores in $[0, 4]$ and $[8, 10]$ are selected as negative and positive examples respectively, serving as the bad and good templates for training in the next stage. Intuitively, an essay with a score beyond eight out of a 10-point scale is considered good, while the one receiving less than or equal to four, is considered to be with a poor quality.

A hybrid deep model for fine-grained assessment. To enable a prompt-dependent assessment, a model is desired to comprehensively capture the ways in which a prompt is described or discussed in an essay. In this paper, semantic meaning, part-of-speech (POS), and the syntactic taggings of the token sequence from an essay are considered, grasping the quality of an essay for a target prompt. The model architecture is summarized in Figure 2. Intuitively, the model learns the semantic meaning of an essay by encoding it in terms of a sequence of word embeddings, denoted as $\overrightarrow{e}_{sem}$, hoping to understand what the essay is about; in addition, the part-of-speech information is encoded as a sequence of POS taggings, coined as $\overrightarrow{e}_{pos}$; ultimately, the structural connections between different components in an essay (e.g., terms or phrases) are further captured via syntactic network, leading to $\overrightarrow{e}_{synt}$, where the model learns the organization of the essay. Akin to (Li et al., 2015) and (Zhou and Xu, 2015), bi-LSTM is employed as a basic component to encode a sequence. Three features are separately captured using the stacked bi-LSTM layers as building blocks to encode different embeddings, whose outputs are subsequently concatenated and fed into several dense layers, generating the ultimate rating. In the following, the architecture of the model is described in details.

- *Semantic embedding.* Akin to the existing works (Alikaniotis et al., 2016; Taghipour and Ng, 2016), semantic word embeddings, namely, the pre-trained 50-dimension GloVe (Pennington et al., 2014), are employed. On top of the word embeddings, two bi-LSTM layers are stacked, namely, the essay layer is constructed on top of the sentence layer, ending up with the semantic representation of the whole essay, which is denoted as $\overrightarrow{e}_{sem}$ in Figure 2.

- *Part-Of-Speech (POS) embeddings* for individual terms are first generated by the Stanford Tagger (Toutanova et al., 2003), where 36 different POS tags present. Accordingly, individual words are embedded with 36-dimensional one-hot representation, and is transformed to a 50-dimensional vector through a lookup layer. After that, two bi-LSTM layers are stacked, leading to $\overrightarrow{e}_{pos}$. Take Figure 3 for example, given a sentence "Attention please, here is an example.", it is first converted into a POS sequence using the tagger, namely, VB, VBP, RB, VBZ, DT, NN; thereafter it is further mapped to vector space through one-hot embedding and a lookup layer.

- *Syntactic embedding* aims at encoding an essay in terms of the syntactic relationships among different syntactic components, by encoding an essay recursively. The Stanford Parser (Socher et al., 2013) is employed to label the syntactic structure of words and phrases in sentences, accounting for 59 different types in total. Similar to (Tai et al., 2015), we opt for three stacked bi-LSTM, aiming at encoding individual phrases, sentences, and ultimately the whole essay in sequence. In particular, according to the hierarchical structure from a parsing tree, the phrase-level bi-LSTM first encodes different phrases by consuming syntactic

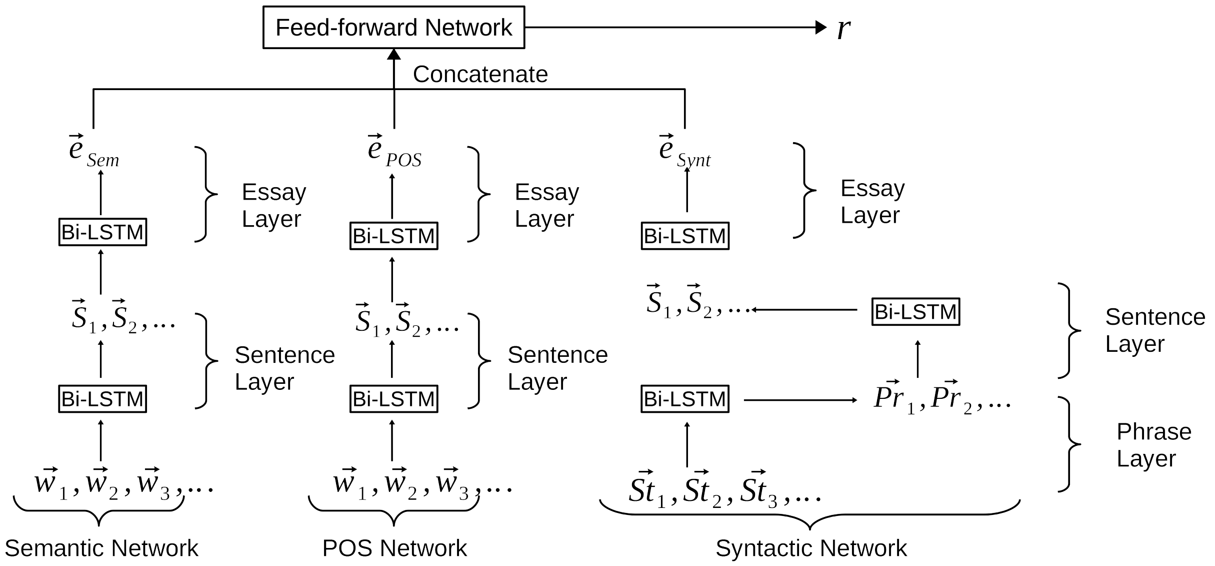


Figure 2: The model architecture of the proposed hybrid deep learning model.

embeddings ($\vec{St_i}$ in Figure 2) from a lookup table of individual syntactic units in the tree; thereafter, the encoded dense layers in individual sentences are further consumed by a sentence-level bi-LSTM, ending up with sentence-level syntactic representations, which are ultimately combined by the essay-level bi-LSTM, resulting in $\vec{e}_{synt}$. For example, the parsed tree for a sentence "Attention please, here is an example." is displayed in Figure 3. To start with, the sentence is parsed into ((NP VP)(NP VP NP)), and the dense embeddings are fetched from a lookup table for all tokens, namely, NP and VP; thereafter, the phrase-level bi-LSTM encodes (NP VP) and (NP VP NP) separately, which are further consumed by the sentence-level bi-LSTM. Afterward, essay-level bi-LSTM further combines the representations of different sentences into $\vec{e}_{synt}$.

```
(ROOT
    (S
        (S
            (NP (VB Attention))
            (VP (VBP please)))
        (, ,)
        (NP (RB here))
        (VP (VBZ is)
            (NP (DT an) (NN example)))
        (. .)))
```

Figure 3: An example of the context-free phrase structure grammar tree.

- *Combination.* A feed-forward network linearly transforms the concatenated representations of an essay from the mentioned three perspectives into a scalar, which is further normalized into $[0, 1]$ with a sigmoid function.

2.3 Objective and Training

Objective. Mean square error (MSE) is optimized, which is widely used as a loss function in regression tasks. Given N pairs of a target prompt p_i and an essay e_i, MSE measures the average value of square error between the normalized gold standard rating $r^*(p_i, e_i)$ and the predicted rating $r(p_i, e_i)$ assigned by the AES model, as summarized in Equation 1.

$$\frac{1}{N}\sum_{i=1}^{N}\big(r(p_i, e_i) - r^*(p_i, e_i)\big)^2 \qquad (1)$$

Optimization. Adam (Kingma and Ba, 2014) is employed to minimize the loss over the training data. The initial learning rate η is set to 0.01 and the gradient is clipped between $[-10, 10]$ during training. In addition, dropout (Srivastava et al., 2014) is introduced for regularization with a dropout rate of 0.5, and 64 samples are used in each batch with batch normalization (Ioffe and Szegedy, 2015). 30% of the training data are reserved for validation. In addition, early stopping (Yao et al., 2007) is employed according to the validation loss, namely, the training is terminated if no decrease of the loss is observed for ten consecutive epochs. Once training is finished,

Prompt	#Essays	Avg Length	Score Range
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

Table 1: Statistics for the ASAP dataset.

akin to (Dong et al., 2017), the model with the best quadratic weighted kappa on the validation set is selected.

3 Experimental Setup

Dataset. The Automated Student Assessment Prize (ASAP) dataset has been widely used for AES (Alikaniotis et al., 2016; Chen and He, 2013; Dong et al., 2017), and is also employed as the prime evaluation instrument herein. In total, ASAP consists of eight sets of essays, each of which associates to one prompt, and is originally written by students between Grade 7 and Grade 10. As summarized in Table 1, essays from different sets differ in their rating criteria, length, as well as the rating distribution[1].

Cross-validation. To fully employ the rated data, a prompt-wise eight-fold cross validation on the ASAP is used for evaluation. In each fold, essays corresponding to a prompt is reserved for testing, and the remaining essays are used as training data.

Evaluation metric. The model outputs are first uniformly re-scaled into $[0, 10]$, mirroring the range of ratings in practice. Thereafter, akin to (Yannakoudakis et al., 2011; Chen and He, 2013; Alikaniotis et al., 2016), we report our results primarily based on the quadratic weighted Kappa (QWK), examining the agreement between the predicted ratings and the ground truth. Pearson correlation coefficient (PCC) and Spearman rank-order correlation coefficient (SCC) are also reported. The correlations obtained from individual folds, as well as the average over all eight folds, are reported as the ultimate results.

Competing models. Since the prompt-independent AES is of interests in this work, the existing AES models are adapted for prompt-independent rating prediction, serving as baselines. This is due to the facts that the prompt-dependent and -independent models differ a lot in terms of problem settings and model designs, especially in their requirements for the training data, where the latter ones release the prompt-dependent requirements and thereby are accessible to more data.

No.	Feature
1	Mean & variance of word length in characters
2	Mean & variance of sentence length in words
3	Essay length in characters and words
4	Number of prepositions and commas
5	Number of unique words in an essay
6	Mean number of clauses per sentence
7	Mean length of clauses
8	Maximum number of clauses of a sentence in an essay
9	Number of spelling errors
10	Average depth of the parser tree of each sentence in an essay
11	Average depth of each leaf node in the parser tree of each sentence

Table 2: Handcrafted features used in learning the prompt-independent RankSVM.

- **RankSVM**, using handcrafted features for AES (Yannakoudakis et al., 2011; Chen et al., 2014), is trained on a set of pre-defined prompt-independent features as listed in Table 2, where the features are standardized beforehand to remove the mean and variance. The RankSVM is also used for the prompt-independent stage in our proposed TDNN model. In particular, the linear kernel RankSVM[2] is employed, where C is set to 5 according to our pilot experiments.

- **2L-LSTM.** Two-layer bi-LSTM with GloVe for AES (Alikaniotis et al., 2016) is employed as another baseline. Regularized word embeddings are dropped to avoid over-fitting the prompt-specific features.

- **CNN-LSTM.** This model (Taghipour and Ng, 2016) employs a convolutional (CNN) layer over one-hot representations of words, followed by an LSTM layer to encode word sequences in a given essay. A linear layer with sigmoid activation function is then employed to predict the essay rating.

- **CNN-LSTM-ATT.** This model (Dong et al., 2017) employs a CNN layer to encode word sequences into sentences, followed by an LSTM layer to generate the essay representation. An attention mechanism is added to model the influence of individual sentences on the final essay representation.

[1] Details of this dataset can be found at `https://www.kaggle.com/c/asap-aes`.

[2] `http://svmlight.joachims.org/`

For the proposed TDNN model, as introduced in Section 2.2, different variants of TDNN are examined by using one or multiple components out of the semantic, POS and the syntactic networks. The combinations being considered are listed in the following. In particular, the dimensions of POS tags and syntactic network are fixed to 50, whereas the sizes of the hidden units in LSTM, as well as the output units of the linear layers are tuned by grid search.

- **TDNN(Sem)** only includes the semantic building block, which is similar to the two-layer LSTM neural network from (Alikaniotis et al., 2016) but without regularizing the word embeddings;
- **TDNN(Sem+POS)** employs the semantic and the POS building blocks;
- **TDNN(Sem+Synt)** uses the semantic and the syntactic network building blocks;
- **TDNN(POS+Synt)** includes the POS and the syntactic network building blocks;
- **TDNN(ALL)** employs all three building blocks.

The use of POS or syntactic network alone is not presented for brevity given the facts that they perform no better than TDNN(POS+Synt) in our pilot experiments. Source code of the TDNN model is publicly available to enable further comparison[3].

4 Results and Analyzes

In this section, the evaluation results for different competing methods are compared and analyzed in terms of their agreements with the manual ratings using three correlation metrics, namely, QWK, PCC and SCC, where the best results for each prompt is highlighted in bold in Table 3.

It can be seen that, for seven out of all eight prompts, the proposed TDNN variants outperform the baselines by a margin in terms of QWK, and the TDNN variant with semantic and syntactic features, namely, TDNN(Sem+Synt), consistently performs the best among different competing methods. More precisely, as indicated in the bottom right corner in Table 3, on average, TDNN(Sem+Synt) outperforms the baselines by at least 25.52% under QWK, by 10.28% under PCC, and by 15.66% under SCC, demonstrating that the proposed model not only correlates better with the manual ratings in terms of QWK, but also linearly (PCC) and monotonically (SCC) correlates better with the manual ratings. As for the four baselines, note that, the relatively underperformed deep models suffer from larger variances of performance under different prompts, e.g., for prompts two and eight, 2L-LSTM's QWK is lower than 0.3. This actually confirms our choice of RankSVM for the first stage in TDNN, since a more complicated model (like 2L-LSTM) may end up with learning prompt-dependent signals, making it unsuitable for the prompt-independent rating prediction. As a comparison, RankSVM performs more stable among different prompts.

As for the different TDNN variants, it turns out that the joint uses of syntactic network with semantic or POS features can lead to better performances. This indicates that, when learning the prompt-dependent signals, apart from the widely-used semantic features, POS features and the sentence structure taggings (syntactic network) are also essential in learning the structure and the arrangement of an essay in response to a particular prompt, thereby being able to improve the results. It is also worth mentioning, however, when using all three features, the TDNN actually performs worse than when only using (any) two features. One possible explanation is that the uses of all three features result in a more complicated model, which over-fits the training data.

In addition, recall that the prompt-independent RankSVM model from the first stage enables the proposed TDNN in learning prompt-dependent information without manual ratings for the target prompt. Therefore, one would like to understand how good the trained RankSVM is in feeding training data for the model in the second stage. In particular, the precision, recall and F-score (P/R/F) of the essays selected by RanknSVM, namely, the negative ones rated between $[0, 4]$, and the positive ones rated between $[8, 10]$, are displayed in Figure 4. It can be seen that the P/R/F scores of both positive and negative classes differ a lot among different prompts. Moreover, it turns out that the P/R/F scores do not necessarily correlate with the performance of the TDNN model. Take TDNN(Sem+Synt), the best TDNN variant, as an example: as indicated in Table 4, the performance and the P/R/F scores of the pseudo examples are only weakly correlated in most cases.

To gain a better understanding in how the quality of pseudo examples affects the performance of TDNN, the sanctity of the selected essays are examined. In Figure 5, the relative precision of

[3] https://github.com/ucasir/TDNN4AES

Eval. Metric	QWK	PCC	SCC	QWK	PCC	SCC	QWK	PCC	SCC
Method	Prompt 1			Prompt 2			Prompt 3		
RankSVM	.7371	.6915	.6726	.4666	.4956	.4993	.4637	.5584	.5357
2L-LSTM	.4687	.6570	.4213	.2788	.6202	.6337	.5018	.6410	.6197
CNN-LSTM	.4320	.6933	.5108	.3230	.6513	.6395	.5454	**.6844**	.6541
CNN-LSTM-ATT	.6256	.7430	.6612	.4348	.7200	.6724	.4219	.5927	.6327
TDNN(Sem)	.7292	.7366	.7190	.6220	.7138	.7372	.6038	.6613	.6714
TDNN(Sem+POS)	.7305	.7413	.7209	.6551	.7276	.7469	.6112	.6706	.6809
TDNN(Sem+Synt)	**.7688**	**.7759**	**.7318**	**.6859**	**.7292**	**.7593**	**.6281**	.6759	**.7028**
TDNN(POS+Synt)	.7663	.7700	.7310	.6808	.7225	.7581	.6219	.6803	.6984
TDNN(All)	.7310	.7584	.7300	.6596	.7210	.7496	.6146	.6772	.6943
Method	Prompt 4			Prompt 5			Prompt 6		
RankSVM	.5112	.6250	.6325	.6690	.7103	.6651	.5285	.5443	.5239
2L-LSTM	.5754	.6527	.6354	.5128	.7375	.7360	.4951	.6528	.6669
CNN-LSTM	.7065	.7564	.7346	.6594	.6722	.6536	.5810	.6460	.6447
CNN-LSTM-ATT	.4665	.7224	.7383	.5348	.6531	.6505	.5149	.6291	.6637
TDNN(Sem)	.7398	.7412	.6934	.6874	.7585	.7323	.6278	.6524	.7205
TDNN(Sem+POS)	.7450	.7601	.7119	.6943	.7716	.7341	.6540	.6780	.7239
TDNN(Sem+Synt)	**.7578**	**.7616**	**.7492**	**.7366**	**.7993**	**.7960**	**.6752**	**.6903**	**.7434**
TDNN(POS+Synt)	.7561	.7591	.7440	.7332	.7983	.7866	.6593	.6759	.7354
TDNN(All)	.7527	.7609	.7251	.7302	.7974	.7794	.6557	.6874	.7350
Method	Prompt 7			Prompt 8			Average		
RankSVM	.5858	.6436	.6429	.4075	.5889	.6087	.5462	.6072	.5976
2L-LSTM	**.6690**	**.7637**	**.7607**	.2486	.5137	.4979	.4687	.6548	.6214
CNN-LSTM	.6609	.6849	.6865	.3812	.4666	.3872	.5362	.6569	.6139
CNN-LSTM-ATT	.6002	.6314	.6223	.4468	.5358	.4536	.5057	.6535	.6368
TDNN(Sem)	.5482	.6957	.6902	.5003	.6083	.6545	.5875	.6779	.6795
TDNN(Sem+POS)	.6239	.7111	.7243	.5519	.6219	.6614	.6582	.7103	.7130
TDNN(Sem+Synt)	.6587	.7201	.7380	**.5741**	**.6324**	**.6713**	**.6856**	**.7244**	**.7365**
TDNN(POS+Synt)	.6464	.7172	.7349	.5631	.6281	.6698	.6784	.7189	.7322
TDNN(All)	.6396	.7114	.7300	.5622	.6267	.6631	.6682	.7176	.7258

Table 3: Correlations between AES and manual ratings for different competing methods are reported for individual prompts. The average results among different prompts are summarized in the bottom right. The best results are highlighted in bold for individual prompts.

Neg/Pos	Metric	QWK	PCC	SCC
$[0, 4]$	Precision	+0.5151	+0.4286	+0.4471
	Recall	- 0.2362	- 0.1363	- 0.3491
	F-score	+0.4135	+0.4062	+0.1703
$[8, 10]$	Precision	+0.3526	+0.3224	+0.3885
	Recall	+0.0063	- 0.0415	- 0.2112
	F-score	+0.8339	+0.6905	+0.4221

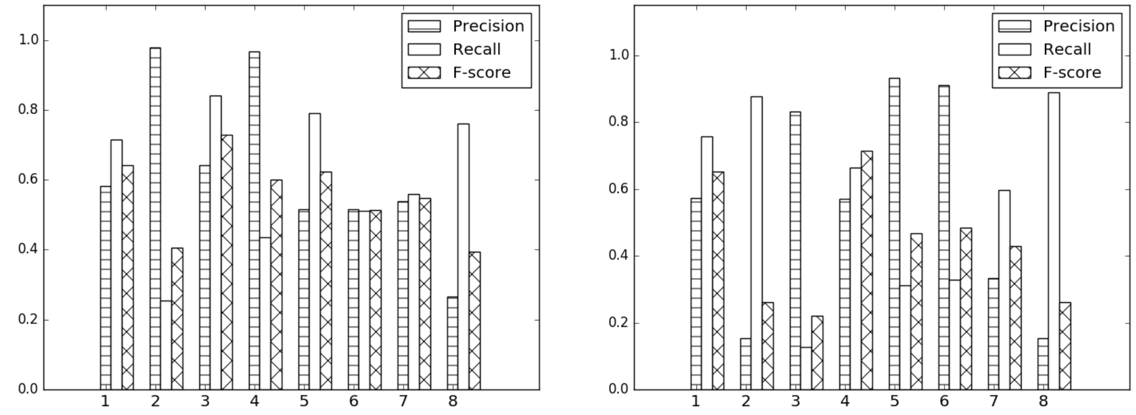
Table 4: Linear correlations between the performance of TDNN(Sem+Synt) and the precision, recall, and F-score of the selected pseudo examples.

Prpt	1	2	3	4	5	6	7	8
Neg	191	245	847	428	501	209	454	60
Pos	623	470	65	295	277	426	267	418

Table 5: The numbers of the selected positive and negative essays for each prompt.

the selected positive and negative training data by RankSVM are displayed for all eight prompts in terms of their concordance with the manual ratings, by computing the number of positive (negative) essays that are better (worse) than all negative (positive) essays. It can be seen that, such relative precision is at least 80% and mostly beyond 90% on different prompts, indicating that the overlap of the selected positive and negative essays are fairly small, guaranteeing that the deep model in the second stage at least learns from correct labels, which are crucial for the success of our TDNN model.

Beyond that, we further investigate the class balance of the selected training data from the first



(a) Negative

(b) Positive

Figure 4: The precision, recall and F-score of the pseudo negative or positive examples, which are rated within $[0, 4]$ or $[8, 10]$ by RankSVM.

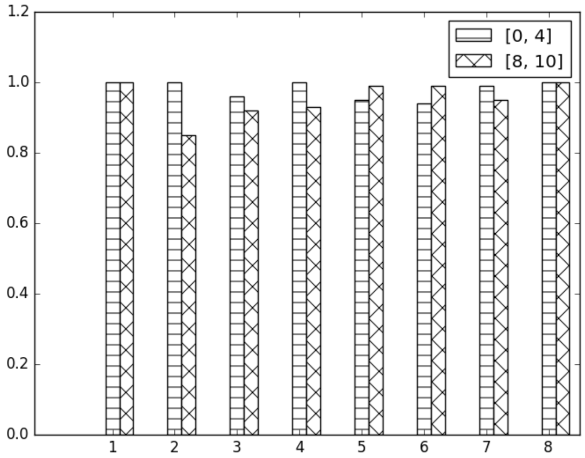


Figure 5: The sanctity of the selected positive and negative essays by RankSVM. The x-axis indicates different prompts and the y-axis is the relative precision.

stage, which could also influence the ultimate results. The number of selected positive and negative essays are reported in Table 5, where for prompts three and eight the training data suffers from serious imbalanced problem, which may explain their lower performance (namely, the two lowest QWKs among different prompts). On one hand, this is actually determined by real distribution of ratings for a particular prompt, e.g., how many essays are with an extreme quality for a given prompt in the target data. On the other hand, a fine-grained tuning of the RankSVM (e.g., tuning C_+ and C_- for positive and negative examples separately) may partially resolve the problem, which is left for the future work.

5 Related Work

Classical regression and classification algorithms are widely used for learning the rating model based on a variety of text features including lexical, syntactic, discourse and semantic features (Larkey, 1998; Rudner, 2002; Attali and Burstein, 2006; Mcnamara et al., 2015; Phandi et al., 2015). There are also approaches that see AES as a preference ranking problem by applying learning to ranking algorithms to learn the rating model. Results show improvement of learning to rank approaches over classical regression and classification algorithms (Chen et al., 2014; Yannakoudakis et al., 2011). In addition, Chen & He propose to incorporate the evaluation metric into the loss function of listwise learning to rank for AES (Chen and He, 2013).

Recently, there have been efforts in developing AES approaches based on deep neural networks (DNN), for which feature engineering is not required. Taghipour & Ng explore a variety of neural network model architectures based on recurrent neural networks which can effectively encode the information required for essay scoring and learn the complex connections in the data through the non-linear neural layers (Taghipour and Ng, 2016). Alikaniotis et al. introduce a neural network model to learn the extent to which specific words contribute to the text's score, which is embedded in the word representations. Then a two-layer bi-directional Long-Short Term Memory networks (bi-LSTM) is used to learn the meaning of texts, and finally the essay score is predicted through a mutli-layer feed-forward network (Alikaniotis et al., 2016). Dong & Zhang employ a hierarchical convolutional neural network (CNN) model, with a lower layer representing sentence structure and an upper layer representing essay structure based on sentence representations, to learn features automatically (Dong and Zhang, 2016). This model is later improved by employing attention layers. Specifically, the model learns text representation with LSTMs which can model the coherence and co-reference among sequences of words and sentences, and uses attention pooling to capture more relevant words and sentences that contribute to the final quality of essays (Dong et al., 2017). Song et al. propose a deep model for identifying discourse modes in an essay (Song et al., 2017).

While the literature has shown satisfactory performance of prompt-dependent AES, how to achieve effective essay scoring in a prompt-independent setting remains to be explored. Chen & He studied the usefulness of prompt-independent text features and achieved a human-machine rating agreement slightly lower than the use of all text features (Chen and He, 2013) for prompt-dependent essay scoring prediction. A constrained multi-task pairwise preference learning approach was proposed in (Cummins et al., 2016) to combine essays from multiple prompts for training. However, as shown by (Dong and Zhang, 2016; Zesch et al., 2015; Phandi et al., 2015), straightforward applications of existing AES methods for prompt-independent AES lead to a poor performance.

6 Conclusions & Future Work

This study aims at addressing the prompt-independent automated essay scoring (AES), where no rated essay for the target prompt is available. As demonstrated in the experiments, two kinds of established prompt-dependent AES models, namely, RankSVM for AES (Yannakoudakis et al., 2011; Chen et al., 2014) and the deep models for AES (Alikaniotis et al., 2016; Taghipour and Ng, 2016; Dong et al., 2017), fail to provide satisfactory performances, justifying our arguments in Section 1 that the application of established prompt-dependent AES models on prompt-independent AES is not straightforward. Therefore, a two-stage TDNN learning framework was proposed to utilize the prompt-independent features to generate pseudo training data for the target prompt, on which a hybrid deep neural network model is proposed to learn a rating model consuming semantic, part-of-speech, and syntactic signals. Through the experiments on the ASAP dataset, the proposed TDNN model outperforms the baselines, and leads to promising improvement in the human-machine agreement.

Given that our approach in this paper is similar to the methods for transductive transfer learning (Pan and Yang, 2010), we argue that the proposed TDNN could be further improved by migrating the non-target training data to the target prompt (Busto and Gall, 2017). Further study of the uses of transfer learning algorithms on prompt-independent AES needs to be undertaken.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (61472391), and the Project of Beijing Advanced Innovation Center for Language Resources (451122512).

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *ACL (1)*. The Association for Computer Linguistics.

Y. Attali and J. Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment* 4(3).

Pau Panareda Busto and Juergen Gall. 2017. Open set domain adaptation. In *ICCV*. IEEE Computer Society, pages 754–763.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *EMNLP*. ACL, pages 1741–1752.

Hongbo Chen, Jungang Xu, and Ben He. 2014. Automated essay scoring by capturing relative writing quality. *Comput. J.* 57(9):1318–1330.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained multi-task learning for automated essay scoring. In *ACL (1)*. The Association for Computer Linguistics.

S. Dikli. 2006. An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment* 5(1).

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring - an empirical study. In *EMNLP*. The Association for Computational Linguistics, pages 1072–1077.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *CoNLL*. Association for Computational Linguistics, pages 153–162.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*. volume 1999, pages 939–944.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*. JMLR.org, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*. ACM, pages 133–142.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Leah S. Larkey. 1998. Automatic essay grading using text categorization techniques. In *SIGIR*. ACM, pages 90–95.

Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard H. Hovy. 2015. When are tree structures necessary for deep learning of representations? In *EMNLP*. The Association for Computational Linguistics, pages 2304–2314.

Danielle S. Mcnamara, Scott A. Crossley, Rod D. Roscoe, Laura K. Allen, and Jianmin Dai. 2015. A hierarchical classification approach to automated essay scoring. *Assessing Writing* 23:35–59.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. ACL, pages 1532–1543.

Peter Phandi, Kian Ming Adam Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *EMNLP*. The Association for Computational Linguistics, pages 431–439.

L. M Rudner. 2002. Automated essay scoring using bayes' theorem. *National Council on Measurement in Education New Orleans La* 1(2):3–21.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *ACL (1)*. The Association for Computer Linguistics, pages 455–465.

Wei Song, Dong Wang, Ruiji Fu, Lizhen Liu, Ting Liu, and Guoping Hu. 2017. Discourse mode identification in essays. In *ACL (1)*. Association for Computational Linguistics, pages 112–122.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *EMNLP*. The Association for Computational Linguistics, pages 1882–1891.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL (1)*. The Association for Computer Linguistics, pages 1556–1566.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*. The Association for Computational Linguistics.

D.M. Williamson. 2009. A framework for implementing automated scoring. In *Annual Meeting of the American Educational Research Association and the National Council on Measurement in Education, San Diego, CA*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *ACL*. The Association for Computer Linguistics, pages 180–189.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation* 26(2):289–315.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-independent features for automated essay grading. In *BEA@NAACL-HLT*. The Association for Computer Linguistics, pages 224–232.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL (1)*. The Association for Computer Linguistics, pages 1127–1137.