Multi-task Learning for Automated Essay Scoring with Sentiment Analysis

Panitan Muangkammuen[1] **and Fumiyo Fukumoto**[2]
Graduate School of Engineering[1]
Interdisciplinary Graduate School[2]
University of Yamanashi
4-3-11, Takeda, Kofu, 400-8511 Japan
{g19tk021,fukumoto}@yamanashi.ac.jp

Abstract

Automated Essay Scoring (AES) is a process that aims to alleviate the workload of graders and improve the feedback cycle in educational systems. Multi-task learning models, one of the deep learning techniques that have recently been applied to many NLP tasks, demonstrate the vast potential for AES. In this work, we present an approach for combining two tasks, sentiment analysis, and AES by utilizing multi-task learning. The model is based on a hierarchical neural network that learns to predict a holistic score at the document-level along with sentiment classes at the word-level and sentence-level. The sentiment features extracted from opinion expressions can enhance a vanilla holistic essay scoring, which mainly focuses on lexicon and text semantics. Our approach demonstrates that sentiment features are beneficial for some essay prompts, and the performance is competitive to other deep learning models on the Automated Student Assessment Prize (ASAP) benchmark. The Quadratic Weighted Kappa (QWK) is used to measure the agreement between the human grader's score and the model's prediction. Our model produces a QWK of 0.763.

1 Introduction

Automatic essay scoring (AES) is the task of grading student essays, using natural language processing to assess quality. The system is designed to reduce time and cost from the human graders' workload. Recently, neural network models based on deep learning techniques have been proposed for AES. These approaches involve the use of both recurrent neural networks, e.g., a basic recurrent unit (RNN) (Elman, 1990), gated recurrent unit (GRU) (Cho et al., 2014), or long short-term memory unit (LSTM) (Hochreiter and Schmidhuber, 1997), and convolutional neural networks (Lecun et al., 1998; Kim, 2014). More specifically, Taghipour and Ng (2016) proposed a convolutional recurrent neural network over the word sequence to construct a document representation. Dong et al. employed hierarchical CNN and LSTM structure (Dong and Zhang, 2016; Dong et al., 2017) to construct sentences and document representation separately. Similarly, several text properties are utilized for scoring an essay, such as grammatical roles (i.e., subject, object) (Burstein et al., 2010), discourse (Song et al., 2017), or coherence (Tay et al., 2017; Mesgar and Strube, 2018).

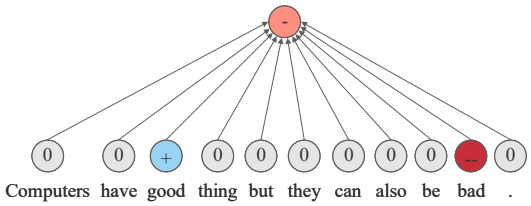

Figure 1: Example of sentiments in a sentence (This sentence is taken from an actual essay, and it is not grammatical).

The divergent and polarizing writers' opinions in their essays create overall essay structure and quality, especially in persuasive and controversial articles (Pang and Lee, 2008). Sentiment analysis has typically been designed for use with specific domains, such as movie reviews (Thongtan and Phienthrakul, 2019), product reviews (Shrestha and Nasoz, 2019), social media (Song et al., 2019), and news (Godbole et al., 2007). Beigman Klebanov et al. (2012) are the first who attempted to incorporate sentiments with essay data. It involves the use of subjective lexicons to recognize the polarity of a sentence. Another notable work (Klebanov et al., 2013) found a way to measure the compositionality of multi-word expression's sentiment profile (relative degree of polarities) in essays. Farra et al. (2015) built essay scoring systems that incorporate persuasiveness based on the analysis of opinions

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 116–123

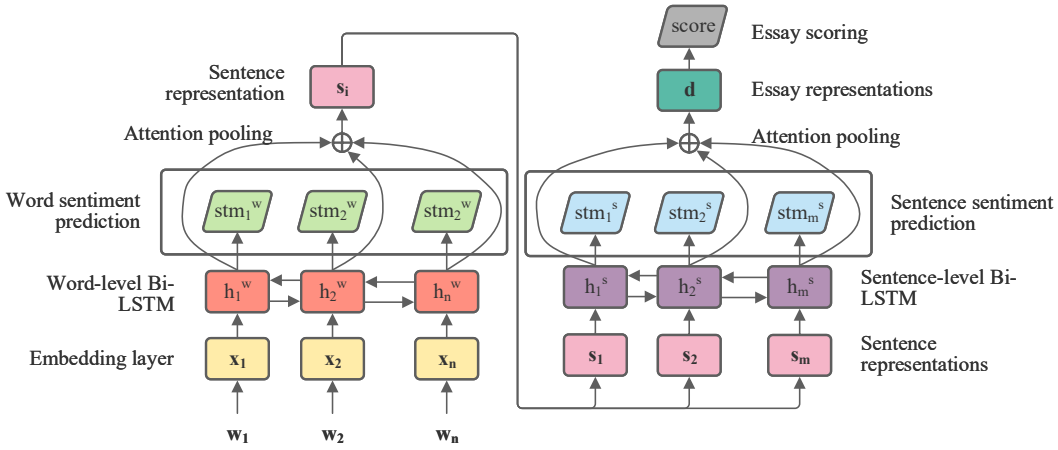

Figure 2: Multi-task Learning Framework.

expressed in the essay. Janda et al. (2019) showed sentiment-based features are in the top-ranked features giving the best performance on essay evaluation. However, these works are based on feature engineering, which must be carefully handcrafted and selected to fit the appropriate model.

Recently, Multi-Task Learning (MTL) approach has been shown promising results in many NLP tasks. The primary purpose is to leverage useful information in multiple related tasks to improve all the tasks' generalization performance (Zhang and Yang, 2017). The objectives are applied together, such as predicting the probability of the sequence and the probability that the sequence contains names (Cheng et al., 2015), the frequency of the next word with part-of-speech (POS) (Plank et al., 2016), surrounding words with other several tasks (Rei, 2017), error detection with additional linguistic information (Rei and Yannakoudakis, 2017). More advanced, Augenstein and Søgaard (2017) explored MTL for classifying keyphrase boundaries incorporating semantic super-sense tagging and identifying multi-word expression. Sanh et al. (2018) introduced a hierarchical model supervising a set of low-level tasks at the bottom layers and more complex tasks at the model's top layers. There are merely a few existing works that utilize multi-task learning in AES (Cummins et al., 2016; Cummins and Rei, 2018).

In this paper, we propose a method to incorporate sentiment analysis and AES. The proposed method utilizes the sentiment aspect for improving an essay scoring system. The sentiment information by both word-based and sentence-based, shown in Figure 1, is applied to enhance textual representations for sentiment perception of the model. To the best of our knowledge, this is the first approach on multi-task learning incorporating sentiment analysis and AES. Our proposed system is based on a hierarchical structure model to learn the features and relations between an essay score and its sentiments. The model is trained to predict a holistic score at the top-level (document-level) along with sentence and word sentiments at the lower levels, i.e., sentence-level and word-level, respectively.

2 Multi-Task Learning

We employ a hierarchical multi-task learning model similar to the model of Farag and Yannakoudakis (2019), shown in Figure 2. The model considers an essay d composed of a sequence of sentences $d = \{s_1, s_2, ..., s_m\}$, and each sentence s_i consists of a sequence of words $s_i = \{w_1, w_2, ..., w_n\}$. We describe each layer in our framework in detail.

2.1 Sentence Representation

Firstly, we consider the left-hand side of the framework in Figure 2. This part aims to learn the context representation of a sentence by taking a word sequence as an input. A word embedding lookup table maps the words in the vocabulary into low dimensional vectors,

$$\mathbf{x}_i = \mathbf{E}\mathbf{w}_i, (i = 1, 2, ..., n), \quad (1)$$

where $\mathbf{E} \in \mathbb{R}^{|V| \times D}$ be the embedding matrix, $|V|$ is the vocabulary size, and D is the word embedding dimension. $\mathbf{x}_i \in \mathbb{R}^D$ is the embedding vector of w_i, and $\mathbf{w}_i$ is a one-hot representation of w_i.

After the word embedding sequence is obtained from the embedding layer, a bidirectional LSTM is applied to the sequence to capture the context representations. In addition to a bi-direction, we concatenate the output vectors from both directions:

$$\begin{aligned} \overrightarrow{h_i^{\mathbf{w}}} &= LSTM(\mathbf{x}_i, \overrightarrow{h_{i-1}^{\mathbf{w}}}), \\ \overleftarrow{h_i^{\mathbf{w}}} &= LSTM(\mathbf{x}_i, \overleftarrow{h_{i+1}^{\mathbf{w}}}), \\ h_i^{\mathbf{w}} &= [\overrightarrow{h_i^{\mathbf{w}}}, \overleftarrow{h_i^{\mathbf{w}}}]. \end{aligned} \tag{2}$$

To construct a representation of a sentence $\mathbf{s}_j$, we follow Dong et al. (2017) to use an attention pooling layer to automatically calculate weights of the word context representations obtained from the intermediate hidden states of Bi-LSTM $\{h_1^{\mathbf{w}}, h_2^{\mathbf{w}}, ..., h_n^{\mathbf{w}}\}$:

$$\begin{aligned} u_i^{\mathbf{w}} &= tanh(W_u^{\mathbf{w}} h_i^{\mathbf{w}}), \\ a_i^{\mathbf{w}} &= \frac{exp(W_a^{\mathbf{w}} u_i^{\mathbf{w}})}{\sum_i exp(W_a^{\mathbf{w}} u_i^{\mathbf{w}})}, \\ \mathbf{s}_j &= \sum_i a_i^{\mathbf{w}} h_i^{\mathbf{w}}, \end{aligned} \tag{3}$$

where $W_u^{\mathbf{w}}$ and $W_a^{\mathbf{w}}$ refer to learnable parameters, $u_i^{\mathbf{w}}$ and $a_i^{\mathbf{w}}$ are the attention vector and the attention weight of the i-th word in the sentence s_j, respectively. The attention mechanism (Xu et al., 2015; Luong et al., 2015) emphasizes the salient words to build better sentence representation $\mathbf{s}_j$.

2.2 Essay Representation

An essay representation is constructed similarly to the sentence representation. Instead of taking a sequence of words $\{w_1, w_2, ..., w_n\}$ as an input, we employ another Bi-LSTM over a sequence of sentence representations $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_m\}$, as shown on the right-hand side of Figure 2:

$$\begin{aligned} \overrightarrow{h_j^{\mathbf{s}}} &= LSTM(\mathbf{x}_j, \overrightarrow{h_{j-1}^{\mathbf{s}}}), \\ \overleftarrow{h_j^{\mathbf{s}}} &= LSTM(\mathbf{x}_j, \overleftarrow{h_{j+1}^{\mathbf{s}}}), \\ h_j^{\mathbf{s}} &= [\overrightarrow{h_j^{\mathbf{s}}}, \overleftarrow{h_j^{\mathbf{s}}}]. \end{aligned} \tag{4}$$

In the same way as constructing sentence representation, attention pooling is used to summarize all of the sentence contexts:

$$\begin{aligned} u_j^{\mathbf{s}} &= tanh(W_u^{\mathbf{s}} h_j^{\mathbf{s}}), \\ a_j^{\mathbf{s}} &= \frac{exp(W_a^{\mathbf{s}} u_j^{\mathbf{s}})}{\sum_j exp(W_a^{\mathbf{s}} u_j^{\mathbf{s}})}, \\ \mathbf{d} &= \sum_j a_j^{\mathbf{s}} h_j^{\mathbf{s}}, \end{aligned} \tag{5}$$

where $W_u^{\mathbf{s}}$ and $W_a^{\mathbf{s}}$ are learnable parameters, $u_i^{\mathbf{s}}$ and $a_i^{\mathbf{s}}$ are the attention vector and attention weight of the j-th sentence in the essay d, respectively.

2.3 Objective

Essay scoring The main task of our model is to predict the score of an essay. It is predicted by applying a fully connected layer to an essay representation $\mathbf{d}$. Then we bound the score in the range [0, 1] with a sigmoid function,

$$\hat{y} = \sigma(W^{\mathbf{d}} \mathbf{d}), \tag{6}$$

where $W^{\mathbf{d}}$ is a learnable weight matrix of a fully connected layer, and $\hat{y}$ is a predicted score. Since it is a regression task, we use mean square error (MSE) as a loss function,

$$L_{sc} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{7}$$

where N is the total number of training data, y_i is the ground-truth score, and $\hat{y}_i$ is a predicted score obtained by the model.

Sentiment Prediction Another objective of the model is to predict the sentiments of the words and sentences. To obtain such a probability distribution over word sentiment classes, we use a fully connected layer normalized by a softmax function over the hidden states of word-level Bi-LSTM $\{h_1^{\mathbf{w}}, h_2^{\mathbf{w}}, ..., h_n^{\mathbf{w}}\}$,

$$P(stm_i^{\mathbf{w}c} | h_i^{\mathbf{w}}) = softmax(W_{stm}^{\mathbf{w}} h_i^{\mathbf{w}}), \tag{8}$$

where $W_{stm}^{\mathbf{w}}$ is a learnable weight matrix of a fully connected layer, $P(stm_i^{\mathbf{w}c} | h_i \mathbf{w})$ is a predicted probability distribution of the word i-th sentiment, and c denotes a class (e.g., positive, negative). A similar method is employed for sentences,

$$P(stm_j^{\mathbf{s}c} | h_j^{\mathbf{s}}) = softmax(W_{stm}^{\mathbf{s}} h_j^{\mathbf{s}}), \tag{9}$$

Prompts	#Essays	Genre	Avg Length	Score Range
1	1783	Persuasive / Narrative / Expository	350	2-12
2	1800	Persuasive / Narrative / Expository	350	1-6
3	1726	Source dependent responses	150	0-3
4	1772	Source dependent responses	150	0-3
5	1805	Source dependent responses	150	0-4
6	1800	Source dependent responses	150	0-4
7	1569	Persuasive / Narrative / Expository	250	0-30
8	723	Persuasive / Narrative / Expository	650	0-60

Table 1: ASAP dataset detail and statistics.

where $W^{\mathbf{s}}_{stm}$ is a learnable weight matrix of a fully connected layer, and $P(stm^{\mathbf{sc}}_j|h^{\mathbf{s}}_j)$ is a predicted probability distribution of the sentence j-th sentiment. The word and sentence sentiment prediction losses are calculated by using the negative log-probability of the correct sentiment labels,

$$L_w = -\sum_i \sum_j \sum_c stm^{\mathbf{wc}}_{ij} logP(stm^{\mathbf{wc}}_{ij}|h^{\mathbf{w}}_{ij}), \tag{10}$$

$$L_s = -\sum_j \sum_c stm^{\mathbf{sc}}_j logP(stm^{\mathbf{sc}}_j|h^{\mathbf{s}}_j), \tag{11}$$

where i indicates a number of words in a sentence, j refers to the number of sentences in an essay. c shows the number of classes of sentiment. $stm^{\mathbf{w}}_{ijc}$ and $stm^{\mathbf{s}}_{jc}$ indicate the ground-truth labels of word and sentence sentiment, respectively.

To learn in a multi-task manner, the model optimizes the total loss of the main and auxiliary objectives with different weight indicators, as shown in Eq. (12).

$$L_{total} = \alpha L_{sc} + \beta L_w + \gamma L_s, \tag{12}$$

where α, β, and $\gamma \in [0, 1]$ are hyperparameters.

3 Experiments

3.1 Dataset

In our experiments, we used the Automated Student Assessment Prize (ASAP)[1] public dataset on Kaggle to evaluate our methods. The dataset contains eight different prompts of the essay, as described in Table 1. The prompts elicit responses of different genres and of different lengths. The essays were written by students ranging from grade 7 to grade 10 and graded by at least two human graders.

We followed Taghipour and Ng (2016) to use 5-fold cross-validation for the evaluation with the same splits. In 5 folds, three folds of the data are used as a training set, one fold as the development set, and one fold as the test set. The final result is then calculated from the average of the five folds.

3.2 Sentiment annotation

We tokenize an essay into sentences and extract its sentiments using the Stanford CoreNLP[2] based on Recursive Neural Tensor Network (Socher et al., 2013). It first split an essay into sentences, then annotate each sentence and the words in it with sentiment labels, as shown in Figure 1. The extracted sentiments are represented within five sentiment classes, i.e., very negative, negative, neutral, positive, and very positive. The model was trained on the Standford Sentiment Treebank dataset extracted from movie reviews.

3.3 Evaluation Metric

The Quadratic Weighted Kappa (QWK) is used as the evaluation metric, which measures correlation or agreement between two raters (Yannakoudakis and Cummins, 2015), since it is the official evaluation metric of the ASAP competition. The QWK score ranges from 0 to 1. It can also become negative if there is less agreement than expected by chance. Therefore, the higher the value of QWK, the better the results. It is calculated using

$$K = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}, \tag{13}$$

where matrices w, O, and E are the matrices of weights, observed scores, and expected scores, respectively. A value of $O_{i,j}$ denotes the number of essays that receive a score i by the first-rater and a score j by the second-rater. The weights are

[1] https://www.kaggle.com/c/asap-aes

[2] https://nlp.stanford.edu/sentiment/

Method	Prompts								
	1	2	3	4	5	6	7	8	Avg QWK
Single-task	0.827	0.667	0.673	0.801	**0.820**	0.814	0.802	0.688	0.762
Multi-task (word sentiment)	**0.833**	0.685	0.690	0.795	0.812	**0.816**	0.798	0.673	0.763
Multi-task (sentence sentiment)	0.818	0.674	0.683	0.786	0.786	0.812	0.786	0.666	0.751
Multi-task (word&sentence sentiment)	0.803	0.658	0.664	0.772	0.799	**0.816**	0.787	0.644	0.743
Dong and Zhang (2016)	-	-	-	-	-	-	-	-	0.734
Taghipour and Ng (2016)	0.775	**0.687**	0.683	0.795	0.818	0.813	**0.805**	0.594	0.746
Dong et al. (2017)	0.822	0.682	0.672	**0.814**	0.803	0.811	0.801	**0.705**	**0.764**
Tay et al. (2017)	0.832	0.684	**0.695**	0.788	0.815	0.810	0.800	0.697	**0.764**

Table 2: Experimental results. Best result is in bold and 2nd best is underlined.

$w_{i,j} = \frac{(i-j)^2}{(N-1)^2}$, where N is the number of possible scores, matrix E is calculated as the outer product between two histogram vectors of the scores. Then matrices O and E are normalized to have the same sum and then calculate the QWK score.

3.4 Baselines

We compare our model with several baselines:

Single-task uses only one objective, which is essay scoring, without any utilization of sentiment analysis. The model is an attention-based hierarchical Bi-LSTM. The loss weight indicator α is fixed to 1, β, and γ are fixed to 0 in Eq. (1).

Multi-task has a combination objective of essay scoring and sentiment prediction. Multi-task (word sentiment) focuses only on the prediction of word sentiment. Hence, the loss weight indicator γ is fixed to 0 in Eq. (1). Similarly, the loss weight indicator β is set to 0 for multi-task (sentence sentiment) for switching off the task.

We also compare our model with several neural deep learning approaches for AES:

Hierarchical CNN (Dong and Zhang, 2016) comprises two layers of CNN, in which one convolutional layer is used to extract sentence representations, and the other is stacked on sentence vectors to learn essay representations. Concatenation of max-pooling and average pooling is used to produce the sentence and essay vectors.

RNN (Taghipour and Ng, 2016) with long short-term memory units (LSTM). LSTM units make use of three gates to forget or pass the information through time. They showed that using a mean-over-time layer is much more effective than using the last state vector or attention mechanism.

Attention-based RCNN (Dong et al., 2017) is similar to hierarchical CNN. Instead, the convolutional layer is replaced by an LSTM layer at the sentence-level to learn global coherence. Above the CNN layer and LSTM layer, an attention pooling layer is employed to acquire sentence representations and essay representations, respectively.

SKIPFLOW (Tay et al., 2017) is based on long short-term memory (LSTM) network. SKIPFLOW mechanism possesses a neural tensor layer to model the relationship between two positional outputs of LSTM across time steps. The tensor generates a coherence feature and also acts as an auxiliary memory. The coherence feature vector is then concatenated with the essay representation obtained from a mean pooling over the entire LSTM layer's hidden states.

3.5 Implementation Setup

In the embedding layer, we used the pre-trained word embedding GloVe[3] (Pennington et al., 2014) trained on 6 billion words from Wikipedia 2014 and Gigaword 5. During the training process, word embeddings are fine-tuned. The vocabulary was set to the 4,000 most frequent words by following Taghipour and Ng (2016) and treating other words as unknown words. We set the number of the essay sentences to the maximum for each essay prompts and the maximum sentence length to 128 and trained the models on batch size 16 for 50 epochs. The following hyperparameters were tuned by using optuna[4] in 100 trials.

- Embedding dimension: {50, 100, 200, 300}
- LSTM dimension: {50, 100, 200, 300}
- Optimizer: {RMSprop, Adam}
- Learning rate: [0.0001, 0.01]
- Dropout rate: [0.1, 0.5]
- α, β and γ: [0, 1]

[3]https://nlp.stanford.edu/projects/glove/
[4]https://optuna.org/

Prompts	Multi-task objective weights								
	word sentiment			sentence sentiment			word&sentence sentiment		
	α	β	γ	α	β	γ	α	β	γ
1	0.9	0.9	-	0.4	-	0.2	0.3	0.2	0.1
2	1.0	0.1	-	0.7	-	0.5	0.7	0.9	0.6
3	0.8	0.2	-	0.6	-	0.2	0.9	0.9	0.5
4	0.5	0.2	-	0.8	-	0.1	0.8	0.1	0.2
5	0.8	0.7	-	1.0	-	0.2	0.7	0.1	0.4
6	0.9	0.1	-	1.0	-	0.1	1.0	0.1	0.1
7	1.0	0.4	-	1.0	-	0.1	0.7	1.0	0.1
8	0.4	0.2	-	1.0	-	0.1	0.8	0.8	0.1

Table 3: Multi-task objective weights after tuning hyperparameters by 5-fold cross-validation. α indicates the scoring task weight, β and γ indicate the weights of the word and sentence sentiment prediction tasks, respectively

4 Results and Discussion

We can see from Table 2 that the results obtained from the Single-task already give good results compared to the existing deep learning approaches. The Multi-task (word sentiment) performed slightly better than the Single-task on four prompts, and the average result is close. The Multi-task (sentence sentiment) performed better than the Single-task only on two prompts and totally worse than Multi-task (word sentiment). The complexity of sentence sentiment might be too difficult to extract beneficial information and affect the model's sharing parameters. We tracked the accuracy of the sentence sentiment prediction during the training process. The model could reach only 70-80% accuracy. In contrast to Multi-task (word sentiment), the model could reach up to 99% accuracy of word sentiment prediction. The Multi-task (word&sentence) performed the worst among Multi-task and Single-task. The sentence sentiment prediction task's difficulty and two auxiliary tasks might make the main objective of the model, essay scoring, unstable.

Table 3 reports the Multi-task objective weights after tuning hyperparameters. We observe the model could find the proper objective weights for Multi-task (word sentiment) and Multi-task (sentence sentiment). The model performed best when the main objective weight α is greater than auxiliary objective weights, β, and γ. In contrast, in Multi-task (word&sentence sentiment), the model seems unable to find proper objective weights to balance the main and auxiliary tasks. As the summation of auxiliary task weights, β, and γ, is larger than that of α.

Comparing with other related neural deep learning models on AES, we also found that Single-task and Multi-task (word sentiment) are better than Dong and Zhang (2016) and Taghipour and Ng (2016) and comparable to Dong et al. (2017) and Tay et al. (2017). Our models perform poorly on prompt 8. One reason is that prompt 8 has the longest average length, and we limit the sentence length too short. We need to utilize the full text of an essay to the models for further improvement.

5 Conclusion

In this paper, we described a neural approach incorporating sentiment analysis and automatic essay scoring (AES). Our method is based on a hierarchical structure multi-task learning model. We compared our approach to several neural deep learning approaches (Dong and Zhang, 2016; Taghipour and Ng, 2016; Dong et al., 2017; Tay et al., 2017) on the Automated Student Assessment Prize (ASAP) benchmark. Overall, our approach is competitive with the best ones. Using word sentiment with multi-task learning, we report better results on four prompts compared to the single-task. We intend to make the model more sophisticated in future work and to apply other auxiliary tasks. We also would like to investigate the use of contextualized embeddings (e.g., BERT Devlin et al. (2019)) that shows amazing performance in many NLP tasks.

Acknowledgements

We thank Dr. Danial Beck for valuable suggestions and feedback. We are grateful to the anonymous AACL reviewers for their insightful comments and suggestions. This work was supported by the Grant-in-aid for JSPS, Grant Number 17K00299, Support Center for Advanced Telecommunications Technology Research, Foundation, and KDDI Foundation Research Grant Program.

References

Isabelle Augenstein and Anders Søgaard. 2017. Multi-task learning of keyphrase boundary classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 341–346, Vancouver, Canada. Association for Computational Linguistics.

Beata Beigman Klebanov, Jill Burstein, Nitin Madnani, Adam Faulkner, and Joel Tetreault. 2012. Building subjectivity lexicon(s) from scratch for essay data. In *Computational Linguistics and Intelligent Text Processing*, pages 591–602, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 681–684, Los Angeles, California. Association for Computational Linguistics.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-domain name error detection using a multi-task RNN. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 737–746, Lisbon, Portugal. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Ronan Cummins and Marek Rei. 2018. Neural multi-task learning in automated assessment. *CoRR*, abs/1801.06830.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained multi-task learning for automated essay scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–799, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring – an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas. Association for Computational Linguistics.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

Jeffrey L. Elman. 1990. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211.

Youmna Farag and Helen Yannakoudakis. 2019. Multi-task learning for coherence modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 629–639, Florence, Italy. Association for Computational Linguistics.

Noura Farra, Swapna Somasundaran, and Jill Burstein. 2015. Scoring persuasive essays using opinions and their targets. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–74, Denver, Colorado. Association for Computational Linguistics.

Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

H. K. Janda, A. Pawar, S. Du, and V. Mago. 2019. Syntactic, semantic and sentiment analysis: The joint effect on automated essay evaluation. *IEEE Access*, 7:108486–108503.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Beata Beigman Klebanov, Jill Burstein, and Nitin Madnani. 2013. Sentiment profiles of multiword expressions in test-taker essays: The case of noun-noun compounds. *ACM Trans. Speech Lang. Process.*, 10(3).

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Mohsen Mesgar and Michael Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339, Brussels, Belgium. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43, Copenhagen, Denmark. Association for Computational Linguistics.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2018. A hierarchical multi-task approach for learning embeddings from semantic tasks. *CoRR*, abs/1811.06031.

Nishit Shrestha and Fatma Nasoz. 2019. Deep learning sentiment analysis of amazon.com reviews and ratings. *CoRR*, abs/1904.04096.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Wei Song, Dong Wang, Ruiji Fu, Lizhen Liu, Ting Liu, and Guoping Hu. 2017. Discourse mode identification in essays. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 112–122, Vancouver, Canada. Association for Computational Linguistics.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *CoRR*, abs/1902.09314.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. *CoRR*, abs/1711.04981.

Tan Thongtan and Tanasanee Phienthrakul. 2019. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044.

Helen Yannakoudakis and Ronan Cummins. 2015. Evaluating the performance of automated text scoring systems. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–223, Denver, Colorado. Association for Computational Linguistics.

Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *CoRR*, abs/1707.08114.