ISSN 2007-9737

Multi-Head Multi-Layer Attention to Deep Language Representations for Grammatical Error Detection

Masahiro Kaneko, Mamoru Komachi

Tokyo Metropolitan University,
Graduate School of Systems Design, Tokyo,
Japan

kaneko-masahiro@ed.tmu.ac.jp, komachi@tmu.ac.jp

Abstract. It is known that a deep neural network model pre-trained with large-scale data greatly improves the accuracy of various tasks, especially when there are resource constraints. However, the information needed to solve a given task can vary, and simply using the output of the final layer is not necessarily sufficient. Moreover, to our knowledge, exploiting large language representation models to detect grammatical errors has not yet been studied. In this work, we investigate the effect of utilizing information not only from the final layer but also from intermediate layers of a pre-trained language representation model to detect grammatical errors. We propose a multi-head multi-layer attention model that determines the appropriate layers in Bidirectional Encoder Representation from Transformers (BERT). The proposed method achieved the best scores on three datasets for grammatical error detection tasks, outperforming the current state-of-the-art method by 6.0 points on FCE, 8.2 points on CoNLL14, and 12.2 points on JFLEG in terms of $\mathrm{F}_{0.5}$. We also demonstrate that by using multi-head multi-layer attention, our model can exploit a broader range of information for each token in a sentence than a model that uses only the final layer's information.

Keywords. Multi-head multi-layer attention, grammatical error detection.

1 Introduction

Neural networks are known to be best exploited when trained on large-scale data. It has been demonstrated that utilizing language representation models pre-trained with large-scale data is effective for various tasks. For example, recent studies have shown a significant improvement using large-scale data to train large deeper models for natural language understanding tasks [2, 4, 11].

In contrast, for grammatical error detection, several studies have adapted large-scale data by creating artificial training data from a large-scale raw corpora [6, 14]. Moreover, there have been studies that have effectively used language representation models for grammatical error detection task [13]. To our knowledge, however, there are no studies that have utilized deep language representation models pre-trained with large-scale data for this task.

Moreover, deep neural networks learn different representations for each layer. For example, Belinkov et al. [3] demonstrated that in a machine translation task, the lower layers of the network learn to represent the word structure, while higher layers are more focused on word meaning.

Peters et al. [11] showed that in learning deep contextualized word representations, constructing representations of layers corresponding to each task by a weighted sum improved the accuracy of six NLP tasks. Peters et al. [12] empirically showed that lower layers are best-suited for local syntactic relationships, that higher layers better model longer-range relationships, and that the top-most layers specialize at the language modeling.

For tasks that emphasize the grammatical nature, such as grammatical error detection, information from the lower layers is considered to be important alongside more expressive information in deep layers. Therefore, we

doi: 10.13053/CyS-23-3-3271

hypothesized that using information from optimal layers suitable for a given task is important.

As such, our motivation is to construct a deep grammatical error detection model that considers optimal information from each layer. Therefore, we propose a model that uses multi-head multi-layer attention in order to construct hidden representations from different layers suitable for grammatical error detection.

Our contributions are as follows:

1. We propose a multi-head multi-layer attention model that can acquire even more suitable representations for a given task by fine-tuning a pre-trained deep language representation model with large-scale data for grammatical error detection.

2. We show that our model is effective at acquiring hidden representations from various layers for grammatical error detection. Our analysis reveals that using multi-head multi-layer attention effectively utilizes information from various layers. We also demonstrate that our proposed model can use a wider range of information for each token in a sentence.

3. Experimental results show that our multi-head multi-layer attention model achieves state-of-the-art results on three grammatical error detection datasets (viz., FCE, CoNLL14, and JFLEG).

2 Related Works

2.1 Grammatical Error Detection with Language Representations

Often, in sequence labeling tasks, recent supervised neural grammatical error detection models are built upon Bi-LSTM [5, 6, 13, 14, 15, 16]. Rei and Søgaard [15] used token-level predictions by Bi-LSTM for self-attention to predict sentence-level labels for grammatical error detection. However, we adopt a transformer block-based model for token-level grammatical error detection, and we build a very deep model for this task.

Rei [13] showed the effectiveness of multitask learning by coupling language modeling and grammatical error detection.

They used an additional objective for language modeling training to learn to predict surrounding tokens for every token in a dataset. In contrast to previous research, we adopt information from deep language representations for grammatical error detection by multi-head multi-layer attention.

Several studies have exploited large quantities of raw data to create additional artificial data. Rei et al. [14] artificially generated writing errors in order to create additional resources to learn a neural sequence labeling model following Rei [13]. Kasewa et al. [6] employed a neural machine translation system to create error-filled artificial data for grammatical error detection. By contrast, we directly adopt a pre-trained language representation model trained with large-scale raw data.

2.2 Using the Layer Representations

Deep Contextualized Word Representations (ELMo) [11] used large-scale data for a deep language representation model. Their model learns task-specific weighting from all fixed hidden layers of the pre-trained bidirectional long short-term memory (Bi-LSTM) to construct contextualized word embeddings optimized to a given task. In other words, ELMo learns task-specific representations exclusively in the first layer, whereas other parameters of a pre-trained model remain unchanged. On the contrary, we construct representations suited for given tasks by fine-tuning all parameters of our pre-trained model, using multi-head multi-layer attention. All parameters and constructed representations of our model are trained to be best-suited for the given task.

Takase et al. [18] employed intermediate layer representations, including input embeddings, to calculate the probability distributions in order to solve a ranking problem in language generation tasks. Similarly, we considered the information of each layer, but our motivation is to seize the optimal information from each layer suitable for a given task using a multi-head multi-layer attention.

Moreover, their model estimated probability distributions from each layer, whereas ours constructs hidden representations from each layer for the output layer.

Furthermore, there is a study that predicts information from the middle layer of each layer of the language model and learns the errors occurring owing to the model [1]. The use of the information of the middle layer of `transformer_block` is common to our research, but the information of each layer is not taken into account at the time of evaluation and is used only for learning. Furthermore, the information on the surface layer is less useful and learning is undertaken so that the influence of the surface layer decreases as learning progresses. In contrast, as the method uses attention, it also lets you learn which layer is utilized in the model itself.

3 Deep Language Representations for Grammatical Error Detection

We propose a model that applies multi-head attention to each layer (multi-head multi-layer attention, MHMLA) to fine-tune pre-trained Bidirectional Encoder Representations from Transformers (BERT) [4]. Architectures of BERT and MHMLA for the grammatical error detection task are illustrated in Figure 1. In this section, we first introduce BERT and then explain our proposed model, MHMLA.

3.1 BERT

BERT is designed to learn deep bidirectional representations by jointly conditioning both the left and right contexts in all layers (Figure 1(a)). It is based on a multi-layer bidirectional transformer encoder [20]. Insofar it is a deep language representation model pre-trained on large-scale data, it can be used for fine-tuning. It achieved state-of-the-art results for a wide range of tasks such as natural language understanding, name entity recognition, question answering, and grounded commonsense inference [4].

BERT has a multi-layer bidirectional transformer encoder and can be used for different architectures, such as in classification and sequence-to-sequence learning tasks. Here, we explain the BERT's architecture for sequence labeling tasks. Given a sequence $S = w_0, \cdots, w_n, \cdots, w_N$ as input, BERT is formulated as follows:

$$\begin{aligned} h_n^0 &= W_\mathrm{e} w_n + W_\mathrm{p}, & (1)\\ h_n^l &= \mathrm{transformer\_block}(h_n^{l-1}), & (2)\\ y_n^{(\mathrm{BERT})} &= \mathrm{softmax}(W_\mathrm{o} h_n^L + b_\mathrm{o}), & (3)\end{aligned}$$

where w_n is a current token, and N denotes the sequence length. Equation 1 thus creates an input embedding. Here, `transformer_block` includes self-attention and fully connected layers [20], and outputs h_n^l. l is the number of the current layer, $l \geq 1$. L is the total number of layers of BERT. Equation 3 denotes the output layer. W_o is an output weight matrix, b_o is a bias for the output layer, and $y_n^{(\mathrm{BERT})}$ is a prediction.

The parameters W_e, W_p and `transformer_block` are pre-trained on a large document-level corpus using a masked language model [19] and predicting a next sentence. Then, BERT uses a different task-specific matrix W_o of the output layer (Equation 3) for a given sequence labeling task. To adapt BERT for specific tasks, all parameters of BERT are fine-tuned jointly by predicting a task-specific label with the task-specific output layer to maximize the log-probability of the correct label.

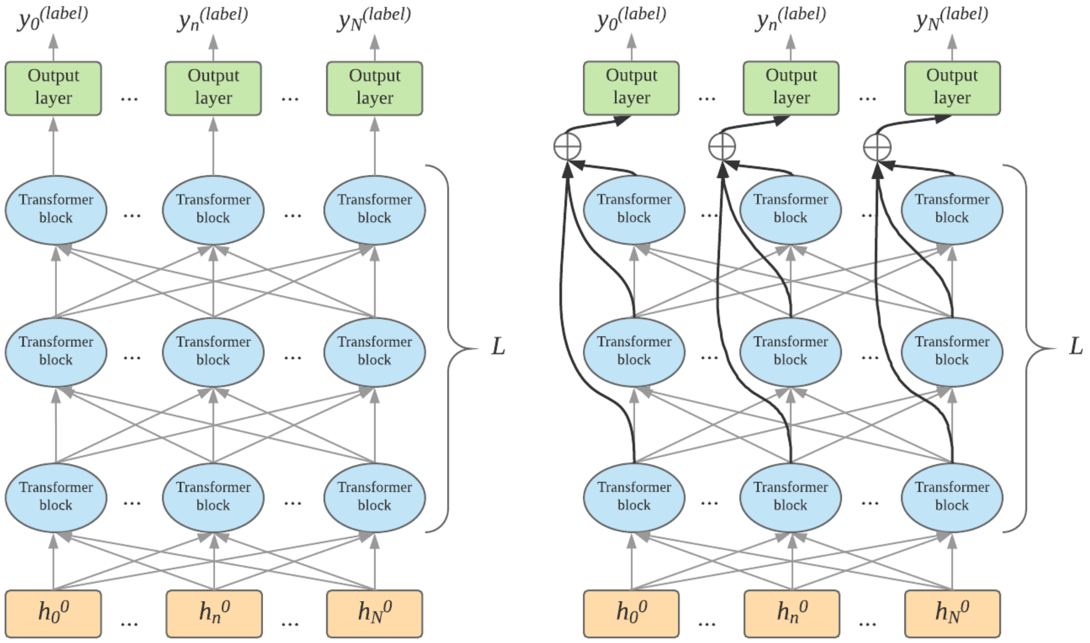
3.2 Multi-Head Multi-Layer Attention to Acquire Task-Specific Representations

Multi-head attention [20] is more beneficial than a single attention function. MHMLA on a sequence labeling model applies attention to each layer l of the output of `transformer_block` h_n^l of Equation 2 (Figure 1(b)). First, we calculate attention value v_n^l:

$$v_{n,j}^l = W_{\mathrm{v}j}^l h_n^l + b_{\mathrm{v}j}^l. \quad (4)$$

Here, W_v is a weight matrix, b_v is a bias, and j is a head number. We apply a non-linear layer to h_n^l to acquire k_n^l. Attention score a_n^l is as follows:

$$\begin{aligned} k_{n,j}^l &= \mathrm{relu}(W_{\mathrm{k}j}^l h_n^l + b_{\mathrm{k}j}^l), & (5)\\ a_{n,j}^l &= W_{\mathrm{a}j}^l k_n^l + b_{\mathrm{a}j}^l, & (6)\end{aligned}$$



(a) BERT.

(b) MHMLA.

Fig. 1. Architectures of BERT and MHMLA for grammatical error detection

where W_k and W_a are weight matrices, and b_k and b_a are biases. Multi-heads are then calculated as follows:

$$\tilde{a}^l_{n,j} = \frac{\exp(a^l_{n,j})}{\sum_{t=1}^{L}\exp(a^t_{n,j})}, \tag{7}$$

$$\mathrm{head}_{n,j} = \sum_{t=1}^{L} \tilde{a}^t_{n,j} v^t_{n,j}, \tag{8}$$

where $\tilde{a}^l$ is the attention weight, normalized to sum up to 1 over all values in the layers. These weights are then used to combine the context-conditioned hidden representations from Equation (5) into a single-token representation c_n:

$$c_n = \mathrm{concat}(\mathrm{head}_{n,1}, \cdots, \mathrm{head}_{n,J}), \tag{9}$$

where J is the total number of heads. Finally, we return task-specific predictions based on this representation:

$$y_n^{(\mathrm{label})} = \mathrm{softmax}(W_\mathrm{o} c_n + b_\mathrm{o}). \tag{10}$$

W_o is an output weight matrix and b_o is a bias of output layer. Our model is optimized by minimizing cross-entropy loss on the token-level annotation.

Table 1. Sentence statistics of used corpora

corpus	train	dev	test
FCE	28,731	2,222	2,720
CoNLL14	-	-	1,312
JFLEG	-	-	747

4 Experiments

4.1 Datasets

We focus on a supervised sequence labeling task: viz., grammatical error detection. Grammatical error detection is the task of identifying incorrect tokens that need to be edited in order to produce a grammatically correct sentence. We evaluated our approach on the three different grammatical error detection datasets. Table 1 shows statistics for each corpus.

FCE. We fine-tuned and searched the parameters of the model and evaluated our system on the First Certificate in English (FCE) dataset [22], which contains error-annotated short essays written by language learners. The FCE dataset is a popular English learner corpus for grammatical error detection. We followed the official split of the data.

CoNLL14. We additionally used dataset from the CoNLL 2014 shared task (CoNLL14) dataset [10] in our evaluation. This dataset was written by higher-proficiency learners on different technical topics. It was manually corrected by two separate annotators, and we report results on each of these annotations (CoNLL14-$\{1,2\}$).

JFLEG. We also evaluated our approach with the JHU FLuency-Extended GUG (JFLEG) corpus [9]. It contains a broad range of language-proficiency levels and focuses more on fluency edits and making the text more native-sounding, in addition to grammatical corrections. JFLEG is not labeled for grammatical error detection. Therefore, we used dynamic programming to label tokens in sentences as correct or incorrect. Because JFLEG is a recently developed corpus, there is only one prior study with experimental results [15]. JFLEG is tagged by multiple annotators, like CoNLL14, so we followed this work to build a version that combines the references: if a token is labeled as an error by any annotator, it is marked as an error[1].

4.2 Experimental Details

We used a publicly available pre-trained deep language representation model, namely the $\mathrm{BERT}_{\mathrm{BASE}}$ uncased model[2]. This model has 12 layers, 768 hidden size, and 16 heads of self-attention. Layer attention has 12 heads (J = 12). We fine-tuned the model over 5 epochs with a batch size of 32. The maximum training sentence length was 128 tokens. We used the Adam optimizer [7] with a learning rate of 5e-05. We applied dropout [17] to h_n^l, $k_{n,j}^l$, and $\tilde{a}_{n,j}^l$ with a dropout rate of 0.3. $\tilde{a}_{n,j}^l$ is referred to as attention dropout. We also used WordPiece embeddings [21]. To make this compatible with sub-token tokenization, we inputted each tokenized word into the WordPiece tokenizer and used the hidden state corresponding to the first sub-token as input to the output layer, as with the original BERT.

We used $\mathrm{F}_{0.5}$ as the main evaluation measure. This measure was also adopted in the CoNLL14 shared task for the grammatical error correction task [10]. It combines both precision and recall, while assigning twice as much weight to precision, because accurate feedback is often more important than coverage in error detection applications [8].

4.3 Baselines and Comparisons

We compare with models of Rei [13], Rei and Søgaard [15], Rei et al. [14], and Kasewa et al. [6] which are based on the Bi-LSTM architecture. The first group, Rei [13] and Rei and Søgaard [15], was trained exclusively on the FCE dataset. The second group, Rei et al. [14] and Kasewa et al. [6] used additional artificial data along with the FCE dataset for training.

Our baseline and proposed models were trained on the transformer architecture. The first three are the descriptions of our baselines, and the fourth is a description of the proposed model:

$\mathrm{BERT}_{\mathrm{BASE}}$ w/o pre-train. This model is trained using only the FCE dataset and with random initialization. This baseline did not use any other corpus for training.

$\mathrm{BERT}_{\mathrm{BASE}}$. This is the original pre-trained model described in Section 4.2 fine-tuned on the FCE dataset. This baseline uses original BERT model [4] and can be seen as surrogated version of the proposed method without multi-layer attention.

[1] Although JFLEG's experimental settings are not described in the paper, we confirmed them with the authors of the paper over e-mail.

[2] `https://github.com/google-research/bert`

doi: 10.13053/CyS-23-3-3271

Table 2. Results of grammatical error detection. These results are averaged over five runs. $*$ and † indicate that there is a significant difference against $\mathrm{BERT}_{\mathrm{BASE}}$ and AvgL, respectively

	FCE			CoNLL14-1			CoNLL14-2			JFLEG		
	P	**R**	$\mathbf{F}_{0.5}$	**P**	**R**	$\mathbf{F}_{0.5}$	**P**	**R**	$\mathbf{F}_{0.5}$	**P**	**R**	$\mathbf{F}_{0.5}$
Rei [13]	58.88	28.92	48.48	17.68	19.07	17.86	25.22	19.25	23.62	-	-	-
Rei and Søgaard [15]	65.53	28.61	52.07	25.14	15.22	22.14	37.72	16.19	29.65	72.53	25.04	52.52
Rei et al. [14]	60.67	28.08	49.11	23.28	18.01	21.87	35.28	19.42	30.13	-	-	-
Kasewa et al. [6]	-	-	55.6	-	-	28.3	-	-	35.5	-	-	-
$\mathrm{BERT}_{\mathrm{BASE}}$ w/o pre-train	48.85	11.30	29.34	11.45	7.80	10.47	18.24	9.31	15.30	58.85	13.22	34.81
$\mathrm{BERT}_{\mathrm{BASE}}$	**69.80**	37.37	59.47	34.08	**33.56**	33.97	46.01	33.89	42.93	**78.06**	36.28	63.45
AvgL	68.09	41.14	60.20	34.97	32.02	34.33	45.33	35.27	42.88	77.35	37.05	63.52
MHMLA	68.87†	**43.45***†	**61.65***†	**35.74***	33.50†	**35.26***†	**46.45**†	**35.47***	**43.74**†	77.74	**38.85***†	**64.77***†

Table 3. $\mathrm{F}_{0.5}$ scores of MHMLA using different number of heads J. These results are averaged over five runs

J	FCE	CoNLL14-1	CoNLL14-2	JFLEG
1	61.16	33.75	42.89	63.98
2	61.62	33.44	42.42	63.72
3	**61.90**	34.50	43.17	64.45
4	61.55	33.74	42.80	64.37
6	61.22	34.26	43.29	64.48
8	61.27	34.72	43.02	64.10
12	61.65	**35.26**	**43.74**	**64.77**

AvgL. This model is called averaged layers, which averages representations after linear transformation of h_n^l (Equation 2) for the output layer of $\mathrm{BERT}_{\mathrm{BASE}}$ model instead of using an attention.

MHMLA. This is the proposed model that is an extension of $\mathrm{BERT}_{\mathrm{BASE}}$, with MHMLA to the pre-trained model while fine-tuning on the FCE dataset.

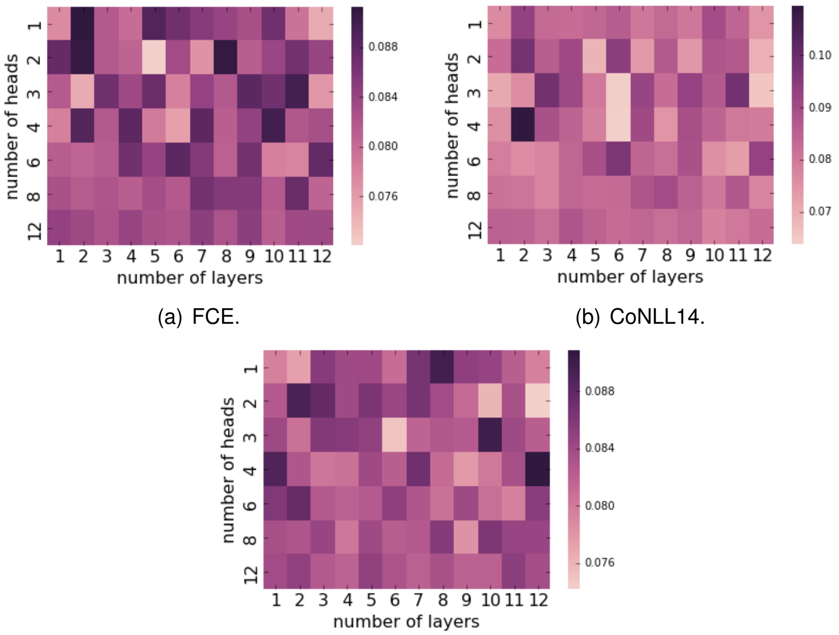
5 Results

Table 2 shows the grammatical error detection results for the FCE, CoNLL14-{1,2}, and JFLEG datasets. Scores for Rei [13], Rei and Søgaard [15], Rei et al. [14], and Kasewa et al. [6] were taken from their respective papers. In FCE, CoNLL14, and JFLEG, the $\mathrm{BERT}_{\mathrm{BASE}}$ model significantly outperformed existing methods and our baseline (without pre-training) in terms of precision, recall, and $\mathrm{F}_{0.5}$. This demonstrates that using a pre-trained deep language representation model is highly effective for grammatical error detection. Furthermore, MHMLA achieved the highest $\mathrm{F}_{0.5}$ on all datasets, outperforming $\mathrm{BERT}_{\mathrm{BASE}}$ by 2.18 points, 1.29 points, 0.81 points, and 1.32 points on FCE, CoNLL14-{1,2}, and JFLEG, respectively. The scores for the AvgL model were lower than that for our proposed MHMLA model, meaning that naively using information from layers is not as effective as using MHMLA. These results show that using MHMLA and learning task-specific representations improves the accuracy.

To verify the effect of MHMLA, we examined the $\mathrm{F}_{0.5}$ value for each head number. We investigated 1, 2, 3, 4, 6, 8, and 12 heads (i.e. the number of heads up to 12 by which the hidden layer size of 768 can be divided). Table 3 shows the $\mathrm{F}_{0.5}$ values for each number of heads on FCE, CoNLL14-{1,2}, and JFLEG datasets. Regarding FCE, the highest $\mathrm{F}_{0.5}$ score was achieved with 3 heads. For CoNLL14-{1,2} and JFLEG, the $\mathrm{F}_{0.5}$ values were highest with 12 heads, demonstrating that adopting multi-head leads to improved accuracy.

6 Analysis of the Effect of MHMLA

The purpose of MHMLA is to construct representations not only from the final layer but also from various layers. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. Therefore, it is considered that increasing the number of heads leads to utilization of information from various layers. Hence, we investigate the effect of the number of heads on each layer by visualizing the averaged score of MHMLA that was calculated by considering the heads j of Equation



(a) FCE.

(b) CoNLL14.

(c) JFLEG.

Fig. 2. Attention visualization of MHMLA on each dataset using a different number of heads. MHMLA with 8 and 12 heads tends to attend to all layers more or less equally for all datasets.

7 for all layers on test sets of the three datasets: FCE, CoNLL14, and JFLEG.

Figure 2 visualizes the average attention score to each layer of MHMLA for each head. The average attention score is calculated by averaging head_n in Equation (8). For all datasets, when there were a fewer numbers of heads, the multi-head attentions learned to attend to different layers but tended to focus on particular layers. For example, as shown in Figure 2(b), multi-head attention with heads of 2, 3, and 4 heads focused more on layers 2 and 3 while hardly attending to layers 5 and 6. Figure 2(b) shows that the same amount of attention is attended to each layer when the number of heads are 8 and 12. In Figure 2(c), attention is sharp, especially with the number of heads being 1, 2, 3, and 4. In contrast, with there are more heads, viz. 8 and 12, attention tended to attend to all layers more or less equally for all datasets. From this visualization, we conclude that our goal of utilizing the information from various layers has been achieved.

7 Conclusion

In this study, we investigated the effect of utilizing a deep language representation model ($\mathrm{BERT}_{\mathrm{BASE}}$) pre-trained on large-scale data for grammatical error detection. Simply fine-tuning our $\mathrm{BERT}_{\mathrm{BASE}}$ model greatly improved $\mathrm{F}_{0.5}$ scores for grammatical error detection task.

Furthermore, we have introduced an approach to learning representations suited for grammatical error detection task from various layers of a pre-trained deep language representation model using MHMLA. Our MHMLA model outperformed previous models for grammatical error detection, establishing new state-of-the-art $\mathrm{F}_{0.5}$ scores. Our analysis demonstrated that we succeeded at learning appropriate representations for a given task using information from different layers.

Future work includes applying MHMLA to other language representation models like Open AI GPT model [2]. Furthermore, with different combination of existing pre-trained language representation models, we hope to obtain even greater improvements. In addition, we will explore whether our layers learned the same syntactic and semantic roles as a previous work [12], also what exactly self-attention learns at a token-level for grammatical error detection.

Acknowledgement

This work was partially supported by JSPS Grant-in-Aid for Young Scientists (B) Grant Number JP16K16117.

References

1. **Al-Rfou, R., Choe, D., Constant, N., Guo, M., & Jones, L. (2019).** Character-level language modeling with deeper self-attention. *AAAI*, Association for the Advancement of Artificial Intelligence.
2. **Alec, R., Karthik, N., Tim, S., & Ilya, S. (2018).** *Improving Language Understanding with Unsupervised Learning*. Technical report, OpenAI.
3. **Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., & Glass, J. (2017).** What do neural machine translation models learn about morphology? *ACL*, Association for Computational Linguistics, pp. 861–872.
4. **Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018).** BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
5. **Kaneko, M., Sakaizawa, Y., & Komachi, M. (2017).** Grammatical error detection using error- and grammaticality-specific word embeddings. *IJCNLP*, Asian Federation of Natural Language Processing, pp. 40–48.
6. **Kasewa, S., Stenetorp, P., & Riedel, S. (2018).** Wronging a right: Generating better errors to improve grammatical error detection. *EMNLP*, Association for Computational Linguistics, pp. 4977–4983.
7. **Kingma, D. P. & Ba, J. (2015).** Adam: A method for stochastic optimization. *ICLR*.
8. **Nagata, R. & Nakatani, K. (2010).** Evaluating performance of grammatical error detection to maximize learning effect. *COLING*, Coling 2010 Organizing Committee, pp. 894–900.
9. **Napoles, C., Sakaguchi, K., & Tetreault, J. (2017).** JFLEG: A fluency corpus and benchmark for grammatical error correction. *EACL*, Association for Computational Linguistics, pp. 229–234.
10. **Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., & Bryant, C. (2014).** The CoNLL-2014 shared task on grammatical error correction. *CoNLL: Shared Task*, Association for Computational Linguistics, pp. 1–14.
11. **Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018).** Deep contextualized word representations. *NAACL*, Association for Computational Linguistics, pp. 2227–2237.
12. **Peters, M., Neumann, M., Zettlemoyer, L., & Yih, W.-t. (2018).** Dissecting contextual word embeddings: Architecture and representation. *EMNLP*, Association for Computational Linguistics, pp. 1499–1509.
13. **Rei, M. (2017).** Semi-supervised multitask learning for sequence labeling. *ACL*, Association for Computational Linguistics, pp. 2121–2130.
14. **Rei, M., Felice, M., Yuan, Z., & Briscoe, T. (2017).** Artificial error generation with machine translation and syntactic patterns. *BEA*, Association for Computational Linguistics, pp. 287–292.
15. **Rei, M. & Søgaard, A. (2019).** Jointly learning to label sentences and tokens. *AAAI*, Association for the Advancement of Artificial Intelligence.
16. **Rei, M. & Yannakoudakis, H. (2016).** Compositional sequence labeling models for error detection in learner writing. *ACL*, Association for Computational Linguistics, pp. 1181–1191.
17. **Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014).** Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958.
18. **Takase, S., Suzuki, J., & Nagata, M. (2018).** Direct output connection for a high-rank language model. *EMNLP*, Association for Computational Linguistics, pp. 4599–4609.
19. **Taylor, W. L. (1953).** Cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, Vol. 30, No. 4, pp. 415–433.
20. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017).** Attention is all you need. In *NIPS*. Curran Associates, Inc., pp. 5998–6008.

21. **Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016).** Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

22. **Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011).** A new dataset and method for automatically grading ESOL texts. *ACL: Human Language Technologies*, Association for Computational Linguistics, pp. 180–189.

Article received on 06/03/2019; accepted on 15/03/2019. Corresponding author is Masahiro Kaneko.