MIHNet: Combining N-gram, Sequential and Global Information for Text Classification

MIHNet: Combining N-gram, Sequential and Global Information for Text Classification

Yingxin Song[1]

[1]University of Science & Technology, Beijing, Beijing, 100083, China

[*]songyingxin960119@163.com

Abstract. Pre-training language model has achieved amazing results in many NLP tasks. In Particular, BERT (Bidirectional Encoder Representations from Transformers) create a new era in NLP tasks. Despite the success, these model perform well at Global Information but weak on n-gram and Sequential information. In this paper, we conduct exhaustive experiments of classical text classification models upon BERT in text classification task and provide a general guide for BERT+ models. Finally, we propose a new text classification model called MIHNet (Multi-dimension Information Integration using Highway network), which integrates Global, n-gram and Sequential information together and get a better performance. Notably, our model obtains new state-of-the-art results on eight widely-studied text classification datasets.

1. Introduction

Text Classification is a classic and important problem in Natural Language Processing(NLP). The task is to assign predefined categories to a given text sequence. In recent years, neural network models that can make use of text representation have been shown to be effective for text classification, including convolution models [1][3][5], recurrent models [6][8], and attention mechanisms [9].

In Natural Language Processing, pre-trained language representations are beneficial for text classification and other NLP tasks, which can avoid training a new model from scratch. And there are two existing strategies for applying pre-trained language representation to downstream tasks: feature-based and fine-tuning. The featured-based approach including word embedding, such as Word2Vec[11] , Glove[12], Cove[13] and ELMO [14]. The fine-tuning approach, such as BERT[15], OpenAI GPT[16], has brought a new breakthrough in Natural Language Processing(NLP) and become one of the hottest research topics.

In this paper, we investigate how to maximize the utilization of BERT+ models for the text classification task. We design exhaustive experiments to make a detailed analysis of BERT+ models and proposed a new model called MIHNet, which integrate Global, n-gram, Sequential information together to enhance performance further.

The contributions of our paper are as follows:

- We analyse the effect of text length on BERT and propose detailed advice to choose the appropriate length based on the length distribution.
- We propose a variety of scenarios for long texts and analyse the performance of each scenario on THUCNews and IMDB.
- We investigate the performance of several text classification models upon BERT and propose a general solution to use those models upon BERT for different datasets.

- We propose a new model named MIHNet and achieve the state-of-the-art results on seven widely-studied English text classification datasets and one Chinese news classification dataset.

2. Related Work

2.1. Pre-trained Language model

Feature-based approaches that learning widely applicable representations of words has been an active area of research for decades, including Word2Vec[11], Glove[12]. There is a serious problem in those classical word embedding approaches: Word ambiguity problem.

ELMO[14] propose to extract context-sensitive features from deep language model to solve word ambiguity problem and achieve new state-of-the-art results in many tasks.

Recently, the method of pre-trained language models with a large amount of unlabeled data has made a big breakthrough in several natural language understanding tasks, such as OpenAI GPT[16], BERT[15]. BERT is the first fine-tuning based representation model that achieves state-of-the-art results for a range of NLP tasks.

Some recent models explore language model research along different dimension. ERNIE[18] and ERNIE[17] propose to integrate knowledge information into pre-trained language model. MASS[19] and UNILM[20] explore how to use pre-trained language model in Natural Language Generation.

2.2. Text Classification

Over the last few years, neural network-based architectures have achieved state of the art in text classification task. TextCNN [1] and DPCNN[4] develop CNN for capturing the n-gram features and getting the state of the art performance in most text classification datasets. RCNN[30] uses LSTM to capture the global information, compared to CNN. Another popular model, Hierarchical Attention Network(HAN)[26] explicitly models hierarchical information from documents to extract meaningful features, incorporating word-level and sentence-level information to classify documents.

3. Model

3.1. BERT for Text Classification

When we adapt BERT fine-tuning strategy to text classification task, we always get a better result than classic text classification models in most datasets. To improve model performance further, a proper strategy combined with BERT is desired. In this paper, we look for the proper combination strategy in the following models.

BERT takes an input of a sequence of no more than 512 tokens and outputs the representation of the sequence. In the section, the input sentence is like that: [CLS] + sentence, where [CLS] contains the special classification embedding. We assume that the output of BERT contains two representations, as follows:

$$h, O = BERT(S) \tag{1}$$

The output of BERT consists of two parts, where h $\in R^d$ is the final hidden state of the first token [CLS] and O $\in R^{|S|\times d}$ is the context representation matrix of the sequence. Beyond on this, we conduct a detailed experiment that explore the performance of different models.

1) BERT-base: In this model, we take the final hidden state as the representation of the whole sequence. A simple Softmax classifier is added to predict the probability of label c:

$$p(c \mid h) = soft\max(Wh) \tag{2}$$

where W is the task-specific parameter matrix. We fine-tune all the parameters from BERT as well as W jointly by maximizing the log-probability of the correct label.

2) BERT + TextCNN: In this model, we take all the final hidden state matrix O of the sentence tokens as the representation of the whole sequence. Then we apply TextCNN to extract the n-gram features to get the final representation and feed it into Softmax classifier.

$$h_{cnn} = TextCNN(O) \tag{3}$$

$$p(c \mid h_{cnn}) = softmax(W_{cnn} h_{cnn}) \tag{4}$$

where W_{cnn} is the specific parameter matrix for BERT + TextCNN.

3) BERT + LSTM: We apply Bi-LSTM to enhance the sequential information. Then, we take the last token hidden state of LSTM as the final representation and feed it into Softmax classifier:

$$h_{lstm} = LSTM(O) \tag{5}$$

$$p(c \mid h_{lstm}) = \text{softmax}(W_{lstm} h_{lstm}) \tag{6}$$

where W_{lstm} is the specific parameter matrix for BERT + LSTM.

4) BERT + RCNN: We use Bi-LSTM to get the contextual representation of each token: c_i^l, c_i^r. To get the final representation of token, we concat the output of Bi-LSTM and the output of BERT like this:

$$x_i = [c_i^l; o_i; c_i^r] \tag{7}$$

We then apply a tanh activation function and max-pooling to get the final representation of the input sentence:

$$h_i = tanh(W_h x_i + b_h) \tag{8}$$

$$y = max_{i=1}^{n} h_i \tag{9}$$

Finally, we use Softmax classifier to get the probability of each class:

$$p(c \mid y) = \text{softmax}(W_{rcnn} y) \tag{10}$$

5) BERT + DPCNN: We use deep convolution to capture deep semantic information and feed it into Softmax classifier.

$$h_{rcnn} = DPCNN(O) \tag{11}$$

$$p(c \mid h_{rcnn}) = \text{softmax}(W_{rcnn} h_{rcnn}) \tag{12}$$

3.2. MIHNet

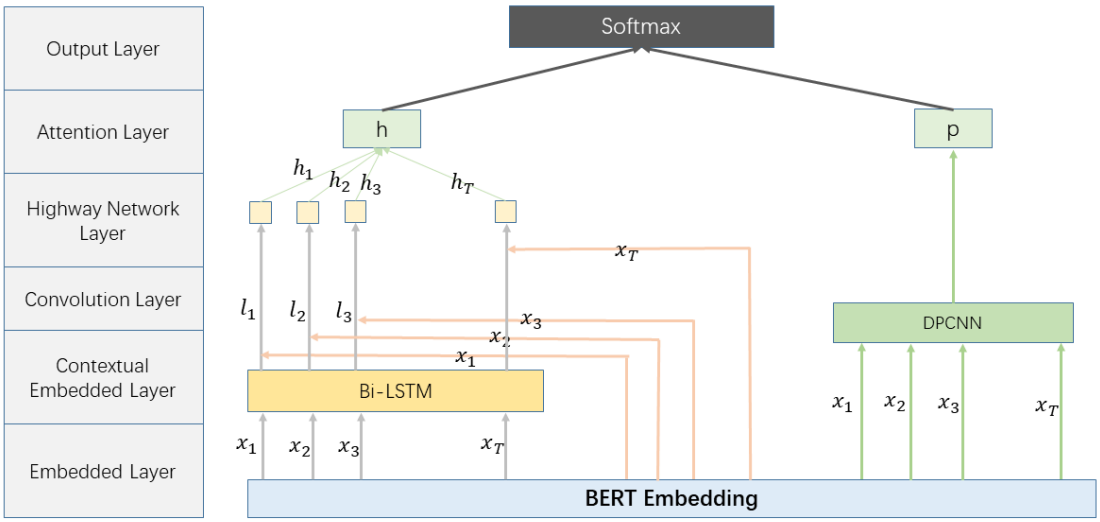

Figure 1. Multi-dimension Information Intergration using Highway Network

For text classification task, global information, n-gram information and Sequential information can play different roles for different datasets. MIHNet is designed to achieve the best performance by reasonably combining three information from different dimensions.

MIHNet is a hierarchical multi-stage process and consists of five layers:

Embedding Layer. Embedding Layer is responsible for mapping each word to a high-dimensional vector space. Let $\{x_1,...x_T\}$ represent the words in the input context paragraph. We obtain the context embedding of each word using BERT[16].The output of this layer is a sentence of d-dimensional vectors, or more conveniently, one matrices: $X \in R^{d\times T}$ for the context.

Contextual Embedding Layer. We use a Bi-LSTM to model the temporal interactions between words. Hence we obtain $L \in R^{2l\times T}$ from the context word vectors X .

Highway Network Layer. The concatenation of the word and contextual embedding vectors is passed to a two-layer Highway Network [27]. The output of the Highway Network is a $d+2l$ dimensional vectors: $H \in R^{(d+2l)\times T}$ for the context. Finally, we use Attention mechanism to get the final representation of the information : $h \in R^{d+2l}$.

Convolution Layer. Convolution layer is responsible for capturing n-gram information from context words. Unlike previously shallow convolution [1], DPCNN [4] can capture n-gram information of different sizes without manually setting the convolution kernel size. The output of convolution layer is a c dimensional vector: $p \in R^c$ for the context.

Output Layer. We connect the output of Highway Network Layer to the output of the Convolution Layer and feed it into the softmax classifier.

$$p(y \mid x) = \text{softmax}(W^T[h; p]) \tag{13}$$

where $W \in R^{d+2l+c}$ is a trainable weight vector.

4. Experiments

4.1. Datasets

We explore all approach on eight widely-studied datasets. These datasets have varying numbers of documents and varying document lengths, covering three common text classification tasks: sentiment analysis, question classification, and topic classification.

Sentiment analysis. For sentiment analysis, we use the binary film review IMDB dataset, two-class version of the SST dataset and five-class version of the Yelp review dataset.

Question classification. For question classification, we evaluate our method on Yahoo! Answers dataset.

Topic classification. For topic classification, we use large-scale AG's News and DBPedia as English datasets. To test the effectiveness of BERT for Chinese text, we create the Chinese dataset using the THUCNews corpus.

Data analysis We analyse the distribution of all eight datasets over different lengths. As show in the table below:

Table 1. Data distribution percentage (%) statistics according to text length.

Dataset	**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**
IMDB	0.01	0.09	0.27	0.49	0.63	0.72	0.75	0.82	0.86	0.89
SST-2	0.99	1	-	-	-	-	-	-	-	-
Yelp	0.18	0.43	0.61	0.73	0.82	0.87	0.90	0.93	0.95	0.96
Yahoo	0.78	0.92	0.96	0.98	0.99	1	-	-	-	-
AG's_News	0.91	0.99	0.99	1	-	-	-	-	-	-
DBPedia	0.45	0.99	0.99	1	-	-	-	-	-	-
THUCNews	0.001	0.57	0.10	0.14	0.18	0.22	0.25	0.30	0.34	0.42

4.2. Exp-I: Investigating text length

For text classification, proper text length settings can reduce training time while achieving optimal results. In the experiment, we limited the text length to test the performance on multiple datasets.

Table 2 shows the effectiveness of different lengths. Longest length achieves the best performance on all datasets due to the Padding Mask mechanism in Transformer [9]. On the other hand, as the length range is expanded, the performance improvement is slower and slower until the growth is stopped within a certain length.

Table 2. Test accuracy rates (%) for different text length using BERT

Dataset	50	100	150	200	250	300	350	400	450	500
IMDB	81.57	86.79	89.82	91.17	92.14	94.80	93.28	93.67	93.77	93.99
SST-2	92.54	93.46	-	-	-	-	-	-	-	-
Yelp_pol	70.23	86.38	92.48	94.82	95.36	96.13	96.84	97.20	97.53	97.76
Yelp_full	56.59	64.41	66.35	66.76	68.14	69.12	69.33	69.65	69.85	69.97
Yahoo	75.96	75.85	76.19	76.20	76.33	-	-	-	-	-
AG's News	94.43	94.18	94.27	94.28	-	-	-	-	-	-
DBPedia	99.12	99.15	99.16	99.25	-	-	-	-	-	-
THUCNews	96.15	96.52	97.06	97.79	97.06	97.26	97.20	97.23	97.29	97.35

4.3. Exp-II: Investigating Long text

The main question of fine-tuning BERT in text classification is how the performance of the model changes as the text length window expands. Since the maximum sequence length of BERT is 512. How to processing the text with a length larger than 512 is vital. We try the following way for dealing with long articles different from [24].

We first make a clause for the article and then connect the sentences to form a paragraph based on the length of the sentence. In the end, K paragraphs are formed, each with a text length of no more than 500 in THUCNews and 100 in IMDB.

We feed the paragraphs into BERT to obtain the representation separately, and then use Mean-pooling, Max-pooling or Attention mechanism to combine the representations of the all fractions, respectively.

Table 3. Test Accuracy rates (%) on IMDB and THUCNews datasets

Model	IMDB	THUCNews
BERT + Attention	93.818	97.495
BERT + Max-pooling	93.979	97.563
BERT + Avg-pooling	93.688	97.118

As show in above table, the performance on IMDB is even worse than the performance of 500 and better than the performance of 100 due to contextual fragmentation problem. However, the performance on THUCNews is better because the model is able to read the full text rather than reading a part of the article.

4.4. Exp-III:Models

We first combine some classical text classification models with BERT and analyse the effects of different models, as described above in 3.1. We then analyse the performance of these models with our model MIHNet on different datasets. The performance are as follows:

Table 4. Test Accuracy rates (%) on different datasets using different models

Model	IMDB	SST-2	Yelp pol	Yelp full	Yahoo	AG's News	DBPedia	THUC
BERT	93.99	93.46	97.76	69.97	76.33	94.28	99.25	97.35
BERT+ TextCNN[1]	97.73	92.95	97.14	69.03	76.00	94.42	99.37	97.58
BERT+ LSTM[6]	94.28	93.84	97.77	69.76	76.40	94.38	99.44	97.41
BERT+ RCNN[27]	94.62	94.28	98.26	70.11	76.47	94.74	99.48	98.25
BERT+ DPCNN[4]	93.41	93.22	97.79	69.92	76.51	94.99	99.43	98.17
MIHNet	94.91	94.37	98.27	70.17	76.54	95.02	99.50	98.59

As shown in the above table, compared with BERT-base, the performance improvement of different model is not a lot, mainly because the simplicity of text classification. Text classification relies heavily on shallow semantic features, while simple linear combinations of shallow semantic features can get good performance.

On the other hand, for Sentiment analysis and Question classification, BERT+RCNN tends to get the best results, while for Topic classification, BERT+DPCNN can get the best performance.

Finally, our model MIHNet achieved the best performance on all eight datasets, which proves that using Highway Network to fuse different granularities of information can capture sentence contextual information more comprehensively.

5. Conclusion and Future work

In this paper, we first explore the impact of text length on BERT and make recommendations for different situations. Then, we conduct extensive experiments to investigate the different BERT+ models to fine-tuning BERT for the text classification task. Finally, we propose a new model named MIHNet which get better performance on most text classification datasets.

One direction of future work involves comparing different pre-trained model like XLNet [34], ERNIE [35] with BERT in text classification task and investigate whether NIHNet is still effective.

Acknowledgments
This research was supported by my teacher GuangPing Zeng.We thank the reviewers for their helpful comments.

References

[1] Meng Kalchbrenner N, Grefenstette E, Blunsom P. (2014) A Convolutional Neural Network for Modelling Sentences. J. Eprint Arxiv, 2014, 1.
[2] Zhang X, Zhao J, Lecun Y. (2015) Character-level Convolutional Networks for Text Classification. C. Neural Information Processing Systems. MIT Press.
[3] Conneau A, Schwenk H, Barrault L, et al. (2016) Very Deep Convolutional Networks for Natural Language Processing. J.
[4] Johnson R, Zhang T. (2017) Deep Pyramid Convolutional Neural Networks for Text Categorization. C. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).
[5] Carin L. (2017) Deconvolutional Paragraph Representation Learning. J.
[6] Liu P, Qiu X, Huang X. (2016) Recurrent Neural Network for Text Classification with Multi-Task Learning. J.
[7] Yogatama D, Dyer C, Ling W, et al. (2017) Generative and Discriminative Text Classification with Recurrent Neural Networks. J.
[8] Seo M, Min S, Farhadi A, et al. (2017) Neural Speed Reading via Skim-RNN. J. 2017.
[9] Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention Is All You Need. J.
[10] Lin Z, Feng M, Santos C N D, et al. (2017) A Structured Self-attentive Sentence Embedding. J.
[11] Mikolov T, Sutskever I, et al. (2013) Distributed Representations of Words and Phrases and their Compositionality. J. Advances in Neural Information Processing Systems, 26:3111-3119.
[12] Jeffrey Pennington, Richard Socher, et al.. (2014). Glove: Global vectors for word representation. J. In Empirical Methods in Natural Language Processing, pages 1532–1543.
[13] Mccann B, Bradbury J, Xiong C, et al. (2017) Learned in Translation: Contextualized Word Vectors. J.
[14] Peters M E, Neumann M, Iyyer M, et al. (2018) Deep contextualized word representations. J.
[15] Devlin J, Chang M W, Lee K, et al. (2018) BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding. J.
[16] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. (2018). Improving language understanding with unsupervised learning. Technical report. J.
[17] Zhang Z, Han X, Liu Z, et al. (2019) ERNIE: Enhanced Language Representation with Informative Entities. C. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.
[18] Sun Y, Wang S, Li Y, et al. (2019) ERNIE: Enhanced Representation through Knowledge Integration. J.
[19] Song K, Tan X, Qin T, et al. (2019) MASS: Masked Sequence to Sequence Pre-training for Language Generation. J.
[20] Dong L, Yang N, Wang W, et al. (2019) Unified Language Model Pre-training for Natural Language Understanding and Generation. J.
[21] Maas A L, Daly R E, Pham P T, et al. (2011) Learning Word Vectors for Sentiment Analysis. C. The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference. Association for Computational Linguistics.
[22] Richard Socher, Alex Perelygin, et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. J. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642.

[23] VOORHEES, Ellen M. (2001) The TREC question answering track. J. Natural Language Engineering, 7(04):361-378.
[24] Sun C, Qiu X, Xu Y, et al. (2019) How to Fine-Tune BERT for Text Classification? J.
[25] Zhang Y, Wallace B. (2015) A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. J. Computer Science.
[26] Pappas N, Popescu-Belis A. (2017) Multilingual Hierarchical Attention Networks for Document Classification. J.
[27] Srivastava R K, Greff K, Schmidhuber, Jürgen. (2015) Highway Networks. J. Computer Science.
[28] Yang Z, Dai Z, Yang Y, et al. (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. J.
[29] Sun Y, Wang S, Li Y, et al. (2019) ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. J.
[30] Lai S, Xu L, Liu K, Zhao J. (2015) Recurrent Convolutional Neural Networks for Text Classification. J.