A Comparison of BERT and ELMo on Text Classification Tasks

Xin Peng, Xinyan He, Xinyu Kang
Department of Computer Science
University of Toronto
{xin.peng, xinyan.he, xinyu.kang}@mail.utoronto.ca

Abstract

With context-aware word embeddings, BERT and ELMo both have achieved impressive performance on various NLP tasks. In recent years, BERT has become one of the most popular state-of-the-art models in this area, whereas ELMo is much less used. However, little work has shown how the two models' performance differs on specific tasks. In this paper, we address this problem. We first present the structures of BERT and ELMo. Then we focus on text classification tasks and test their performance on a variety of scenarios. We found that fine-tuned BERT models outperform ELMo in all scenarios introduced in this paper.

Github Repo for data and code: `https://github.com/XinyanHe/CSC413-Final-Project`

1 Indroduction

Due to the huge vocabulary size of human languages, natural language processing tasks often demand training on large corpora to have satisfactory results. This process is extremely time-consuming. In recent years, pre-trained models are introduced where they are trained on large-scale datasets, and the resulting word representations can be reused and fine-tuned for specific tasks [6]. These models are now commonly used to solve a variety of NLP problems [6]. Among them, BERT and ELMo both have achieved state-of-the-art performance on generating context-aware word representations.

Unlike the bidirectional LSTM layers used by ELMo [7], BERT adopts Transformers in its architecture and uses a masked language modelling approach to further learn the context around each word [2]. It is shown that such a structure and method can achieve incredible results on many NLP tasks [2]. The NLP community often considers BERT as a milestone in this area, and it has become extremely popular in recent years. However, little work has shown enough evidence that BERT should always be the default approach [1, 4], some researchers even questioned its superiority on some tasks [3, 5]. It is important to have empirical evidence to choose the suitable model for each problem. We focus on text classification and investigate the performance of BERT and ELMo, trying to provide some insight on which model is better for this task. In this paper, we evaluate the performance of BERT of ELMo on text classification by carrying out three different experiments.

2 Related Work

There are limited comparisons between BERT and ELMo. Researchers have evaluated the two models in the field of Biomedical Natural Language Processing [1] and concluded that the BERT model pre-trained on biomedical datasets performs better than most state-of-the-art models. However, some results from other tasks disagree. Rodina et al. [4] investigated BERT and ELMo in semantic change detection for Russian corpora. Their results showed that there is no significant difference between the performance of the two models to conclude which one is better.

Preprint. Under review.

Experiments also disagree on whether BERT can always achieve better performance than traditional models. When it comes to text classification tasks, González-Carvajal et al. [3] have compared BERT with traditional TF-IDF and suggested that BERT should be the default option for NLP tasks. However, there are also experiments suggesting the performance of the models depends on the tasks and the corpus [5], where it is found that a simple bidirectional LSTM can perform much better than BERT when the training dataset is small.

Based on these reasons, further comparisons are needed between BERT and ELMo. We will test the two models on a series of new datasets to determine whether there should be a significant preference towards one model when it comes to text classification tasks.

3 Methodology

3.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a contextualized word representation model that is designed to pre-train deep bidirectional representations [1, 2]. There are two steps in BERT's framework: *pre-training* and *fine-tuning*[2]. In pre-training, BERT is trained on unlabeled large data using different pre-training tasks. It can be fine-tuned using one additional output later for a specific task [2]. It is first initialized with the pre-trained parameters, which are fine-tuned later based on the labelled data for the specific task [3].

3.1.1 Pre-training BERT

BERT is pre-trained using two unsupervised tasks, Masked LM and Next Sentence Prediction (NSP). In Masked LM, some input tokens are masked randomly, and the task is to predict those masked tokens. Next Sentence Prediction consists of predicting whether sentence B follows sentence A, which helps in understanding the relationship between two sentences [2].

In this paper, we initialized BERT with pre-trained BERT provided by Huggingface. It is trained on lower-cased English text, with 12 layers, 768 hiddens, 12 heads, and 110M parameters.

3.1.2 Fine-tuning with BERT

The self-attention mechanism in the Transformer allows BERT to model many downstream text-mining tasks, and task-specific inputs and outputs can be simply plugged into BERT and then we can fine-tune all the parameters [2].

For text classification task, we used BertForSequenceClassification provided by Huggingface for fine-tuning. It is the normal BERT model with a linear layer on top of the pooled output for classification. When we plug the inputs into the model, all the parameters and the additional untrained layer would be trained on the specific text classification task.

3.2 ELMo

Similar to BERT, ELMo is a deep contextualized word representation model that is pre-trained on a large corpus, and can be easily fine-tuned for specific tasks [7]. Its structure (Appendix C) is rather simple comparing to BERT. With the raw text input as its first layer, x, the encoder stacks several bidirectional RNN layers, say L layers, where the intermediate word vectors output from each layer is fed to the next layer as input. The hidden layers of both forward and backward passes of all RNN are stored as the internal representation. That is, for each input token x_k, its representation R_k is the set

$$R_k = \{x_k, \vec{h}_{k,j}, \overleftarrow{h}_{k,j} | j = 1 \dots L\} = \{h_{k,j} | j = 0 \dots L\}$$

In the last layer, it generates task-specific softmax weights for linearly combining the internal representation representation. For input token x_k, its ouput for the current *task* would be

$$\text{ELMo}_k^{task} = \lambda^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}$$

where s^{task} are the softmax-normalized weights for $task$.

4 Experiments

We perform three classification tasks including sentiment analysis, topic labeling, text quality analysis. The following three datasets are used: IMDB Dataset, AG News Classification Dataset, Stack Overflow Questions Dataset. Table 1 provides detailed information about each dataset.

We preprocessed the datasets as follows. For BERT, we constructed a tokenizer using “bert-base-uncased" version of BertTokenizer. Then we converted training and testing text to the encoded form, and limited each text input to the first 256 tokens. After getting the encoded form, we used DataLoader to combine dataset and sampler. And for BERT's implementation, we directly used pre-trained BERT model with version “bert-base-uncased" from Huggingface Transformer. It is a pretrained model on English using a masked language modeling (MLM). When prepossessing datasets for ELMo, we encoded the label first before doing the training and testing steps when there are multiple classes labels. Then using the pre-trained ELMo model which was trained on 1 Billion Word Benchmark from Tensorflow-Hub to create an embedding layer. Next constructing a 2-Dense model adding the embedding layer we created before and use the model to make prediction.

Table 1: A summary of the four datasets, including the number of classes, the number of training set entries, the number of testing set entries.

Dataset	Number of Classes	Training	Testing
IMDB Dataset	2	1200	300
AG News Classification Dataset	4	4000	1000
Stack Overflow Questions Dataset	3	4000	1000

4.1 IMDB Sentiment Analysis

We used IMDB Dataset from the following website, which contains movie reviews from IMDB, labeled by two sentiment classes: positive and negative.

We compare the performance of BERT and ELMo primarily using accuracy scores, since the training and testing datasets are all balanced. We also report precision, recall and F1 score:

- **Accuracy** refers to the ratio of correctly classified instances over the total number of instances.
- **Precision** refers to the ratio of true positives T_p over the number of true positives T_p plus the number of false positives F_p.
- **Recall** refers to the ratio of true positives T_p over the number of true positives T_p plus the number of false negatives F_n.
- **F1 Score** is defined as follows:

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Table 2: Performance on the IMDB Sentiment Analysis

Model	Accuracy	Precision	Recall	F1 Score
BERT	0.90	0.88	0.93	0.90
ELMo	0.84	0.84	0.84	0.84

In Table 2, the classification performance of the two models is displayer. We have observed that BERT has outperformed ELMo with respect to all metrics.

4.2 AG News Topic Labeling

This experiment deals with topic labeling. The AG News Classification Dataset we used here is from the following website. It contains news articles gathered from more than 2000 news sources by

ComeToMyHead. The task to solve here is a multiclass classification, and each news is classified into four topics, including World, Sports, Business and Science/Technology.

Since this task involves multiple classes, we report the precision, recall, and F1 score for each class (see Appendix A for precision, recall, and F1 score for each class).

Table 3: Performance on the AG News Topic Labeling

Model	Accuracy
BERT	0.92
ELMo	0.91

The results show that BERT performs slightly better than ELMo in this task.

4.3 Stack Overflow Questions Quality Analysis

This final experiment deals with the Stack Overflow Questions written in English. The dataset is from the following website. This dataset is collected from Stack Overflow from 2016 to 2020. THE questions are classified into three categories: LQ_CLOSE, LQ_EDIT, HQ, where LQ_CLOSE denotes low-quality posts that were closed by the community without a single edit, LQ_EDIT denotes low-quality posts with a negative score, and multiple community edits, but they still remain open after those changes, and HQ denotes high-quality posts with a total of 30+ score and without a single edit.

Results of this experiment are given in table 4, and Appendix B.

Table 4: Performance on the Stack Overflow Questions Quality Analysis

Model	Accuracy
BERT	0.90
ELMo	0.79

5 Conclusion

In this work we compare the BERT's and ELMo's performance on text classification tasks. We have shown that BERT has outperformed ELMo in all three different text classification scenarios we have introduced, adding evidence of BERT's superiority in NLP problems w.r.t. text classification. A direction for future work is to further investigate BERT and ELMo's performance on domain specific text classification task, and non-English text classification task. Such work could potentially further compare the two model's performance in a wider range of text classification task.

6 Contributions

Xin Peng introduced the methodology for BERT model and designed and conducted all experiments for BERT; Xinyan He designed and conducted all experiments for ELMo; Xinyu Kang introduced the methodology for ELMo model, reviewed related work and helped in the design of experiments. Xin Peng, Xinyan He, and Xinyu Kang edited paper.

7 References

[1] Peng, Y., Yan, S., & Lu, Z. (2019). Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*.

[2] 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*.

[3] González-Carvajal, S., & Garrido-Merchán, E. C. (2020). Comparing BERT against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

[4] Rodina, J., Trofimova, Y., Kutuzov, A., & Artemova, E. (2020). ELMo and BERT in semantic change detection for Russian. *arXiv preprint arXiv:2010.03481*.

[5] Aysu Ezen-Can (2020). A Comparison of LSTM and BERT for Small Corpus. *arXiv preprint arXiv:2009.05451*.

[6] Zhou, M., Duan, N., Liu, S., & Shum, H. (2020). Progress in Neural NLP: Modeling, Learning, and Reasoning. *Engineering, 6(3)* , 275-290. doi:10.1016/j.eng.2019.12.014

[7] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, M., Lee, K., &, Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.0536*

Appendix A BERT and ELMo's performance on the AG News Topic Labeling

Table 5: BERT Performance on the AG News Topic Labeling

Model	Class	Precision	Recall	F1 Score	Accuracy
BERT	World(1)	0.92	0.93	0.93	0.92
	Sports(2)	0.96	0.96	0.96	
	Business(3)	0.93	0.86	0.90	
	Science/Technology(4)	0.89	0.95	0.92	

Table 6: ELMo Performance on the AG News Topic Labeling

Model	Class	Precision	Recall	F1 Score	Accuracy
ELMo	World(1)	0.96	0.86	0.91	0.91
	Sports(2)	0.93	1	0.96	
	Business(3)	0.87	0.87	0.87	
	Science/Technology(4)	0.89	0.93	0.91	

Appendix B BERT and ELMo's performance on the Stack Overflow Questions Quality Analysis

Table 7: BERT Performance on the Stack Overflow Questions Quality Analysis

Model	Class	Precision	Recall	F1 Score	Accuracy
ELMo	LQ_CLOSE(0)	0.85	0.85	0.85	0.90
	LQ_EDIT(1)	1.00	1.00	1.00	
	HQ(2)	0.87	0.87	0.87	

Table 8: ELMo Performance on the Stack Overflow Questions Quality Analysis

Model	Class	Precision	Recall	F1 Score	Accuracy
ELMo	LQ_CLOSE(0)	0.74	0.88	0.80	0.79
	LQ_EDIT(1)	0.78	0.56	0.65	
	HQ(2)	0.88	0.94	0.91	

Appendix C ELMo

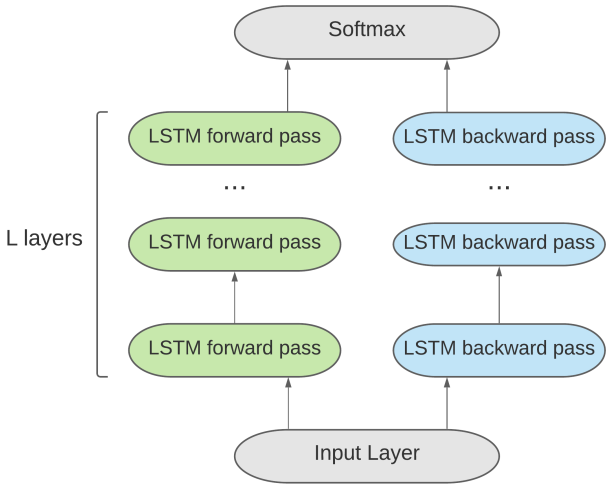

Figure 1: The structure of ELMo