Utilizing Enhanced Deep Contextualized Word Embeddings for Downstream Tasks

Stanford CS224N Default Project

Renaldo Venegas
Department of Mathematics
Stanford University
renaldov@stanford.edu

Ethan Yuen
Department of Computer Science
Stanford University
ehyuen@stanford.edu

Abstract

The Bidirectional Encoder Representations from Transformers, more commonly known as BERT, is a Language model which uses transformers to produce contextualized word embeddings. Deep bidirectional models have been shown to encode meanings in the hidden layers of BERT, which gave rise to ELMo: word representations which incorporate the hidden states of a neural network. We aim to improve on this paradigm by allowing a larger variability in these deep contextualized word embeddings for higher semantic understanding and learnability. We take inspiration from ELMo, producing a new form of contextualized embeddings which are given by trainable parameters.

1 Key Information to include

- Mentor: Timothy Dai
- Sharing project: No
- Contributions: Ethan implemented the minBERT and the baseline, helped with training, and wrote parts of the report. Renaldo implemented the contexual embeddings, multitask training, ran most of the training and testing, and wrote parts of the report.

2 Introduction

The introduction of the Transformer architecture in 2017 (Vaswani et al. (2017)) gave way to many groundbreaking improvements in the field of Natural Language Processing. One of the first language models to encapsulate this was the Bidirectional Encoder Representations from Transformers (BERT) in 2018, (Devlin et al. (2018)) which showed state of the art performance on Natural Language understanding tasks through pre-training a bidirectional model with transformers to capture context windows and semantic meaning from both directions.

However, despite the improvements, it is still difficult to train models which aim to perform well on multiple different tasks. We aim to incorporate techniques in the literature to fine-tune our BERT and create extended word embeddings on to perform three different NLP tasks: Sentiment Analysis, Paraphrase detection, and Semantic text similarity.

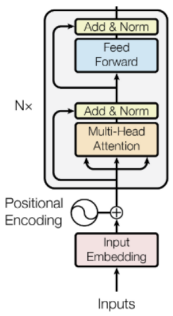

Figure 1: Encoder layer of a BERT

3 Related Work

3.1 ELMo

ELMo (Embeddings from Language Models) is a novel approach for word representation in natural language processing (NLP) tasks which generates word representations that are deeply contextualized, meaning the representation of a word depends on the entire sentence in which it appears, whereas traditional word embeddings like Word2Vec and GloVe assign a single vector to each word, regardless of its context. The paper on this by Peters et al. Peters et al. (2018) finds that hidden states within a bidirectional language model such as BERT carry semantic meaning separate from the final encoding of a word. Thus, our paper incorporates a linear combination of the hidden layers into the word embeddings to attempt to improve on this paradigm.

3.2 Multitask Training

Multitask training is a novel framework designed to enhance news recommendation systems by leveraging BERT's ability to capture rich semantic information and applying multitask learning to simultaneously address related tasks such as news classification, user click prediction, and recommendation. In the paper "Mtrec: Multi-task learning over bert for news recommendation" (Bi et al. (2022)), BERT is used to generate contextualized embeddings for news articles, which are then processed by task-specific layers. By sharing representations across tasks, the model learns more robust features, leading to better generalization and improved recommendation accuracy. The end-to-end training optimizes for multiple tasks jointly, resulting in superior performance compared to traditional models and single-task approaches.

3.3 Sentence-BERT

In the paper "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," (Reimers and Gurevych (2019)) the authors present a method to derive semantically meaningful sentence embeddings by fine-tuning BERT using a Siamese network architecture. Traditional BERT embeddings are not directly suitable for tasks requiring sentence-level semantics due to high computational costs, so sentence-BERT modifies BERT by employing a Siamese network that processes sentence pairs, enabling efficient computation of similarity scores between sentences. The approach involves fine-tuning BERT with a combination of classification and regression objectives on various datasets. After obtaining the sentence embeddings, cosine similarity is used to measure the similarity between two sentences by calculating the cosine of the angle between their embedding vectors. This method effectively captures semantic similarities, allowing SBERT to significantly improve performance

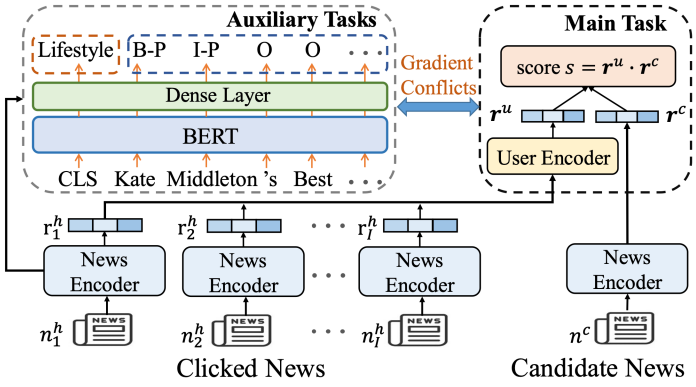

Figure 2: Sentence-BERT architecture with cosine similarity

on semantic textual similarity tasks and various sentence embedding applications while reducing computational overhead.

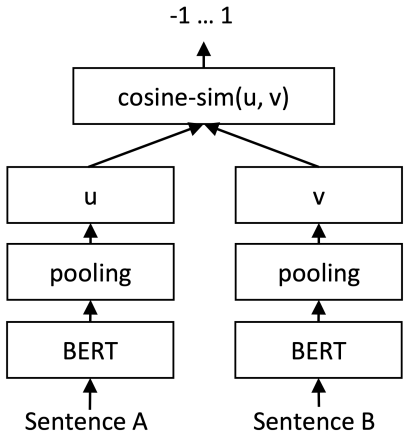

Figure 3: Sentence-BERT architecture with cosine similarity

3.4 Fine-tuning

In the paper "How to Fine-Tune BERT for Text Classification?" (Sun et al. (2020)), the authors present methods to fine-tune the learning rate and remedy overfitting. Lower layers of the BERT model are said to contain more general information, and thus ought to be fine-tuned with different learning rates. Parameters θ are split into $\{\theta^1, \cdots, \theta^L\}$, where θ^l contains the parameters of the lth later of BERT. This gives us the following update rule:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^t} J(\theta) \tag{1}$$

where the learning rate of the lth layer η^l is calculated via $\eta^{k-1} = \xi \cdot \eta^k$, with decay factor $\xi \leq 1$. Assigning lower layers a lower learning rate ($\xi = 0.95$) is found to have the best performance on the IMDb dataset. The paper then examines further pre-training possibilities, exploring within-task, in-domain, and cross-domain pre-training, all of which are found to be generally beneficial, though within-task pre-training seems to have detrimental effects on certain datasets. In addition, a multi-task

learning approach which extends the multi-task deep neutral network model by incorporating BERT as its shared text encoding layers is outlined, an approach which we pursue in our model

4 Approach

4.1 Expanding on ELMo

The approach of ELMo, Embeedings from Language Models, was the following. They found that the hidden states within a bidirectional language model such as BERT carry semantic meaning separate from the final encoding of a word. Thus, this paper incorporated a linear combination of the hidden layers into the word embeddings to produce better results for tasks such as sentiment analysis. The extended representation is represented by $[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$ where x_k is the word embedding of the kth token and

$$\mathbf{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$

where γ is a trainable parameter, L is the number of layers, s are the softmax-normalized weights, and $h_{k,j}^{LM}$ is the hidden layer of token k at layer j.

We added higher variance to this contextualized word embedding, noting that for some sentence, it may be that certain hidden layers represent the meaning of some tokens better and consequently perform better on NLP tasks. Instead of forcing the linear combination of the hidden layers to be taken over the softmax, our original approach was make each of these weights individual, trainable parameters, which we normalize with a softmax to prevent us from overscaling the embedding. More formally, we have

$$\mathbf{b}_k^{task} = \gamma \sum_{j=0}^{L} \gamma_j h_{k,j}^{LM}$$

and we pass in $[\mathbf{x}_k; \mathbf{b}_k^{task}]$ as contextual representations, similar to ELMo.

In regards to ELMo, the γ variable from the original ELMo equation is said to be "of practical importance to aid optimization, due to the different distributions between the biLM internal representations and the task specific representations." It is also said by Peters et al. that "without this parameter, the last-only case performed poorly (well below the baseline)" Peters et al. (2018). Although we believe there is potential for stronger embeddings with more expressive power to come out of our approach, we also recognize that the additional variance may cause too different of a distribution from the task specific representation, and could lead to worse results than desired.

4.2 Multitask fine-tuning

Our team had to tackle another question: how do we train a model focused on multiple tasks at once? Doing so naively was our first approach, training on one task, after the other. However, this proved to not produce good results. We instead referred to a common method, which adds the losses of the multiple tasks: Bi et al. (2022)

$$\mathcal{L} = \mathcal{L}_i + \mathcal{L}_j$$

would be the loss at a certain training step, where $\mathcal{L}_i$ represents the loss of task i.

4.3 Cosine Similarity/Binary Cross entropy loss

Although the sst dataset uses cross-entropy in order to measure the loss, in order to fit the binary nature of paraphrase classification and comparing by similarity, we employ binary cross entropy loss and cosine similarity loss, respectively.

5 Experiments

5.1 Data

The datasets we used for training and evaluation of the minBERT model were the Stanford Sentiment Treebank (SST) dataset, which consisted of 11.855 single sentences from movie reviews parsed into 215,154 unique phrases, each of which was labeled according to their sentiment, and the CFIMDB dataset, which consisted of 2,434 highly polar movie reviews, each labeled negative or positive. The only task which was trained and evaluated on minBERT was sentiment classification.

The datasets we used for training and evaluation of the full model were the same SST dataset for sentiment analysis, the Quora dataset, which consisted of 404,298 question pairs with labels indicating whether particular instances are paraphrases of one another, for paraphrase detection, and the SemEval STS dataset, which consisted of 8.628 different sentence pairs labeled according to their similarity, for semantic textual analysis.

5.2 Evaluation method

We evaluated our method by expanding on the provided `evaluation.py` file. Instead of working directly with the logit, we applied the sigmoid function for non-linearlity and normalization, and scaled to get a range of similarity scores from 0 to 5. By doing so, we gained predictions we were able to directly compare with the source files to evaluate accuracy.

5.3 Experimental details

One issue we ran into when implementing our approach was dataset size: the Quora dataset was much greater in magnitude in comparison to the other two. This was not a problem with our native approach of training for one task one after another, but it became an issue when needing data from each source at once to calculate the total loss. We decided to use all the data, and resorted to cycling through the remaining data until all three sources were exhausted. Although this resulted in longer

5.4 Results

Below are the results of our experiments, listing the performance on each respective dev sets, where CE denotes contextual embeddings, MTL denotes multi-task learning, and lr denotes learning rate.

	CE, lr = 1e-5	MTL, lr = 1e-5	MTL, CE, lr = 1e-5	MTL, CE, lr = 4
Sentiment Accuracy	0.264	0.225	0.254	0.272
Paraphrase Accuracy	0.484	0.630	0.368	0.370
Similarity Accuracy	-0.022	-0.08	-0.030	-0.042

Our final test leaderboard submission was the mulititask learner without contextual embeddings, using a learning rate of 1e-5. Unfortunately, the embedding seems to only worsen the performance, and we got worse results than expected in general.

6 Analysis

We suspect that the usage of the trainable γ parameter being neccessary in the original ELMo embeddings is the same reason that our embeddings were not effective in the natural language tasks: there was too much variance and as a result the embeddings deviated from the original semantic meaning. This is supported by the fact that not having the contextual embedding gave us our best performance. We had originally noted this as a possibility, as the addition of a these many trainable parameters is a strong tool for possibly increasing expressive power, but also can transform the loss landscape to make it much more volatile.

Additionally, we noted that the lack of convergence could potentially be as a result our learning rate being too low: When we increase the number of dimensions by so much, travelling with the gradient with a learning rate meant for half the number of dimensions could have caused our model to not converge. Thus, we tested with a learning rate of 0.0004, as outlined in the original ELMo paper, which did not improve our performance.

7 Conclusion

Although the hidden layers of Bidirectional models like BERT do contain useful semantic meaning and information, when working with sensitive word embeddings, adding extra parameters to already pre-trained embeddings may have the opposite effect. It is more beneficial to take a more intentional weighting of the hidden layers, as shown by the performance of ELMo vectors and advanced fine-tuning techniques which decreases the weight placed on each layer. Although there is still potential on tuning the best weights for these layers, it may be more productive to focus on less computationally expensive paradigms which perform better on Natural Language tasks.

8 Ethics Statement

8.1 Representation and bias

The main ethical concern with our project lies in representativeness - insufficient or unequal representation of data. More specifically, we are likely to encounter inherent biases in our training data, by virtue of our datasets coming mostly from English-language websites (e.g., IMDB, Quora). BERT itself is primarily trained on a large corpora of text from the internet and will thus tend to contain biases present in society. This biases could have an impact on the outputs of the model, which, depending on what the model is used for, can perpetuate discrimination and harmful stereotypes. We could try to remedy these issues by training our model on more diverse corpora — for example, on datasets from different languages, or from sources other than the internet.

8.2 Environmental impact

It's important to take into account our project's impact on the environment. Training and evaluating the models requires heavy consumption of computational resources and therefore produces a large carbon footprint, and so to offset this, we want to try to make our code as intelligent and efficient as possible, doing incremental testing on smaller datasets on our local machines to minimize heavy compute. Unfortunately, one of our approaches involved increasing the use of data in order to be able to sum losses, which is a more computationally expensive algorithm, so in future steps we would likely look at employing more efficient algorithms instead.

References

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. Mtrec: Multi-task learning over bert for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.