

**Project Name:**

**Comparing BERT and ELMo on a Sentiment Analysis Task using IMDb Reviews Database**

Students name: Tamir Shoresh & Dan Goren

Lecturer name: Dr. Yair Horesh

Submission date: 19.09.22

Abstract:

This paper will investigate the use of Natural Language Processing (NLP) for sentiment analysis as a way to classify text based on their content and context. Specifically, the use of Bidirectional Encoder Representations from Transformers (BERT) and Embeddings from Language Model (ELMo)

This paper offers a review of two popular word embedding algorithms, ELMo and BERT, followed by a comparison on a sentiment analysis task using a movie review dataset. Word embeddings are real-valued word representations able to capture lexical semantics and trained on natural language corpora. Models proposing these representations have gained popularity in recent years, but the issue of finding the most adequate evaluation method remains unsolved. In this study, BERT and ELMo will be evaluated according to their performance on the IMDb movie review dataset which contains 50K reviews that have been categorized by sentiment as positive of negative. A pre-trained BERT-Base architecture and a standard 2-layer bi-directional ELMo were used to implement the embedding and were fine-tuned to solve the sentiment analysis problem. The best performing model was BERT with an accuracy of 89% on the test set followed by ELMo with an accuracy of 69%. The validation and performance evaluation of our solutions demonstrate promising results for both word embedding models. Furthermore, the comparison offered in this paper can be expanded to other datasets to support the claims of this study.

1. Introduction:

Sentiment Analysis is the computational interpretation of opinions, sentiments, and subjectivity of text (Medhat et al., 2014). a sentiment analysis task can be interpreted as a classification task where each category represents a sentiment (Prabowo & Thelwall, 2009). The analysis of these sentiments and opinions has become popular in many fields such as consumer information, marketing, product review, and social media. Sentiment analysis has become a hot topic in decision-making, with hundreds of thousands of users depending on online sentiment reviews (Chalothom & Ellman, 2015; Hussein, 2018). A database such as IMDb can greatly benefit from analyzing the reviews posted to its platform. The large amount of data may be difficult to analyze manually, making automated tools an ideal candidate for the task.

Over the past decade, there has been an increase in papers published regarding sentiment analysis using Natural Language Processing (NLP) techniques. For example, Xu et al., 2019 tune a Bidirectional Encoder Representations from Transformers (BERT) for review comprehension and aspect-based sentiment analysis using a large customer review dataset.

Word embeddings are real-valued word representations able to capture lexical semantics and trained on natural language corpora. Models proposing these representations have gained popularity in the recent years, but the issue of finding the most adequate evaluation method remains unsolved.

This paper offers a review of two popular word embedding algorithms, ELMo (Embeddings from Language Model) and BERT. A comparison of the two models will be carried out using reviews from the IMDb dataset for sentiment analysis.

1. Models:
   1. **Context Independent Embedders:**

These embedders do not make use of the word’s position in a sentence, nor the surrounding context, when calculating the word vectors in a sentence. Context independent embedders, such as GLoVe (Pennington et al., 2014) and Word2vec, are able to provide semantic knowledge about known dictionary words. However, their disadvantage lies in the fact that identical words used in different contexts can acquire the same embedding for all instances.

* 1. **ELMo:**

ELMo is a context inclusive word embedder. ELMo word vectors are functions of the entire input sentence and are computed on top of a two-layer bi-directional language model (Peters et al., 2018). ELMo makes use of a concatenation right-to-left and left-to right LSTMs which means one layer reads in one direction and another layer reads in the other. Additionally, ELMo makes use of a character-based input, providing vectors for each character that are usually combined through a model.

Expanding, ELMo’s architecture contains two layers of bi-directional LSTM’s, each containing two opposite-directed LSTMs, for capturing the right-to-left and the left-to-right context of the input sentence. Each sentence is embedded into a vector using a naïve embedder. The two output vectors of the two first-layer LSTMs are then concatenated and used as the input for the second layer, which in turn, outputs two more vectors. Lastly, the output vectors of the first layer and the second layer, along with the naïve embedding vector, undergo a weighted sum to get the final word embedding.

An example of such an architecture can be seen in Figure 1.

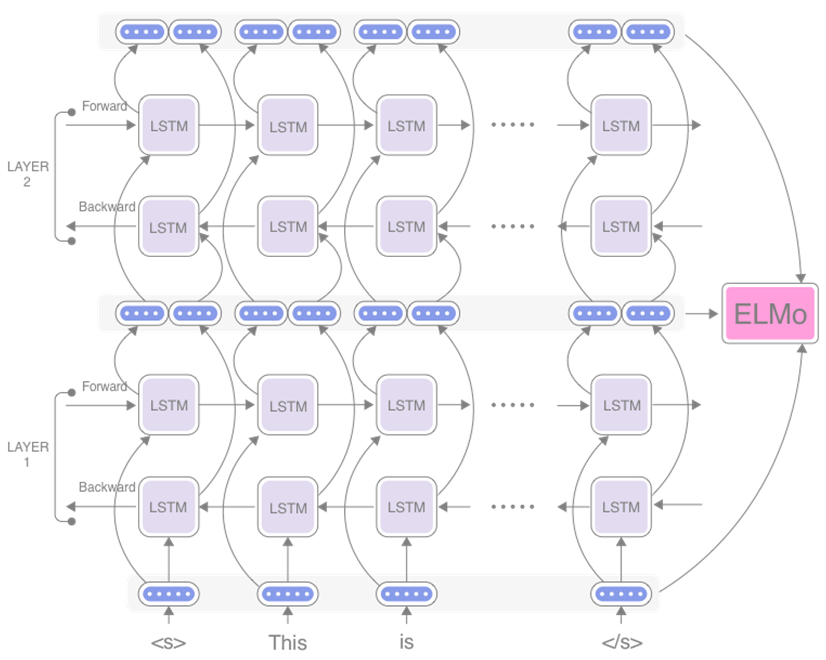


Figure - Architecture of ELMo

* 1. **BERT:**

BERT creates embeddings by pre-training on two unsupervised tasks simultaneously. The first task is Masked Language Modeling (MLM), in which 15% of all input tokens are randomly selected and replaced by a [MASK] token, enabling the representation to fuse the left and the right context from both sides of the mask (Munikar et al., 2019). The second task is Next Sentence Prediction (NSP), in which given two sentences, A and B, the model needs to decide if sentence B follows sentence A. This allows BERT to understand the relationship between two sentences, which is not even captured by language modeling.

Using a pre-trained model, BERT can be fine-tuned to perform a specific task. Fine-tuning is done by replacing the fully connected output layer by a new fully connected layer that corresponds to the desired task. For example, when considering a positive-versus-negative sentiment analysis task, the new fully connected layer only needs two output neurons. In this process, most of the learning is done by the top layers. To retain the learning to the top layers, it is not recommended to fine-tune the model for more than 2-4 epochs as most of the bottom layers have already been perfectly weighted from the previous pre-trained step and we do not wish to change them too much.

A schematic of the BERT architecture can be seen in Figure 2.

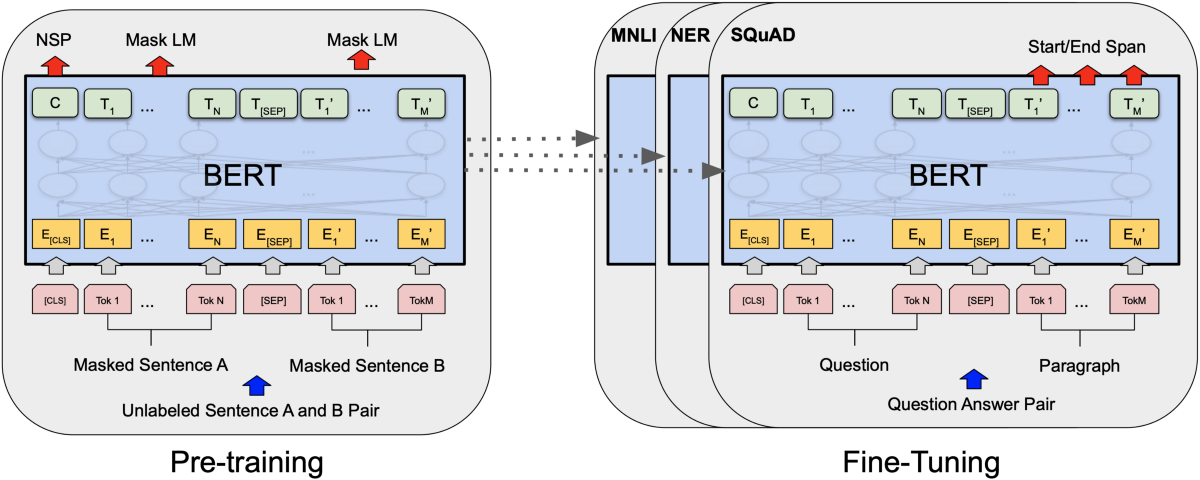


Figure - Left: a schematic of BERT architecture while pre-training. Right: a schematic of BERT architecture while fine-tuning on a question answering task

In practice, BERT is pre-trained on both MLM and NSP simultaneously be receiving as an input a sequence of two successive sentences which have already been tokenized and masked. These tokens are embedded into vectors by adding together 3 vectors: the token embedding vector, which is a pre-trained embedding vector, the segment embedding vector, which is the sentence number (A or B) encoded into a vector, and the position embedding vector, which is the position of the word in that specific sentence encoded into a vector. The segment and position embeddings are required for temporal ordering since all the word vectors are fed in simultaneously into BERT and language models need the sentence order preserved.

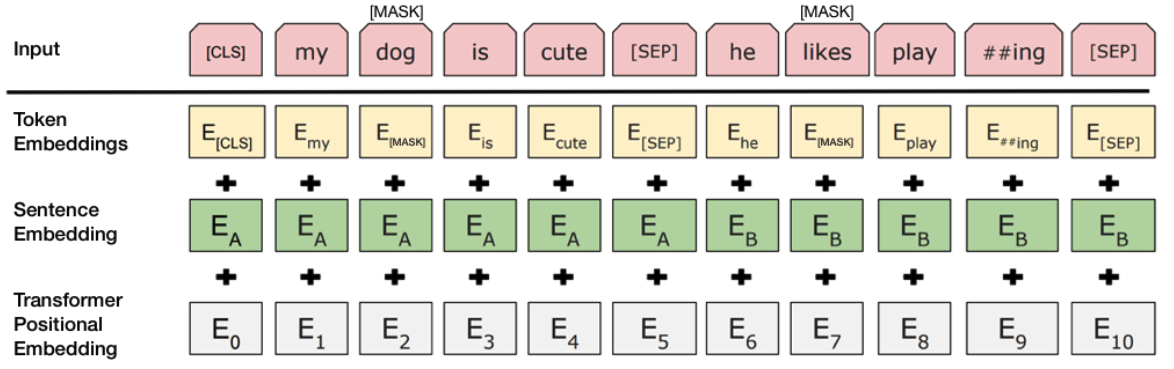


Figure - an example of input embedding required for BERT

Added to the beginning of each sequence is a classification token ‘[CLS]’ and ‘[SEP]’ to separate or finish a sequence.

On the output side, “C” is the binary output for next sentence prediction, returning ‘0’ if sentence B does not follow sentence A and ‘1’ if it does. In addition, each “T”s is a word vector corresponding to the outputs of the MLM task.

1. Dataset:

The IMDb dataset contains movie reviews along with their associated binary sentiment polarity labels. It is intended to serve as a benchmark for sentiment classification. The core dataset contains 50,000 reviews split evenly into 25k train and 25k test sets. The overall distribution of labels is balanced (25k positive and 25k negative). In the entire collection, no more than 30 reviews are allowed for any given movie because reviews for the same movie tend to have correlated ratings. Further, the train and test sets contain a disjoint set of movies, so no significant performance is obtained by memorizing movie-unique terms and their associated with observed labels. In the labeled train/test sets, a negative review has a score <= 4 out of 10, and a positive review has a score >= 7 out of 10. Thus, reviews with more neutral ratings are not included in the train/test sets. In this study, a sentiment analysis task will be trained classifying the sentiment of each review into 2 classes – positive or negative.

1. Methodology:
   1. **Pre-processing:**

In this stage, both training and testing data need to be preprocessed to turn review sentences into the required format for BERT and ELMo. Firstly, each sentence is tokenized. This step is done by normalizing text (convert whitespace characters to spaces, convert to lowercase and remove accent markers) and punctuation splitting, which adds whitespace around punctuation. To finish the tokenization, WordPiece tokenization was used to convert longer versions of words to simpler sub-words. For example, the word “playing” will be turned to “play” + ”##ing”. This is done so similar versions of words, for example -play, playing, player, can be treated the same. Specifically for BERT, the step after tokenization is to apply [CLS] and [SEP] to the start and end of the tokenized sentence, respectively. Lastly, we need to convert tokens to ids and pad data with ‘0’s.

* 1. **Training ELMo:**

For this study, a standard architecture of ELMo from TensorFlow hub was used. The module outputs fixed embeddings at each LSTM layer, a learnable aggregation of the 3 layers, and a fixed mean-pooled vector representation of the input. architecture achieves state of the art results on several benchmarks.

ELMo’s exclusive task is word embedding, meaning that in order to perform a sentiment analysis task, a fully connected layer must be added to the architecture. The ELMo model used in this study has an output layer dimension of 1024. To this output layer, a fully connected layer of 256 neurons was concatenated, followed by a fully connected layer of 2 neurons with a softmax activation, corresponding to the two classes of the sentiment analysis task.

* 1. **Training BERT:**

For this study, the lowercase BERT-Base was chosen due to case sensitivity not impacting the review context and the large model being excessive for this type of text sentiment. BERT-base consists of a 12-layer transformer, an output dimension of 768, and 12 self-attention heads, summing up to 110 million model parameters. Additionally, maximum sequence length is an important parameter for BERT. Capping at 512, the lower the maximum length, the quicker the network can be trained. It is important to set this as low as possible while not affecting the dataset since sequences over the maximum value are truncated. In this study, a maximum length of 196 was chosen.

The final architecture consisted of an input layer takes the pre-processed ids as its own input and feeds them into the BERT layer. After this, to reduce the dimensionality of the layer, a Lambda layer is used to flatten the middle dimension, to the size of the maximum sequence length, whilst ignoring the other dimensions. The resulting tensor is fed through dropout and dense layers, finally ending with a dense layer of with 2 output neurons, corresponding to the two classes of the sentiment analysis task at hand. The last layer has a softmax activation function which should be used when doing classification and outputs probabilities for each class that all sum up to 1. Dropout layers are used in this model to prevent overfitting of data.

1. Results:

Both BERT and ELMo were fine-tuned for a duration of 3 epochs using the 25K reviews available in the test set.

In Figure 4 the loss over 3 training epochs is shown for the BERT model. Additionally, in Figure 5, the accuracy is shown for the same 3 epochs. It can be concluded that the model does not overfit the data given that the validation loss continues to drop at the third epoch. Alternatively, it can be shown that the validation accuracy increases in the third epoch.



Figure - Loss over training epochs for BERT model

Chart, line chart

Description automatically generated

Figure - Accuracy over training epochs for BERT model

After training, BERT was used to predict the sentiments of the test set reviews. The results can be seen in Figure 6. BERT was able to predict the correct sentiment of the reviews with an accuracy of 89.236%.

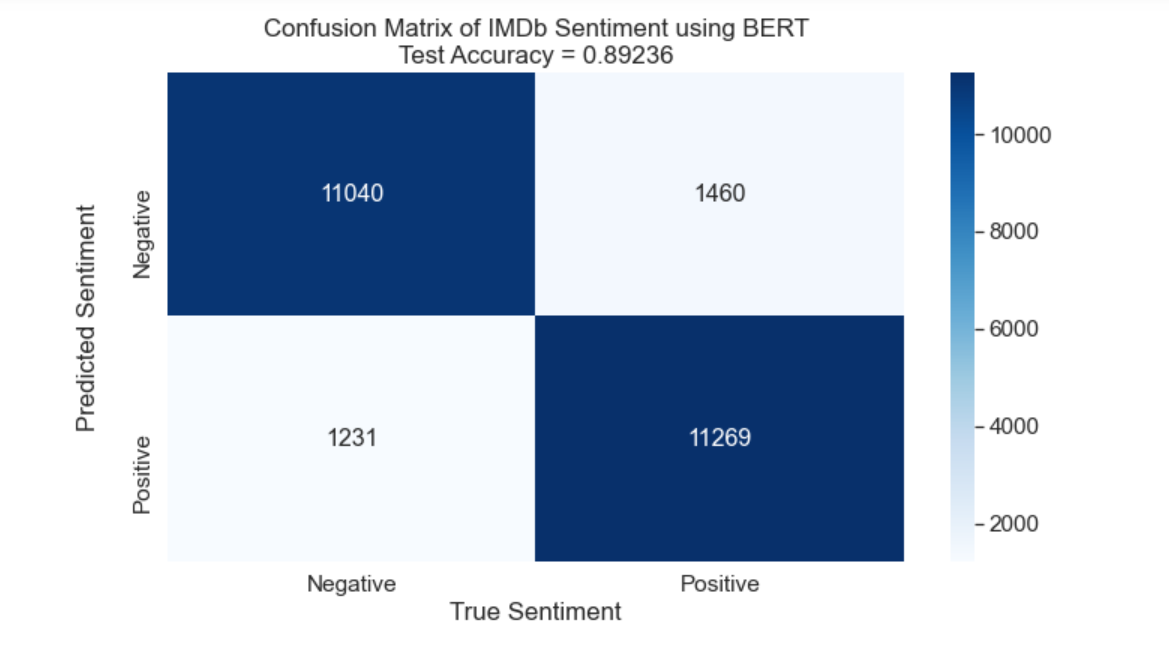


Figure - Confusion matrix of sentiment classification of test set using BERT model

Lastly, the model was used to predict an additional 6 sentences as a visual example of its abilities. The sentences and their predicted sentiment can be seen in Table 1.

|  |  |
| --- | --- |
| Review | Predicted Sentiment |
| This movie is rubbish, would not recommend. | 0 |
| Amazing! Loved this so much! Great film. | 1 |
| Average movie with a slow plot | 0 |
| I probably wouldn’t go see it again, but overall I guess it was alright | 1 |
| Not a big fan of these type of movies. Don’t really know how I got dragged into this. | 0 |
| Blew my absolute mind off! Best film I’ve seen this year | 1 |

Table - Made-up reviews and their predicted sentiment using BERT model

In Figure 7 the loss over 3 training epochs is shown for the ELMO model. Additionally, in Figure 8, the accuracy is shown for the same 3 epochs. It can be concluded that the model does not overfit the data given that the validation loss continues to drop at the third epoch. Alternatively, it can be shown that the validation accuracy increases in the third epoch.



Figure - Loss over training epochs for ELMo model

Chart, line chart

Description automatically generated

Figure - Accuracy over training epochs for ELMo model

After training, ELMO was used to predict the sentiments of the test set reviews. The results can be seen in Figure 6. ELMO was able to predict the correct sentiment of the reviews with an accuracy of 69.44%.

Chart, treemap chart

Description automatically generated

Figure - Confusion matrix of sentiment classification of test set using ELMo model

Lastly, the model was used to predict an additional 6 sentences as a visual example of its abilities. The sentences and their predicted sentiment can be seen in Table 1.

|  |  |
| --- | --- |
| Review | Predicted Sentiment |
| This movie is rubbish, would not recommend. | 0 |
| Amazing! Loved this so much! Great film. | 1 |
| Average movie with a slow plot | 0 |
| I probably wouldn’t go see it again, but overall I guess it was alright | 0 |
| Not a big fan of these type of movies. Don’t really know how I got dragged into this. | 0 |
| Blew my absolute mind off! Best film I’ve seen this year | 1 |

Table - Made-up reviews and their predicted sentiment using ELMo model

1. Conclusions:

After comparing the results of both models, BERT can be seen outperforming ELMo for sentiment analysis on review text. Both models were trained on the same dataset with the same number of epochs and same batch sizes, but BERT’s test accuracy came out to be over 89% while ELMo’s accuracy came out at about 69.5%.

Nevertheless, both models seem to capture the sentiment from simpler texts as shown in Table 1 and Table 2.

This project was a success due to the effectiveness of the sentiment analysis model, the evaluation of it and the objectives that were met. Even though Word2Vec and GloVe were not tested, we can estimate that, like ELMo, BERT will outperform them due to a superior understanding of context.

1. Future Goals:

The proposed evaluation can be used to compare any NLP model to another. In future studies, one might wish to compare the larger BERT variation, BERT-Large, to both BERT-Base and ELMo.

1. Bibliography:

Chalothom, T., & Ellman, J. (2015). Simple approaches of sentiment analysis via ensemble learning. In *Information science and applications* (pp. 631–639). Springer.

Hussein, D. M. E.-D. M. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, *30*(4), 330–338.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113.

Munikar, M., Shakya, S., & Shrestha, A. (2019). Fine-grained sentiment classification using BERT. *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, *1*, 1–5.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. ArXiv preprint. *ArXiv Preprint ArXiv:1802.05365*.

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143–157.

Xu, H., Liu, B., Shu, L., & Yu, P. S. (2019). BERT post-training for review reading comprehension and aspect-based sentiment analysis. *ArXiv Preprint ArXiv:1904.02232*.