# Sentiment Analysis Of IMDB Movie Reviews

**Jin Yan**
The Division of Statistics and Machine Learning (STIMA)
Department of Computer and Information Science (IDA)
Linköping University
jinya425@student.liu.se

## Abstract

Sentiment analysis is a promising sub-field in text mining since there is a considerable demand for insights into user-generated information for corporations, governments, or even individuals. The advent of the Large Language Model provides more possibilities in this field. The article aims to analyze the most up-to-date methods in this field and help future users or researchers to choose the most appropriate one. Based on this aim, the article surveys the performance of Bidirectional-LSTM and two large Language Models (Bert and Zephyr). The used dataset is' imdb' on the HuggingFace community. Given that the Bert model and the Zephyr model are of different structures, "fine-tuning" and "prompt engineering" are applied to them, respectively. Notably, "Prompt Engineering" consists of "zero-shot engineering" and "few-shots engineering." It turned out that Bi-LSTM can get the 0.86180 accuracy. The best results of fine-tuned-version Bert and Zephyr can be 0.87 and 0.90, respectively. There are no significant differences in RNN structure and transformer structure. However, considering the training time and other elements, Bi-LSTM is still meaningful in practice in this LLM era. Code for fine-tuning and Prompt Engineering:

https://github.com/yj313155521/LLM-Project.git

## 1 Introduction

As a Natural Language Processing(NLP) technique, Sentiment Analysis places more emphasis. This is because Internet service providers, consumer companies, and even governments can improve their services with insights from the user's opinions. As the most promising machine learning tools, neural networks have shown their power in dealing with such tasks. Bi-directional LSTM Network showed good results on the Amazon Product Review dataset. (Mahadevaswamy and Swathi, 2023) Transformer architecture (Vaswani et al., 2023) makes two types of models possible. The first one is BERT, which Google AI Language introduces. This model was designed to pre-train deep bidirectional representations from unlabeled tests by joint conditioning on both left and right contexts in all layers. This unique architecture allows fine-tuning the model without substantial modifications.(Devlin et al., 2019). Another LLM used in this article is Zephyr, a fine-tuned version of the Mistral-7B-v0.1 model according to the model card in the huggingface community. In terms of Mistral, it can be utilized for generating human-like text and is also based on transformer architecture.(Jiang et al., 2023). It is a good candidate for testing different "prompt engineering" strategies. In this situation, it is interesting to investigate if one of these excellent structures can show great advantages over the others regarding accuracy and computational costs. However, some papers did similar research on this topic, like (Wankhade et al., 2022), but they still need to do such a detailed comparison.

## 2 Theory

### 2.1 Bidirectional LSTM Network

LSTM Network, namely Long and Short Memory Network, is a Recurrent Neural Network(RNN) variation. This new structure can effectively eliminate the vanishing/ exploding gradients problems in RNN. Each LSTM unit is composed of a "Forget Gate," "Input Gate," and "Output Gate." "Forget Gate" controls how much long-term memory from the last unit would remain in this unit. "Input Gate" is used for generating new long-term memory. The new long-term memory will be used by the last gate, "Output Gate," to create new hidden value. An LSTM Network consists of many such units that join end-to-end. Creating a Bidirectional LSTM requires that LSTM neurons be divided into two directions—one for forward states and the other for backward states.(Mahadevaswamy and Swathi,

2023) The two directions in the network can allow input data from both the past and future of the present time frame.

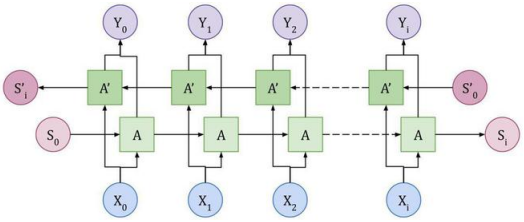

Figure 1: The Bidirectional LSTM - model architecture.

## 2.2 Transformer

The main structure is shown in Figure 2. From the figure, we can see that this structure is composed of two parts. On the left bottom left corner, the structure first converts the input tokens to vectors of dimension $d_{model}$, and then these vectors("word embeddings") are added to "positional encodings." The results are then put into an "encoder." A stack of N = 6 identical layers compose the encoder. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. To simplify our explanations, we mainly discuss the one-head self-attention mechanism. In this mechanism, the input for one token will be replicated into three identical ones, and then these three ones are projected to three vectors with the same dimension (64). They represent "Query," "Key," and "Value." The output of the self-attention mechanism(self-attention value) for one token is the weighted summation of different values of all tokens "Value". The Weights in the calculation are derived by calculating the distances between the token's "Query," and all token's "Values," including its own "Value." The advantage of using this mechanism is that it can reduce computational complexity and the path between long-range dependencies in the network. Also, it can increase the amount of computation that can be parallelized.

## 2.3 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It is a multi-layer bidirectional Transformer encoder based on the above-mentioned Transformer Model. It is worth noting that it is mainly based on the "encoder" part of the Transformer. The parameters used in this model are different from Transformers. Take the model used in the article, $BERT_{base}$, as an example. The

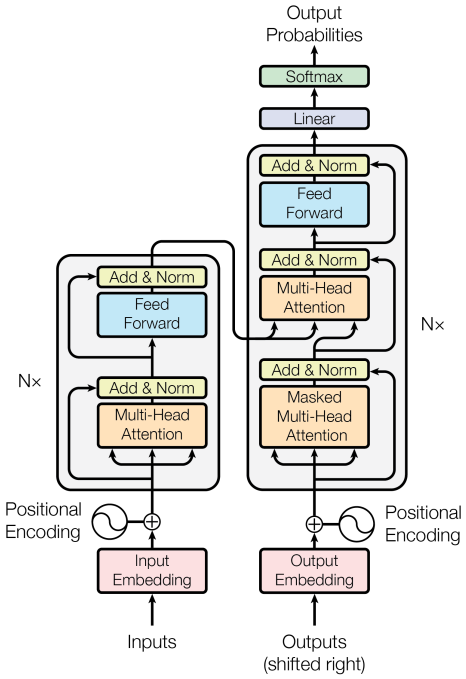


Figure 2: The Transformer - model architecture.

number of layers for the encoder is 12, the hidden size is 768, and the self-attention heads are 12. By contrast, in the Transformer model, these figures are 6, 512, and 8, respectively. In order to get such a pre-trained model, the researcher used two tasks. One is "Masked LM". That is masking some percentage of the input tokens randomly and then predicting those masked tokens. The final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary. Another task for pre-training is sentence prediction. After getting the pre-trained model, the model needs to be fine-tuned for each task to improve the performance of downstream tasks. In Fine-tuning Procedure, most model hyperparameters are the same as pre-training except for batch size, learning rate, and number of training epochs.

## 2.4 Mistral 7B

Mistral 7B is based on Transformer architecture.(Jiang et al., 2023) This model outperforms Llama 1(the best open 13B model) across all evaluated benchmarks and Llama2(the best released 34B model) in reasoning, mathematics, and code generation. Both Llama 2 and Llama 1 are from Meta. Also, this model uses "GQA" and "SWA" and thus can effectively handle sequences of arbitrary length with a reduced inference cost.

### 2.5 Prompt Engineering

As for dealing with NLP tasks, there are three main stages. The first stage is characterized by feature engineering and architecture engineering. The critical point is to find the salient features and the appropriate inductive bias. The second stage is famous for the pre-train and fine-tune paradigm. In this paradigm, the model is pre-trained as a language model (LM) and then adapted to all downstream tasks. The process may involve introducing and fine-tuning new parameters using task-specific objective functions. The third one is the "pre-train, prompt, and predict" paradigm. Instead of objective engineering, this stage is famous for "prompt engineering." When we want to find the sentiment of a film review, we can input "[review], and it contains [ ] emotion." and ask the model to fill in the square bracket. At the same time, there are already different training strategies, like Tuning-free strategy and Fix-prompt LM Tuning. The strategy used in this article, namely the Tuning-free strategy, would avoid the risk of catastrophic forgetting since it does not change the parameters of the pre-trained models.

## 3 Data

The dataset I used in this article is 'imdb,' a large sentiment analysis dataset containing 25,000 reviews for training and 25,000 reviews for testing.

The training and test datasets have all been used for the Bidirectional Neural Network. Nevertheless, that is different for the other two structures.

In the case of fine-tuning the BERT model, the size of the training Data set and evaluation data set used are both 1000. Meanwhile, since the limitation of the input for the BERT model is 512, the experiment truncated the input sentences and ensured the input length was within 250. This way, the number of tokens created can satisfy this requirement.

By comparison, in the case of prompt engineering, the test data set size is ten since this model would take longer to create an answer, including a desired label, and this size can ensure an appropriate test time (about 30 minutes). Like BERT, Zephyr also has its input length limitation. This way, the length of the truncated sentence is within 200 to ensure that even if a few shots are added to reviews, the length requirement can be satisfied.

## 4 Method

In this section, the article will show the methodology employed in this thesis project.

For the first one, the neural network in this field consists of four layers. The first is an embedding layer, which can transform tokens into vectors. The next one is the Bidirectional Layer. After one Dense Layer, the final one is a Dense Layer with a sigmoid function, which creates the "predicted label."

The Second one is Fine-tuning. In order to realize this, the article used the class, the Transformers Trainer. The process is much easier, and the trained model [1] is saved in the Huggingface community.

The last one is the usage of prompt engineering called Tuning-free Prompting. (Luo, 2023). We use three prompt templates to explore the results of this process. Each template consists of three parts: "System," "Human," and "Assistant." Details are shown in Table 1, Table 2, and Table 3.

In template 1, at first I introduce a review and then give the task description to the language model and this would make the model understand the task very well. Finally the model is asked to classify the review I mention at the beginning of Human part.

In template 2, instead of presenting the task descriptions, I show the model few demonstration examples. In each example, the model can learn how to label film reviews. This process is also called augmentation method.(Liu et al., 2023) or "in-context learning".

In template 3, I do not give much information about the tasks, and just let model classify the input review according to its understanding. And thus, the result created can be used as a baseline to assess the effects of the other two prompt engineering strategies.

## 5 Results

This section contains four parts. The first part shows the accuracy of the Bidirectional LSTM network. The second part compares the Bert Base model to the fine-tuned model.4. The result is shown in table 5. The last part compares different strategies in table 6. The final result compares the best results of these several methods, including the training time.

[1] jinya425/bert-base-cased_for_sentiment_analysis

| Parts of Prompt | Content |
|---|---|
| `### System:` | You are an AI assistant that follows instruction extremely well. |
| `### Human:` | [Text] Given the above review we have two two classes.<br>class_0: In such film reviews, customers complain about something they do not like and think this film is not satisfactory.<br>class_1: In such film reviews, customers show positive emotions, they might mention something that they like.<br>Please classify this film review into one class out of these two classes,and just output the label without anymore word. |
| `### Assistant:` | |

Table 1: Template 1 (with Detailed Instruction, zero-shot)

| Parts of Prompt | Content |
|---|---|
| `### System:` | You are an AI assistant that follows instruction extremely well. |
| `### Human:` | First review: Brilliant and moving performances by Tom Courtenay and Peter Finch.<br>Second review: This is a great movie. Too bad it is not available on home video.<br>Third review: Primary plot!Primary direction!Poor interpretation.<br>Fourth review: Read the book, forget the movie!<br>Above, the first two reviews belong to 'class_1'; the last two reviews belong to 'class_0'.<br>Please classify the following film review into one class out of these two classes,and the output format should be the same as 'This review belongs to 'class_X'.<br>New review:[text] |
| `### Assistant:` | |

Table 2: Template 2 (with demonstration examples, extremely few-shots)

| Parts of Prompt | Content |
|---|---|
| `### System:` | You are an AI assistant that follows instruction extremely well. |
| `### Human:` | [text] Given the above review we have two two classes. 'class_0' and 'class_1'.<br>Please classify this film review into one class out of these two classes,and just output the label without anymore word. |
| `### Assistant:` | |

Table 3: Template 3 (Baseline prompt)

| Name of Models | Accuracy/Training Time |
|---|---|
| `Bidirectional Network` | 0.86180 / 6.1 min |

Table 4: Result of Bidirectional Network for sentiment analysis

| Name of Models | Accuracy/Training Time |
|---|---|
| `bert-base-cased` | 0.51 / 0 min |
| `fine-tuned version of Bert` | 0.87 / 101.5 min |

Table 5: Results of different Bert Models for sentiment analysis

| Template No. | Accuracy/Traning Time |
|---|---|
| 1 | 0.85 / 35.8 min |
| 2 | 0.90 / 25.9 min |
| 3 | 0.50 / 35.6 min |

Table 6: Results of different prompt engineering tactics for sentiment analysis

| Model Name | Accuracy/Training Time |
|---|---|
| Bidirectional Network | 0.86180 / 6.1 min |
| Bert | 0.87 / 101.5min |
| Zephyr | 0.90 / 25 min |

Table 7: The best performance of different models for sentiment analysis

## 6 Discussion

The performance of the Bidirectional LSTM Network is competitive, especially considering the training time used in the whole process.

As for Bert, it is clear that with the implementation of the fine-tuning method, the accuracy of the sentiment analysis increased a lot. This further proves the power of fine-tuning for the LLM method.

After fine-tuning, although LLM can do a good job, we must recognize that training such a model needs a lot of computational resources and time. Most importantly, we cannot ensure that enough labeled datasets can be offered each time. With enough data points, we can reach the same goal.

In order to solve the drawbacks mentioned above, we can consider another generative pre-trained model, Zephyr in this case. From the final result, compared to the baseline prompt template 3, template 1 (Zero-shot) and template 2 (Extremely few-shots) can provide exciting results. Most interestingly, the performance of template 2 is the best, and the computational time is the least. This shows that using examples can make it easier for this model to learn than the general description of the classification task.

However, it differs from the result in (Luo, 2023). In that paper, the zero-shot strategy achieves better results. The main reason is the differences between the two classification tasks. In that paper, the data set consists of 14 labels. Among them are 13 specific labels and another label representing those conversations that cannot be categorized into the first 13. By comparison, in this article, there are only two labels for the model to distinguish.

As for the limitations, the first one should be the amount of training data. For the Bidirectional LSTM model, all of the data from 'imdb' is used to train and validate. However, considering the computational resources limitations, the data put into the Bert Training process is only about 1000 or so. For the Zephyr model, the situation is "worse". If we look at the "few shots" case, only two positive examples and two negative examples are given to the model. Meanwhile, to satisfy the requirement of the input length, the input reviews are truncated, which means not all information is considered by the model when making predictions.

Second, the scope of the models and the prompt engineering tactics are limited. Besides what the article mentioned above, many large language models and prompt engineering tactics can still be compared. Here are only representatives of them.

## 7 Conclusion

In this article, the accuracy of Bidirectional LSTM is pretty high, and the computational time is relatively minor. That means even though in the era of the Large Language Model, the usage of RNN structure is meaningful. Regarding the LLMs, the training time could be more competitive. However, considering the limited training data, LLMs have excellent potential. If the computational resources are enough, the advantages of such models can be more significant.

In general, no method is perfect and overwhelms others. When we consider which one to use, the situation should be considered. If the computational resources are limited, Bidirectional LSTM is a good choice. By contrast, if high accuracy is preferred, LLM is better. When the training dataset only contains a few labeled input reviews, Zephyr is the best one. Notably, searching for a good template is time-consuming, which should be addressed.

## References


Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. ArXiv:2310.06825 [cs].

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 55(9):195:1–195:35.

Hengyu Luo. 2023. *Prompt-learning and Zero-shot Text Classification with Domain-specific Textual Data*.

U. B. Mahadevaswamy and P. Swathi. 2023. Sentiment Analysis using Bidirectional LSTM Network. *Procedia Computer Science*, 218:45–56.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. ArXiv:1706.03762 [cs].

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780.