Mini project : Sentiment analysis with state of the art models in NLP

César FABIANI
ENSAE Paris

Abstract

In this article, we aimed at testing state of the art technics to predict the sentiment of a movie reviews using the IMDB dataset. We start from a benchmark model and show that simple embeddings trained on our dataset are not really great for sentiment analysis since they struggle with capturing context. Transformers based models, such as BERT, are more context-aware and give promising results for this task.

1 Introduction

This project aims to predict whether a review published on imdb is positive or negative. Maas et al (2011), who introduced the dataset, proposed an approach to deal with the difficulty of sentiment analysis. We aim to test a different approach using the latest state-of-the-art techniques to perform sentiment analysis. In this report, we will present the state of the art in sentiment analysis and take a loot at the dataset. We make an initial analysis of the data. We then present the models and finally the results.

2 The state-of-the-art about sentiment analysis review

Sentiment analysis is a common topic in NLP and can be challenging because the notion of sentiment can be difficult to define. Of course, the first step is to look at the vocabulary used in a document to identify positive or negative terms. But phrases such as 'Although the critics said it was terrible, I really enjoyed this film' can be more difficult to assess. It's clear that the sentence is about a film review, from the vocabulary used ('critcs', 'movie', 'enjoyed'), but the sentiment is harder to guess. You need to understand the context in which the terms are used. The methods used to analyse sentiment have evolved and the most recent techniques, such as embeddings and LLMs, have been used to achieve this task.

Early approaches, using classical NLP models such as support vector machines, the Naives Bayes classifier or random trees associated with TF-IDF, achieved an accuracy of around 87-88% (see Maas et al. 2011). The use of convolutional neural networks (CNNs) then improved accuracy to 91-92% (Johnson and Zhang, 2015). A specific type of neural network, long-term memory networks (LSTMs), which are able to capture long-term context and are therefore better suited to examining complex sentences, further increased accuracy to around 93% (Dai and Le, 2015).

Finally, transform-based models, such as BERT, introduced by Devlin et al. (2019), have been able to dramatically increase the success of text analysis across a very wide range of tasks, including sentiment analysis, where they achieve near-perfect accuracy. Thanks to self-attention mechanisms, and without being sequential like LSTMs, transformer models are able to understand the relationships between words regardless of their position. As a result, they are able to obtain a better contextual

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

meaning, which is absolutely necessary for the analysis of film reviews in view of the text examples presented above.

3 The IMDB reviews database

3.1 The dataset

The dataset is composed of 50000 reviews, 25000 in the train set and 25000 in the test set. Those reviews come from the IMDB website, which allows for movie reviews by user. A note, going from 1 to 10 is generally attached to the review. The dataset does not contain the note given to the movie, and the label is simply "positive" if the review got a note from 7 to 10 or "negative" if the review got a note up to 4.

3.2 A rapid exploration of the data

Length We first look at the length of negative and positive reviews to see if the difference can be partly established in this way. We could guess than negative reviews can be longer because people want to explain what bothered them about the film for examples. The graphs 1 and 2 show no real difference in length between positive and negative reviews. They cannot therefore be used for predictive purposes.

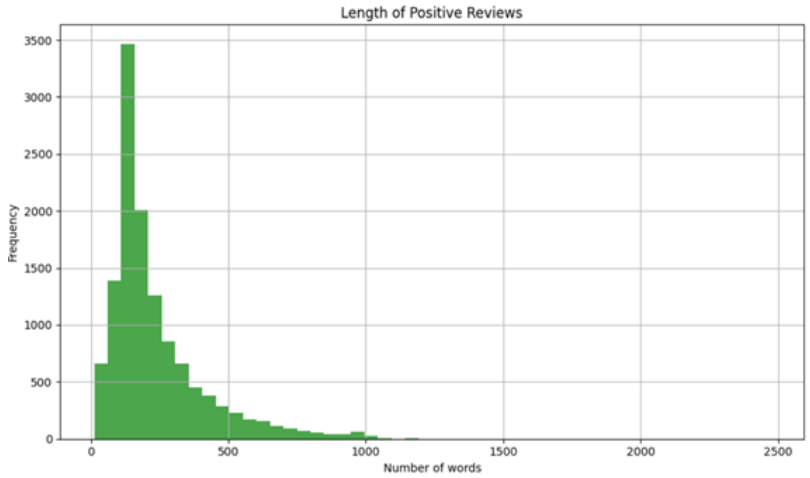

Figure 1: Length of positive reviews

Terms used We look at the difference between the words found in negative and positive reviews to see if a pattern emerges. Table 1 shows the 20 most common words (excluding empty words) found in both positive and negative reviews. Interestingly, the word ‘good’ is more common than ‘bad’ in negative reviews, confirming what we explained earlier about the difficulty of conducting sentiment analysis on film reviews. People may use positive terms in negative reviews to show what they liked, even if they didn’t like the film as a whole, or they may use it in a negative form (‘it wasn’t really good’). Our models therefore need to take context into account.

As most of the most frequent words are common English words, we use the td-idf transformation to see which words are most used in both types of film reviews. The graph 2 gives the most weighted words. Once again, it seems difficult to use only these words to predict sentiment as, for example, the word ‘like’ is used a lot in negative reviews. Nevertheless, it seems that in the positive reviews, people often talk about the synopsis of the film (‘living’, ‘drinking’), whereas in the negative reviews they focus on the analysis of the film (‘character’, ‘acting’, ‘plot’).

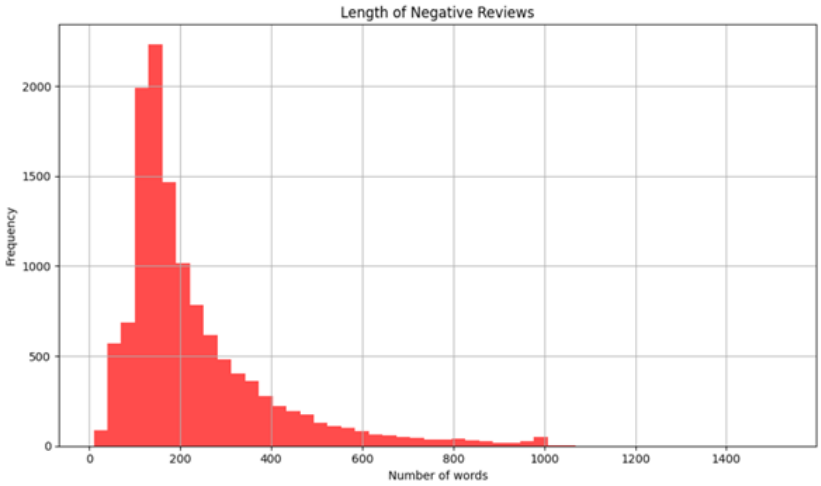

Figure 2: Length of negative reviews

Rank	Positive	Negative
1	(br, 49235)	(br, 52636)
2	(film, 20284)	(movie, 24097)
3	(movie, 18498)	(film, 18474)
4	(one, 13279)	(one, 12614)
5	(like, 8778)	(like, 10967)
6	(good, 7452)	(would, 7672)
7	(story, 6562)	(even, 7664)
8	(great, 6327)	(good, 7206)
9	(time, 6051)	(bad, 7139)
10	(see, 5872)	(really, 6240)
11	(would, 5715)	(time, 5848)
12	(well, 5631)	(could, 5697)
13	(also, 5514)	(see, 5406)
14	(really, 5462)	(story, 5021)
15	(even, 4945)	(get, 5003)
16	(first, 4634)	(much, 4993)
17	(much, 4618)	(people, 4749)
18	(people, 4419)	(make, 4552)
19	(best, 4237)	(made, 4354)
20	(get, 4232)	(first, 4248)

Table 1: Top 20 most common words in positive and negative reviews with frequencies

4 Our model

In this section, we present the approach of using state-of-the-art methods for our task. We first present a reference model, with a very simple approach. Next, we use simple word embeddings to show that they are not well suited to sentiment analysis. Finally, we introduce transformer models, with a compact version of Bert, to show that contextual embeddings such as transformers are best suited to our task.

4.1 Baseline model

We first do a very simple benchmark model. We convert raw text into numerical feature vectors based on term frequency-inverse document frequency (TD-IDF) then we use first a Naive Bayes Classifier algorithm. We do the same again but with a linear support vector machine algorithm. These are not state-of-the-art models but are great for comparison and work quite well according to the literature.

Rank	Positive	Negative
1	(box, 1331)	(br, 1468)
2	(narrative, 676)	(movie, 786)
3	(flick, 625)	(film, 690)
4	(living, 323)	(like, 375)
5	(gross, 317)	(just, 372)
6	(struggle, 278)	(really, 274)
7	(token, 256)	(time, 264)
8	(release, 250)	(don, 248)
9	(magazine, 215)	(story, 246)
10	(pile, 214)	(movies, 224)
11	(beowulf, 207)	(people, 222)
12	(narrator, 201)	(acting, 216)
13	(wastes, 194)	(plot, 214)
14	(thriller, 194)	(make, 213)
15	(lit, 193)	(watch, 200)
16	(severe, 185)	(think, 189)
17	(drinking, 183)	(way, 188)
18	(wayne, 182)	(seen, 187)
19	(floating, 180)	(characters, 184)
20	(check, 177)	(character, 171)

Table 2: Top 20 most weighted words in positive and negative reviews

4.2 Embeddings

Embeddings, developed since the 2010s could seem useful for sentiment analysis since they allow for a more profound analysis or word meanings. But since embeddings are able to map words in a way that words close in meaning are "close" in the embeddings, it doesn't analyse much context. Or, as we said, context is essential for sentiment analysis. Therefore simple embeddings may be not the best model for this task. To make sure, we use a classic embedding model, Word2Vec, and we then introduce a model closer to our needs.

Word2Vec Word2vec is a family of models that create neural embeddings. They were introduced by Mikolov et al. (2013). As other embeddings Word2vec is used to map words in vector space composed of several hundred dimensions (but way less dimensions than words in the vocabulary). Word2Vec is able to modelled the semantic relationships between words (so vectors) by arithmetic operations. We train the Word2Vec embedding with our train set. We then train a simple one layer neural network to make the prediction using the mapping created by the embedding. Compared to before, the embeddings replace the td-idf transformation earlier. We could have used the pretrained version of Word2Vec, which would have maybe given better results because the sample and the training would have been better, but this part is for comparison.

SSWE While traditional embedding models such as Word2Vec have proven effective at seeing semantic relationships, they do not encode sentiment information well. Tang et al. (2014) proposed a Sentiment-specific Word Embeddings (SSWE), designed to also reflect sentiment polarity. For example, traditional embeddings may have difficulties separating "terrible" from "incredible", since they are close on a semantic basis, both expressing strong feelings and often being in similar contexts.

The architecture of SSWE models are based on a neural network that takes as input a sequence of words as a window around a central a word and outputs a score for syntactic compatibility and a preduction of the sentiment label. The training objective is a combination of two components: a language modeling objective that captures syntactic coherence, and a sentiment objective that ensures that the embedding of the target word contributes to the correct classification of sentiment. During training, SSWE minimizes a hinge loss for syntactic modeling and a cross-entropy loss for sentiment prediction, updating word vectors so that sentimentally similar words are close together in the vector space. We train a SSWE and use after a simple neural network for prediction such as with Word2Vec.

SSWE embeddings remain models that turn words into vector which limit the ability to understand subtle context. That's why our main model should use more advanced, context-aware transformers.

4.3 DistillBERT

DistillBERT is a compact version introduced by Sanh et al. (2019) of BERT (Bidirectional Encoder Representations from Transformers), a transformer model proposed by Devlin et al. (2018). We use this model since it retains 97% of the language understanding capabilities while being able to be 60% faster. Due to computation limitations, reduced but still present, we train the model on a sample of our dataset, containing 500 reviews (and we also keep 500 reviews for the test samples, although it doesn't change things much to use the whole test sample). We use the common hyperparameters and set 2 epochs, once again to balance computational requirements, risk of overfitting and complexity. We have a slight regularization in order to not have too much importance to one weight. We use a L2 regularization, with $\lambda = 0.01$, which penalize heavy weights (especially true if there is a small sample).

$$L_{new} = L_{original} + \lambda \sum_{i=1}^{n} w_i^2 \tag{1}$$

The sample is very small and there is a risk of overfitting, but it already takes times (around 20 minutes) and the results are convincing.

5 The results

In this part we give a general overview of the performance of each model before giving more comments.

5.1 A general overview

Table 3 shows the results for each model used. First we see that the baseline model (TD-IDF + SVC) gives good result comparable to the literature. Then, we see, as predicted, that simple embeddings such as Word2Vec are not the greatest for sentiment analysis. If the database was bigger, or had we not trained it ourselves, the results would have surely been better , but it shows the limitations of those type of models for this task. SSWE works better because it is able to take in more context and is therefore a relevant architecture for sentiment analysis if we want to use simple embeddings. Finally DistillBERT gives the best results. Yet, they are well-below what could have been predicted. However, our sample was very small and they may have been overfitting so the results would have surely been much greater with more data and more computations. Therefore, state-of-the-art transformer based models seem to be the best-suited models for movie review analysis.

Model	**Accuracy (%)**
TD-IDF + Naive Bayes	0.74
TD-IDF + SVC	0.87
Word2Vec + NN	0.81
SSWE + NN	0.85
DistillBERT	0.88

Table 3: Accuracy comparison

5.2 Some comments

We plot below the confusion matrix of each models. In general those matrix show very similar performance for negative and positive reviews (except a slight imbalance for embeddings) [1]

[1] Making the confusion matrix is still useful. At one point, I tried a model with a sample of the dataset and couldn't understand why I got perfect accuracy. I though it was overfitting but couldn't see why. Plotting the confusion matrix showed me that I had forgotten to take a **random** sample and had only taken the first 1000 reviews for the train and the test, which were all positive. Indeed, the model achieved perfect accuracy with this sample !

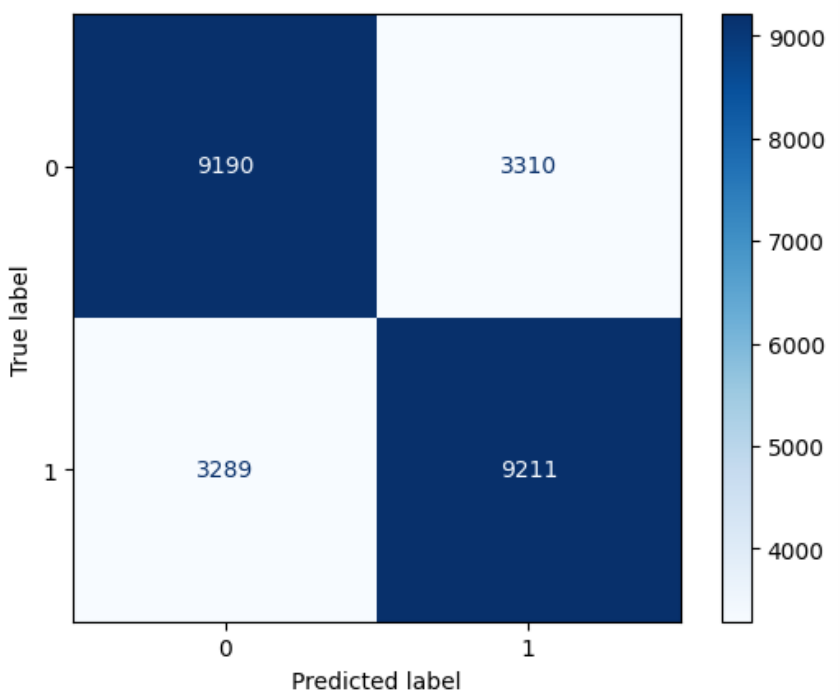

Figure 3: Confusion matrix for TD-IDF + NBC

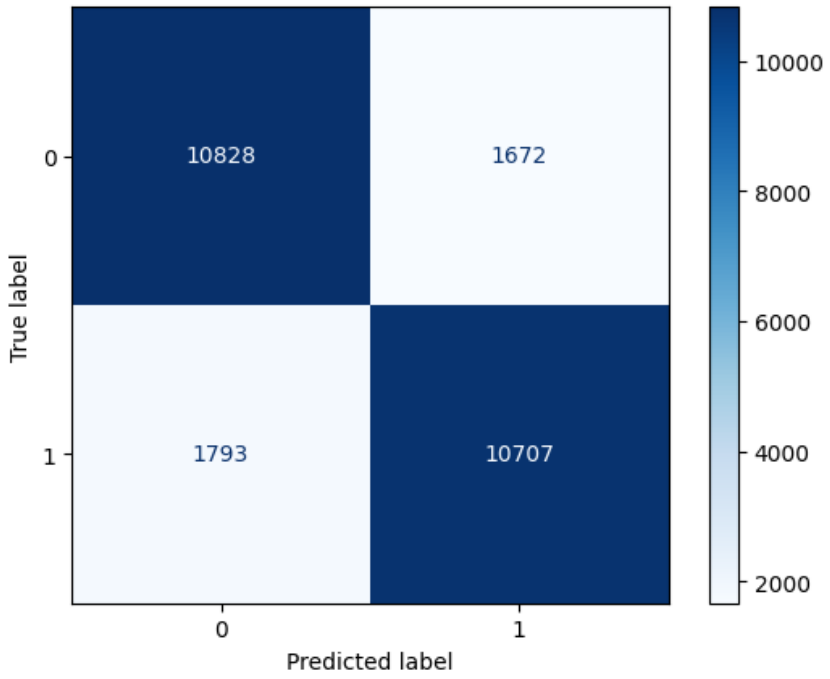

Figure 4: Confusion matrix for TD-IDF + SVC

A more interesting thing to look out for is how the models perform for some specific sentences. In 4, we see that the SSWE embeddings has trouble with sentences 3 and 4 where there is a crucial need to understand the context. In 5, we see that the model succeeds with sentence 3. It still struggles with

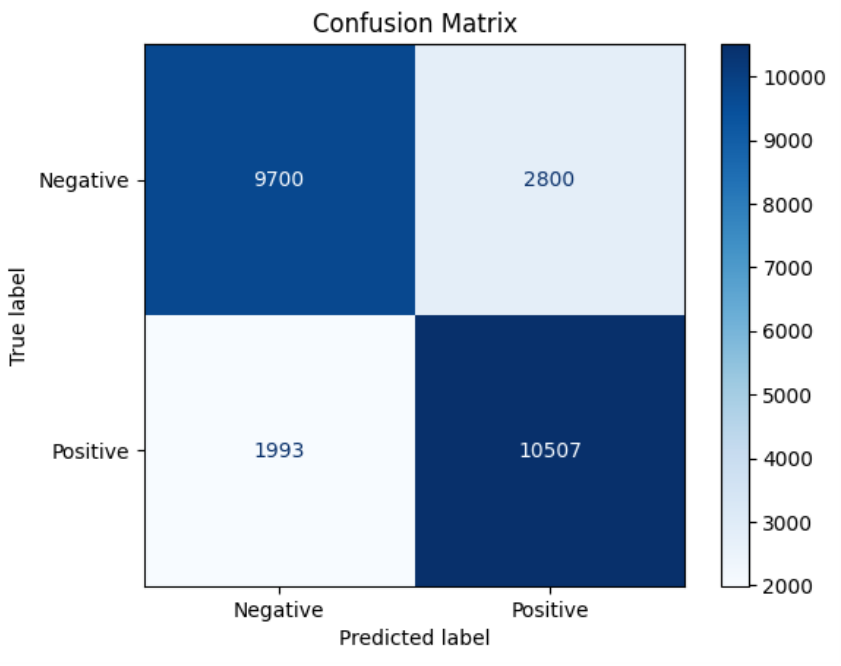

Figure 5: Confusion matrix for Word2Vec + NN

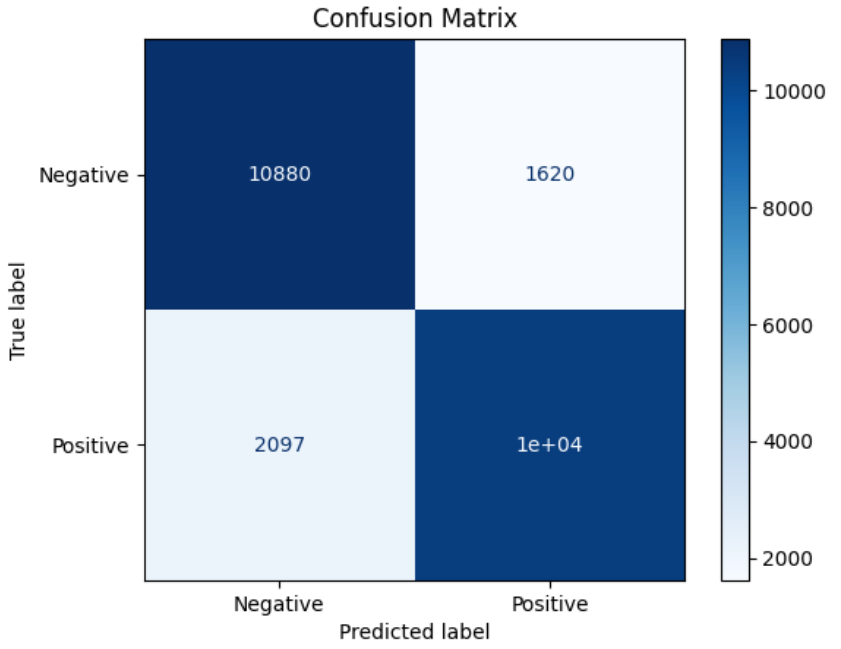

Figure 6: Confusion matrix for SSWE + NN

sentence 4, but it is a very hard sentence where there is no positive terms and yet the sentiment is positive.

Sentence	**Predicted Sentiment**
I really did not enjoy this movie, it was horrible!	Negative
Although people found it terrible, I think it was a masterpiece	Positive

Although people found it a masterpiece, I think it was terrible	Positive
The critics saying that the movie was bad were wrong	Negative

Table 4: Prediction test for SSWE

Sentence	**Predicted Sentiment**
I really did not enjoy this movie, it was horrible!	Negative
Although people found it terrible, I think it was a masterpiece	Positive
Although people found it a masterpiece, I think it was terrible	Negative
The critics saying that the movie was bad were wrong	Negative

Table 5: Prediction test for DistillBERT

6 Conclusion

In this project, we explore different methods for sentiment analysis. We saw that context-awareness is key for sentiment analysis and that a work on the words used are not sufficient. Simple embeddings are not well suited for this task. State-of-the-art models such as BERT are very promising. Nonetheless, even those very powerful LLMs do not get perfect accuracy, meaning there is still room for improvement and proving our subtle the human language can be.

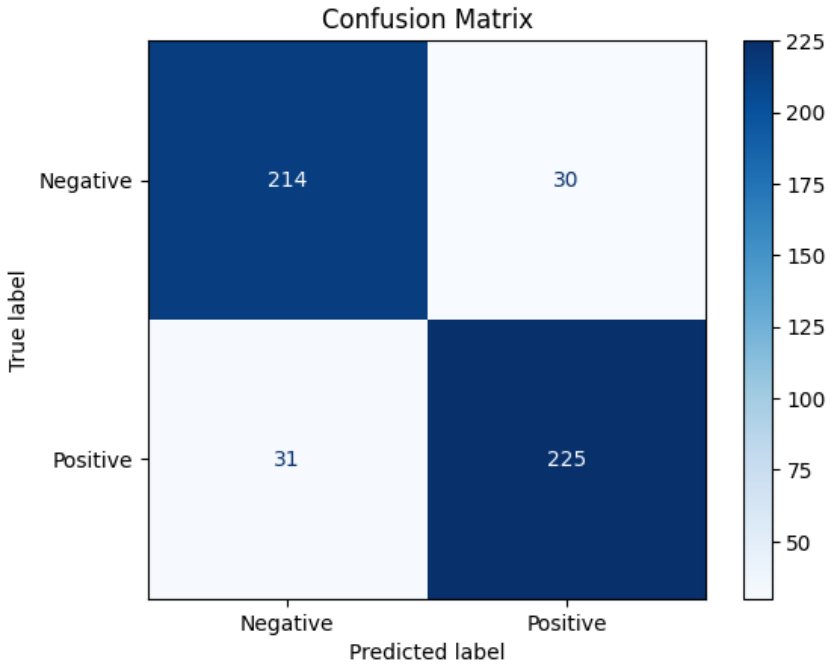

Figure 7: Confusion matrix for DistillBERT

References

[1] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). *Learning word vectors for sentiment analysis*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 142–150.

[2] Johnson, R., & Zhang, T. (2015). *Effective use of word order for text categorization with convolutional neural networks*. In Proceedings of NAACL-HLT.

[3] Dai, A. M., & Le, Q. V. (2015). *Semi-supervised sequence learning*. In Advances in Neural Information Processing Systems, 3079–3087.

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4171–4186.

[5] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781.

[6] Tang, D., Qin, B., & Liu, T. (2014). *Learning semantic representations of users and products for document level sentiment classification*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 1014–1023.

[7] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108.