Utilising BERT Embeddings and Unsupervised Techniques on User-Generated Reviews

Jordan O'Donovan
School of Computing
National College of Ireland
Dublin, Ireland
x19372016@student.ncirl.ie

***Abstract*—Developments in the field of natural language processing have been increasing tremendously over the past decade. One notable example of this has been the introduction of Bidirectional Encoder Representations from Transformers, BERT. In this paper I will be utilising BERT's ability to create insightful embeddings on Amazon and Yelp reviews which I will then apply different clustering techniques to. A book recommender was created, and sentiment analysis was carried out, both successfully. My findings demonstrate how effective BERT embeddings can be when combined with unsupervised methods.**

Keywords—BERT, Recommendation Systems, Cosine Similarity, Transformers, Unsupervised Learning, Python, PyTorch, TensorFlow, K-means Clustering, Sentiment Analysis

I. Introduction

Amazon was founded by Jeff Bezos in Washington, July 1994 [1]. Initially an online marketplace for books, it has now expanded into almost any product conceivable, making it one of the first US based trillion-dollar companies. They are one of the largest private employers in the world and own several other billion-dollar subsidiaries such as Twitch, Ring and Whole Foods.

Yelp, founded in 2004 by former PayPal employees Russel Simmons and Jeremy Stoppelman is a user-generated review website for local businesses. Google offered to buy them multiple times from 2005 to 2009 for up to half a million US dollars [2]. They are currently publicly traded with a market cap of over 2.2 billion dollars. On their website, users rate each business on one of a five-star system. Just over half of these ratings are five starts (52%) while the second most common rating seen on the website is one star, making up 18% of the total [3].

For this paper I will perform cleaning and pre-processing on thousands of reviews. I will conduct exploratory data analysis (EDA) on both datasets before utilising my chosen transformer and unsupervised algorithms. Following this I will evaluate the performance of my code before documenting them along with my findings in this paper.

II. Data

A. Amazon Book Reviews

This dataset is a subset of three million book reviews taken from Amazon's marketplace. The original dataset this originates from contains 142.8 million reviews for many types of items from May 1996 – July 2014.

This dataset contains eleven columns: ID (unique identifier of book), the book title, the price of the book, the ID of the user, the name of the user, the helpfulness ratio of the review as given by other users, the review score, the original user gave, the time the review was submitted at, a summary of the review and finally the full review.

This dataset contains no labels thus I will be performing unsupervised techniques on this unlabelled data.

B. Yelp Reviews

This Yelp dataset is a subset of a subset of the much larger Yelp Dataset. A subset was created from this to add the polarity of over one and a half million of the reviews. The subset I will be using for this paper contains two hundred and eighty thousand of these English reviews. This dataset is much more elementary than the aforementioned one, it only has two columns: the review and the sentiment.

As this dataset contains labels for the sentiment of each review, I will perform unsupervised learning then compare the predictions to the labels to measure the performance of my models.

III. Aims and Objectives

The objective of this assignment is to take reviews from both websites. The reviews from Amazon will contain only English reviews on books while the reviews from Yelp will be English reviews for any business. I will then create Bidirectional Encoder Representations from Transformers (BERT) embeddings using these reviews and will use unsupervised methods to create a recommendation system for the Amazon books and to perform sentiment analysis on the Yelp reviews.

For the recommendation system, the aim is to create a model which will accurately recommend the best books for a user to read next, listed in descending order of their probability of liking said book. I will likely use cosine similarity to produce said recommendations.

With the sentiment analysis, the aim is to create a model which will accurately be able to predict whether a given review is positive or negative. Once my embeddings are created, I will likely utilise K-means clustering to separate the embeddings by sentiment.

IV. Overview of Methods Selected

A. Bidirectional Encoder Representations from Transformers

BERT is a deep learning natural language processing (NLP) model used to perform several NLP tasks such as text classification, sentiment analysis and distinguishing words with different meanings [4]. BERT was primarily trained on two tasks: language modelling and next sentence prediction. As BERT was built upon Transformers, it is able to look at both sides of a word thus the model is able to understand the full context of each word. It is the first NLP technique to rely solely on self-attention mechanism, again due to the

bidirectional Transformers at BERT's foundation. Transformers are made up of encoder and decoders however BERT only uses the encoder here as using a decoder is not necessary for tasks performed by this model [5].

B. Unsupervised Models

Unsupervised learning is a type of algorithm which finds patterns from unlabelled data. It does this by using machine learning algorithms to analyze and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention [6]. Unsupervised learning is generally more inaccurate than its labelled counterpart and tends to require more computational power to train.

1) K-means Clustering

K-means clustering utilises unsupervised learning to group similar data into clusters. A data point is assigned to a cluster based on how close to the center of the nearest cluster it is. The aim of K-means clustering is to find the optimal number of clusters while having the lowest mean distance between all data points and their assigned cluster.

2) Cosine Similarity

Cosine similarity is the measure of similarity between two vectors in an inner product space. The cosine of the angle between the two vectors is measured, with the range being between -1 and 1. A value of 1 means the vectors are equal, 0 means they're perpendicular to each other and -1 means they're completely dissimilar.

V. Related Work

In 2021, Budi Juarto and Abba Suganda Girsang attempted to combine neural collaborative filtering with a sentence BERT model for a news recommender system. What they found was that after adding the sentence BERT model to the neural collaborative model, the changes were very minimal. The most drastic change was that the recall of their results dropped by 0.6% when sentence BERT was utilised. It did however increase the precision by 0.38%. The hit ratio was of note. During the fiftieth epoch, the hit ratio was 13% more than just the collaborative model, increasing the ratio from 61% - 74%. From a table they documented showing the results from three epochs, it appears that as epochs in their deep learning model increase, the hit ratio increases significantly faster than the collaborative model. They didn't perform any hyper-parameter tuning on this model but they mention that in future work it can be carried out [7].

In 2022, Li Zhang, Wei Lu, Haihua Chen, Yong Huang and Qikai Cheng investigated the performance of using BERT models to recommend similar biomedical articles. Here, they used two versions of the RELISH dataset and the TREC Genomics dataset. They compared two BERT models, BioBERT and SPECTER to several traditional NLP methods, such as document embedding models LDA and Doc2Vec and sentence embedding models WikiSentVec and BioSentVec. From their results they proved that with the RELISH dataset, BERT with fine-tuning performed better than the other methods, whereas with the TREC Genomics dataset, BERT performed almost as well as the best method, BioSentVec when using article-oriented article recommenders but it performed up to 7% less when the task was user-oriented recommenders [8].

In October 2019 Zhengjie Gao, Ao Feng, Xinyu Song and Xi Wu published a paper comparing eleven different BERT models to seven embedding models and three hand crafted features for target-dependant sentiment classification (also known as aspect based sentiment analysis). Here they created their own target-dependant BERT (TD-BERT) model which consistently outperformed other BERT models, some of which they claimed were state-of-the-art models at the time. Based off of their results they were correct. For five different tests with their accuracy and Macro-F1 scores measured, one their TD-BERT models didn't come out on top for two of the six Macro-F1 results although at least one of the three TD-BERT models was less than a percent away from the top result in both of these instances (both times another BERT architecture had the best performance). In all cases the TD-BERT models had the best accuracy. This paper has taught me a significant amount about the various BERT architectures out there to be explored, and it has enlightened me on how well a custom BERT model can perform [9].

In May 2019 Hu Xu, Bing Liu, Lei Shu and Philip S. Yu, published a paper using BERT post-training for review reading comprehension and aspect-based sentiment analysis. For this task they had over two and a half thousand questions and almost one thousand reviews, both of which were related to laptops and restaurants. They compared their four BERT models to three state-of-the-art models which had been released a year or two prior. What they found was that for their six scenarios, their BERT-PT model. This article taught me more about transfer learning and post-learning, which was a concept I was vaguely familiar with before this but I had never seen it be referred to with this name [10].

VI. Data Cleaning and Pre-Processing

For all of the code to be written for this paper, I shall be using Python. I will create a single Jupyter Notebook which will contain all of the cleaning, pre-processing, EDA and results for these two datasets.

I will use the Python module Pandas to convert the two comma-separated values (CSV) files to DataFrames, as this will be by far the easiest way to manipulate, read and analyse the data. Regex will be used when cleaning the data along with the Natural Language Toolkit (NLTK) module. NumPy will be used for calculations and to create NumPy arrays, while Matplotlib's Pyplot and Seaborn will be used to create visualisations. TensorFlow or PyTorch may be used to create my embeddings. Hugging Face's Transformers module will be used to import the pre-trained BERT models (also known as transfer learning). Scikit-learn will be used for my unsupervised learning models.

A. Amazon Book Reviews

This dataset, when initially downloaded, contained three million instances of English reviews on books. As this was far too many for my hardware to be able to load, I at first attempted to work with the first two hundred thousand rows.

The first step need to be taken was to encode the title of each book, which I inserted into a new column named "title_id". There were many users which had an ID of -1, which upon further analysis appeared to be an error thus I removed any instance with said value. I then encoded the "user_id" column as in its initial state, the values contained letters.

For some reason many of the books had multiple reviews from the same user thus to not confuse my models later, I limited each user to only one review per book ID.

Once this data was cleaned, I moved onto pre-processing it in preparation for the BERT encoding. I at first cleaned the reviews by using Regex, removing stopwords and performing stemming and localisation, but I later decided that it would be much more important to keep the full context in each review for BERT to analyse.

I tokenised by words using both the TensorFlow based TFBertTokenizer and the encode_plus models in order to evaluate their speed and outputs. What I found was that encode_plus was easier to work with when separating the input IDs, attention masks and token type IDs. Truncation had to be applied as some of the reviews extended beyond the 512 tokens BERT allows as its maximum. Next, padding had to be utilised so that each review and its three key metrics were of an equal length. These three were then converted to TensorFlow constants before being put into the TensorFlow model to create the embeddings. What I found was that my laptop would crash if I tried to embed over three hundred and fifty reviews, so I tried running the Notebook in Google Colab, where it would also run out of memory whenever I attempted to run my deep learning model.

I then attempted to use PyTorch to create the embeddings, which was successful, but because my GPU is produced by AMD thus it's not CUDA, it took a tremendous amount of time for my PyTorch code to run.

I wasn't happy with this performance however, which was when I came across the bert-base-nli-mean-tokens model. This was by far the easiest and quickest to run thus far so I decided to stick with this model to create my embeddings.

I managed to get a GPU running through Google Colab which made a significant difference to the processing time required, but it still took a very long time to create the embeddings.

B. Yelp Reviews

Fortunately, this dataset was much simpler than the aforementioned. Here, it was initially just two columns, Review and Sentiment. The Review column contained the full text for each review while the Sentiment column contained a 1, representing a negative sentiment or a 2, representing a positive.

I reduced the scope of this dataset from two hundred and eighty thousand reviews to just one hundred thousand, to save on computational power and in an attempt to not repeat the hardware crashing experienced previously.

As there were no null values and only two columns, combined with my newly gained knowledge from working with the previous dataset, there was very little cleaning and pre-processing to perform on the Yelp reviews.

Despite there being far more five-star reviews on an average business listing on Yelp than one or two stars, in this dataset reviews are slightly skewed towards the negative side, with negative reviews making up 53.2% of the total.

VII. Models and Results

A. Amazon Book Reviews

Once I had my BERT embeddings created, I was finally able to experiment with unsupervised learning methods. I chose to use cosine similarity as it's efficient with sparse data, such as these book reviews given that each user will only review a very small subset of the total number of books present. After running my unsupervised learning model and creating my function to recommend books, it was clear that using BERT embeddings was quite effective for creating recommendation systems.

For "Romeo and Juliet," the model predicts that "Prodigal Son" by Dean Koontz is the book most similar. Next is "The Scarletti Curse" by Christine Feehan before Isaac Asimov's "Foundation." Upon further research it is found that Asimov has written two books on Shakespeare's works, totalling over eight hundred pages.

""Dr. Bernstein's Diabetes Solution: The Complete Guide to Achieving Normal Blood Sugars Revised & Updated" has a seemingly strange top recommendation, that being "Economics in One Lesson." The next three books however, at least by title, are much more similar to this book, with these books being 'The Gluten-Free Bible: The Thoroughly Indispensable Guide to Negotiating Life without Wheat' by Jax Lowell, 'Semi-Homemade Cooking: Quick, Marvelous Meals and Nothing is Made from scratch' by Sandra Lee and 'Neuropsychology of Weight Control: Personal Progress Journal'.

For the book "Stopped at Stalingrad: The Luftwaffe and Hitler's Defeat in the East, 1942-1943 (Modern War Studies)", the book predicted to be most similar was curiously a manual for an aircraft boardgame. "Crimson Skies: Aircraft Manual" is the name of this book. While a board game book has very little in common with a book about World War II, the game is set in the 1930's and is about airborne military conflict. The next three books were much more similar to the original, with those being "Lenin's Tomb: The Last Days of the Soviet Empire", "Dick Bong: Ace of Aces", a book about a fighter pilot during World War II, "The Birth of Britain volume One", and finally "Homer or Moses?: Early Christian Interpretations of the History of Culture (Hermeneutische Untersuchungen zur Theologie)".

The last book had the most interesting and disappointing results. "Economics in One Lesson", which was the book predicted to be most similar to the diabetes book, but that book does not show up in the results here. Four of the five books most strongly recommended were fiction books although the second book recommended was "How to Own Your Home Years Sooner - without making extra interest payment".

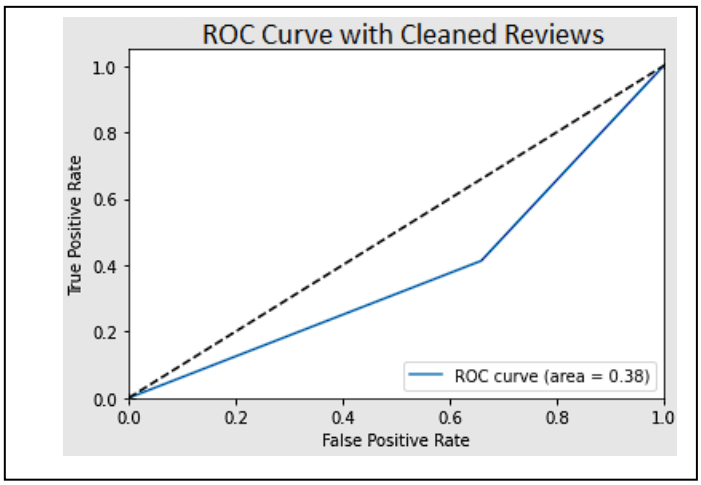
B. Yelp Reviews

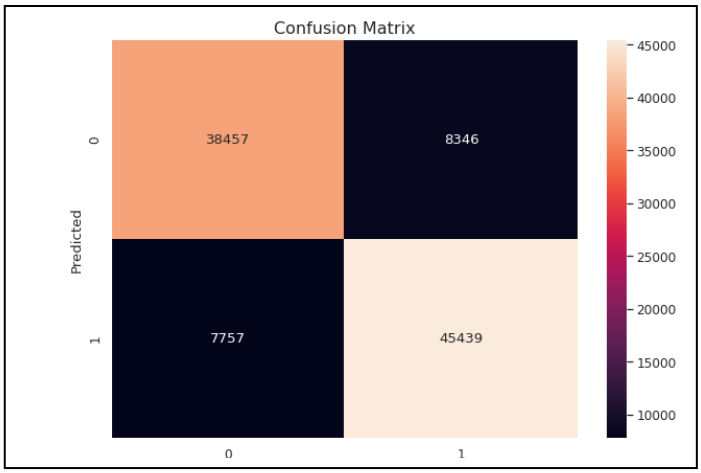
For the Yelp Reviews I used the same BERT embedding model as previously, due to the lower computational power required along with it's positive performance with the previous dataset.

Once the embeddings were created, I used K-means clustering to separate the data into two clusters. I chose two because the sentiment could only be one of two values. Just like with the actual data, there was a slight skew towards negative sentiment, although in the predictions it had increased from 53.2% to 53.79%.

Overall, this unsupervised machine learning method was 83.9% accurate with its predictions. When we investigate further, for the positive reviews the predictions were 82.17% accurate. This number rises to 85.42% for the negative reviews. When we look again at both sentiments, the precision was 84.48%, the recall was 85.42% and the F1 score was 85%.

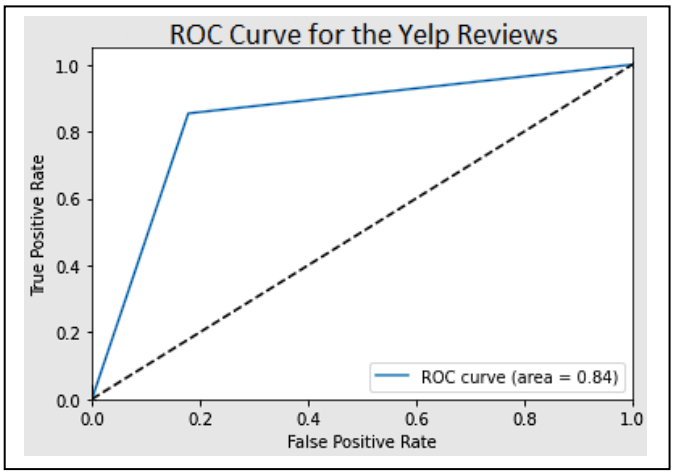
As you can see from the ROC curve, the curve score is quite high at 0.84. The confusion matrix also confirms this, showing that the model was quite accurate when predicting the sentiment of each review.





I then decided to "clean" the reviews to see how well the model would perform. To do this I removed stopwords and performed stemming and lemmatization. I then embedded these cleaned reviews using the same model and performed k-means clustering on them.

As I suspected earlier with the Amazon dataset, performing these actions on the reviews made the predictions much less accurate. Here, the accuracy was 37.92%.



VIII. Conclusions and Future Work

A. *Amazon Book Reviews*

It is clear that by using BERT encodings and cosine similarity, the unsupervised learning model with a smaller than ideal dataset size was able to distinguish the history books from the cooking while also being able to distinguish the fiction books from the other two genres. The economics book had the most interesting results though. Further research needs to be done into it to discover why four of the top five books were fiction, while the other book was very similar. More analysis needs to be done to discover if there are any other economics books in the dataset which in theory should have been recommended. 'Neuropsychology of Weight Control: Personal Progress Journal'.

It's likely that some of the books in the dataset had too few reviews or there was very little overlap between certain books or users thus an accurate recommendation could not be given.

Overall, I believe I was successful in building a recommendation system. If I were to continue this work, I would try to run my code on better hardware so that I could create many more embeddings thus increasing the robustness of my recommender. I would also experiment with more BERT models as it's still quite a new technology whose full potential is yet to be reach as well as finding a labeled dataset so that I could thoroughly measure the performance of my models.

B. *Yelp Reviews*

The key takeaway from the Yelp reviews is that when you perform some standard NLP practices on data before utilising BERT embeddings and unsupervised methods, you drastically reduce your ability to predict the sentiment.

Overall, I certainly feel as though this was a success, given the results of my "uncleaned" reviews. If I were to continue on this work I would try to encode more reviews and I would experiment more with other NLP techniques to see if I could manipulate the reviews to increase prediction accuracy without overfitting.

References

[1] Guevara, N. (2020) *Amazon's John Schoettler has helped change how we think of corporate campuses*, Available at: https://www.bizjournals.com/seattle/news/2020/11/17/skyline-shaper-john-schoettler.html [Accessed: January 26, 2023].

[2] Luckerson, V. (2014) *Yelp CEO on why he turned down google* Available at: https://time.com/3611053/yelp-ceo/ [Accessed: January 27, 2023].

[3] *Fast facts* (2022) Available at: https://www.yelp-press.com/company/fast-facts/default.aspx [Accessed: January 27, 2023]

[4] Lutkevich, B. (2020) *Bert Language Model* Available at: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model [Accessed: January 27, 2023].

[5] Wolfe, C. (2022) *Language understanding with bert*, Available at: https://towardsdatascience.com/language-understanding-with-bert-c17a453ada1a [Accessed: January 27, 2023].

[6] *What is unsupervised learning?* Available at: https://www.ibm.com/topics/unsupervised-learning [Accessed: January 27, 2023].

[7] Juarto, B. and Suganda Girsang, A. (2021) 'Neural collaborative with sentence Bert for News Recommender System,' *JOIV : International Journal on Informatics Visualization*, 5(4), p. 448. Available at: https://doi.org/10.30630/joiv.5.4.678.

[8] Zhang, L. *Lu W. Chen H. Huang Y. Cheng Q* (2022) 'A comparative evaluation of biomedical similar article recommendation,' *Journal of Biomedical Informatics*, 131, p. 104106. Available at: https://doi.org/10.1016/j.jbi.2022.104106.

[9] Gao, Z. *Feng A. Song X. Wu W.* (2019) 'Target-dependent sentiment classification with bert,' *IEEE Access*, 7, pp. 154290–154299. Available at: https://doi.org/10.1109/access.2019.2946594.

[10] Xu H. Liu B. Shu L. Yu P. S. (2019) 'BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis' Available at: https://doi.org/10.48550/arXiv.1904.02232