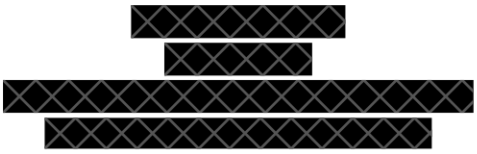
Opinion mining on Amazon product reviews



Google Drive Folder link - contains the train and test Colab notebooks, trained tokenizers and models saved for testing and data iterations used in the study:

Link to the zip file - contains all the python code and data required to run our two best-trained models on a test set (The zip file also contains a README.txt file for further details on implementation):

1 Problem statement and motivation

As the world is getting more and more digitized, e-commerce is developing at an exponential rate - enabling consumers to obtain products without leaving the comfort of their homes. Further, the benefits of e-commerce transactions and purchases extend to providing a consumer the opportunity to leave written reviews for products and also evaluate other consumers' reviews before deciding on a purchase. However, extracting information from reviews can be a tedious task and is neither efficient in terms of time nor effort. This is because products can have thousands of reviews - both positive and negative, describing varying aspects of the product - quality, value, and durability to name a few. To combat this, there is a need to develop quick and efficient ways of mining these reviews.

Our study aims to create a supervised learning model that will polarise reviews on the basis of their content as positive or negative - classes 1 and 0, respectively. This organisation of customer satisfaction will not only support the consumers in their hunt for the most suitable product, but will also enable businesses to monitor reviews, take corrective action and improve their products in an informed manner.

This motivation helps us define our research question as: How do different feature extraction techniques and classification models impact the accuracy and efficiency of supervised learning models for polarizing e-commerce reviews as positive or negative?

2 Research hypothesis

Our research hypothesis is that a transformer-based model, specifically BERT, will provide the highest accuracy for sentiment analysis of e-commerce product reviews. We hypothesize that BERT's ced capability to successfully capture the wider and deeper semantic relationships between the words in the text will enable it to capture subtle nuances in the context of product reviews, which will lead to more accurate classifications. Additionally, we believe that owing to their pre-trained contextual embeddings and advanced architecture, transformer-based models have better ability to generalize to new and unseen data - making them better suited for practical implications. (Hendrycks et al., 2020) We will test this hypothesis by comparing the accuracy of our proposed model to those achieved in previous studies using other supervised learning approaches such as SVM, MNB, and RNN-LSTM with different vectorizers like BoW, TF-IDF and Word2Vec.

As we test the validity of our research hypothesis, we will experiment with different feature extraction techniques and classification models to compare their performance and determine which approach works best - this will in turn enable us to gain insights to answer our original research question.

3 Related work and background

The task at hand, as described in the text by Popescu and Etzioni (2007), is that of opinion min-

ing. Further, they break this task down into four bite sized pieces that were customized to inform the trajectory of our study. These can be described as: (i) Identifying products being reviewed, (ii) Identifying product features that are described in the review, (ii) Identifying opinions regarding the features, and (iv) Determining the polarity of those opinions. To perform these tasks, the researchers use OPINE, an unsupervised information extraction system.

We continued the literature survey to further inform the trajectory of our study. In another study, Haque et al. (2018) utilise a supervised learning approach. They pre-processed data by executing tokenization, removal of stopwords, and POS tagging. Post this, the study uses a mix of Bag of Words, TF-IDF and Chi-square vectorizers to extract features, they then plug the outputs into Multinomial Naive Bayesian (MNB) and Linear Support Vector Machine (SVM) models. In terms of the usage of unigrams vs. bigrams, unigrams provided an accuracy higher than that of bigrams by roughly 15% - we will utilise this intel to inform our pre-processing steps. The classification task is performed using hold out cross validation and a 70:30 train-test split. This is in alignment with the supervised learning approach we will be taking in our study. They successfully divided the products being reviewed into 3 categories and achieved accuracies as high as 94% using the Linear SVM model and a 10 fold cross validation.

Another study by Bansal and Srivastava (2018) employed a Word2Vec approach to extract similar features using cosine distance - within this they identified CBOW to be a higher performing model than the skip-gram. Additionally, it concluded that Random Forests rendered a superior accuracy when compared with SVM, Logistic regression and MNB. A study by J. and Kumaran (2021) took a different approach to the Word2Vec vectorizer - using the skip-gram and TF-IDF vectorizers to extract features and used them with an SVM and RNN-LSTM model. The results were compared across several metrics including F1, Recall, Precision and Accuracy - the RNN-LSTM model outperformed the SVM in all of them. We will also be employing several metrics, as our aim is to understand the effect of various classifiers on identifying the True Positives, True Negatives and False Positives correctly. Another study by Gope et al. (2022) utilised the RNN-LSTM model to perform their classification task and concluded that this enhanced their analysis' accuracy over traditional classifiers. In terms of the architecture of the LSTM model, the general approach across all studies has been to use a word embedding layer with subsequent LSTM and regularization layers - ultimately concluding with a dense fully connected feed-forward layer and an activation function enabling the model to execute the classification. An interesting approach we came across in a study by Alsharif (2022) was to employ two LSTM layers - one with fewer units than the next to capture data patterns of varied dimensions. Through continued monitoring, we aim to employ various such methods to curate the best possible layer combination.

Upon further review, we realised that most studies pertaining to sentiment analysis on product reviews employed traditional machine learning classification models as illustrated before. Plus, all the studies that employed transformer based models for this domain and task were all conducted fairly recently. One such study by Xu et al. (2020) employed a Neural Network model developed with the Bidirectional Encoder Representations from Transformers (BERT) features. The study concluded that this implementation enabled the researchers to successfully represent the data for classification despite avoiding heavy pre-processing. We also recognized the potential of the transformer based models through various other studies that compared their performances against those of traditional models and concluded that BERT is the "ultimate advancement" in domains like Question Answering, Text Classification, and Sentiment Analysis owing to its robust nature. (Geetha and Karthika Renuka, 2021) While most studies carried out in this domain employed the BERT Uncased model, a study by Hendrycks et al. (2020) compared the performances of several other BERT models on out of sample test data to assess the performance of each. One of the conclusions of the study was - bigger models are not always better. This is why we will be using the DistilBERT model to obtain our vectors in one of the experiments in our study in order to keep our study manageable and computationally efficient. DistilBERT - a distilled version of BERT - is 40% smaller than BERT, making it 60% faster than BERT, all while retaining 97% of its capabilities. (Sanh et al., 2020)

3.1 Accomplishments

- Task 1: Transform the dataset to suitable format for pre-processing and pre-process it to filter out punctuation, special characters, stopwords, and convert to lowercase – Completed
- Task 2: Tokenised the dataset - Completed
- Task 3: Experiment with the implication of using Stemming and Lemmatization with the pre-processing - Completed
- Task 4: Build and train a Random Forest, Support Vector Machine, and Multinomial Naive Bayes models on collected dataset (without any pre-processing) using a Count Vectorizer, and examine their performance on a validation set - Completed
- Task 5: Experiment with the use of n-grams within the application of the Count Vectorizer to quantify the impact of uni-grams vs. bi-grams and tri-grams.
- Task 6: Build and train a Random Forest, Support Vector Machine, and Multinomial Naive Bayes models on various iterations of pre-processed datasets using a Count Vectorizer and a TF-IDF vectorizer, and examine their performance on a validation set to identify the best performing model - Completed
- Task 7: Apply Parts-Of-Speech tagging and Named Entity Recognition to the pre-processed data and examine the performance of the best performing machine learning model from earlier - Completed
- Task 8: Experiment with Word2Vec implications on the best performing machine learning model - Failed due to time constraints
- Task 9: Experiment with tree based feature crossing and Aspect Based Sentiment Analysis - Failed due to time constraints
- Task 10: Utilise GLoVE vectorizers to train an RNN-LSTM model and compare the performance of the same with traditional machine learning approaches - Completed
- Task 11: Experiment with various LSTM architectures and hyperparameter combinations, and analyse their implications on our study - Completed
- Task 12: Implement a variation of a BERT transformer to assess whether this is the best classifier out there for a sentiment analysis task such as ours - Completed
- Task 13: Perform in-depth analysis and various evaluation metrics to figure out what kinds of examples our approach struggles with and why - Completed.

4 Approach and Methodology

Since, we are utilising a standard dataset, our approach in this sentiment analysis study capitalised on pre-existing research to establish a pipeline that enables us to compare the performance of standard machine learning algorithms and transformer based models. Simultaneously, we wanted to identify approaches that will enable us to combat the most common obstacles in completing such tasks successfully. As described in the study by Hussein (2018), the top five challenges include: (1) negation, (2) domain dependence, (3) spam and fake review detection, (4) huge lexicon and (5) extraction of feature/keywords. We use various approaches to tackle these, as described below (The list below does not reflect the order of implementation):

1. Negation: Applying n-grams helps capture the context of the review, rather than simply relying on the presence of negative words. Additionally, we used a pre-trained BERT model that is known to perform well in understanding the context of negation. We also experimented with an RNN-LSTM as it can encapsulate the long-term dependencies in the input sequence, thereby capturing the context surrounding negated words - this can often reverse the polarity of the sentiment.
2. Domain dependence: We trained our models on a large dataset of product reviews from various categories and industries, ensuring that the models can capture the idiosyncrasies of different product domains and improving generalizability of models. Additionally, we used parts-of-speech tagging to identify domain-specific language by identifying nouns, adjectives, etc. to capture specific syntactic and semantic characteristics of the text.
3. Spam and fake: We pre-process the data heavily and normalize it. This helps remove irrelevant words and improves the quality of the data.

4. Huge lexicon: We experimented with the use of bag-of-words vectorizer with an SVM model as this provides a simplified representation of the text - an effective approach in dealing with large vocabularies. We also used a pre-trained transformer-based model (BERT) that has the ability to learn complex textual representations and is less dependent on a fixed vocabulary - allowing models to capture the meaning and context of data that may not have been encountered during training.

5. Extracting features/keywords: We use feature extraction techniques such as BoW, TF-IDF, and Word2Vec, which help capture relevant keywords indicative of sentiment. Additionally, we used POS tagging and NER to further improve the quality of our feature extraction.

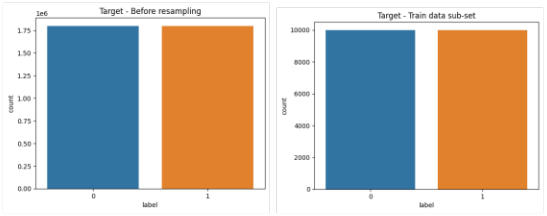
In addition to matplotlib, pandas and numpy, we utilised the nltk, sklearn, gensim, transformers, keras and tensorflow libraries for the tasks described above.

With respect to baselines, we utilise a sophisticated baseline of using a BOW representation with a Naive Bayes classifier. (Refer to section 6) The primary limitations of this baseline are, a) the inability to capture the sequential nature of language, and b) the independence assumption made by Naive Bayes, which assumes that all features (words) are independent - this may not hold true in practice. In contrast, the limitations of the project approach include the need for a large amount of training data, the potential for overfitting, and the difficulty of interpreting the results of deep learning models. These models are often considered "black boxes" because of their complex internal structures, making it difficult to interpret the results. Overall, while we do not expect our approach to fail in similar ways to our baseline, there are still limitations to consider in our approach.

5 Dataset

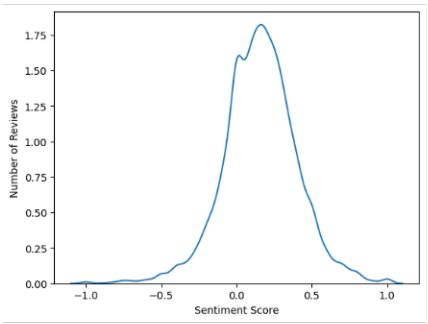
The data utilised for this study is an extensive Amazon reviews dataset acquired from the following public website: https://huggingface.co/datasets/SetFit/amazon_polarity This data was found in the .jsonl format and is annotated in entirety. The train and test files contained four sub-categories of data - the review title, the review, its label in numerical format, and its label in descriptive format. The dataset was imported directly into the notebook from the huggingface website in the arrow dataset format using the 'load_dataset("SetFit/amazon_polarity")' command.

With over 3.6 million rows of data in the training set and 400,000 rows in the test set, it would have been computationally costly to use the entirety of this for our study. Therefore, we assessed the class distribution of the main train dataset - identified it to be a balanced dataset and then, used a subset of 20,000 training data rows for our study while retaining the class balance (0: 10,000, 1:10,000). We extracted a subset of 5,000 rows from the test set - resulting in a 75:25 train:test split. Additionally, from the remaining train dataset, 5,000 rows were extracted for validation. These subsets were saved as dataframes and utilised for the study.

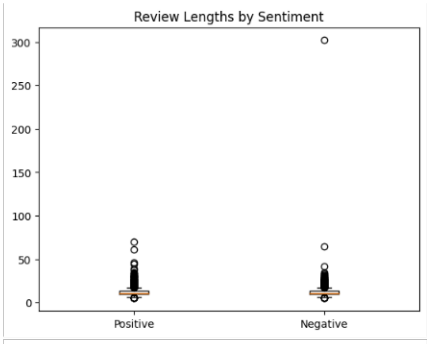


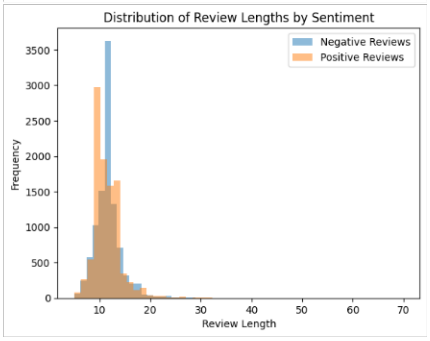
The sub-categories from earlier were translated into columns in the translation of an arrow dataset to a dataframe. We then dropped the redundant label_text column and utilised the review topics to curate WordClouds of the most common words used in the Positive reviews versus the Negative Reviews.

We observed that the most common topics in negative reviews were about cost efficiency, product quality and suitability. Another observation was that negative reviews contained a significant amount of positive words - this may restrict the model from correctly predicting the negative class samples. This observation was further reinforced by our sentiment score plot displaying the presence of more positive sentiment in the reviews despite there being an equal number of positive and negative class samples in the training set.



We also analysed the length reviews belonging to the two classes. This enabled us to identify that while there was an outlier in the negative reviews with a review length of 302 words, the reviews generally tend to be under 100 words. When this outlier was excluded, we found that positive reviews tend to be generally longer than negative reviews - this increased review length may hamper the ability of some models to correctly identify positive class instances.





Lastly, we combined the review title and text columns to curate our training data to keep our study organized and manageable without the loss of any crucial information.

5.1 Dataset preprocessing

With respect to our pre-processing, we curated a general function that was modified as per the requirement of different classification models. This function included lowercasing, removing punctuation, removing URLs, tokenizing, and removing english stopwords - helping us normalize the text data and reduce noise.

We then experimented with the implications of stemming vs. lemmatizing the output of this preprocessing function. (Refer to Supplementary material document) Ultimately, the use of lemmatization gave us the highest accuracy and recall rates - therefore our final general pre-processing function employed lemmatization.

		SVM: Validation Scores		
Dataset iteration	Vectorizer Type and Configuration	Accuracy	Recall: 0	Recall: 1
Original Dataset	Count	86.72	88	85
Pre-processed + Stemming	Count + TF-IDF	86.82	87	87
Pre-processed + Lemmatization	Count + TF-IDF (unigrams)	**86.98**	**87**	**86**

The difficulty associated with selecting a suitable set of preprocessing techniques is that there is no one-size-fits-all approach. Therefore, while this function was applied to most of our implementations, it was tweaked to, a) exclude the removal of stopwords when executing NER on the data - the presence of stopwords can provide contextual information that may be relevant, b) only include lowercasing and removal of punctuation for the DistilBERT implementation - this can prevent the loss

of crucial data that this highly powerful transformer model could learn from.

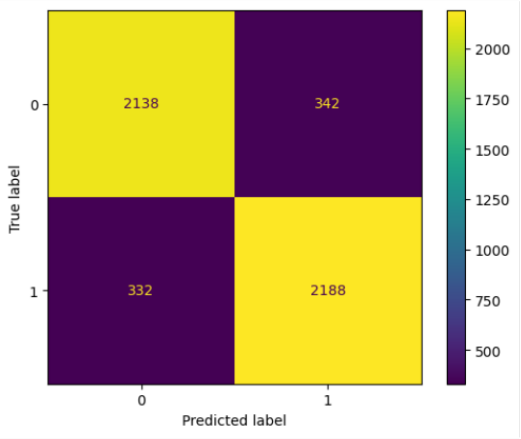
6 Baselines

We used several different baselines to guide the course of our study. For our balanced dataset, we initially started with a random guess baseline. While this baseline served as a simple and quick way to establish a benchmark for our classification task, a 50% baseline accuracy seemed like it may not be a challenging enough baseline for the model. This was deduced as a simplified BOW vector representation gave us an accuracy higher than 82% across various classification models. Additionally, the use of this baseline is tied with the assumption that positive and negative classes are equally likely to occur in any given text, which may not reflect reality. Therefore, we adapted our baseline to be a BOW representation of the train data used in conjunction with a Naive Bayes classifier. This baseline is simple in its implementation and computationally efficient. It also achieves a reasonable performance (84.8% accuracy and 80% positive class recall rate) to serve as a starting point to improve upon by tackling limitations of this implementation.

7 Results, error analysis

Post establishing a pre-processing strategy, we established a final baseline accuracy at 84.8% and a recall rate of classes 0 and 1 at 89% and 80%, respectively.

1. We began our model experimentations with use of the CountVectorizer and TF-IDF Vectorizer in conjunction with the SVM model - this classification model was identified to be the most suitable during the pre-processing and baselining stages. We used the two vectorizers together for two primary reasons, a) handling rare words, and b) accurately weighting features.



Additionally, we observed that uni-grams and bi-grams used together in the CountVectorizer rendered a higher accuracy than the use of uni-grams alone - the validation accuracy increased from 86.98% to 87.14%. Moreover, this model had a significantly higher recall rate of predicting the positive class (1) at 87% as compared with the baseline of 80%.

While we initially anticipated that the higher length of the positive reviews may pose a problem in classifying via CountVectorizer as they could cause sparsity in the feature space and make it harder for the model to learn patterns in the data, we believe the additional overlayed use of TF-IDF vectorizer and n-grams provided significant support to this implementation.

Error Analysis:

- Misclassified to 0: "2745 We might be wrong thinking those sick, vicious minds are far from our lifes and daughters...This terrible book is A WARNING for all of us who have daughters or young sisters.The stories are true, and in them you will see they are all around..."
  Here, the customer' negative comments about the content of the books are causing the misclassification, as the vectorizer has mistaken them for an overall negative review of the product.
- Misclassified to 1: "3459 GREAT FUN WHILE IT LASTED! My 7 year old son got it for Christmas. He was really excited. It is definitely a neat toy. The whole family was into it. However, it only lasted one day. It would no longer charge and I am just going to take it back

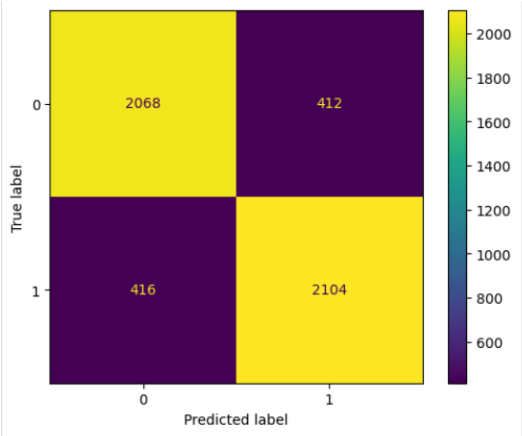
because I am in no position waste $20.00 just for one day."

This review has a significant amount of positive sentiment words - the BOW and TF-IDF approaches do not consider the ordering of words and their context resulting in the meaning of a sentence to be lost/distorted.

2. Next, we applied the POS tagging to the review text and utilised the text pre-processed using the general strategy established earlier. (Refer to Supplementary material document) We attained a validation accuracy of 83.66%. While this is lower than our baseline of 84.8%, the recall rate of the positive class is at 84% - showing an improvement from the baseline of 80%. We hypothesize that the longer positive reviews contain more complex language structures or idiomatic expressions that are better captured using a technique such as POS tagging. However, the recall rate of the negative class decreased from 89% to 83% - this may be due to the presence and tagging of positive sentiment adjectives in the negative class reviews as was illustrated in our EDA.

   In the case of NER, we used the topics of the reviews to extract the most valuable information and improve the accuracy of the NER task. We also tweaked pre-processing, eliminating the step where stopwords were filtered. However, this implementation still gave us a fairly low accuracy of 75.32%.

   Therefore, we stacked the vectors obtained from both implementations above and used them in the SVM classifier together. This implementation resulted in a 84.56% validation accuracy and a 83.44% test accuracy. The recall rates of both classes were 83% in the test run.



While this may have been our lowest performing model, it was interesting to see how stacking the two vectors enabled us to increase the accuracy and recall rates. We suspect this was possible due to the ability of each vectorizer capturing a different aspect of the review - leveraging both potentially enabled us to assimilate a more complete representation of the text in question.

Error Analysis:

- Misclassified to 0: "429 You Won't Regret Purchasing ... A MUST HAVE! To put it simply, I was in awe when I first received my copy ofAfrican American Fraternities and Sororities: The Legacy and theVision. "Insightful," "thorough," "tasteful," and "definitive" all come to mind as I think about ways to describe this comprehensive book. Every member of these organizations should have this on their shelves. Also, very interesting for anyone interested in African Americana ... It is a must read, a must!!!!"
  The review is very positive and contains several adjectives, therefore we conclude that the misclassificatio 0 may have been caused by a factor associated with the NER.
- Misclassified to 1: "609 Dissappointed read If you are a Jackie fan, i suggest you not to read this book. The book portrays her as a spoiled and phony person, who just happens to be rich, powerful and very tactful. My personal "favorite" Jackie story from this book is about how she wrote a "heartfelt" thank you letter to the person gifted her a beautiful cake that she "enjoy it very much", and turns

around to instructed secret service to "destroy it"! HOW CLASSY!"

This review has significant amount of positive sentiment words but those are being used to describe the content of the product. The POS tagging and NER approach is understandably not effective in this scenario.

3. Next, we curated an RNN-LSTM model used with pre-trained GloVe embeddings. These embeddings have been trained on large text corpora enabling them to be better at generalization and capturing nuanced relationships within text. Additionally, these embeddings are designed to capture semantic similarity. (M. Mohammed et al., 2021)

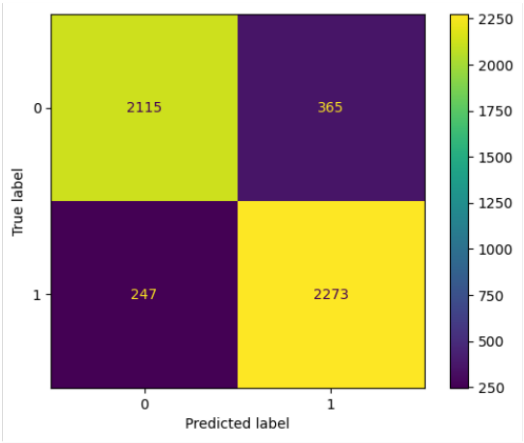
   For this, we utilised the general preprocessing approach and experimented with varied lengths of padding for the data sequences. Contrary to our anticipation, the most suitable value was 100, not the maximum sequence length of the training data. This may be because padding to the maximum length may be excessive, adding sparsity to the sequences. We then defined our vocabulary size, embedding dimension and embedding matrix. Post this, we experimented with the different layer configurations of the model over a run of 30 epochs.

	LSTM 1 units	LSTM 2 units	Dropout	Recurrent Dropout	Validation Accuracy	Validation Loss
Iteration 0	128				87.6	0.561
Iteration 1	128		0.2 x 3	0.2	88.26	0.523
Iteration 2	64	128	0.2 x 2	0.2	87.62	0.639
Iteration 3	128	256	0.2 x 3		87.5	0.528
Iteration 4	16	256	0.2 x 2		87.96	0.313
Iteration 5	**8**	**128**	**0.2 x 3**		**88.12**	**0.299**

As can be seen in the table above, we started with a model of 128 LSTM units in a single layer. We observed that despite attaining a 99% train accuracy, our validation accuracy was fairly low - to combat this we added regularization through using several instances of dropout to randomly set some of the neurons to zero during training. (Refer to Supplementary material document) We then experimented with stacking two LSTM layers to capture patterns of varied dimensions, thereby increasing the model's capacity to capture complex sequential patterns. Additionally, this also helps with regularization. (Alsharif, 2022)

The best model was iteration 5 gave us a validation accuracy score of 88.12%. While this was the slightly lower than iteration 1, the validation loss of iteration 5 was the lowest of all - indicating towards a better generalization performance.

The test run of this model gave us a high accuracy of 87.4% and high recall rates of 87% and 88% for classes 0 and 1 respectively. This model performed significantly better than the previous models - pointing towards the strong capabilities of pre-trained embeddings and an LSTM model. However, we believe the model could further be reconfigured to attain an even higher accuracy in the future.



Error Analysis:

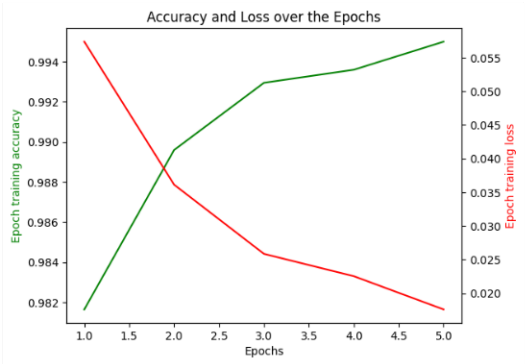
- Misclassified to 0: "4736 This is not an alanis interveiw Dont not buy this product if you think you are going to get hours of Alanis talking and being asked questions. Thats not what it is. That is what i was exprecting but no i was wrong. This CD is just this British woman talking about Alanis' life and her career. It only has 15 second clips of Alanis talking at the beginning of each track. This CD is not what it may seem. The Mini poster is kind of cool though, but overall it is not worth it."

  This review has a great deal of negation which may have caused the misclassification - proving that our LSTM model is not optimized to successfully handle such data.
- Misclassified to 1: "3459 What a surprise! This album has twice the substance of the current Sixpence None the Richer album. While it lacks a strong

pop song like "There She Goes Again", it more than makes up for it with deeply introspective lyrics and edgy guitar riffs. Leigh's vocals are breathy, haunting whispers which beg you to crank up the volume.Slap on the headphones, close your eyes and crank it as high as you can take it!"

In this case, the model may have focused too heavily on positive keywords despite the overall negative connotation of the review. It is important to note that this appears to be a complex classification.

4. Lastly, we implemented DistilBERT - a transformer based neural network. This version of BERT is smaller and faster to implement and promises high performances for general applications. With minimal pre-processing applied, we fed our training and validation data to the pre-trained 'distilbert-base-uncased' tokenizer. We used the AdamW optimizer as that incorporates weight decay - helping prevent overfitting. Moreover, we used the cross entropy loss function and initially set the number of epochs to 3 - this was later increased to 5 post monitoring training performance. This implementation enabled us to get the highest accuracies and recall rates with the least amount of efforts and computational costs.



	Number of epoch	Epoch training loss	Epoch training accuracy	Epoch training time
0	1	0.057372	0.98165	540.337843
1	2	0.036126	0.98960	540.008368
2	3	0.025848	0.99295	540.222098
3	4	0.022530	0.99360	539.789356
4	5	0.017589	0.99500	539.667577

Despite not tweaking weights and layers of the model, we attained a validation accuracy and average recall rate of 94%. In the test run these metrics were found to be at 93%. This is significantly higher than our baseline and can further be improved by added customizations.

Error Analysis:

- Misclassified to 0: "4355 Suprised This was not the type of book I expected from Joyce Carol Oates. It is a very dark tale. A 30 somthing gay male is seeking love and unquestioned devotion and thinks a zombie would meet this standard. Since there aren't too many zombies available he tries to create his own."

  This review is generally describing the content of the product in a negative light, and the initial 'surprised' may be being read as sarcasm because of this - ultimately misclassifying the review as negative.
- Misclassified to 1: "4745 Long Speech The book is a long speech to convince you that you need to re-parent yourself, and doesn't explain what this actually means. It's like going to a restaurant and talking with the waiter all evening about the menu...."

  The model perhaps was unable to understand the usage of the simile figure of speech at the end of the review. Additionally, the review may have been classified as positive due to negation and the way the book has been described.

	Test Scores of the best iterations		
Classification model configuration	Accuracy	Recall: 0	Recall: 1
Baseline: Count Vectorizer + MNB	84.8	89	80
Count Vectorizer + TF-IDF Vectorizer + SVM	86.52	86	87
POS tagging + NER + SVM	83.44	83	83
GLoVe embeddings + RNN-LSTM	87.8	85	90
DistilBERT	**93**	**92**	**94**

8 Lessons learned and conclusions

In conclusion, this study aimed to explore different approaches and the effect of compounding them to improve sentiment analysis on Amazon product reviews. We established a strong baseline accuracy using CountVectorizer and TF-IDF Vectorizer in combination with SVM, which we further improved by incorporating bi-grams. As was deduced by our initial hypothesis, the top performing model was transformer based. Coming in close after this were the GloVe with RNN-LSTM, and the CountVectorizer+TF-IDF Vectorizer with SVM models.

Furthermore, our findings suggest that while the use of POS tagging and NER can provide valuable insights into the sentiment analysis of product

reviews, it is crucial to select the parts of speech or entities to include carefully, as including irrelevant information can lead to decreased accuracy and increased noise. With respect to misclassifications, we identified the primary reason for this to be the content of products such as books and movies being described in the review - if the content described is negative, the review can be misclassified as negative despite the customer having liked the product. Additional features such as a numerical rating given to the product could be added to the data to help support the classification training in such scenarios.

Overall, the techniques explored in this study have the potential to improve sentiment analysis on product reviews, which can be valuable for businesses and consumers alike. Future research such as, a) could further hyperparameter tuning using grid search on machine learning classifiers and the RNN-LSTM model, b) fine-tuning the DistilBERT model, and c) utilising LDA for topic modelling to support aspect based sentiment analysis.

References

Nizar Alsharif. 2022. Fake opinion detection in an e-commerce business based on a long-short memory algorithm - soft computing.

Barkha Bansal and Sangeet Srivastava. 2018. Sentiment classification of online consumer reviews using word vector representations. *Procedia Computer Science*, 132:1147–1153. International Conference on Computational Intelligence and Data Science.

M.P. Geetha and D. Karthika Renuka. 2021. Improving the performance of aspect based sentiment analysis using fine-tuned bert base uncased model. *International Journal of Intelligent Networks*, 2:64–69.

Joy Chandra Gope, Tanjim Tabassum, Mir Md. Mabrur, Keping Yu, and Mohammad Arifuzzaman. 2022. Sentiment analysis of amazon product reviews using machine learning and deep learning models. In *2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, pages 1–6.

Tanjim Ul Haque, Nudrat Nawal Saber, and Faisal Muhammad Shah. 2018. Sentiment analysis on large scale amazon product reviews. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness.

Doaa Mohey El-Din Mohamed Hussein. 2018. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338.

Sangeetha J. and Dr. U. Kumaran. 2021. Comparison of sentiment analysis on online product reviews.

Shapol M. Mohammed, Karwan Jacksi, and Subhi R. M. Zeebaree. 2021. A state-of-the-art survey on semantic similarity for document clustering using glove and density-based algorithms. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(1):552.

Ana-Maria Popescu and Orena Etzioni. 2007. *Extracting Product Features and Opinions from Reviews*, pages 9–28. Springer London, London.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Shuzhe Xu, Salvador E. Barbosa, and Don Hong. 2020. Bert feature based model for predicting the helpfulness scores of online customers reviews. In *Advances in Information and Communication*, pages 270–281, Cham. Springer International Publishing.