Statistical Natural Language Processing Project - Group 6

Iunia Burlan 16096156
Malika Aidasani 17053245
Shreya Ruparelia 20146145
Tejal Gohil 20132424

Abstract

Online reviews generate vast amounts of data that can be used to gauge sentiment and inform user choices about a specific product. In this case, the dataset is comprised of movie reviews posted on the Rotten Tomatoes platform. This paper utilizes deep learning architectures in an attempt to classify these reviews into a specific sentiment ranked sequentially from positive to negative. We have employed bidirectional long short-term memory networks, BERT and XLNet and have found that, despite imbalanced data, they generate robust classifications, particularly when looking at per class accuracy.

1 Introduction

Movie reviews and ratings are useful ways to measure the performance of a particular movie and to decide whether a movie is worth one's time. 63% of US adults sometimes, mostly or always read reviews before seeing a movie (2018) [1]. A particular movie review can contain both positive and negative comments related to different topics. Sentiment analysis is a part of Natural Language Processing that extracts subjective information from the review and classifies it in a pre-defined manner. Hence, sentiment analysis is often used to assess the overall attitude of the reviewer [2]. In this project, we aim to perform sentiment analysis on the Rotten Tomatoes movie reviews dataset. The reviews in this dataset are split into 5 classes – "Negative", "Somewhat Negative", "Neutral", "Somewhat Positive" and "Positive". Previous papers on this specific dataset have used various classification techniques such as Logistic Regressions, (Multinomial) Naïve Bayes and Support Vector Machines to classify the sentiment of these movie reviews. Whilst these methods achieved satisfactory results, deep learning approaches, such as neural networks, were also employed in an attempt to improve accuracy, however have generally failed to do so for this task [3]. The objective of this paper is therefore to perform sentiment analysis on this dataset using more complex deep learning models such as Bidirectional LSTMs with an attention mechanism, BERT and XLNet. Given that these models are bidirectional in nature, they should allow us to improve the classification accuracy of movie reviews, whilst also employing an attention mechanism that places more emphasis on words that are more important given the context of the data.

2 Related work

Paper [3], published in 2012, discusses methods of predicting sentiment on a Rotten Tomatoes' movie reviews dataset. The authors focus on fine grained sentiment analysis by involving the sentiment of constituent sub-phrases. The methods implemented include simpler machine learning based text classifiers such as Multinomial Naïve Bayes and Support Vector Machines. In addition, a more complex 3-layer neural network with a logistic regression classifier was tested, where hyperparameters were tuned to optimise cross validation prediction accuracy. Multinomial Naïve Bayes gave a prediction accuracy of 75.2%, which was higher than the SVM accuracy of 72.3%. The neural network failed to give a better prediction accuracy, most likely due to the inability of the network to classify context-dependent polarity, as it failed to consider the constituents of the larger n-grams. Sentiment can be viewed as the aggregation of the components of a sentence, however information from smaller n-grams was lost when larger models were created, as they were treated independently. The paper displayed the potential of deep learning on sentiment analysis but showed that it would require more complexity

than a basic neural network architecture. Paper [4] approaches sentiment classification of movie reviews with a lexicon-based approach called SentiWordNet (SWN) along with machine learning based methods which include Naïve Bayes and SVM. SentiWord Net is an opinion lexicon containing synsets from WordNet (a lexical database), where each term is attributed a numerical score to reflect opinion polarities. The sentiments are scored across three categories: objective, positive and negative. Issues arise when using SentiWordnet to classify polarity as there is ambiguity surrounding the best method of choosing the terms to extract and how to combine the sentiment values of individual terms to determine an overall sentiment for the whole document. Here, adverbs and adjectives were extracted for use, and four variations of weighting adverbs and adjectives were explored. However, similar issues as in Paper [3] arise, where word sentiments are not considered in context of surrounding words. The accuracy of the SWN methods peaked at 65.9% when prescribing different weightings to adjective and adverbs. This proved to be much lower than the Naïve Bayes and SVM accuracies which were 81.07% and 76.78% respectively. Results from SentiWordNet concluded that adjectives are the most defining feature of text when considering sentiment and is useful when classifying polarity across different types of text; however, when the style of text is similar (as in the case of movie reviews), the accuracy cannot meet the benchmark set by Naïve Bayes and SVM. Paper [5] used the "VADER" (Valence Aware Dictionary and Sentiment Reasoner) lexicon. It is pre-trained to recognise most positive and negative words, as well as interpret punctuation and negation in this context, which is particularly useful in social media applications. As expected, the model performs very well on social media with an accuracy of 91%, but when applied to movie review data the accuracy fell to 61%. In the 1990's popularity of purely statistical models for sentiment analysis rose and the use of n-grams became integral in keeping pace with the influx of online text. Paper [6] introduces deep learning techniques such as LSTMs and compares them with the results of non-deep learning techniques such as SVM and Naïve Bayes. All the deep learning models outperformed the alternative methods when used on the IMBD dataset consisting of 50,000 movie review files. The dataset had an equal number of positive and negative reviews and the LSTM reported an accuracy of 86.64% when classifying a binary outcome variable. A similar paper [7] performed sentiment analysis using the same IMBD dataset and also posed the problem as a binary classification task. It produced an accuracy of 88.46% with 100 LSTM units. In 2017, Google gave rise to the transformer architecture, which was a paradigm shift from standard RNN's and LSTMS's. The transformer was based entirely on attention mechanisms and consequently gave rise to Google's Bidirectional Encoder Representations from Transformers. Unlike the previous models, it is 'deeply bidirectional, unsupervised language representation, pretrained using a Wikipedia and books corpus.' This is advantageous as it can learn the context of words by analysing the surrounding words. Paper [8] used BERT on an IMDB movie review dataset and produced 89% accuracy. XLNet is a generalized autoregressive pretraining method which utilizes permutation language modelling, where all tokens are predicted but in a random order, in contrast to the autoencoder language modelling as utilised by BERT. Paper [9] contained a comparison of the performance of XLNet and BERT, where the models were trained with the same data and hyperparameters to induce fairness. XLNet consistently outperformed BERT on all the considered datasets, evidencing the substantial improvement over the BERT model. Clearly, BERT and XLNet models perform the best when faced with general sentiment analysis tasks, in comparison to less complex methods featured in earlier papers such as Naïve Bayes. There does not seem to be an in-depth analysis on the performance of pre-trained models on the Rotten Tomatoes movie reviews dataset, which is why this was chosen as the topic for this project.

3 Methods

3.1 Data

The dataset is comprised of movie reviews submitted by critics and published on the website Rotten Tomatoes. The data is pre-labelled, with labels ranging from 0 to 4 for each phrase, as follows: 0 – negative, 1 – somewhat negative, 2 – neutral, 3 – somewhat positive, 4 – positive. Each sentence has been parsed into separate phrases using the Stanford Sentiment Treebank, a corpus of fully labelled parse trees that help decipher the mean-

ing of longer sentences in a principled way [10]. This way of pre-processing the data has been introduced in response to poor performance of previous models, which had only been considering words in isolation and regardless of their position in the sentence. The dataset, which was first introduced by Pang and Lee [11], consists of 8,544 sentences parsed into a total of 156,060 unique phrases. All text had been lower-cased and tags/non-English sentences had been removed. Once split into underlying phrases, each phrase had been separately annotated by a total of three human judges [10]. Many of the shorter phrases tend to be allocated to the "neutral" category, with more extreme labels being generated for the longer, more complex sentences. Considering the nature of the data, this could be due to the use of stylistic features such as negation, sarcasm, etc. and serves to prove the importance of utilizing a classification method that is sophisticated enough to process more complex groups of words. Additionally, this generates an imbalance in the data, where the vast majority of the labels fall into the "neutral" category. This is presented in Figure 1 below.

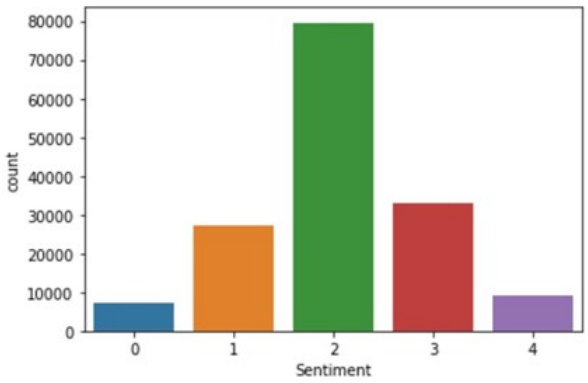

Figure 1: Sentiment votes

Given the source of the dataset, test sentences have not been labelled and, therefore, we decided to exclude them from this project. Instead, we have removed 20% of the data from the train set and have used that for testing purposes. The large number of sentences in the dataset ensures that, even if shortened to 80% of its initial size, the training set remains appropriately sized to guarantee a robust analysis.

3.2 Pre-processing

We used a total of three models for sentiment classification in this project, as detailed in Section 3.3.

The baseline method, namely a bidirectional long short-term memory ("bidirectional LSTM") network, required pre-processing of the data prior to its application. For these purposes, we first "cleaned" the data by transforming all words to lowercase and removing all pre-specified symbols. We then tokenized each phrase using the keras package tokenizer on the "cleaned" text. This tokenizer first creates a vocabulary based on the collection of words present in the dataset and assigns a numerical index to each word depending on its frequency. It then transforms each phrase into a list of integers corresponding to the indices of the respective words that form that phrase [12]. Following tokenization, we also padded/truncated each phrase in order to achieve a consistent phrase length across the data.

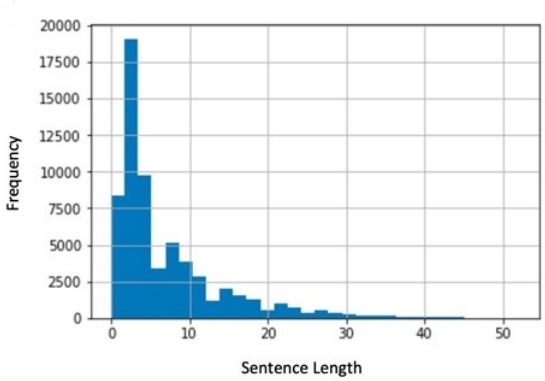

Figure 2: Frequency of sentence length

The length was chosen according to the frequency of sentence lengths (Figure 2); clearly the majority of sentences have length less than 35 therefore, to optimise training time, the maximum sentence length was chosen to be 35. We split the data thus processed into training and testing data (as detailed above) and finally, performed a one-hot encoding transformation of the labels of each phrase. This final transformation was carried out in order to facilitate the sequential classification process, given that labels are split into five categories, whereas the standard is typically two.

On the other hand, both BERT and XLNet employ their own pre-defined tokenizers, which perform all of the aforementioned pre-processing tasks automatically. In these two instances, we also padded/truncated all phrases to achieve a consistent phrase length across the entire dataset.

3.3 Model selection

3.3.1 Baseline model

As a baseline method, we utilized bidirectional LSTMs with an added attention mechanism. The concept of long short-term memory builds upon recurrent architectures and attempts to solve the issue of exploding or vanishing gradient by enforcing constant errors throughout the internal states of each unit [13]. LSTMs essentially consist of different cells, each with an input, output and forget gate, which pass or block the data they receive based on the strength of its signal. Bidirectional networks will process and run the data in both directions, once forward and once backwards, which captures a larger extent of the dependencies between words (ibid.). This also provides the network with more information which is beneficial as it improves the context available to the model. The input (1), forget (2) and output (3) gates are characterized by the following equations, where z_t represents a concatenation of the input at a specific time step and the hidden state at the previous state:

$$i_t = sigm(W^i z_t + b_i) \quad (1)$$

$$f_t = sigm(W^f z_t + b_f) \quad (2)$$

$$o_t = sigm(W^o z_t + b_o) \quad (3)$$

We also incorporated an attention layer into the LSTM architecture, which enabled the model to emphasize the words with the highest significance in attributing a sentence to a certain class. The main idea underpinning attention mechanisms is to calculate respective weights for each word in the vocabulary (4, 5), using a softmax function to account for the sequential classification. These are then utilized to calculate a weighted sum of the encoder outputs at each time step (6), which is called a context vector and is used by the decoder to compute a probability distribution over all possible outcomes [14]. The use of attention has thus allowed us to encode the relevance of all input elements in a more compact way, which can be difficult to achieve for data that is characterized by longer, more complex formulations.

$$e_{ti} = f(s_{t-1}, h_i) \quad (4)$$

$$a_{ti} = \frac{exp(e_{ti})}{\sum_{j=1}^{T} exp(e_{tj})} \quad (5)$$

$$c_t = \sum_i a_{ti} h_i \quad (6)$$

However, even in the presence of an attention mechanism, bidirectional LSTMs have been shown to suffer in practice, as the sequential information flow can affect the model's ability to capture longer term dependencies in more complex sentences [15].

3.3.2 Transformers

In an attempt to improve the baseline model, bidirectional transformer-based models - BERT and XLNet were used. As mentioned above, extreme labels were generated for longer sentences, therefore models that handle long dependency issues better than the bidirectional LSTMs are required for this sentiment analysis task. Transformers are appropriate as they are non-sequential and process each sentence as a whole as opposed to one word at a time.

Transformers have a similar architecture to LSTMs, but with a pre-encoded attention mechanism. They consist of a series of encoders and decoders where each encoder has two layers: self-attention and a feed-forward neural network.

BERT is a model which is pre-trained on unlabeled data for a range of different tasks. Its parameters are then fine-tuned using labeled data. BERT carries out the following pre-training tasks simultaneously: Masked Language Model (MLM) and Next Sentence Predictions (NSP). MLM enables the model to learn text bidirectionally and also "masks" 15% of the tokens randomly during training with the objective to correctly predict them. NSP on the other hand explores the relationship between pairs of sentences. It predicts whether the next sentence is actually the true next sentence. [16].

Since the aim of this project is to perform multi-class sentiment analysis, a BERT-base Uncased model for sequence classification was used. The BERT classifier was fine-tuned using the Adam optimizer.

The BERT model outputs logit predictions. In order to compute the probability that a sentence belongs to a certain class, a softmax function was applied to the logit predictions. After applying the softmax, the class with the highest probability was assigned to each sentence.

Due to the masking architecture of BERT, dependencies between masked tokens are not captured. To improve on this, XLNet, a generalised autoregressive model, was developed. It introduces permutation language modelling and has

been shown to result in an increase in prediction metrics in 20 language tasks when compared to BERT. Permutation language modelling predicts all tokens in a random order, whereas the more traditional language models predict tokens sequentially. In this way, XLNet is an improvement over BERT as BERT only predicts the masked 15% of tokens. Using this mechanism, XLNet is able to capture bidirectional context, similar to BERT, with the additional advantage of the absence of masking. Furthermore, XLNet is trained on 130GB of data; a much larger text corpus than BERT.

4 Experiments

4.1 Data/Sampling setup

As previously mentioned, the dataset is imbalanced such that the majority of the sentiment labels are categorized as "2 – neutral". As this can negatively affect the performance of the classification task, we have taken steps to partially mitigate this issue. Firstly, we have chosen a dataset that is large enough to ensure robust representation for all five labelling classes. Secondly, we performed a randomized 80-20 train-test split using the scikit-learn package, so as to avoid any potential bias in selecting specific chunks of the data for either the training or the testing set. For BERT and XLNet, we also employed the DataLoader function to sample batches randomly during the training process, again attempting to minimize bias in selecting specific batches in specific orders.

4.2 Parameter fine-tuning

All models were trained using a cross entropy loss refined for classification tasks with more than two classes. Additionally, we employed the Adaptive Moment Estimation ("Adam") optimizer for both the baseline and the improving models. Adam calculates adaptive learning rates for each parameter of the model by using both an exponentially decreasing average of past squared gradients and an exponentially decreasing average of past gradients, normalized for bias. Adam has been shown to work well in practice and compares favourably to similar optimization techniques [17].

4.3 Evaluation metrics

To evaluate our models, we have produced normalized confusion matrices, which are more broadly discussed in Section 5. Additionally, accuracy is calculated by dividing the number of correctly classified examples to the total number of examples in the test set. Whilst it has been used for a long time as the preferred evaluation metric in machine learning due to its ease of implementation, it has been shown to be significantly affected by class imbalances [18]. As a result, we decided to supplement our evaluation with values for precision and recall for each class for each model. Precision measures specificity and is defined as the proportion of correctly classified examples in a class and the total instances classified as being in that class. Recall measures coverage and is defined as the proportion of correctly classified examples in a class and the total instances whose true label corresponds to that class. Model selection using precision is usually used when the cost of false positive (predicted as a certain class when it does not belong to that class) is high whereas it is more useful to select a model using recall when the cost of false negative (not predicted as a certain class when it actually belongs to the class) is high [19]. In this instance, the cost of false positives is neither higher nor lower than the cost of false negatives, hence we have used both precision and recall for evaluation.

4.4 Limitations of the experiment

One particular limitation in our experiment, which has been discussed earlier in the paper, is the imbalance in the labels of our data. Imbalanced classes can affect both the accuracy of the models and some of the evaluation metrics that are most widely used for these models. Furthermore, limited computational resources also represent a challenge in terms of potential training and fine-tuning of our models. This has affected BERT and XLNet in particular, given that they are significantly more complex and require more GPU time and resources to run.

5 Results and Discussion

The baseline LSTM model achieves 65.9% accuracy, with recall and precision for individual classes provided in Table 1. Clearly the neutral reviews were the most accurately classified of all sentiments, at 79% accuracy. In addition, neutral sentiment gave the highest recall and precision (75.6% and 78.7%). Similarly, when looking on a per class basis, XLNet and BERT have highest prediction accuracy for neutral sentiment reviews

compared to other classes. The overall accuracy for BERT was slightly lower (1%) than XLNet. Whilst it is expected for XLNet to perform better than BERT, the difference in accuracy in this situation is not substantial. Generally, it seems like both models performed better in terms of accuracy for classes that have a higher frequency of data.

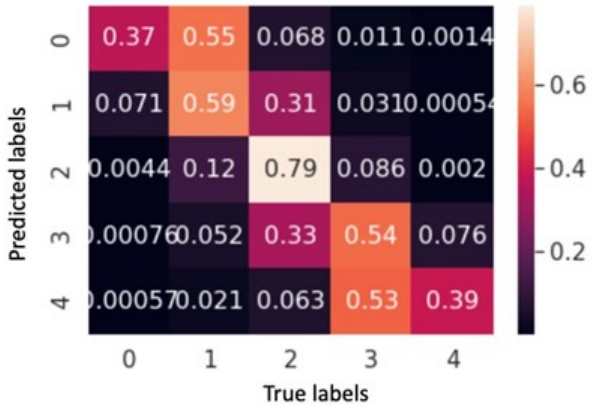

Figure 3: Bidirectional LSTM confusion matrix

Sentiments	Recall (%)	Precision (%)
0	52.73	36.72
1	51.24	59.02
2	75.57	78.69
3	58.89	54.24
4	56.20	38.80
Accuracy: 65.92%		

Table 1: Bidirectional LSTM results

However, it is interesting to see that all three models have the tendency to confuse "somewhat negative" sentences with "negative" sentences. 47% and 42% of the "somewhat negative" sentences are actually predicted as "negative" in the BERT and XLNet models respectively. The same applies for the positive classes as over 30% of the "somewhat positive" sentences are actually predicted as "positive". Therefore, it would perhaps be possible to achieve more accurate results by combining the negative classes and the positive classes into one negative and one positive class in order to only have three classes in total (negative, neutral and positive).

The recall and precision for the different classes are quite similar for BERT and XLNet, with XLNet performing slightly better. Similar to the results for accuracy, higher recall and precision were obtained for the "neutral" class whilst lower

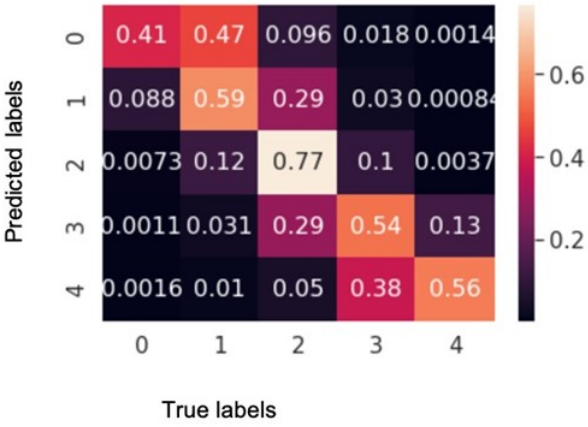

Figure 4: BERT Confusion Matrix

Sentiments	Recall (%)	Precision (%)
0	48.23	41.44
1	56.16	58.82
2	75.22	76.81
3	59.01	54.41
4	52.81	55.97
Accuracy: 65.76%		

Table 2: BERT results

values corresponded to the "negative" and "positive" classes. In general, the LSTM model gave lower precision values when compared to BERT and XLNet, except for the "neutral" sentiment reviews. This suggests that, compared to the other models, LSTMs are more likely to predict a certain class when reviews do not actually belong to the class.

The accuracies for all three models were fairly similar, and all three were slightly lower than accuracies seen in past papers. This may be due to limitations of the dataset itself, such as imbalance in the sentiment categories and number of classes. The increased number of classes (5) compared to binary sentiment in previous papers may have resulted in ambiguity between the two positive classes and the two negative classes. This can be seen in the confusion matrices. Furthermore, there seems to be a large number of neutral reviews misclassified as somewhat positive and somewhat negative. This may be a result of the presence of words with usually positive or negative connotations, whose larger attention weighting influences the overall sentiment prediction.

Figure 1 shows a clear majority of neutral reviews; the dataset contained sub-sections for each

review with their own ratings which led to an increased number of neutral reviews (particularly for the smaller size n-grams which did not carry any sentiment). Furthermore, there were very few extremely positive or extremely negative reviews. Due to this imbalance, sentiments were more likely to be predicted as 'neutral' and less likely to be predicted as 'positive' or 'negative', regardless of the underlying true sentiment. To check if this had a significant impact on model evaluation metrics, a new dataset was developed with 40,000 randomly selected reviews with neutral sentiment removed, resulting in better balance across the middle three sentiments. Train, validation and test data were extracted from this new dataset and new BERT and XLNet models were trained. In both cases, although accuracy decreased (to be expected with reduced training data), the precision and recall for all sentiment classes (aside from neutral) increased. This suggests that a more balanced dataset leads to a more reliable model when identifying non-neutral sentiments. To improve on this further, reviews with extreme underlying sentiment (0 or 4) should be generated to ensure balance across all sentiments.

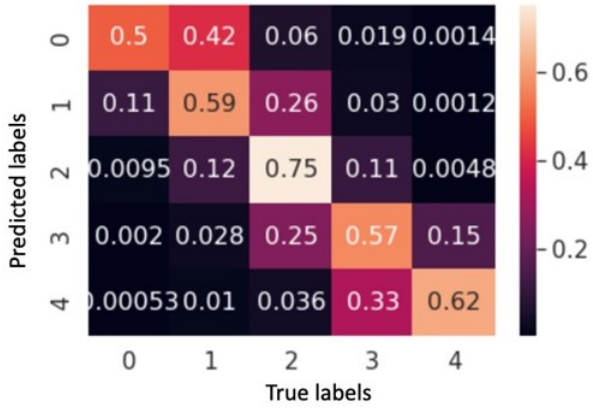

Figure 5: XLNet Confusion Matrix

Sentiments	Recall (%)	Precision (%)
0	46.48	49.90
1	56.77	59.40
2	77.66	75.26
3	59.72	56.97
4	51.51	62.40
Accuracy: 66.43%		

Table 3: XLNet results

6 Conclusion and Further Work

This paper builds on existing work in sentiment analysis by employing the transformer architecture on the movie reviews dataset published by Rotten Tomatoes. Previous analysis had focused more on non-deep learning methods as well as recurrent architectures, all of which tend to struggle with more complex phrase dependencies. Despite challenges with the setup of the data, which resulted in slightly lower accuracies for both BERT and XLNet compared to the baseline method, we have shown herein that transformers can generate robust results in sequence classification and, with further computational power, could outperform more standard methods. The research carried out in this paper suggests that XLNet performs better relative to Bidirectional LSTMs and BERT when tasked with classifying polarity of movie reviews.

This work could be further expanded through the fine-tuning of BERT and XLNet using the SST-2 corpus of the General Language Understanding Evaluation benchmark ("GLUE"). GLUE represents a collection of natural language understanding tasks that help evaluate models across a range of different NLP tasks [20]. Additionally, further features that can address sarcastic formulations, ambiguity in phrasing, etc. would also be useful in modelling this particular dataset. Sarcasm detection has been at the forefront of NLP research recently, however the lack of existing labelled data for this task in the movie reviews space precluded us from incorporating such a feature in this project. Lastly, aspect-based sentiment analysis could also be explored to further our analysis on this dataset and it would potentially be extremely valuable to users of Rotten Tomatoes that are interested in different aspects of a movie.

References

[1] Julia Stoll. Movie review readers u.s. 2018. Available at https://www.statista.com/statistics/898999/reading-reviews-before-viewing\protect\discretionary{\char\hyphenchar\font}{}{}movies-united-states/ (2021/01/13).

[2] Amey Parulekar Ankit Goyal. Sentiment analysis for movie reviews. Available at https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf (2005/06/12).

[3] Yuanyuan Pao Jean Y. Wu. Predicting sentiment from rotten tomatoes movie reviews. Unpublished Manuscript, 2012.

[4] V.K. Singh, R. Piryani, A. Uddin, and P. Waila. Sentiment analysis of movie reviews and blog posts. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 893–898, 2013.

[5] Paschalis Frangidis, Konstantinos Georgiou, and Stefanos Papadopoulos. Sentiment analysis on movie scripts and reviews. In Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis, editors, *Artificial Intelligence Applications and Innovations*, pages 430–438, Cham, 2020. Springer International Publishing.

[6] Nehal Ali, Marwa Hamid, and Aliaa Youssif. Sentiment analysis for movies reviews dataset using deep learning models. *International Journal of Data Mining Knowledge Management Process*, 09:19–27, 05 2019.

[7] Shareef Shaik Jyostna Devi Bodapati, N. Veeranjaneyulu. Sentiment analysis from movie reviews using lstmss. *Ingenierie des Systemes d'Information*, 24:125–129, 11 2018.

[8] Saad Abdul Rauf, Yan Qiang, Syed Basit, and Waqas Ahmad. Using bert for checking the polarity of movie reviews. *International Journal of Computer Applications*, 177:37–41, 12 2019.

[9] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[10] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[11] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1–2):1–135, January 2008.

[12] Keras. Text data preprocessing. Available at https://keras.io/api/preprocessing/text/#tokenizer-class (2021/05/30).

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[14] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention, please! A critical review of neural attention models in natural language processing. *CoRR*, abs/1902.02181, 2019.

[15] Yue Zhang, Qi Liu, and Linfeng Song. Sentence-state LSTM for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[18] Brendan Juba and Hai S. Le. Precision-recall versus accuracy and the role of large data sets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4039–4048, Jul. 2019.

[19] Koo Ping Shung. Accuracy, precision, recall or f1? Available at `https://towardsdatascience.com/accuracy-precision-recall-or-\f1-331fb37c5cb9` (2018/03/15).

[20] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.