Deep Sequence Models for Subjectivity and Polarity Classification

Filippo Momesso (229298)

University of Trento

filippo.momesso@studenti.unitn.it

Abstract

This document contains the report for the final project developed for the course Natural Language Understanding of the M.Sc. in Artificial Intelligence Systems at the University of Trento. The objective is to develop Sentiment Analysis models for solving subjectivity detection and polarity classification tasks. The code is available at https://github.com/Momofil31/NLU-SA-Project.

1. Introduction

Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. It is relevant in several applications, ranging from marketing and customer feedback analysis to politics. The focus of this work is on two particular tasks of sentiment analysis: subjectivity detection and polarity classification.

Historically, lexicon-based [1], heuristics-based [2] or term-counting-based [3] approaches have been used and research has shown that shallow machine learning algorithms such as Naive Bayes or SVMs have competitive performance [4, 5]. More recently, deep learning took the lead in many fields, from Computer Vision to Natural Language Processing, and sequence models such as RNNs and Transformers made a huge step forward in terms of performances in several NLP tasks [6].

In this work, I use various deep learning models to tackle subjectivity detection and polarity classification tasks and compare their performances showing that transfer learning on pretrained large transformer models gives astonishing results in terms of accuracy, at the cost of higher computational needs, while smaller recurrent neural networks with an attention mechanism offer a reasonable trade-off between the two.

2. Task Formalisation

I solve two sentiment analysis tasks in this project: subjectivity detection and polarity classification.

2.1. Subjectivity detection

Subjectivity detection is a binary text classification task in which, given a sentence, the goal is to classify if the sentence is *objective* or *subjective*. An objective sentence is a sentence that expresses a fact or objective reality, rather than a personal opinion, for example, the sentence *"I bought an iPhone a few days ago."*. A subjective sentence, instead, expresses a personal opinion or perspective, such as the sentence *"It was such a nice phone."*.

2.2. Polarity classification

Polarity classification is a binary[1] text classification task in which, given a sentence or a document, the goal is to classify if the sentence is *positive* or *negative*. For example, predict if a movie review is positive ("thumbs up") or negative ("thumbs down").

3. Data Description & Analysis

For this project, I use the *movie reviews* and the *subjectivity* datasets [4], both available in the NLTK library. In Table 1, a summary of the two dataset statistics is available.

3.1. Subjectivity

The subjectivity dataset first used in [4], is a dataset of sentences divided into 5000 subjective sentences (or snippets) and 5000 objective sentences. Each sentence is at least 10 tokens long. Each token is down-cased and separated by whitespace; punctuation and stopwords are not removed.

To build the dataset, the authors extracted sentences from snippets of movie reviews from Rotten Tomatoes and plot summaries of movies from IMDb. The labeling procedure is performed automatically by assigning all sentences from Rotten Tomatoes reviews as subjective and all sentences from plot summaries as objective. This choice, however, introduces mislabeling errors since plot summaries can contain subjective sentences and reviews might contain objective sentences.

As shown in Table 1, subjectivity datasets sequences are on average relatively short, having a length of 24 tokens with very low variance, hence extreme cases such as 120 tokens maximal length have a very low frequency. After removing NLTK stopwords and punctuation, the lexicon size is 23737 words, and the 10 most common words are almost all nouns for both classes. The number of words that belong to the lexicons of both classes is 6243, hence 17494 words are belonging to a single class[2], roughly divided into equal parts. This means that more words are part of the lexicon of a single class than the number of words that are shared by the two lexicons.

3.2. Polarity

The movie reviews dataset (polarity dataset v2.0), used for the first time in [4], is a dataset composed of 1000 negative and 1000 positive movie reviews. Each review can be composed of multiple sentences, and, similarly to the subjectivity dataset, each token is down-cased and separated by whitespace.

The reviews that compose the dataset are taken from the IMDb archive of the *rec.arts.movies.reviews* group processed

[1] Note that polarity classification is considered, in some cases, a ternary classification task in which the classes are *positive*, *negative*, *neutral*.

[2] This peculiarity justifies the performances of the baseline model described in Section 5.

Table 1: *Datasets statistics.*

Dataset	# of sequences	Sequence length			Lexicon size	Intersection lexicon size	Class 0 only lexicon size	Class 1 only lexicon size
		Avg	Max	Min				
Subjectivity	10000	24.06±9.94	120	10	23737	6243	9074	8420
Movie Reviews	2000	791.91±347.25	2879	19	39587	18954	11282	9351
Movie Reviews Filtered	2000	501.37±257.57	2169	19	30088	13667	7949	8472

automatically to remove the rating information. However, if the original review text contains several instances of rating, potentially in different forms, those not recognized as valid IMDb ratings remain part of the review text. To label the reviews the authors assigned to the positive class a review, according to the first rating identified when rated as:

- 7 and up if grading is from 0 to 10,
- 3.5 and up stars if grading is up to five stars,
- 3 and up stars if grading is up to four stars,
- B or above in the case of a letter grade system.

Complementary cases are labeled as negative.

As shown in Table 1, the sequences are considerably longer than the ones of the subjectivity dataset reaching an average of 791 words per sequence with relatively high variance, hence sequences exceeding 1000 words are quite common. Moreover, the lexicon size is almost doubled, and behavior similar to the subjectivity dataset can be observed in the number of tokens belonging to the lexicon of only one class, with the difference that the number of tokens belonging to the intersection of the two classes' lexicons is roughly doubled. This means that positive and negative reviews share more words than the number of words that are not common to both.

3.3. Polarity Filtered

The approach used in [4] to address polarity classification leverages a pre-processing of reviews that removes the objective sentences (such as plot summaries) to obtain "an *extract* that should better represent a review's subjective content". Moreover, removing objective sentences reduces the review's total sequence length. To remove the objective sentences I use the baseline subjectivity detector described in Section 4.

Table 1 shows that the average sequence length is reduced by roughly 400 tokens, and a lexicon considerably smaller is obtained. The proportion between the size of the lexicon intersection and the number of words belonging to only one class remains approximately the same as the original movie reviews dataset.

4. Models

This section describes the architecture of the models developed and the pipeline used to apply them to the datasets described in Section 3. During the development of the project, I followed an incremental approach, starting from simpler models and gradually increasing the complexity to reach higher accuracy. This section follows the same order. The code is based primarily on PyTorch and Scikit-Learn libraries.

The focus of this project is deep sequence models, however, as a baseline, I use a simple Multinomial Naive Bayes model with a bag-of-words representation of the sequences similar to what has been done in [4], as required by the project instructions. The baseline pipeline is as follows: 1. tokenize the sequences and remove stop-words, 2. vectorize using a *CountVectorizer* which converts the sequence to a matrix of token counts, and 3. train a Multinomial Naive Bayes classifier.

4.1. Small Sequence Models

The first idea employed a recurrent neural network model to address the sequences directly without the need to use a bag-of-words representation. Hence I developed a two layers Bidirectional GRU [7]. The *Gated Recurrent Unit* is an improvement of the standard recurrent architecture, similar to the LSTM [8], which employs the use of gates to decide how the information is remembered and passed to the output. Moreover, a bidirectional GRU processes the sequence in both directions. Compared to LSMTs, GRUs have fewer parameters, therefore more memory efficient and faster.

The following step was to add an attention mechanism to the Bidirectional GRU so that the model is able to give more importance to certain words or groups of words in the sequence. In order to do so I developed a soft-attention module based on the approach proposed in [9] for neural machine translation, which is a fully-connected layer with dropout that provides probabilities scores α_i for each token in the sequence. These α_i scores are then used to compute a weighted sum of the tokens' word embeddings which is the input to the final classification layer.

The last model developed for the "small models" category is TextCNN [10], a simple convolutional neural network that employs one layer of 1D convolutions with multiple filters with different sizes.

All the models described rely on learnable word embeddings to represent the tokens of the sequences, on top of which the recurrent or convolutional architecture is built. To learn the embeddings I experimented with two approaches: 1. learning from random initialization while training the network, 2. using GloVe [11] pre-trained embeddings to relieve the training from the burden of learning the token representations.

Furthermore, the sequences are tokenized using NLTK *WordPunctTokenizer* which tokenized the text based on a regex formula before feeding them to the model.

4.2. Transformer models

The introduction of the Transformer [12] was a game changer in the NLP community in terms of performance on several tasks, including text classification, founding a new class of large models based on self-attention such as the BERT [13] and GPT [14, 15, 16] families. Thus, I decided to experiment with large pre-trained transformer models available in the HuggingFace library.

For the subjectivity dataset, I used an instance of DistilBERT [17], a distilled version of pre-trained BERT Base, appending a classification head to output the class logits for this specific task. For the polarity and polarity-filtered dataset, I used an instance of RoBERTa [18] pre-trained on 58 million tweets and already finetuned for sentiment analysis on the

TweetEval [19] benchmark. However, this model is trained for a three-class task (positive, negative, and neutral), hence, I substituted the classification head with a new binary one specific for the movie review polarity classification task.

RoBERTa, which is based on BERT, has a sequence length limit of 512 tokens, while, as discussed in Section 3, sequences in the movie review dataset exceed, on average, this value. For this reason, performing sequence truncation is needed in order to feed the sequence to the model. This operation introduces information loss, therefore I investigate various truncation strategies, as described in [20], such as head, tail, and head-tail truncation.

The self-attention mechanism of the Transformer, and inherently all BERT-based models, has quadratic complexity relative to the sequence length, hence the limit to 512 token sequences. To account for this limitation, research proposed several transformer models with linear complexity attention obtained through different techniques, such as low-rank approximation [21, 22], local-global attention [23, 24], and using softmax as a kernel [25, 26, 27].

To address long sequences, minimizing the need for sequence truncation as much as the GPU at my disposal allowed, I decided to use an instance of Big Bird [24], which reduces the self-attention complexity to linear maintaining the same expressivity of standard self-attention by: *a*) restricting each token to only attend to a window of size w (local attention), *b*) adding global attention on a selected subset of tokens and, *c*) introducing "random attention", where each token in the sequence attends to a few randomly selected tokens. With the GPU at my disposal, I was able to process sequences up to 1024 tokens long, therefore avoiding the truncation of a much larger number of reviews, reducing considerably the information loss.

5. Evaluation

To perform experiments on the models presented in Section 4 I used Nvidia Tesla T4, Quadro RX5000, and RTX A4000 GPUs with 16GB of memory each provided by Google Colab and Paperspace Gradient depending on what was given by the service at that moment.

Each experiment consists of training and evaluation of the model via 5-fold cross-validation[3] in order to average the evaluation metrics and obtain values that are less dependent on the data split choice and random initialization of the weights. All the models are optimized with Adam [28] optimizer through binary cross-entropy loss. For each fold, the best model checkpoint in terms of accuracy was kept and used for the final evaluation. The models introduced in Section 4.1 have been trained for 30 epochs while models introduced in Section 4.2 have been trained for 10 epochs. Hyperparameters for each run are available in the WandB project.

The metrics used for evaluations are binary classification accuracy and F-1 score, computed as follows:

$$accuracy = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y_i = \hat{y}_i) \tag{1}$$

$$F_1 = 2\frac{precision * recall}{precision + recall} \tag{2}$$

where N is the number of samples, y_i and $\hat{y}_i$ are the predicted and the ground truth class for the i-th sample. Accuracy represents the percentage of correctly predicted samples, while the F-1 score is the harmonic mean of precision and recall. Table 2 displays the results in terms of accuracy and F-1 score for the experiments performed.

[3] 2-fold in the case of Big Bird due to time restrictions.

5.1. Baseline

As shown in Table 2, the simple Multinomial Naive Bayes classifier used as the baseline presents a really high accuracy and F-1 score on the subjectivity dataset while reaching lower but rather strong results on the polarity task with roughly a 3% increase in both the metrics, consistently with what is described in [4].

Such results on the subjectivity dataset can be obtained thanks to the relatively large size of the dataset (compared to the polarity dataset), which leads to reduced overfitting, and the observation made in Section 3.1 on the substantial difference between the lexicons of the two classes in terms the words that compose them. A bag-of-word approach, indeed, can benefit a lot from data of this kind.

Although the accuracy obtained is high, the model is not really able to learn what a subjective sentence is but, rather, behaves like an "implicit term counting" approach by learning to ignore common words and focusing on the words belonging to a single class. To corroborate this hypothesis, I generated a set of adversarial sentences that should mislead the model using ChatGPT, available in Appendix B. These sentences are either subjective or objective but are generated using words that belong only to the lexicon of the opposite class, extracted as explained in Section 3. As expected, the baseline classifier is not able to correctly classify those sentences.

On the movie review dataset, performances are lower probably due to the considerably higher length of the sequences and the fact the abovementioned phenomenon is less pronounced.

5.2. Small Sequence Models

In all tasks, the simple bidirectional GRU is not competitive with respect to the baseline approach and fails especially in the polarity classification task, where the average sequence length is much higher, with an accuracy that is slightly higher than random guessing. The reason behind this behavior is discussed by Badhanau *et al.* when, in relation to sequence-to-sequence models, they state that "A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus." [9].

In my case, it is highly probable that the GRU suffers from the same issue. In other words, using the last hidden state as input to the classification layer gives poor results since the RNN forgets older important words, and the hidden state representation does not account for them. The RNN in fact encodes the entire sequence in the final hidden state which can cause information loss as all information needs to be compressed into a global vector representation.

By implementing a soft-attention mechanism on the token representation, delegating the production of a feature representation of the entire sequence to a weighted sum of the token-wise output representations, as described in Section 4.1, the BiGRU with attention model manages to slightly improve the accuracy over the baseline.

In Appendix A, attention heatmaps of correctly classified sentences with high-confidence (Figures 1, 2) for polarity task, show that the model correctly attributes importance to crucial

Table 2: *Comparison on subjectivity, polarity, and polarity after removing objective sentences tasks. The highest performances, both in accuracy and f-1 score, are obtained with transformer-based models pre-trained on larger datasets. (*Pretrained models)*

Model	GloVe	Subjectivity		Polarity		Polarity-filtered	
		Accuracy	F-1	Accuracy	F-1	Accuracy	F-1
Baseline		92.02 ±0.73	92.13 ±0.74	81.45 ±1.60	81.11 ±1.82	84.40 ±2.15	84.13 ±2.30
BiGRU		89.48 ±0.83	89.55 ±0.98	62.60 ±2.84	61.76 ±4.30	63.35 ±2.30	63.61 ±2.45
	✓	93.07 ±0.27	93.07 ±0.25	78.65 ±4.24	78.59 ±6.40	80.10 ±1.23	80.26 ±2.15
BiGRUAttention		89.70 ±0.56	89.91 ±0.65	83.10 ±2.21	83.14 ±2.25	85.25 ±2.00	85.29 ±2.47
	✓	**93.67** ±0.34	**93.70** ±0.33	**89.80** ±1.62	**89.69** ±1.65	**89.85** ±1.42	**89.92** ±1.62
TextCNN		91.02 ±0.38	90.99 ±0.36	80.30 ±2.40	80.44 ±2.20	83.15 ±2.48	82.87 ±3.09
	✓	92.02 ±0.58	91.99 ±0.63	85.45 ±1.59	85.54 ±1.57	86.60 ±1.49	86.83 ±1.79
DistilBERT*		**96.70** ±0.23	**96.72** ±0.23	–		–	
RoBERTa* (head)		–		91.45 ±1.05	91.55 ±0.98	95.25 ±0.90	95.27 ±0.90
RoBERTa* (tail)		–		94.65 ±0.80	94.58 ±0.84	95.70 ±0.84	95.65 ±0.84
RoBERTa* (head+tail)		–		94.05 ±0.72	94.13 ±0.78	95.00 ±0.50	94.99 ±0.46
Big Bird*		–		**96.62** ±0.13	**96.61** ±0.15	**96.75** ±0.50	**96.80** ±0.51

words or group of words. In Figure 3, which reports a misclassified review, it appears that the model is able to find the meaningful parts of the sentence but outputs a wrong label. However, the prediction appears to be correct with respect to the contents of the reviews. By inspecting more reviews, I observed that similar cases are common. Conversely, Figure 4 reports a wrongly classified review that seems to have a ground truth label that is consistent with the contents. In this case, it seems that the model is able to identify the important parts of the review but fails in capturing their meaning, therefore leading to a wrong prediction.

The TextCNN model is able to reach performances similar to the baseline model but still lower in all tasks. In this case, the problem is intrinsic to the CNN architecture which lacks the ability to capture long-term dependencies between words due to its local inductive bias.

As predicted, removing the objective sentences in the polarity classification leads to an increase in accuracy and F-1 score for all the models in this category. The intuition is that shorter sequences are easier to process, and the model does not have to identify which parts need to be ignored (eg. plot summaries are mainly objective and usually contain information less relevant to the review polarity). This is confirmed by the attention heatmap on a correctly classified review in Figure 5, which shows that the attention scores are high for almost all the words in the sequence, which contains only subjective sentences.

As expected, all models benefit from the use of pre-trained GloVe word embeddings, obtaining a substantial increase in accuracy, allowing all the models to beat the baseline in all tasks except for the simple Bidirectional GRU.

Independently on the use of GloVe vectors the models discussed in this section succeed in classifying correctly the adversarial examples described in Section 5.1. This observation is a strong indicator that these sequence models are able to actually capture the meaning and contents of the reviews, instead of simply learning the dataset's word distribution.

5.3. Transformer models

As shown in Table 2, large pre-trained Transformer models introduced in Section 4.2 manage to outperform by a large measure the small sequence models. The main reason behind this behavior is the strong ability of these models to understand the nuances of meanings in natural language thanks to the unsupervised multi-task pre-training on huge datasets. Moreover, they are able to fit sentences or snippets in a global context thanks to self-attention, therefore are less prone to misinterpret some words or expressions.

As discussed in [20], in a text classification scenario, the truncation strategy may directly influence the model's performance since crucial information might be cut out. For this reason, I investigated several truncation approaches for sequences longer than 512 tokens.

Table 2 shows that, on this dataset, the best approach among the three proposed is to keep the tail of the sequence, truncating the beginning section. This works better because movie reviews usually begin with a plot summary or in general less opinionated sentences. At the same time, the middle section and especially the end contain final considerations on the film that strongly drive the polarity of the review.

The best approach overall in terms of pure metrics is Big Bird which outperforms all the previously discussed models. Its strength is dependent on the features common to the Transformers of the BERT family described at the beginning of this Section, as well as the ability to process sequences that are doubled in length with respect to RoBERTa. This leads to much less information loss and better generalization capabilities after the training.

Moreover, it is possible to notice that, similarly to the other methods, the accuracy of the polarity classification task after filtering the objective sentences is slightly higher.

This improved performance comes at the cost of longer training time and higher computational resources required for the Transformer models, especially Big Bird, which makes hyperparameter search and experiments much more time-consuming, as shown by Table 3 in Appendix C.

6. Conclusion

In conclusion, this report discusses deep learning approaches for subjectivity detection, and polarity classification tasks by using deep learning, showing that, differently than shallow machine learning methods, deep models can fully understand the contents of a text and classify it accordingly.

Furthermore, a reasonable compromise between computational requirements and accuracy can be obtained by Bidirectional GRUs with a soft-attention mechanism, while the best performance is obtained by large pre-trained transformers such as Big Bird at the cost of higher computational needs.

7. References

[1] A. Esuli and F. Sebastiani, "SENTIWORDNET: A publicly available lexical resource for opinion mining," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf

[2] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, pp. 216–225, May 2014. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550

[3] A. Kennedy and D. Inkpen, "Sentiment classification of movie and product reviews using contextual valence shifters," 2005.

[4] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *Computing Research Repository - CORR*, vol. 271-278, pp. 271–278, 07 2004. [Online]. Available: https://arxiv.org/abs/cs/0409058

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, Jul. 2002, pp. 79–86. [Online]. Available: https://aclanthology.org/W02-1011

[6] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, "Natural language processing advancements by deep learning: A survey," 2020. [Online]. Available: https://arxiv.org/abs/2003.01200

[7] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014. [Online]. Available: https://arxiv.org/abs/1409.1259

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: https://arxiv.org/abs/1409.0473

[10] Y. Kim, "Convolutional neural networks for sentence classification," 2014. [Online]. Available: https://arxiv.org/abs/1408.5882

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[14] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018.

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[17] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[19] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "Tweeteval: Unified benchmark and comparative evaluation for tweet classification," 2020. [Online]. Available: https://arxiv.org/abs/2010.12421

[20] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" 2019. [Online]. Available: https://arxiv.org/abs/1905.05583

[21] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," 2020. [Online]. Available: https://arxiv.org/abs/2006.04768

[22] Y. Xiong, Z. Zeng, R. Chakraborty, M. Tan, G. Fung, Y. Li, and V. Singh, "Nyströmformer: A nyström-based algorithm for approximating self-attention," 2021. [Online]. Available: https://arxiv.org/abs/2102.03902

[23] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[24] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big bird: Transformers for longer sequences," 2020. [Online]. Available: https://arxiv.org/abs/2007.14062

[25] C. Zhu, W. Ping, C. Xiao, M. Shoeybi, T. Goldstein, A. Anandkumar, and B. Catanzaro, "Long-short transformer: Efficient transformers for language and vision," 2021. [Online]. Available: https://arxiv.org/abs/2107.02192

[26] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, "Rethinking attention with performers," 2020. [Online]. Available: https://arxiv.org/abs/2009.14794

[27] H. Peng, N. Pappas, D. Yogatama, R. Schwartz, N. A. Smith, and L. Kong, "Random feature attention," 2021. [Online]. Available: https://arxiv.org/abs/2103.02143

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

A. Bidirectional GRU Attention Heatmaps

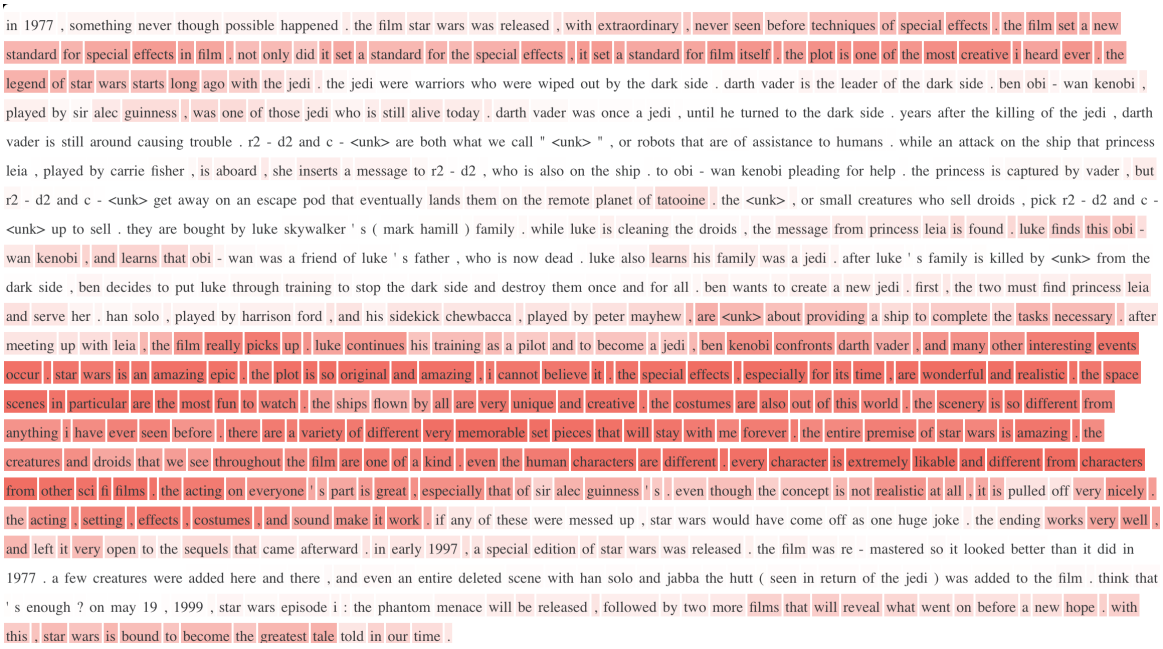

Figure 1: *Attention heatmap on a correctly classified positive review for the polarity classification task.*

Figure 2: *Attention heatmap on a correctly classified negative review for the polarity classification task.*

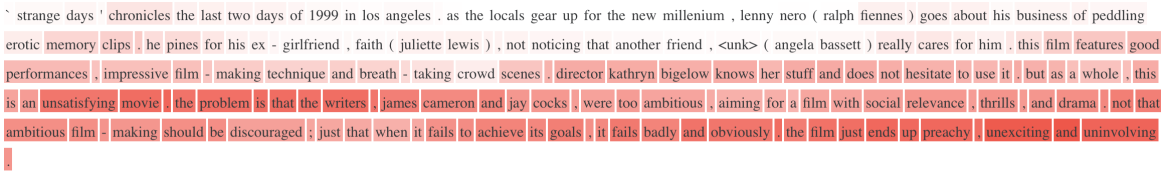

Figure 3: *Attention heatmap on a wrongly classified positively labeled review for the polarity classification task. This review, however, has a negative connotation, and therefore it is highly probable that the ground truth is wrong.*

Figure 4: *Attention heatmap on a wrongly classified negative review for the polarity classification task.*

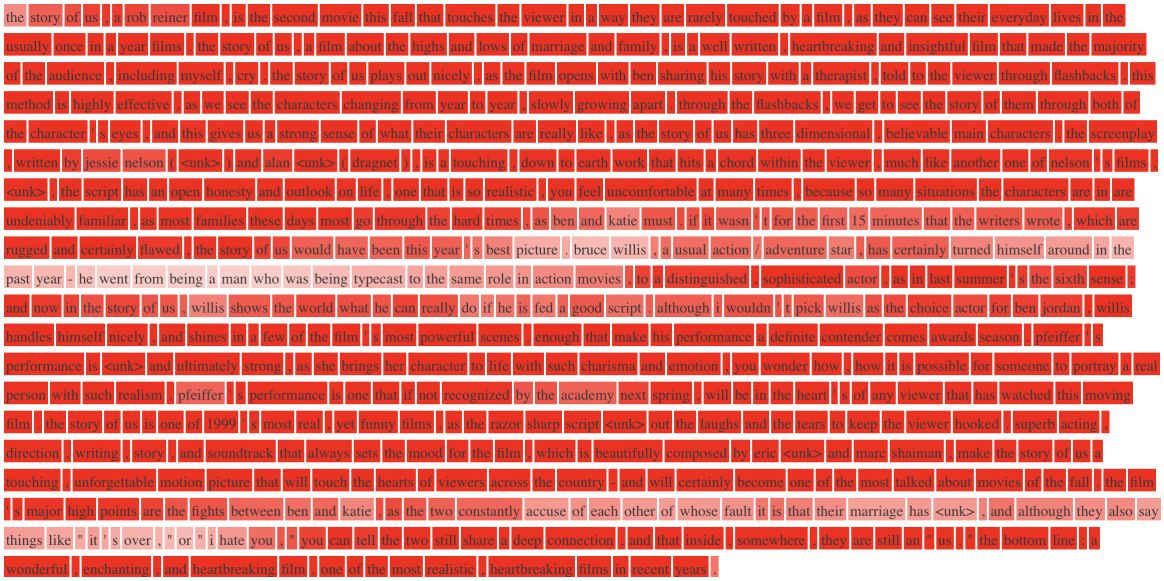

Figure 5: *Attention heatmap on a correctly classified positive review for the polarity-filtered task.*

B. Adversarial generated samples for subjectivity detection.

The following are two objective sentences generated by ChatGPT using words that appear only in the subjective class lexicon.

- *"The widely reserved, self-determination and simplicity of the 12-step program have proven to be an effective life-affirming method for those seeking to overcome addiction and achieve reconciliation with themselves and others."*
- *"The artist-agent's creative approach to marketing and promotion has helped to boost the success and stylishness of numerous music and entertainment projects."*

The following are two subjective sentences generated by ChatGPT using words that appear only in the objective class lexicon.

- *"I was shocked to discover that the financial webcams we had been using were actually part of a scheme known as 'frodes', and I couldn't believe that Daddy's client would scoff at the idea of being caught up in such a bale of trouble."*
- *"I felt betrayed and stunned, but I knew I had to move on and find a new situation-based opportunity, even if it meant leaving behind the familiar Composers' Castle and the territorial Marjorie and Margaret."*

C. Training times

Model	Subjectivity	Polarity	Polarity-filtered
BiGRU	2	3	3
BiGRUAttention	2	11	7
TextCNN	1	2	2
DistilBERT (subjectivity)	60	-	-
RoBERTa	-	91	91
BigBird	-	659	631

Table 3: *Training times in seconds per epoch.*