Final NLU project report

Thomas De Min (mat. 229356)

University of Trento

thomas.demin@studenti.unitn.it

1. Introduction

Polarity classification consists of determining whether an opinion is overall positive or negative. This task can be performed at the document, sentence, or aspect level [1]. Furthermore, to ease the task, Pang et al. [2] proposed to remove objective sentences from the document: ”This can prevent the polarity classifier from considering irrelevant or even potentially misleading text”.

With this project, I focus on sentiment analysis at the document level and I present my approaches to face the problem. To do so I analyze the performances of Naive Bayes, GRU with and without the Attention mechanism, and fine-tuned Transformer Encoders. I also analyze the empirical results of removing objective sentences from the document.

2. Task Formalization

”Sentiment analysis or opinion mining is the computational study of people’s opinions, sentiments, emotions, appraisals, and attitudes toward entities such as products, services, organizations, individuals, issues, events, topics, and their attributes” [1]. Sentiment analysis can be studied at three different levels: document, sentence, or aspect level [1]. The first of the three involves the categorization, as a whole, of an opinionated document as positive or negative (or neutral). It is assumed that the document has an opinion about a single entity (such as a movie), which can be positive or negative. At the sentence level, instead, the categorization is performed for each sentence in the document. In this case, we cannot assume that each sentence is opinionated, and, for this reason, objective sentences are usually removed from the document. Other approaches instead treat the task as a three-class classification problem, where each sentence can be positive, negative, or neutral. Sentiment Analysis at aspect level is more fine-grained compared to the previous ones. It involves the extraction and summarization of people’s opinions expressed on entities and aspects/features of entities [1]. Sentiment Analysis is known to be a highly domain-dependent problem [3]. Depending on the domain, users may express their sentiment using different expressions. Let us take for example the word *easy*. In the electronic domain, it is used to positively describe a product, whereas in the movie domain it can be used in the opposite way: ”the plot is easy to guess” [3]. When this problem is considered, it is necessary to train/fine-tune a classifier based on the application domain (or use multidomain approaches like Loshchilov et al. 2017 [3]).

The focal point of this project is related to movie reviews. These, in particular, are made up of objective and subjective sentences. Objective sentences are used to provide a brief abstract of the plot and other information about the actors and directors. For this reason, objective sentences are not useful in classifying the polarity of a document, since they do not carry opinions. B. Pang and L. Lee [2] proposed to improve performance by training a classifier that can predict the subjectivity of each sentence and then filtering out objective sentences from the dataset.

With this project, I face the task of classifying the polarity, at the document level, of movie reviews by also analyzing the pros and cons of removing objective sentences from each document. Moreover, I investigate the benefits of using different kinds of models and the use of attention.

3. Data Description & Analysis

For the task, two datasets are given. The subjectivity dataset (simply called *subjectivity*), by Pang and Lee, consists of 10,000 sentences, of which 5,000 are objective and 5,000 are subjective. This dataset is used to train a binary classifier able to tell whether a sentence conveys objective or subjective information. The vocabulary contains 497 words. The average sentence length is 24 words while the minimum and maximum are respectively 10 and 120 words. Instead, the polarity dataset (called *movie_reviews*), also by Pang and Lee, is made up of only 2,000 examples. Each example represents a document that contains a review of a movie. Each example is paired with a binary label, ”Positive” or ”Negative”, which refers to the ground truth polarity of the document itself. The objective of this dataset is to train a classifier, also binary, which should tell whether an opinion, on a movie, is positive or negative. Regarding the statistics, 1000 examples are negative reviews while the remaining ones are positive. The vocabulary size is larger, 5316 words. The average number of sentences per document is 33 while the minimum and maximum are respectively 1 and 115 sentences. Regarding the number of words, on average a document presents 792 words with a minimum of 19 and a maximum of 2879 words. To train each approach, Cross-Validation with ”Stratified K-Fold” is used. The latter provides approximately the same label distribution for each fold, and since both classes in both datasets are balanced, each fold will be (approximately) balanced. For each setting, only 5 folds are used. The reason is that Polarity classification, especially with Transformer Encoders, requires a lot of time for each training iteration, so I needed to reduce the number of experiments. To make all approaches comparable, the same random state is used in each experiment.

4. Model

As briefly mentioned in the Introduction, I will present three models. Each one is trained for both subjectivity and polarity classification. Moreover, each polarity classifier is trained with and without filtering the objective sentences in order to compare the performance.

4.1. Multinomial Naive Bayes

This first implemented baseline is the same as that proposed in the project description. Its implementation follows the same steps seen in laboratories.

4.2. GRU

The second baseline involves the implementation of a Gated Recurrent Unit. The choice between this model and the LSTM was justified because the former has fewer parameters than the latter, and thus it leads to faster training. Furthermore, the data is limited, which means that having too many parameters can affect performance. The input consists of the word sequence in *document*. Each word is first encoded into an integer representation ad then embedded in a vector of 256 values. The hidden representation, of the GRU, is composed of 128 values. Regarding the GRU, I used a bidirectional unit like in [4] but with 2 layers since I experienced better performances. For each input word, the hidden representation of the last of two layers is used. Since a bidirectional unit is used, each hidden representation will be:

$$h_i = \overrightarrow{h_i} \parallel \overleftarrow{h_i} \tag{1}$$

where h_i is the hidden representation for the i-th word in the *document*, $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are respectively the forward and backward representation of the word. The symbol $\parallel$ represents the concatenation. The obtained hidden representation are then summed up and sent to a classifier.

As stated in [4], "Different parts of the document usually have different importance for the overall sentiment". For this reason, the attention mechanism was implemented. The implementation I chose is similar to the one in [4], but I omit attention at the sentence level. I made this decision because the gain in accuracy obtained by the authors with the sentence level was very little. Moreover, in my case, data were limited, thus using also the attention at sentence level could have decreased my performance. So, I tried implementing the attention using a two-layers feedforward network, but with disappointing results. In the end, by looking at the attention implementation in [3], I opt for an implementation with a single feedforward layer with Dropout at the bottom, for regularization, and a Softmax at the top. The computation of attention for document k is as follows:

$$\begin{aligned} e_i &= f_a(h_i) \\ \alpha_i &= \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \\ c_k &= \sum_{j=1}^{n} \alpha_j \cdot h_j \end{aligned} \tag{2}$$

where h_i is the hidden representation for the i-th word in the document, e_i is the score calculated by the attention layer prior to normalization, α_i is the output of the Softmax layer (which is a probability distribution) and c_k is the resulting vector (Figure 1).

At the top of the network, I plug in a dropout for regularization and a classifier with a single output neuron. The loss function used is the binary cross-entropy with logit loss, which is numerically more stable than a simple binary cross-entropy [5]. The optimizer used is Adam, but I choose not to use Weight Decay as it seemed to reduce the performance and increased the required number of epochs to train the model. In the end, each GRU is trained for 50 epochs, which are sufficient in order to overfit the training set, with a learning rate of 10^{-3}. Batch sizes for subjectivity and polarity classification are respectively 4096 and 512. One of the main problems I encountered was the batch size itself. Since I started developing with small batches of 32 samples, to speed up the process, I ended up with a GRU that was unable to learn anything, not even a small dataset. While the problem seemed to be caused by errors in the data-loading procedure or during evaluation, the batch dimension was the cause. More precisely, I figured that to be able to train it successfully, the batch dimension should be set to a number ≥ 256.

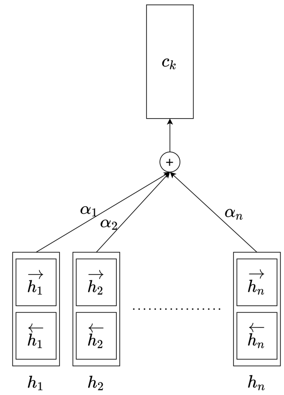

Figure 1: *Attention Mechanism*

4.3. Transformer Encoder

The last of the three models is the Transformer Encoder. In this case, I fine-tune pre-trained weights for computation reasons, data scarcity, and also to get better performance. Two different models are employed based on the task. Here, the motivation relies on the availability of pre-trained weights on Huggingface. For subjectivity classification, I use the weights proposed in [6]. It is a BERT base trained on the Wiki Neutrality Corpus [6], the objective of which is to debias the text by suggesting edits that would make it more neutral. For the polarity classification, I use those proposed in [7]. It is a RoBERTa base trained on around 58M tweets and then fine-tuned on the *TweetEval* benchmark for sentiment analysis. To match the requirements of polarity classification in movie reviews, the weights have to be fine-tuned again on *movie_reviews* dataset. The same classification layer and loss function of the GRU is used here. I fine-tune each Transformer Encoder for 10 epochs (to save computation) with a learning rate of 5^{-5}, as suggested in [8]. Using a slightly lower learning rate, the Transformer Encoder was unable to learn anything. The optimizer used is AdamW [9], which is a "modified" version of Adam that solves some problems related to Weight Decay. The coefficient for the Weight Decay is set to 10^{-2}. Batch sizes for subjectivity and polarity classification are, respectively, 64 and 16, which are the biggest batch sizes allowed by Colab Free GPU. Regarding the tokenizers, I use those proposed by the articles from which I take the pretrain weights, but I do not fine-tune them.

5. Evaluation

I compare the baseline approach, or else a Naive Bayes, with the two proposed: GRU and Transformer Encoder. All these approaches are trained for subjectivity classification and polarity classification, and for the latter, both scenarios (with and without filtering objective sentences) are evaluated. Moreover, for the GRU, the benefits of using attention are also taken into account during the evaluation. As briefly mentioned in Data Description & Analysis each model is trained with "Stratified K-Fold" Cross-Validation using 5 splits in order to compare the performance of each approach on the same sets of data. The

Model	Accuracy	F1-score
Multinomial NB	92.0 ± 0.5	92.0 ± 0.1
GRU	90.6 ± 0.7	90.6 ± 0.7
GRU+Attention	90.5 ± 0.8	90.6 ± 1.0
BERT	$\mathbf{96.9 \pm 0.5}$	$\mathbf{96.9 \pm 0.5}$

Table 1: *Accuracy and F1-score on subjectivity classification.*

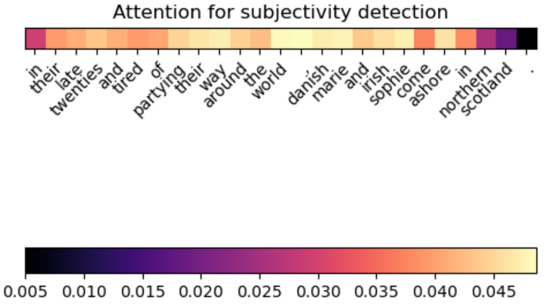

Figure 2: *Attention heatmap for subjectivity detection.*

metrics used are the Accuracy and the F1 score, respectively:

$$\begin{aligned} \text{Acc} &= \frac{TP + TN}{TP + TN + FP + FN} \\ \text{F1} &= 2 \cdot \frac{P \cdot R}{P + R} \end{aligned} \tag{3}$$

where $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$.

5.1. Results on subjectivity classification

In Table 1, we can appreciate the accuracy and F1-score in the subjectivity classification of, from top to bottom, Multinomial Naive Bayes, GRU without attention, GRU with attention, and BERT. As we can see in Table 1, the pre-trained BERT fine-tuned on the subjectivity dataset has been able to outperform the other baselines obtaining a lower error. Contrary to what I expected the performance of the GRU are quite disappointing, in fact it is not capable to reach the Multinomial Naive Bayes accuracy and F1-score. The reason behind these poor performances is probably related to the scarcity of data. In fact, Neural Networks require a considerably large amount of data in order to be properly trained from scratch. However, although BERT counts $\approx$ 110M parameters, it can obtain a low error even with a small dataset due to pretraining in a larger one. Pretraining the GRU on a bigger dataset would have probably led to a more accurate model.

As for the attention mechanism, we can see that it is not able to outperform the base version of the GRU. This behavior is probably caused by the length of each input sequence, which, since it is only a sentence, limits the necessities of attention. As we can see in Figure 2, attention does not focus on particular words, quite the opposite. In fact, scores are all very small[1] and have very low differences from one to the other. That means that attention is not helping the classifier by putting more focus on relevant words, leaving it to deal with the entire sentence.

Model	Fold	Accuracy	F1-score
GRU	1	91.4	91.5
GRU+Attention	1	91.1	91.4

Table 2: *Best sets of weights for subjectivity classification with GRU.*

Model	Fold	Accuracy	F1-score
BERT	1	97.4	97.4

Table 3: *Best set of weights for subjectivity classification with BERT.*

5.2. Results on polarity classification

Since Pang et al. [2] demonstrated the beneficial properties of removing objective sentences, to filter them out the following approach has been used:

- Multinomial NB: Once the Stratified K-Fold has been run and the performance measures have been obtained, the model is trained again on the whole dataset. The final model is used to filter out the objective sentences.
- GRU: The correspondent best set of weights obtained with Cross-Validation is used to filter out the objective sentences (see Table 2). For example, for the polarity classification with attention, the correspondent GRU with attention, trained in subjectivity detection, has been used to filter out objective sentences.
- Transformer Encoder: The best BERT base obtained in the subjectivity classification task has been used (see Table 3).

In Table 4 we can see the accuracy and F1-score in polarity classification of, from top to bottom, Multinomial Naive Bayes, Multinomial Naive Byes with filtered objective sentences, GRU with and without attention, and for both of them with and without filtering objective sentences, RoBERTa and RoBERTa with filtered objective sentences. In Table 4 we can see how the Transformer Encoder (RoBERTa) can achieve better performance over GRU and Naive Bayes, both with and without filtering the objective sentences. We can appreciate how the work of Pang et al. [2] manages to obtain higher performance by filtering out objective sentences (Multinomial NB and RoBERTa).

Even here the performance of the GRU is disappointing and it can be seen that it matches the accuracy score of the baseline. Most importantly, it is easy to notice that the accuracy and F1-score are lower when the filtering mechanism is used. The issue is related to the quality of filtering. Although the filtering mechanism should ease the training for the polarity classification by

Model	Accuracy	F1-score
Multinomial NB	81.6 ± 2.1	81.0 ± 3.0
Multinomial NB+Filtering	84.0 ± 1.7	84.0 ± 3.0
GRU	82.0 ± 0.8	82.0 ± 1.4
GRU+Attention	84.1 ± 1.2	84.3 ± 1.6
GRU+Filtering	77.4 ± 2.7	76.2 ± 3.2
GRU+Attention+Filtering	79.9 ± 2.4	79.8 ± 3.0
RoBERTa	88.2 ± 2.5	88.1 ± 2.5
RoBERTa+Filtering	$\mathbf{93.6 \pm 1.9}$	$\mathbf{93.5 \pm 1.9}$

Table 4: *Accuracy and F1-score on polarity classification.*

[1] Color scale is not in the range $[0, 1]$ otherwise the heatmap would have appeared very dark.

removing useless sentences and, as a consequence, by reducing the input dimensionality, the actual filtering is not accurate. This is a direct consequence of the error of the subjectivity detector, which is relatively high. As a result, it can remove useful sentences while keeping unaltered useless ones. Although this is true for the GRU, the same cannot be said for RoBERTa. The reason why RoBERTa gets better performance using the filtering mechanism is related to the input size limitations of the model itself. If fact, RoBERTa base support sequences whose length is $\leq$ 512 words and this limitation forces the truncation of the document during the tokenization pipeline. Now, by looking at the movie reviews in the dataset it is worth noticing that the majority of them start with the movie plot, which is all objective sentences, and it continues with the actual opinion. Truncating the document after 512 words makes it difficult to capture a large portion of subjective sentences because most of them are objective ones. When the objective sentences are filtered out, the truncated text contains almost only useful sentences to classify the document. Here, also the quality of the subjectivity classification is crucial in order to obtain a clean and usable document.

The first evaluation of RoBERTa consists of the interpretation of errors. I printed out the wrong predictions for fold number 2 to analyze them. The main pattern I can see is that the model makes mistakes when the movie reviews present both positive and negative aspects of the movies. For example, a reviewer criticized every technical aspect of the movie ”Armageddon” before saying it is its favorite movie. However, in other wrong predictions this pattern has not shown and in fact, I cannot say for sure which is the cause of a wrongful prediction in those cases. Probably is caused by a lack of data or by the use of words and idioms that were not present in the training set. Moreover, although the dataset has been filtered before use, some movie reviews are still truncated, thus, it may be the case that some important aspects have been cut off.

To further evaluate the fine-tuned RoBERTa, I decided to collect some movie reviews from a couple of friends. The rationale behind this relies on the fact that Sentiment Analysis is highly domain-dependent. Although I still collected movie reviews, they have been written by Italians who can express concepts differently from native English speakers and who can also make mistakes in writing[2]. I was able to collect just seven of them, but they are made of objective and subjective phrases, and they are short enough to check whether filtering out objective sentences can affect the prediction. Seven reviews are not enough to assess the correctness of these statements, but at least they can give some sort of indication. In subjectivity detection, BERT was able to correctly classify all sentences except one, which is: *The director wants to explain how, nowadays, we don't care enough about natural phenomena that could lead to extinction.* BERT predicted it as subjective while I would personally classify it as objective. Regarding the task of polarity classification, RoBERTa was able to correctly predict the polarity of all reviews. As the last test, I checked whether RoBERTa (trained on filtered documents) was able to correctly classify unfiltered documents, and it was able to do that.

Another important aspect worth noting is the boost in performance provided by the attention mechanism with the GRU (Table 4). Contrary to what happens in the subjectivity classification task, the input sequences here are much longer and, in fact, contain entire movie reviews. Here, attention can play a better role in removing words that are not useful for the polarity classification task. As stated in [4], ”[...] the sentiment of a document is usually decided by a relatively small part of it. In a long review document, the user might discuss both the advantages and disadvantages of the product. The sentiment will be confusing if we consider each sentence of the same contribution”. In Figure 3 we can see two attention heatmaps for the polarity classification of a review I collected. The heatmap on top represents the attention scores of an objective sentence in the review while the one in the bottom shows those of a subjective sentence. Clearly we are dealing with a movie review that has not been filtered by a subjectivity detector. We can appreciate how the attention places more focus on words/sentences that are more meaningful in classifying the document while neglecting, or reducing the importance, of those that are not. In fact, almost all the words in the objective sentence have lower attention scores than those in the subjective sentence. Although this is promising, we still see that attention is quite inaccurate when scoring. The word *liked* has a relatively low score compared to *in*. I think the problem is caused by a lack of data, after all the accuracy and F1-score are quite low, and by consequence also the attention block is not capable of predicting a good probability distribution.

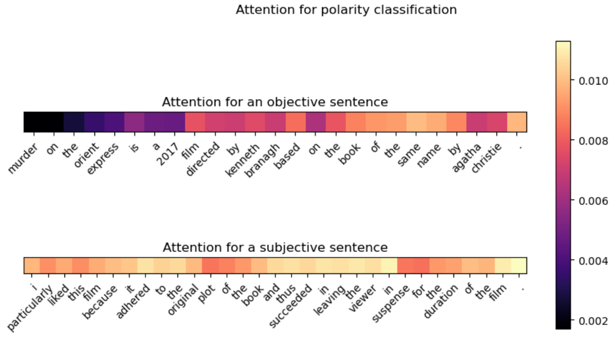

Figure 3: *Attention heatmap for polarity classification.*

6. Conclusion

In this report, I presented three approaches to dealing with polarity classification and I also investigated the effects of filtering out objective sentences from movie reviews. The first one is a Multinomial Naive Bayes. The following one is a Bidirectional GRU, with two layers, powered with attention mechanism. The last one, instead, is a fine-tuned Transformer Encoder. The experimental results showed that a Transformer Encoder was able to obtain better performance over the other two approaches. Moreover, due to issues related to the length of the input sequences, filtering out objective sentences allowed the Transformer Encoder to obtain higher accuracy scores by partially solving this problem. Another important aspect analyzed on this report is the influence of attention (in the GRU) in both tasks. As I showed, attention improved the performance of polarity classification by giving more importance to words/sentences that contain opinions about a movie. On the other hand, attention was shown not to be of relevant importance in the task of subjectivity detection, where its contribution lowered the average accuracy and F1-score of trained models.

[2] One review contains also an acronym: *ofc*.

7. References

[1] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[2] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *arXiv preprint cs/0409058*, 2004.

[3] Z. Yuan, S. Wu, F. Wu, J. Liu, and Y. Huang, "Domain attention model for multi-domain sentiment classification," *Knowledge-Based Systems*, vol. 155, pp. 1–10, 2018.

[4] X. Zhou, X. Wan, and J. Xiao, "Attention-based lstm network for cross-lingual sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 247–256.

[5] "Binary cross-entropy with logits loss," https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html, accessed: 2022-07-28.

[6] R. Pryzant, R. D. Martinez, N. Dass, S. Kurohashi, D. Jurafsky, and D. Yang, "Automatically neutralizing subjective bias in text," in *Proceedings of the aaai conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 480–489.

[7] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "Tweeteval: Unified benchmark and comparative evaluation for tweet classification," *arXiv preprint arXiv:2010.12421*, 2020.

[8] "Fine-tune a pretrained model," https://huggingface.co/docs/transformers/training, accessed: 2022-07-29.

[9] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.