Explainability Methods for Natural Language Processing: Applications to Sentiment Analysis (Discussion Paper)

Francesco Bodria[1], André Panisson[2], Alan Perotti[2], and Simone Piaggesi[2,3]

[1] Scuola Normale Superiore, Pisa, Italy, francesco.bodria@sns.it,
[2] ISI Foundation, Turin, Italy, {andre.panisson,alan.perotti}@isi.it
[3] Università di Bologna, Bologna, Italy, simone.piaggesi2@unibo.it

Abstract. Sentiment analysis is the process of classifying natural language sentences as expressing positive or negative sentiments, and it is a crucial task where the explanation of a prediction might arguably be as necessary as the prediction itself. We analysed different explanation techniques, and we applied them to the classification task of Sentiment Analysis. We explored how attention-based techniques can be exploited to extract meaningful sentiment scores with a lower computational cost than existing XAI methods.

Keywords: eXplainable Artificial Intelligence, Natural Language Processing, Sentiment Analysis

1 Introduction

The World Wide Web is a great invention, as it connects everyone in the world, but simultaneously, it is a double-edged sword. In such an open-by-design architecture, everyone can express their feelings from joy to anger. In Social Networks, negative sentiments can derail into hate speech; that can be shared easily, quickly and anonymously, thus becoming a problem. To contain the generation and diffusion of such undesired content, social network companies had to deploy special teams that regularly watch the net and block this phenomenon. These monitoring and flagging tasks are mostly carried out by human employees, thus making the process inefficient: semi-automating this pipeline would allow for a significant speed-up and, consequently, better coverage of contents shared on social media. The research area that deals with this kind of tasks is Sentiment Analysis (SA): Sentiment Analysis is a sub-field of Natural Language Processing (NLP) that, combining tools and techniques from linguistics and Computer

Science, aims at systematically identifying, extracting, and studying emotional states and personal opinion in natural language. However, how can an algorithm know if a text is expressing positive or negative sentiment? This tricky question is not easy to answer, especially in the very complex and ambiguous domain of feelings in which also humans sometimes struggle. Machine Learning algorithms can be applied to Natural Language Processing. Still, the resulting models can be very complicated, becoming black-boxes that provide no information about how the sentiment classification task is performed. How can we trust the results of a black box? Trusting the model is necessary, especially if there is a need to deploy it on a large scale. eXplainable AI (XAI) is a recent research field that deals with this kind of issue.

Our Research Question is, therefore, the following: *is it possible to equip SA algorithms with XAI techniques, in such a way that sentiment labels are explained, and in a computationally feasible way?*

We start by applying state-of-the-art explainability methods to the field of Sentiment Analysis. Then we explore an attention-based method capable of extracting explanations which are both similar to the black-box predictions and computed in a small amount of time. We evaluate our methods and other XAI techniques by comparing them with the original black-box model predictions and show examples where explanations are in contrast with black-box predictions.

2 Related Work

Recently, the use of neural networks in the development of Natural Language Processing (NLP) tasks has become very popular [1]. The standard approach is to transform the input words into semantic vectors called Word Embeddings. These vectors can then be used as input for other algorithms: in our case, they will be fed into a Sentiment Analysis classifier.

Currently, the most effective techniques to create Word Embeddings are the Transformers models, which rely on the attention mechanism. The attention mechanism allows looking over all the information the original sentence holds and then create the proper word embedding according to the context. Transformers models incorporate this by encoding each word position, so it is possible to link two very distant words [2]. The Transformer model utilised in this work is BERT [3], which is one of the most popular models in NLP. The suggested approach to make Sentiment Classification is by applying Transfer Learning. As indicated by BERT authors, the learning procedure is divided into two parts: Pre-Training and Fine-Tuning. The Pre-Training phase is an unsupervised learning process. It consists of showing the model a large sentence corpus, masking a random word, and trying to predict the same word embedding of the masked input (Generative Pre-Training [4]). The second part is called the Fine-Tuning phase. It consists of stacking, after BERT, a linear layer that acts as a classifier and training the whole resulting model using a Sentiment Analysis dataset.

Explaining text classification might look like an easy task for a human, but not for a machine. The best performing models in Text Classification are deep neural networks composed of billions of parameters, and explaining such complex models is difficult or computationally expensive. Radford et a.l. [5] proposed an original approach to this problem. While training their linear model with L1 regularisation, they noticed it used surprisingly few of the learned units. Further analysis revealed a single "sentiment neuron" that was highly predictive of the sentiment value. Using the output of this neuron, they can create scores that explain each word's sentiment in a sentence. In general, there are several types of methods to explain a text classification [6]. For our case, the most suitable is the feature importance method.

Sentiment Analysis often deals with binary sentiment classification: a *negative* sentence is labelled as 0, and a *positive* one as 1. The sentiment prediction task provides a binary label to a sentence. The XAI method outputs a heatmap visualising the contribution of each word of the prediction, as shown in Figure 1.

Fig. 1: Example of a heatmap applied on text.

As a method for creating such an explanation, LIME (Local Interpretable Agnostic-Model Explanations) [7] relies on a straightforward intuition: the model may be very complex globally, but it is easier to approximate it around the vicinity of a particular instance. While treating the model as a black-box, LIME perturbs the instance and trains a local linear classifier. The weights of the linear interpretable classifier create the heatmap.

Integrated Gradients [8] is another technique that has a different approach to the problem. Formally, suppose we have a function $F : R^n \rightarrow [0, 1]$ that represents a deep network. Specifically, let x be the input at hand, and x' be the baseline input. For image networks, the baseline could be the black image, while for text models, it could be the zero embedding vector. IntGrad considers the straight-line path from the baseline x' to the input x and computes the gradients at all points along the path. Integrated gradients are obtained by cumulating these gradients. Specifically, integrated gradients are defined as the path integral of the gradients along the straight-line path from the baseline x' to the input x.

3 Methodology

In this section, we introduce our approach to extract explanation scores from a sentiment analysis classifier in a computationally feasible way. The approach used here to classify documents is based on a pre-trained BERT model. Since BERT uses attention scores, we exploit these scores for explainability purposes.

Two parts compose the BERT model: the embedding creation part and the classifier. The first one creates a vector representation of text while the latter is built on top of the first and perform classical classification. Since BERT is

an attention-based model, we decide to add an attention layer between the two parts to have better insight on the model decision, as shown in Figure 2.

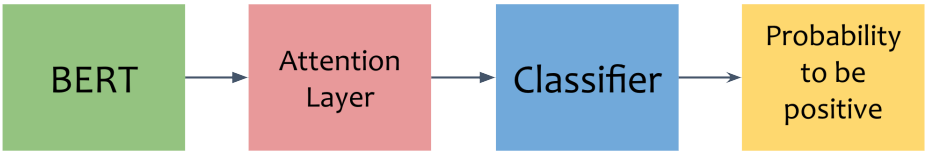


Fig. 2: Adjustment applied to the BERT model. We inserted an attention layer between BERT (encoder) and the Classifier.

Through this attention layer, the model assigns the importance of each word for the prediction task by weighing them when constructing the representation of the text. For instance, a word such as *'amazing'* is likely to be very informative of the emotional meaning of a text, and it should thus be treated accordingly. We use a simple approach inspired by Bahdanau [9] and Yang [10] with a single parameter per input channel:

$$(1) \quad a_t = \frac{exp(h_t w_a)}{\sum_{i=1}^{T} exp(h_i w_a)} \qquad (2) \quad v = \sum_{t=1}^{T} a_t h_t$$

BERT outputs an embedding vector h_t for every word t of a sentence of T words. The attention layer learns importance scores, a_t, for each word by multiplying the representations h_t with a weight vector w_a, learned during the process. The output is normalised using a softmax function to construct a probability distribution over the words. Lastly, the output representation vector for the text, v, is computed with a weighted summation over all the words using the attention importance scores as weights. This representation vector obtained from the attention layer is a high-level encoding of the entire text, and it is used as input to the classifier: a feed-forward layer with a sigmoid activation, $\sigma(W_{classifier} \cdot v)$.

The learned attention scores a_t are the output of a softmax, so they are all positive and do not incorporate the signal from the classifier. To overcome this, we multiply the attention scores obtained from the attention layer by the weights of the classifier. We take the sign and multiply it by the scores:

$$\hat{a}_t = a_t * \text{sign}(W_{classifier} \cdot (a_t h_t))$$

This is our definition of explanation scores.

For the learning phase, we freeze the BERT weights as the original pre-trained model and optimise only the attention and the classification layer.

4 Experiments

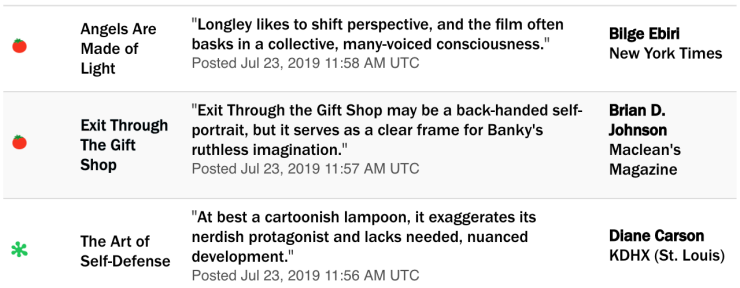


Fig. 3: Rotten Tomatoes Movie Review Dataset Examples, positive scores are expressed with a fresh tomato, while the negative with a rotten one.

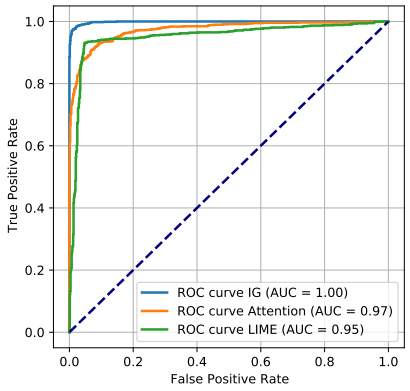
Dataset. For our experiments we used, as data, movie reviews; in particular the Stanford Sentiment Treebank (SST) [11] which contains 11855 sentences in movie reviews. Movie reviews are an excellent source for our task, since users both provide text reviews and a score, thus creating a labelled dataset.

The movie reviews have initially been posted on http://rottentomatoes.com/, a popular movie fan website where users can comment and express a positive/negative sentiment, using a fresh tomato 🍅 or a rotten tomato ✻ (see Figure 3). We split the original dataset of 11855 sentences in a training set (6920), a validation set (872) and a test set (1821).

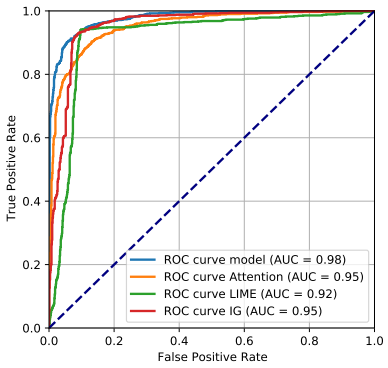
Training. First, we trained our modified version of BERT as a sentiment classifier for the SST dataset mentioned before; we adopted the hyperparameters as described in the original paper [3].
Second, we used the trained BERT as a black-box classifier and labelled all reviews in the test set; we then computed the explanation scores for all labels by running LIME, IntGrad and Attention.
Third, we compared these explanation scores with the ground truth labels and the predictions on the test set of the original black-box model. To do this we have to produce a prediction from the explanation scores, so we built a simple classifier which computes the label y of a sentence j taking the sign of the sum of the score s of the words t of the sentence ($y_j = \text{sign}\left(\sum_{t=0}^{N} s_{tj}\right)$).

Validation. To compare these three explanation scores, we used the *Fidelity* [6], to capture how much each XAI technique can mimic the behaviour of the black-box model it is explaining. We also measured the similarity of our explanation scores with the ground truth labels. Both our evaluation criteria were measured using the ROC AUC scores. The results are shown in Figure 4.

IntGrad is the XAI method that best replicates the model prediction and the one that works better on the test set. This technique seems the best choice,



(a) Explanation scores versus black-box predictions.



(b) Explanation scores and black-box prediction versus ground truth labels.

Fig. 4: Comparison between explainable methods.

but it still suffers from high computational costs. We evaluated the time for each method to be performed on the entirety of the test set. For the attention layer, we only needed two minutes. For IntGrad, we had to call the black-box model for every step from the baseline, so the time raises to 102 minutes. LIME is the method that performs the worst in terms of time for a total of 1440 minutes. For every sentence, we produce a synthetic neighbourhood, then call the black-box model to label it and finally learn an interpretable local classifier. Our method is less accurate but much faster, as the attention layer is part of the black-box model and does not need any additional model call.
To better appreciate the difference and similarities between the XAI methods, we show in Figure 5 an example of score assignment. The sentence is taken from the test set, and it has negative sentiment as ground truth: *"We never really feel involved with the story, as all of its ideas remain just that: abstract ideas"*.

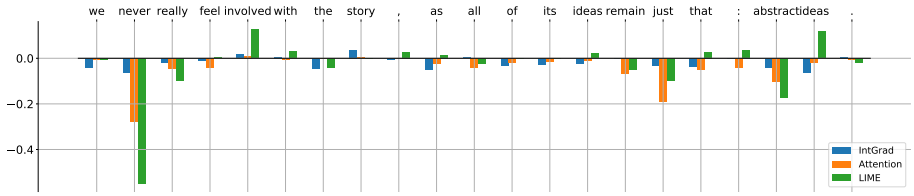


Fig. 5: Comparison between attention score $\hat{a}$ (orange), LIME Scores (blue), and Integrated Gradients scores (green) for the sentence: *"We never really feel involved with the story, as all of its ideas remain just that: abstract ideas."*

Sometimes, these XAI methods conflict with each other. In Figure 6, we show the box-plot of the distribution of correlation coefficients between all the scores along the test dataset used. We can see that overall the scores are strongly

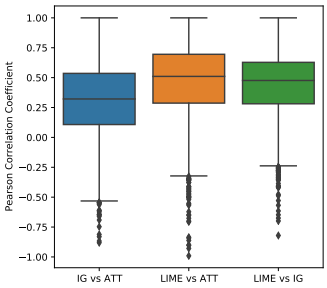


Fig. 6: BoxPlot of the Pearson Correlation Coefficients between XAI methods.

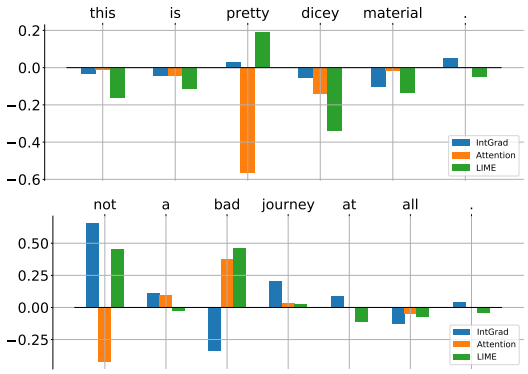


Fig. 7: Comparison between modified attention score $\hat{a}$ and LIME Scores for the sentences: *"This is pretty dicey material."* and *"Not a bad journey at all."*

correlated with each other, but we have a non-negligible number of samples that are negatively correlated. Further analyses highlight how, in some cases, LIME, IntGrad, and attention are negatively correlated in different ways. In Figure 7, we can see two different examples of this uncorrelated behaviour. The top chart shows the negative sentence *"This is pretty dicey material"*, and we can see that neither LIME nor IntGrad could capture the particular use of the word *dicey*. In contrast, the bottom one shows the positive sentence *"Not a bad journey at all"*, and in this case, IntGrad fails at capturing the use of the adjective *bad*. In these examples, explanation scores conflict with the model predictions. The model-explanation concordance is, in general, not guaranteed and has to be taken into account when developing and evaluating XAI techniques.

5 Conclusions

In this work, we show how to use attention layers to extract explanation scores about model predictions. Our method can provide explanations matching the predictions of the black-box model but requires a much lower computational time than the state-of-the-art benchmark methods of LIME and IntGrad. We found that attention scores can be used to explore the internal behaviour of deep neural network models and have XAI capabilities comparable to other approaches, requiring less computational resources. Choosing this approach is a matter of trade-off between performance and time. Different datasets, classification tasks, and black-box NLP models are to be considered in order to explore this trade-off further.

We conclude that many XAI techniques can be applied to the field of NLP to understand better the sentiment classification process (and other NLP tasks in general). However, there is much room for improvement: XAI techniques can

be an enabling factor for the explanation and deployment of ML models, but the current state of the art has not yet reached the desired maturity to be applied at scale.

Acknowledgements

AP and AP acknowledge partial support from Research Project Casa Nel Parco (POR FESR 14/20 - CANP - Cod. 320 - 16 - Piattaforma Tecnologica Salute e Benessere) funded by Regione Piemonte in the context of the Regional Platform on Health and Wellbeing and from Intesa Sanpaolo Innovation Center. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. Richard Socher, Yoshua Bengio, and Christopher D Manning. Deep learning for NLP (without magic). In *Tutorial Abstracts of ACL 2012*, pages 5–5.
2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
3. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
4. Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Tech.Rep., OpenAI*, 2018.
5. Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.
6. Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.
7. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD Int.l Conf. on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.
8. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proc. of the 34th Int.l Conf. on Machine Learning*, volume 70, pages 3319–3328, 2017.
9. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
10. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proc. of the 2016 Conf. of the North American chapter of the ACL: Human Language Technologies*, pages 1480–1489, 2016.
11. Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.