Explanability of Text Classification using SHAP

Sangeet Sagar
Universität des Saarlandes, Germany
sasa00001@stud.uni-saarland.de

Abstract

In recent years, Convolutional neural networks (CNN) (Wason, 2018) have been increasingly popular in the field of natural language processing (NLP) in tasks like text classification, machine translation, sequential decision, etc. The strategy of end-to-end training in deep learning frameworks has led to outperform traditional machine learning approaches. Yet, the lack of interpretability and explainability of these models have posed a challenge to understanding their exceptional performance and how can they be improved. In this paper, we have tried to solve this challenge with the help of a common NLP task- text classification. We describe methods to assess the importance of local features in a CNN-based model using the SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) tool. We compute Shapley values to allow generating multiple model interpretability graphics (Zhao et al., 2020). This would then be followed by using the obtained results from SHAP to demonstrate the local explanation and thus analyze how Shapely values support the findings obtained from CNN.

Keywords: CNN, SHAP, Shapely, text-classification, explainability, interpretability

1 Introduction

CNNs have been very popular since their first use for computer vision years ago. Since then it has been very common among researchers for obvious reasons. Over the recent years, CNN has been a hot topic among computational linguists. Apart from being popular and outperforming other commonly used models in NLP like Long-Short Term Memory (LSTM) or Bidirectional Recurrent Neural Networks (Bi-RNN), CNNs have been shown to achieve greater accuracy. (Kim, 2014) reported significant improvement in results using CNNs compared to then used traditional methods for sentence classification. Since then CNN based models are effective for most natural language processing tasks like sentiment analysis, entity recognition, sequence labeling, etc.

However, CNNs have been often termed as a black box for failing to understand what goes inside it and how can the results be improved. The explainability and interpretability of CNNs is still a hot research topic. The capability to understand and interpret any neural network is still a subject to discussion with the latest tools like SHAP and LIME (Ribeiro et al., 2016) allowing us to do the very much needed task: compute the contribution of a feature into making a certain prediction. Attempts to explain what goes inside a CNN black box were first made by (Zeiler and Fergus, 2013). Their contribution to image classification was reported to be remarkable. They introduced network activation architecture visualization techniques to explore the performance contribution from different model layers. They observed that visualization of CNN gives a better understanding of what is learned and how the model can be tweaked to count a greater contribution of such features.

In this work, we investigate into understanding how CNNs perform a classification task with text data using SHAP (Lundberg and Lee, 2017). The paper is organized as follows. In Section we review the major contribution of deep learning in NLP and related work done to interpret a neural network model. In Section 2 we introduce the CNN model used for text classification and in Section 3 we discuss the explainability methods used to interpret a model. In Section 4 we describe, in order, the experiments performed to train a CNN model and how can it be interpreted using SHAP. Next, we analyze the results in Section 5. Finally, with Section **??** we conclude the paper.

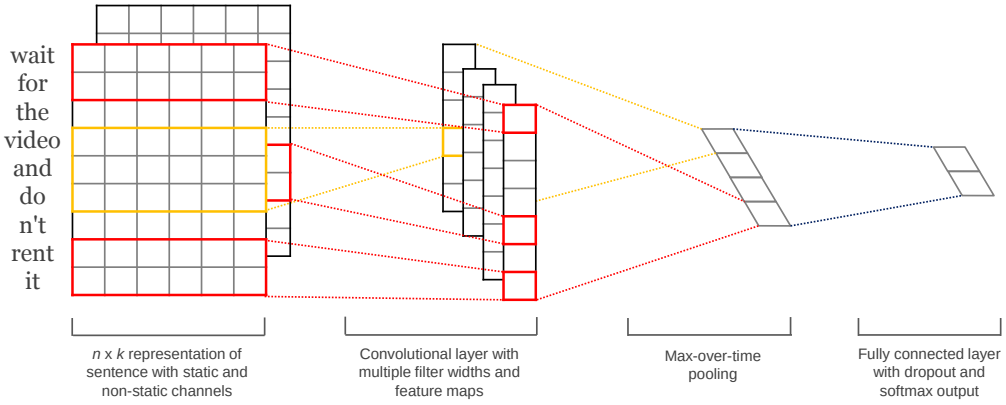

Figure 1: Proposed CNN model architecture.

2 Models

In this work, we make use of the work proposed by (Kim, 2014) that uses a single-layered CNN on top of pre-trained word vectors which consists of 1 million word vectors trained using fastText (Mikolov et al., 2018). (Kim, 2014) in his work demonstrated the conventional approach (Figure 1) for sentence classification using CNN. He reports that a single layer CNN with pre-trained word embeddings word2vec (Mikolov et al., 2013) showed promising results not just in sentence classification but also sentiment analysis and question classification.

3 Explainability and Interpretability

There are several approaches for model interpretability and can be grouped broadly into three categories:

- Intrinsic
- Model agnostic or model specific
- Global or local

Intrinsically interpretable models are the ones that are interpretable by design. These models have a pre-built mechanism for simpler interpretability such as linear models. One example for intrinsic or surrogate model-based diagnostics is LIME (Local interpretable Model-agnostic Explanations) (Ribeiro et al., 2016). In the model, agnostic-based approaches such as SHAP can be used to interpret any machine learning model. While certain tools are capable of interpreting only a specific model, such methods are known as model specific. Examples are linear SHAP, low-order SHAP, Max SHAP, Deep SHAP. Global interpretation-based models are capable to explain the model as a whole, while local interpretation explains the local features behind a single prediction.

In this work, we have made use of SHAP (Lundberg and Lee, 2017) that exploits the Shapley values to investigate the contribution of each feature in order to make a prediction for a class. SHAP first proposed by Lundberg and Lee in 2017 is designed to use Shapley values from game theory to explain machine learning models. (Lundberg and Lee, 2017) reports that there is an "Additive feature attribution method" (LIME belongs to this framework as well) which is basically trying to build a local model that is just a linear combination of the original features.

The Shapley values for an ith are calculated using the equation (Lundberg and Lee, 2017):

$$\phi_i := \frac{1}{M} \sum_{S \subseteq F \setminus \{i\}} \frac{1}{M - 1|S|} C(i|S) \qquad (1)$$

where $C(i|S) = f(S \cup \{i\}) - f(S)$, ϕ_i denotes the Shapley value for the ith feature, M is the total feature set, S is the subset of features.

The above equation sums over the marginal contribution of a specific feature for all possible permutations of features and tracks the contribution of that feature for a prediction. Implementation of the above formulation can be computationally expensive (Lundberg et al., 2019b), but SHAP has optimizations for different models (linear, trees, etc.)

3.1 Kernel SHAP

Kernel SHAP is a model-agnostic method that approximates the SHAP values. It addresses some

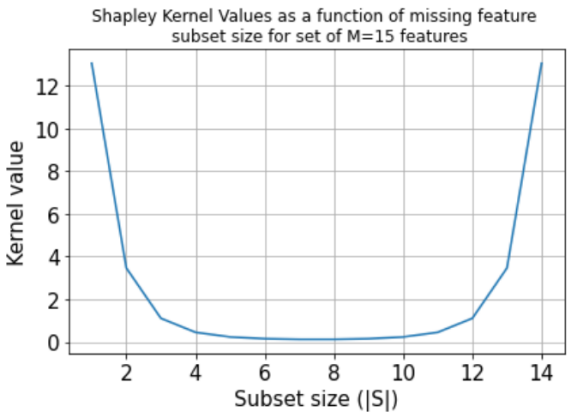

Figure 2: Shapley Kernel values as a function of missing feature subset.

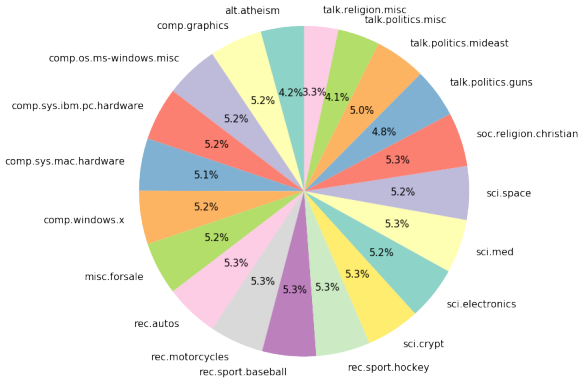

Figure 3: Pie-chart representing the categorical distribution of the dataset over 20 categories.

of the issues described above. Looking at Equation 1, it can be observed that the complexity of the equation is $O(M2^M)$ for the reason that the summation contains 2^M terms. Another issue with the above-proposed method is that it would require training the model an exponential number of times since missing data can not be fed into most of the machine learning models during inference.

The Kernel SHAP method counters these issues by making approximations that the values background data are used to replace the missing features.

The shapely kernel is given by the equation (Lundberg and Lee, 2017) below:

$$\pi_x(S) = \frac{M-1}{\binom{M}{|S|}|S|(M-|S|)} \tag{2}$$

3.2 Tree SHAP

Tree SHAP (Lundberg et al., 2019a) is a model-specific interpretation approach used to improve the interpretability of tree-based models like random forests, decision trees, etc. It focuses on measuring local features and global model structures based on these local features or explanations. It defines the function value $f_x(x)$ using the conditional expectation $E_{X_S|X_C}(f(x)|x_S)$ instead of the marginal expectation unlike Kernel SHAP.

4 Experiments

4.1 Dataset

In this work, we have used the 20Newsgroups dataset (empty, empty) from Scikit-Learn library package (Pedregosa et al., 2011). It consists of a huge archive of news collected from various sources and broadly categorized into 20 different categories like aethism, autos, politics, etc. The data was loaded and tokenized using Python library Keras (Chollet, 2015) and further padding and truncation were carried out to make each sequence a uniform length of 100.

4.2 Word embedding

We also made use of pre-trained word embeddings from (Mikolov et al., 2018). It consists of 1M word vectors trained on various Wikipedia articles, UMBC web base corpus, and stamt.org news dataset which collectively constituted about 16B tokens. We load these pre-trained embeddings of dimension 300 in the Embedding layer when defining the model architecture. Words found in the input that are absent in the pre-trained embeddings are initialized randomly and can be updated accordingly during training using appropriate arguments.

4.3 Training

We followed the CNN architecture as proposed by (Kim, 2014). It uses a total of three filters of sizes 3, 4, 5 that emulate the bi-gram, tri-gram, and four-gram models to obtain multiple feature representations. All these features are further passed through the max-pooling layer of stride 32 to reduce the dimension of the input feature representation. These are further concatenated, flattened and a dropout probability of 0.25 is applied to prevent any overfitting. The output features are passed through a fully connected softmax layer to get the output probability distributed over 20 targets. Figure 2 describes in detail the training architecture used in this experiment. Table 1 describes the training parameters used to carry out the experiments.

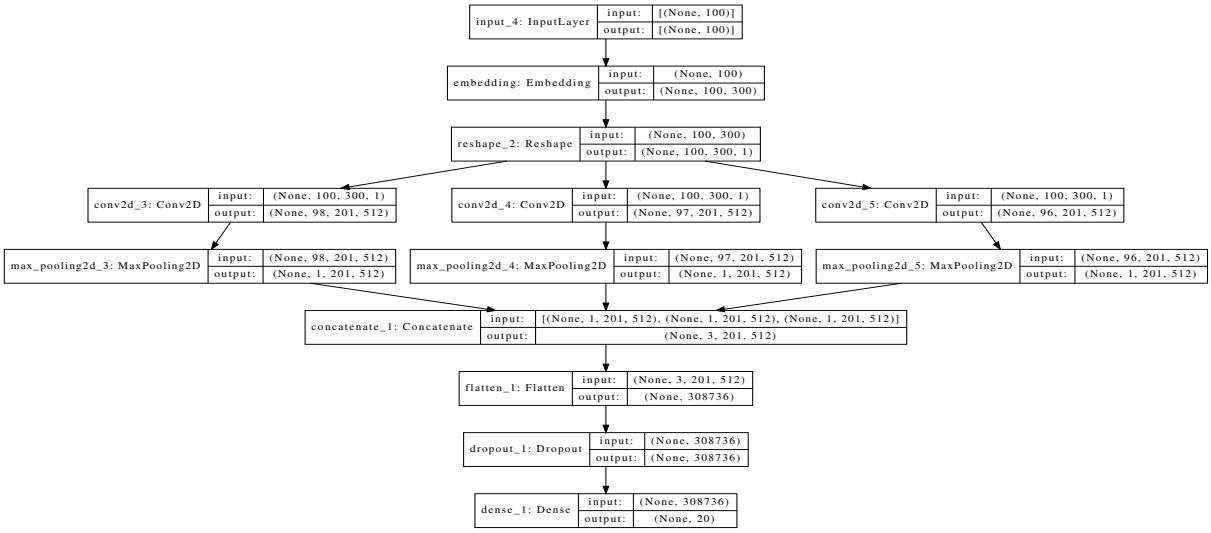

Figure 4: Proposed training architecture.

Parameter	Value
Learning rate	0.001
Training epochs	10
Dropout probability	0.25
Embedding Sizes	300
Number of filters	512
Batch Size	32
Maximum sequence length	100
Filter Sizes	[3, 4, 5]

Table 1: Default training parameters.

4.3.1 CNN SHAP-Model interpretation

To interpret the above-trained model we use SHAP's (Lundberg and Lee, 2017) DeepExplainer and Kernel explainer. DeepExplainer is an optimized algorithm for deep neural networks to compute the SHAPely values. Kernel explainer uses the Kernel SHAP as discussed in Section 3.1 to explain the predicted output of the trained model. It computes the importance of each feature and its contribution to a prediction.

5 Results and Analysis

We trained a CNN-based model on the 20newsgroup dataset using pre-trained embeddings. The overall accuracy was observed to be 91.06 % on the validation set and 91.11 % on the test set and the total training took around 400 s. The training time can be further optimized in presence of GPUs.

We further performed a local model explanation using SHAP. We constructed a summary plot (Figure) that shows the mean absolute SHAP values across all 20 target labels. It explains the contribution of each feature by classifying it into different targets. For example, for the word "university", we see that it has contributed highest for the target label "comp.windows.x".

Figure 6 shows the individual SHAP values for different words from a specific class. The output value is the predicted value for that feature. The base value is the value that would be predicted if features for the current output were to be absent. We use a selection of 100 samples from the training set and was plotted for the class 'comp.os.ms-windows.misc'. The blue color is a representation of a higher prediction and the red color is the representation of a lower prediction. Words like 'abortion','cruel', 'injection' etc have a higher prediction.

Figure 7 represents the kernel explainer. It takes input the compiled model and a set of samples from the train set and the plot explains why it makes single prediction. . Figure 8 explains all predictions for different features. This step can quite expensive depending upon the number of samples chosen from the train set

6 Conclusion

In this paper, we described how a CNN-based text classification model can be interpreted using SHAP. We observed picture visualization representing features that contribute to a model. We analyzed the SHAP values for a single prediction as well as multiple predictions (Figure 8). We were able to provide a more clear and coherent explanation of

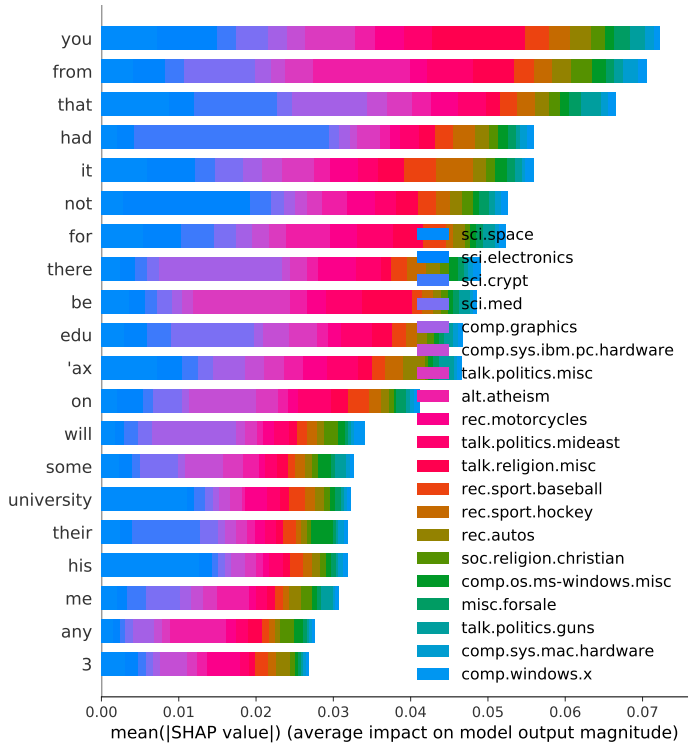

Figure 5: Summary plot for 20 classes with the absolute SHAP values along X-axis.

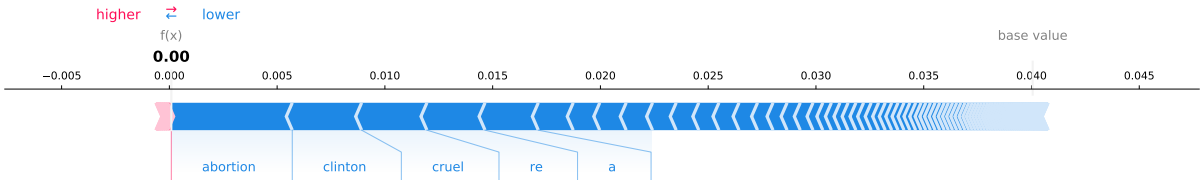

Figure 6: Explanation of a given prediction for the class- ‘sci.med’.

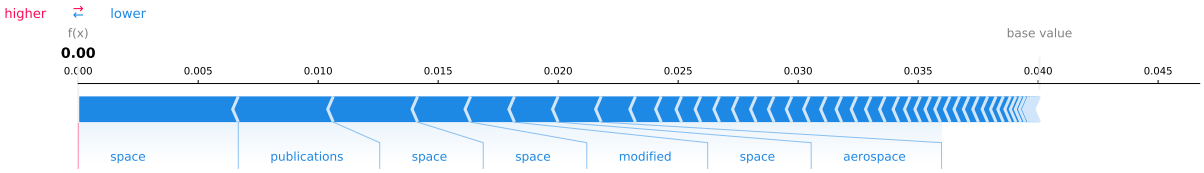

Figure 7: Kernel explainer.

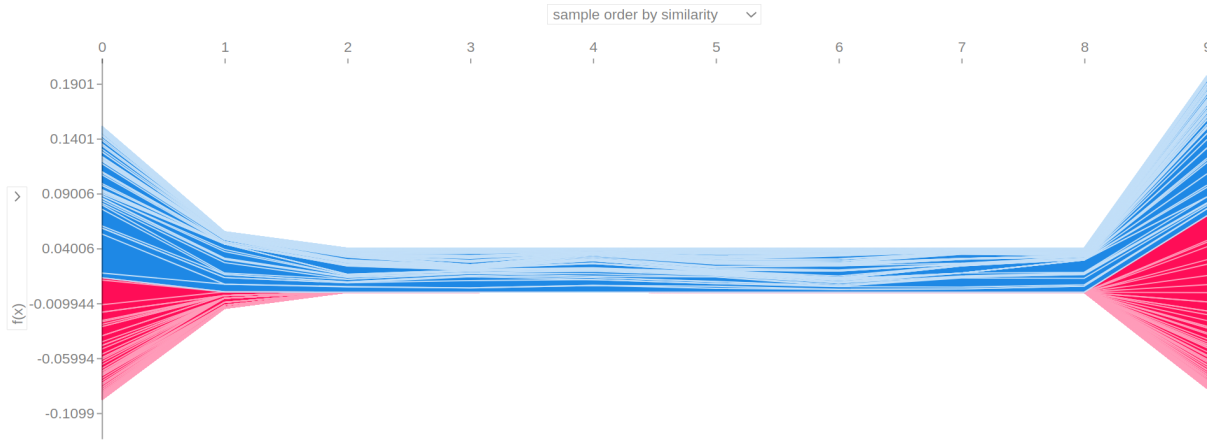

Figure 8: Output expectation for the class- 'sci.med'

the trained model.

References

François Chollet. 2015. keras. https://github.com/fchollet/keras.

empty. empty. 20 newsgroups dataset.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2019a. Explainable ai for trees: From local explanations to global understanding.

Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. 2019b. Consistent individualized feature attribution for tree ensembles.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Ritika Wason. 2018. Deep learning: Evolution and expansion. *Cognitive Systems Research*, 52:701–708.

Matthew D Zeiler and Rob Fergus. 2013. Visualizing and understanding convolutional networks.

Wei Zhao, Tarun Joshi, Vijayan N. Nair, and Agus Sudjianto. 2020. Shap values for explaining cnn-based text classification models.