Записки научных
семинаров ПОМИ
Том 530, 2023 г.

A. Rogov, N. Loukachevitch

AUTOMATIC EVALUATION OF INTERPRETABILITY METHODS IN TEXT CATEGORIZATION

Abstract. Neural networks have begun to take over more and more of a person's everyday life, and the complexity of neural networks is only increasing. When tested on collected test data, the model can show quite decent performance, but when used in real-life conditions, it can give completely unexpected results. To determine the cause of the error, it is important to know how the model makes its decisions. In this work, we consider various methods of interpreting the BERT model in classification tasks, and also consider a method for evaluating interpretation methods using vector representations fastText and GloVe.

§1. Introduction

Neural networks have begun to produce more and more results comparable to the human level and even outperform humans in some tasks, but at the same time the models have become much more complicated, and the number of parameters involved in the network has begun to reach huge levels. Deep learning models themselves are not easy to interpret because of their "black box" nature, and with such developments it is virtually impossible. The benefit of interpretability in machine learning is that it increases the credibility of the model. People are often afraid to rely on machine learning models when solving certain critical tasks. In particular, there may be situations when a person is faced with a new technology, and such an attitude can slow down the pace of its implementation.

Interpretability methods can be evaluated from three points of view: application-grounded, functionally-grounded, or human-grounded [12, 3]. Application-grounded evaluation estimates the consequences in the target environment, for example explanations in bank services. Functionally-grounded evaluation aims to check how well the explanation reflects the model. Human-grounded evaluation estimates whether the explanations are understandable to humans.

Key words and phrases: interpretability, BERT, classification.

In this work, we consider post-hoc interpretation methods in the text categorization task and assume that to be human-grounded the explanation should be semantically related to the category's name. A user should see the semantic similarity between the explanation and the category. We compare several known ways of interpreting the results of deep learning models: LIME [14], SHAP [11], and the self-attention mechanism of BERT as a way of interpreting the results [6] using this approach.

To compare the results of the interpretation methods with the category name, we use word embeddings and a metric originating from information retrieval, namely normalized discounted cumulative gain (NDCG) [7]. The result of interpretation methods considered in this work is a list of words with weights, where weight means the contribution of this word in the model's decision, so this is a ranked list. As a result, we select the top $N \in \{1, 3, 5, 10\}$ words that have made the most positive contribution to the output of the classification result of the interpreted model and compare them with the category's name in semantic similarity using word embeddings.

§2. Related work

Various explanation methods have been proposed to address the need for interpretability of machine learning methods. However, it is quite difficult to understand which method is the most trustworthy. Yalcin and Fan [15] analyzed explanations given by SHAP and LIME methods for the classification of poisonous mushrooms in the mushroom dataset. They found that for more than a third of samples, SHAP and LIME give different explanations when comparing the most important feature.

The authors of [4] study local explanation methods on a wide range (304) of OpenML datasets using six quantitative metrics. They revealed that LIME's and SHAP's approximations are particularly efficient in high dimensions and generate intelligible global explanations, but they suffer from a lack of precision regarding local explanations.

Natural language processing tasks have their own characteristic features, therefore approaches to their interpretability should be studied separately.

In [9], the authors consider several intepretability methods in text categorization. Three tasks for evaluating intepretability were considered: (1)

determining the best classification model from several ones based on explanations; (2) identifying the category of an example based on the explanation; (3) help in the analysis of examples with low probabilities. It was found that the LIME method showed the best results in the second task, where it finds the best evidence for the class independent of the class correctness. The study was implemented for two text categorization datasets (Amazon reviews and arXiv papers) and involved crowdsourcing in the first case and post-graduate student assessors in the second case.

Our study provides an automatic evaluation of task 2 defined in the above-mentioned work [9]. We calculate the semantic similarity of words extracted by interpretability methods with the category's title.

§3. Interpretation algorithms

In our experiments we consider three known algorithms of interpretation: LIME [14], SHAP [11], and self-attention weights [6].

3.1. LIME. LIME [14] (Local Interpretable Model-agnostic Explanations) is a method of local interpretation independent of the machine learning model. Local interpretability implies knowing the reasons for a specific decision. LIME presents a locally faithful explanation by fitting a set of perturbed samples near the target sample using a potentially interpretable model, such as linear models and decision trees [10]. The interpreted explanation in LIME is presented in the form of a binary vector showing the participation of any parameter in the result. For example, a possible interpretable representation for text classification is a binary vector indicating the presence or absence of a word, even though the classifier may use more complex (and incomprehensible) features such as word embeddings [14].

Let $x \in R^d$ be the instance being explained, the explained model be denoted by $f : R^d \to R$ and the explanation of the model be presented as a model $g \in G$, where G is a class of interpreted models such as linear models

$$g(x') = \phi_0 + \sum_{i=1}^{d'} \phi_i x'_i \tag{1}$$

where $x' \in \{0,1\}^{d'}$ is a binary vector of the interpretable representation of x and $\phi_i \in R$.

Interpreted model g tries to ensure $g(z') \approx f(x)$ whenever $z' \approx x'$. Since not every $g \in G$ can be simple enough to be interpreted, a measure of the complexity $\Omega(g)$ of the explanation of $g \in G$ is introduced. For

linear models, $\Omega(g)$ may be the number of non-zero weights. Next, $\pi_x(z)$ is used as a measure of proximity between the perturbed sample z and x. $L(f, g, \pi_x)$ will be a measure of how incorrectly g approximates f in the locality defined by π_x:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g). \tag{2}$$

Thus, the essence of the LIME approach is that we approximate the prediction of the model f of the test case x by a simpler, easily interpreted model g, which uses a simplified representation. The resulting explanation $\xi(x)$ interprets the target sample x with linear weights when g is a linear model.

3.2. SHAP. SHAP [11] (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. Shapley regression values are important characteristics for linear models in the presence of multicollinearity. To calculate the Shapley value, it is required to retrain the model for all subsets of features $S \subseteq F$, where F is the set of all features. These values assign an importance value to each feature, which means the importance of including this feature in the model forecast.

To calculate the Shapley value, the $f_{S\cup\{i\}}$ model is trained with the presence of this feature, and the other f_S model is trained without the feature. Then the predictions from the two models are compared at the current input $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$, where x_S represents the values of the input features in the set S. Feature retention depends on other features in the model, the previous differences are calculated for all possible subsets of $S \subseteq F\backslash\{i\}$. Shapley values are weighted averages of all possible differences:

$$\phi_i = \sum_{S \subseteq F\backslash\{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)]. \tag{3}$$

Exact computation of Shapley values is challenging. The work [11] introduces a new perspective that unifies Shapley value estimation. They propose SHAP values that are the Shapley values of a conditional expectation function of the original model. Let $x \in R^d$ be the instance being explained, and let $x' \in \{0, 1\}^M$ denote a binary vector for its interpretable representation and h be the mapping function $x = h_x(x')$. SHAP values

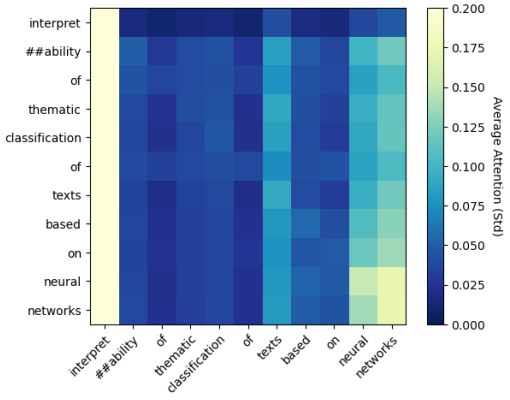

Figure 1. A sample matrix of weights in the form of an attention mechanism.

are the solution of

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}[f(h_x(z')) - f(h_x(z' \setminus i))] \qquad (4)$$

where $z' \in \{0,1\}^M$, $z' \setminus i$ denote setting $z'_i = 0$, $|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'. In [11], the authors propose that $f(h_x(z')) = E[f(z)|z_S]$, where S is the set of non-zero indexes in z'.

3.3. Self-Attention. This method is an attempt to understand whether it is possible to use weights in the attention mechanism as a local interpretation of models with Transformer-based architectures. The method is based on the study [6] of the possible relationship between self-attention and feature selection methods from different points of view, including the coincidence of vocabulary, similarity of ranking, relevance of the subject area, stability of features, and the effectiveness of classification. First, for each input sequence, average weights for the 12 attention heads in the last hidden layer are calculated. Next, a new matrix of weights is generated, grouping subwords in a word by averaging the weights of the subwords. The vertical average is taken as the weight of the word. Next, the top 10 words are considered as an interpretation.

This method does not depend on the response of the model, but is specific only to Transformer-based models. To illustrate this statement, Figure 1 depicts the mean weights of the 12 self-attention heads in the last hidden state of the trained *bert-base-uncased* [2] BERT model and fine-tuned on the WOS [8] dataset for "*Interpretability of thematic classification of texts based on neural networks*". From the plot we can clearly see the so-called vertical pattern, where a few tokens receive most of the attention, such as *training, deep, transformer, language*, and *understanding*. In the work [6] the authors did not include special tokens $< SEP >$ and $< CLS >$ because the amount of attention received by these tokens will make the attention received by the other tokens barely noticeable.

§4. Method for automatic evaluation of interpretability

After applying interpretation methods and getting the results in a convenient form for human perception, it is necessary to understand how satisfactory the interpretation result is. As a method of evaluation, one can ask experts to assess how clear the interpretation of the result is to them, but this is a rather resource-intensive and expensive method.

We suppose that the more the explanation is similar to the category's name in the text categorization task, the more this explanation will be understandable for a human. This allows us to evaluate explanation methods automatically. Since all above-mentioned methods of interpretation return as a result a ranked list of words with weights, we can compare these lists with the category name using the NDCG measure adapted from information retrieval [7]. We consider the top $N \in \{1, 3, 5, 10\}$ output words sorted by the weights assigned to them by the methods. These weights mean the importance of the word when making a decision.

Let $D^i = \{d^i_1, ..., d^i_m\}$ be the set of items retrieved for the query q^i and $\{rel^i(1), ..., rel^i(m)\}$ be their relevance labels. Let $\sigma = \{v_1, ..., v_m\}$ be the ordered items and the $DCG@k$ metric (discounted cumulative gain at position k) of the ordering is:

$$DCG@k(\sigma) = \sum_{j=1}^{k} rel(v_j)D(j), \tag{5}$$

where v_j is the identifier of the item retrieved at the position j and $D(j) = 1/log(1 + j)$ is the discount function. The $NDCG@k$ metric is

$NDCG@k(\sigma) = DCG@k(\sigma)/DCG@k_p$, where $DCG@k_p$ is the discounted cumulative gain of the ideal ordering according to true relevance labels $rel(i)$.

In our case D^i is a list of words of the text that we interpret, q^i is the category label of the text that our deep learning model predicted, $rel(i)$ is the cosine similarity between word embeddings for d^i_1 and q^i. To calculate the ideal word ordering, we extract all words from the target text and arrange them in descending order of embedding similarity to the category's label: this gives us maximal DCG for the target text.

As embeddings, we use the pre-trained GloVe[1][13] and fastText[2][5] models.

§5. Datasets

Experiments were conducted on two datasets: 20NewsGroup [1] and WOS [8].

Web Of Science (WOS) dataset is a collection of abstracts for academic articles that contains three corpora (5736, 11967, and 46985 documents) with 11, 34, and 134 topics respectively. In our work we use WOS-11967 with 34 topics. In this dataset, 70% of the samples are used for training and 30%, for validation.

20NewsGroup dataset includes 18846 documents with maximum length of 1000 words. In this dataset, 14846 samples form the training set and 4000 samples are used for validation. We have also cleaned the text from newsgroup-related metadata contained in them in order to obtain more realistic data and so that the model does not learn to classify them.

For multiword category names in the *WOS* dataset (for example, *Machine learning*), we averaged their component word embeddings.

§6. Experiments

For classification, we used the *bert-base-uncased* BERT model [2] and fine-tuned it on the datasets. For 20NewsGroup we obtained 71.3% accuracy, and for WOS-11967 we obtained 86.3% accuracy.

[1] `http://nlp.stanford.edu/data/glove.840B.300d.zip`

[2] `https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.en.zip`

Table 1. Interpretation of a text about baseball from the 20NewsGroup dataset.

ideal_glove	ideal_ft	SHAP	LIME	Self-Attention
pitchers	catcher	relieve	inning	catcher
catcher	pitchers	catcher	pitchers	ball
inning	inning	mound	not	mound
ball	reliever	pitchers	maine	pitchers
pitch	pitch	inning	of	allowed
reliever	ball	ball	are	warm
mound	mound	pitch	hall	off
pick	pick	ups	is	pitch
university	university	throws	reliever	inning
hall	hall	required	pitch	reliever
1	–	0.74	0.77	0.93
–	1	0.74	0.75	0.88

After we had trained the models, we used standard methods from the SHAP[1] and LIME[2] libraries. For SHAP, we used the universal *Explainer* method, which itself determined to use PartitionSHAP, faster version of KernelSHAP that hierarchically clusters features. For LIME, it was set that the maximum number of features present in an explanation equals 50 and the size of the neighborhood to learn the linear model equals 500. For both methods, we provided the label that the model predicted, not the actual one. The output of interpretation models can contain words with repetitions, so we decided to conduct an experiment also for a unique output, when repeated words are removed from the explanation list.

Table 1 shows the output list from the interpretation models for a text from a 20NewsGroup dataset whose label is "baseball" and the model also predicted the baseball category. The _glove postfix means GloVe and _ft postfix means fastText embeddings were used for calculating scores. The last two columns represent the NDCG values of interpretation results, where 1 in the cell means used embeddings.

[1] `https://github.com/slundberg/shap`
[2] `https://github.com/marcotcr/lime`

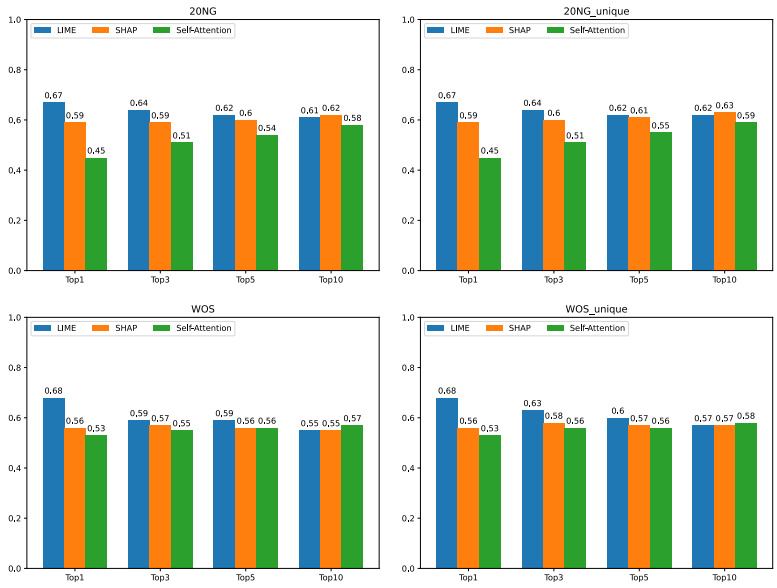

Figure 2. Interpretation results using GloVe embeddings.

The text itself is as follows: "*Pitchers are required to pitch (or feint or attempt a pick-off) within* 20 *seconds after receiving the ball, not 15. Pitchers are required to pitch their warm-up throws within a one minute time frame, beginning after each half inning ends, not two minutes. And the reason why a reliever should be allowed warm-ups is simple: Different mound, different catcher. Ryan Robbins Penobscot Hall University of Maine IO20456@Maine.Maine.Edu*".

We can see in the example (Table 1) that the first word predicted by the SHAP method ("relieve") is quite general, but the word "catcher" high-scored by Self_Attention is very specific for the baseball domain. The overall NDCG@10 for Self_attention is close to ideal, NDCG@10 for SHAP is much lower.

The results of the experiments are shown in Figure 2 for GloVe embeddings and Figure 3 for fastText embeddings. The _unique postfix means that the top $N \in \{1, 3, 5, 10\}$ interpretation output contains only unique words.

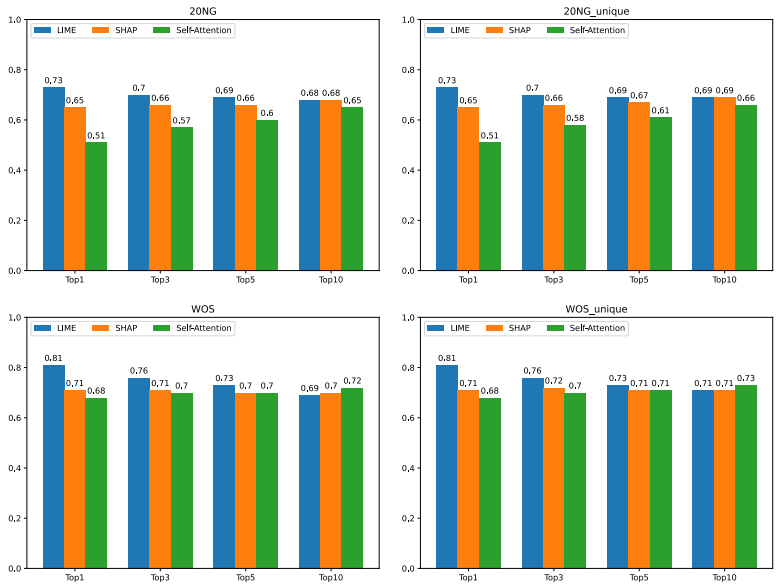

Figure 3. Interpretation results using fastText embeddings.

We can see that the first word in LIME explanations is much closer semantically to the category label for both datasets and both embeddings. LIME NDCG scores are higher at all levels than for SHAP, and almost at all levels for Self_Attention (except NDCG@10 for the WOS dataset). SHAP NDCG scores are much lower at all levels than both other scores. This agrees with the findings from [9] based on human evaluation that LIME was the best method in identifying the category of an example based on the explanation.

§7. Conclusion

In this work, we have suggested an automatic method to measure human-grounded explainability of interpretation techniques in text categorization tasks. We calculate the semantic similarity of explanations with the category label using word embeddings and NDCG measure adapted from

information retrieval. We applied our approach to two datasets: 20NewsGroup and WOS dataset of scientific articles. We compared three well-known methods of explanation: LIME, SHAP, and Self_Attention. We used GloVe and fastText embeddings to calculate the semantic similarity.

We found that the LIME technique achieves the best NDCG scores of semantic similarity for the both datasets and both embeddings, which means that LIME is better suited to explain the obtained category for a specific example.

In the future, we plan to continue the study of interpretability for machine learning models, including trying to adjust the parameters of LIME and SHAP methods and considering whether these results can be improved. In particular, we plan to use Sentence Transformers to compare the formulation of a category's name and word lists generated by interpretation methods.

References

1. *20 newsgroups dataset*, `http://people.csail.mit.edu/jrennie/20Newsgroups/`.
2. J. Devlin, M. Chang, K. Lee, and K. Toutanova, *BERT: pre-training of deep bidirectional transformers for language understanding*, CoRR **abs/1810.04805** (2018).
3. F. Doshi-Velez and B. Kim, *Towards a rigorous science of interpretable machine learning*, stat **1050** (2017), 2.
4. E. Doumard, J. Aligon, E. Escriva, J.-B. Excoffier, P. Monsarrat, and C. Soulé-Dupuy, *A quantitative approach for the comparison of additive local explanation methods*, Information Systems **114** (2023), 102162.
5. Facebook AI, *fastText: Library for fast text representation and classification*, 2016.
6. A. Garcia-Silva and J. M. Gomez-Perez, *Classifying scientific publications with BERT: Is self-attention a feature selection method?*, Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I, Springer, 2021, pp. 161–175.
7. K. Järvelin and J. Kekäläinen, *Cumulated gain-based evaluation of IR techniques*, ACM Transactions on Information Systems (TOIS) **20** (2002), no. 4, 422–446.
8. K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, , M. S. Gerber, and L. E. Barnes, *Hdltex: Hierarchical deep learning for text classification*, Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on, IEEE, 2017.
9. P. Lertvittayakumjorn and F. Toni, *Human-grounded evaluations of explanation methods for text classification*, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 5195–5205.
10. X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, *Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond*, Knowledge and Information Systems **64** (2022), no. 12, 3197–3234.

11. S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, Advances in neural information processing systems **30** (2017).
12. A. Madsen, S. Reddy, and S. Chandar, *Post-hoc interpretability for neural NLP: A survey*, ACM Computing Surveys **55** (2022), no. 8, 1–42.
13. J. Pennington, R. Socher, and C. D. Manning, *GloVe: Global vectors for word representation*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
14. M. T. Ribeiro, S. Singh, and C. Guestrin, *"Why should i trust you?" Explaining the predictions of any classifier*, Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
15. M. O. Yalcin and X. Fan, *On evaluating correctness of explainable ai algorithms: an empirical study on local explanations for classification*, 2021.

Bauman Moscow State
Technical University,
Moscow, Russia
E-mail: rogov.alisher@gmail.com

Lomonosov Moscow State University,
Moscow, Russia
E-mail: louk_nat@mail.ru

Поступило 6 сентября 2023 г.