

Evaluating the Performance of Interpretability Methods in Text Categorization Task

A. A. Rogov^{1*} and N. V. Loukachevitch^{2**}

(Submitted by E. K. Lipachev)

¹*Bauman Moscow State Technical University, Moscow, 105005 Russia*

²*Lomonosov Moscow State University, Moscow, 119991 Russia*

Received January 9, 2024; revised January 31, 2024; accepted February 14, 2024

Abstract—Neural networks are progressively assuming a larger role in individuals daily routines, as their complexity continues to grow. While the model demonstrates satisfactory performance when evaluated on the test data, it often yields unforeseen outcomes in real-world scenarios. To diagnose the source of these errors, understanding the decision-making process employed by the model becomes crucial. In this paper, we consider various methods of interpreting the BERT model in classification tasks, and also consider methods for evaluating interpretation methods using vector representations fastText, GloVe and Sentence-BERT.

DOI: 10.1134/S1995080224600699

Keywords and phrases: *interpretability, BERT, classification.*

1. INTRODUCTION

As neural networks have advanced, they have demonstrated achievements comparable to, and in some instances surpassing, human performance in specific domains. This progress, however, has been accompanied by the escalating complexity of models, characterized by an exponential growth in the number of parameters integrated into the network. The interpretability of deep learning models is inherently challenging due to their “black box” nature, a challenge that is exacerbated with increasing complexity [1]. The significance of interpretability in machine learning lies in its capacity to enhance the credibility of the model. There exists a prevalent apprehension toward relying on machine learning models, especially in critical tasks. This reluctance may arise when individuals encounter novel technologies, and such hesitancy can impede the expeditious adoption of these advancements.

Large Language Models (LLMs) have significantly improved their ability to generate coherent explanations [2, 3]. Unlike conventional deep learning models, the expansive scale of LLMs, in relation to both parameters and training data, presents intricate challenges and promising prospects for research in explainability. Primarily, as these models increase in size, the comprehension and interpretation of their decision-making processes become more challenging owing to heightened internal complexity and the extensive nature of training data. Moreover, this complexity imposes substantial demands on computational resources required for generating explanations [4]. However, there remains a question of how well these explanations truly align with reality and accurately interpret information. As natural language processing and text generation technologies continue to advance, it becomes crucial to thoroughly investigate the quality and reliability of information provided by these models. This is essential not only for enhancing their effectiveness across various domains but also for ensuring the trustworthiness of results obtained through their utilization. Thus, the question of how proficient large language models are in delivering precise and plausible explanations remains open and necessitates further research and refinement.

*E-mail: rogov.alisher@gmail.com

**E-mail: louk_nat@mail.ru

Evaluation of interpretability methods can be approached from three distinct perspectives: application-grounded, functionally-grounded, or human-grounded, as highlighted in previous works [5, 6]. Application-grounded evaluation focuses on assessing the implications of interpretability within the specific target environment, such as examining the utility of explanations in financial services. Functionally-grounded evaluation is directed at gauging the alignment of the explanation with the underlying model. In parallel, human-grounded evaluation aims to ascertain the comprehensibility of explanations from the perspective of human understanding.

In this study, we focus on post-hoc interpretation methods applied to the text categorization task, asserting that for an explanation to be human-grounded, it should exhibit semantic relevance to the category's name. A user's perception of semantic similarity between the explanation and the category is deemed crucial. We compare various established approaches for interpreting the outcomes of deep learning models, including LIME [7], SHAP [8], and the self-attention mechanism of BERT for interpretation [9], employing the aforementioned human-grounded criterion.

The interpretation methods considered in this paper yield a ranked list of words with corresponding weights, which indicate the contribution of each word to the model's decision-making process. Subsequently, we select the top N words, where N can take values from the set $\{1, 3, 5, 10\}$, that have contributed most positively to the classification output of the interpreted model. In order to compare these interpretation results with the category name, we employ two approaches.

In first approach, we utilize word and sentence embeddings, along with the Normalized Discounted Cumulative Gain (NDCG) metric from information retrieval [10]. This allows us to evaluate the semantic similarity between the interpretation results and the category name.

In second approach, we pass the entire interpretation result as a sentence to the Sentence-BERT model [11], which generates a single embedding representation. We then compare these selected words as sentence with the category name in terms of their semantic similarity using embeddings.

2. RELATED WORK

Numerous explanation methodologies have been proposed to enhance the interpretability of machine-learning models. However, determining the most reliable method can be challenging. Yalcin and Fan [12] conducted an analysis of explanations provided by SHAP and LIME methods in the classification of poisonous mushrooms using the mushroom dataset. Their findings revealed disparities in explanations, especially regarding the most crucial feature, for over a third of the samples.

In the study by Doumard et al. [13], local-explanation methods were scrutinized across a broad spectrum of 304 OpenML datasets using six quantitative metrics. The research highlighted the efficiency of LIME and SHAP approximations in high-dimensional scenarios, producing intelligible global explanations. However, these methods exhibited a lack of precision in local explanations.

Given the distinctive nature of natural language processing tasks, interpretability approaches necessitate separate exploration. Lertvittayakumjorn et al. [14] delved into interpretability methods for text categorization. Three evaluation tasks were considered: 1) determining the best classification model based on explanations; 2) identifying the category of an example through explanations; 3) aiding in the analysis of examples with low probabilities. Notably, the LIME method demonstrated superior performance in the second task, excelling at identifying the most relevant evidence for a class, regardless of class correctness. The study, encompassing Amazon reviews and arXiv papers, involved crowdsourcers and post-graduate students for the respective datasets.

In [15] the authors present ERASER, a dataset for evaluation of explainable approaches in NLP. This benchmark includes several datasets and tasks for which human annotations of explanations (supporting evidence) are available. Several metrics were proposed to measure explainability. The authors evaluated several known methods of post-hoc explanation including simple gradients, attention weights, and LIME. They found that "both simple gradient and LIME-based scoring yield more comprehensive rationales than attention weights". In [16], the authors propose a novel attention mechanism in neural networks. They compare explainability of their attention approach on the ERASER dataset and found that the explanation become more comprehensible and sufficient than basic attention maps but still worse than LIME explanations according to metrics.

The authors of [2] evaluate explanation abilities of current large language models in the sentiment analysis task. They found that ChatGPT's self-explanations perform on par with traditional ones. Ye and

Durrett [3] in similar evaluation on two NLP tasks that involve reasoning over text (question answering and natural language inference) conclude that LLM's internal "reasoning" does not always align with explanations that it generates. Besides, the explanations might do not correspond to facts described in the provided prompt. The LLM explanations are written in good English and look very convincing, but they may deceive users into believing the incorrect system's responses.

This present study focuses on the automatic evaluation of the second task outlined in the aforementioned work by Lertvittayakumjorn et al. [14]. Specifically, 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 [7], SHAP [8] and self-attention weights [9].

3.1. LIME

LIME (Local Interpretable Model-agnostic Explanations), introduced by Ribeiro et al. [7], is a method for local interpretation that operates independently of the underlying machine learning model. Local interpretability involves understanding the rationales behind a specific decision. LIME achieves this by providing a locally faithful explanation, constructing a set of perturbed samples near the target sample and fitting them to a potentially interpretable model, such as linear models or decision trees [1].

The interpreted explanation generated by LIME is expressed as a binary vector that illustrates the involvement of each parameter in the outcome. In the context of text classification, this binary vector serves as a potentially interpretable representation, indicating the presence or absence of specific words. It's noteworthy that, despite the classifier utilizing more complex and less understandable features like word embeddings, LIME simplifies the explanation to a binary vector for better interpretability [7].

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

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

where $x' \in \{0, 1\}^{d'}$ is a binary vector of 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 approaches f in the locality defined by π_x ,

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

Indeed, the core idea behind the LIME approach is to approximate the prediction (f) of the model for a given test case (x) by employing a simpler and easily interpretable model (g), which operates based on a simplified representation. Consequently, the generated explanation ($\xi(x)$) elucidates the interpretation of the target sample x using linear weights, particularly when g takes the form of a linear model.

3.2. SHAP

SHAP [8] (SHapley Additive exPlanations) is a game theoretic approach designed to elucidate the output of any machine learning model. It establishes a connection between optimal credit allocation and local explanations through the utilization of classic Shapley values from game theory and their relevant extensions.

Shapley regression values hold significance for linear models, especially in the presence of multicollinearity. The computation of Shapley values necessitates the retraining of the model for all subsets

of features $S \subseteq F$, where F denotes the set of all features. These values assign an importance value to each feature, indicating the significance of including that feature in the model's prediction.

To calculate the Shapley value, the $f_{S \cup \{i\}}$ model is trained with the inclusion of the specific feature, while the f_S model is trained without it. Subsequently, 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 importance depends on other features in the model, and the aforementioned differences are calculated for all possible subsets $S \subseteq F \setminus \{i\}$. Shapley values are then computed as a weighted average of all potential differences:

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

The exact computation of Shapley values is challenging. The work [8] 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 $x' \in \{0, 1\}^M$ will denote a binary vector for its interpretable representation and h be the mapping function $x = h_x(x')$. SHAP values are the solution of

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

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 [8] 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 represents an endeavor to explore the feasibility of employing weights in the attention mechanism as a local interpretation for transformer architecture models. The methodology is grounded in the investigation conducted in [9], which explores the potential relationship between self-attention and feature selection methods from various perspectives. These perspectives include the alignment of vocabulary, similarity in ranking, relevance to the subject area, stability of features, and the efficacy of classification.

In the initial step, the average weights for the 12 heads of attention in the last hidden layer are computed for each input sequence. Subsequently, a new matrix of weights is generated by grouping subwords within a word and averaging the weights of these subwords. The vertical average is then considered as the weight of the word. The top 10 words are subsequently regarded as the interpretation.

It is important to note that this method is independent of the model's response and is specific to transformer models. To illustrate this, Fig. 1 depicts the mean weights of the 12 self-attention heads in the last hidden state of the trained *bert-base-uncased* [17] BERT model, fine-tuned on the WOS [21] dataset for “*Interpretability of thematic classification of texts based on neural networks*”. The plot distinctly reveals a vertical pattern where a few tokens, such as *training*, *deep*, *transformer*, *language*, and *understanding*, receive the majority of attention. Notably, special tokens $\langle SEP \rangle$ and $\langle CLS \rangle$ are excluded from the analysis in [9] due to the potential dominance of attention received by these tokens, making attention to other tokens barely noticeable.

4. METHOD FOR AUTOMATIC EVALUATION OF INTERPRETABILITY

After applying interpretation methods and obtaining results in a format conducive to human perception, it becomes imperative to assess the adequacy of the interpretation outcome. While one method of evaluation involves soliciting expert opinions to gauge the clarity of the interpretation, this approach is notably resource-intensive and expensive.

We posit that the greater the similarity between an explanation and the category's name in a text categorization task, the more comprehensible the explanation is for a human. This perspective enables us to conduct automatic evaluations of explanation methods. We focus on the top $N \in \{1, 3, 5, 10\}$ output words, sorted by the weights assigned to them by the methods. These weights signify the importance of the word in the decision-making process.

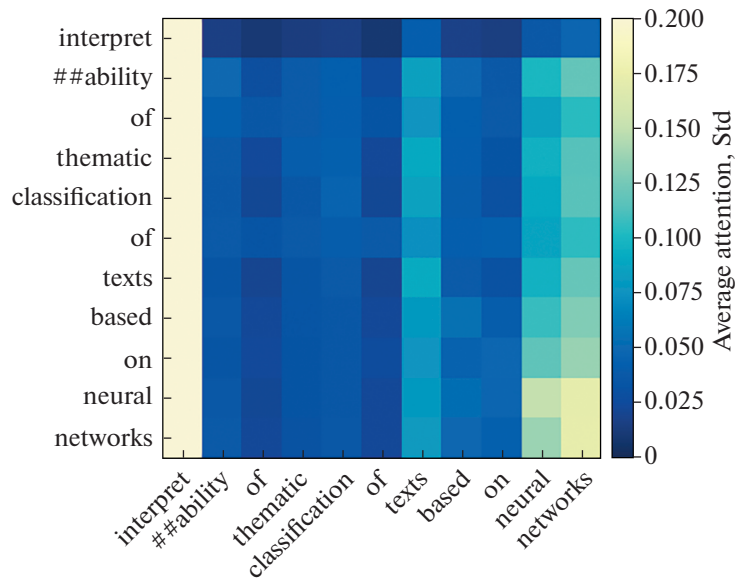


Fig. 1. An example of a matrix of weights in the form of an attention mechanism.

4.1. NDCG

Since all the above-mentioned methods of interpretation return as a result a ranking list of words with weights, we can compare these lists with the category name using the NDCG measure adapted from information retrieval [10].

Let $D^i = \{d_1^i, \dots, d_m^i\}$ be the set of items retrieved for the query q^i and $\{rel^i(1), \dots, rel^i(m)\}$ be their relevance labels. Let $\sigma = \{v_1, \dots, v_m\}$ be the ordered items and the $DCG@k$ metric (discount cumulative gain at the position k) of the ordering is

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

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 a discount 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 d_1^i and q^i embeddings. For calculation of 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 GloVe¹⁾ [18], fastText²⁾ [19] and unigram_Sentence-BERT³⁾ [11] models. The unigram prefix before Sentence-BERT means that we passed only one word to the model.

The selection of embedding models such as GloVe and fastText was made based on several factors, including their historical significance in the field of natural language processing and their widespread use as benchmarks in various NLP tasks. There are newer models like MiniLM-L12-H384 offer improvements in terms of size and quality, our decision to include these established models was driven by the desire to provide a baseline for comparison. These models serve as reference points for assessing the performance of newer approaches. We recognize the importance of incorporating the latest models and advancements in our research. In future work, we plan to expand our comparison to include more recent models.

¹⁾<http://nlp.stanford.edu/data/glove.840B.300d.zip>

²⁾<https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.en.zip>

³⁾<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

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

ideal-unigram_ Sentence-BERT	ideal-GloVe	ideal-fastText	SHAP	LIME	Self-Attention
pitchers	pitchers	catcher	relieve	inning	catcher
inning	catcher	pitchers	catcher	pitchers	ball
ball	inning	inning	mound	not	mound
catcher	ball	reliever	pitchers	maine	pitchers
pitch	pitch	pitch	inning	of	allowed
mound	reliever	ball	ball	are	warm
reliever	mound	mound	pitch	hall	off
throws	pick	pick	ups	is	pitch
ryan	university	university	throws	reliever	inning
hall	hall	hall	required	pitch	reliever

4.2. Sentence-BERT

Sentence-BERT [11] is a modified version of the BERT network that utilizes siamese and triplet networks to generate sentence embeddings that capture semantically meaningful information. Unlike traditional methods that treat sentences as a sequence of words, Sentence-BERT takes into account the entire sentence as a unit during the encoding process. This allows it to capture the overall meaning and nuances of the sentence, rather than just focusing on individual words or tokens. By training the network on various tasks, such as sentence similarity or text classification, Sentence-BERT learns to produce sentence embeddings that are semantically meaningful and can be used for downstream applications like information retrieval, clustering, or sentiment analysis.

In light of the possibility of using Sentence-BERT to obtain a vector representation of sentences, we hypothesized that passing the results of interpretation methods as sentences and converting them to vector representations using Sentence-BERT could be useful. In addition, we can obtain a vector representation of a text category using Sentence-BERT and calculate the semantic proximity between the interpretation result and the category name. This will allow us to evaluate the effectiveness of the interpretation methods. For calculation the semantic similarity we use cosine similarity.

4.3. Example of Applying Interpretation And Evaluation Methods

The interpretation models were applied to a text sample from the 20NewsGroup dataset with the “baseball” label, and the model successfully predicted the category as “baseball” as well. The generated output list from the interpretation models is presented in Table 1. Each column represents the output result of interpretation, presented as a ranking list of words ordered in the rows by their weights. The NDCG@10 evaluation results of Table 1 are presented in Table 2.

The text itself: *“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”*

In the provided example (Table 1), we observe that the SHAP method predicts a relatively general word (“relieve”) as the top interpretation, while the Self-Attention method assigns a high score to the more specific term “catcher”, which is highly relevant to the baseball domain. The overall NDCG@10 score for the Self-Attention method is close to the ideal value across all embeddings, indicating its strong performance. On the other hand, the SHAP method achieves a significantly lower NDCG@10 score, but when used with Sentence-BERT embeddings, it approaches the ideal value.

Table 2. NDCG results for text about baseball

	SHAP	LIME	Self-Attention
ideal-unigram_Sentence-BERT	0.93	0.75	0.90
ideal-GloVe	0.74	0.77	0.93
ideal-fastText	0.74	0.75	0.88

In terms of the evaluation using Sentence-BERT (SBERT) approach, where the interpretation results are passed as a sentence to Sentence-BERT and the semantic similarity is calculated, the results closely align with the NDCG scores, indicating a similar trend. Let SHAP sentence be “relieve, catcher, mound, pitchers, inning, ball, pitch, ups, throws, required”, the LIME sentence be “inning, pitchers, not, maine, of, are, hall, is, reliever, pitch”, the Self-Attention sentence be “catcher, ball, mound, pitchers, allowed, warm, off, pitch, inning, reliever” and the category sentence be “baseball”.

$$\begin{aligned} \cos(\text{SBERT}(\text{SHAP}), \text{SBERT}(\text{category})) &= 0.56, \\ \cos(\text{SBERT}(\text{LIME}), \text{SBERT}(\text{category})) &= 0.51, \\ \cos(\text{SBERT}(\text{Self} - \text{Attention}), \text{SBERT}(\text{category})) &= 0.58. \end{aligned}$$

5. DATASETS

Experiments were conducted on two datasets: 20NewsGroup [20] and WOS [21].

Web Of Science (WOS) dataset is a collection of academic articles’ abstracts that includes three corpora (5736, 11967, and 46985 documents) corresponding to 11, 34, and 134 topics, respectively. In our work, we utilized the WOS-11967 dataset with 34 topics. For this dataset, 70% of the samples were used for training, and 30% were reserved for validation. For multiword category names (e.g., *Machine learning*), we employed the averaging of component word embeddings for GloVe and fastText models.

20NewsGroup dataset comprises 18846 documents, each with a maximum length of 1000 words. In this dataset, 14846 samples were allocated for training, while 4000 samples were set aside for validation. To ensure more realistic data and prevent the model from learning to classify newsgroup-related metadata, the text was cleaned.

6. EXPERIMENTS

For classification, we used *bert-base-uncased* BERT model [17] and fine-tuned it on the datasets. For 20NewsGroup we got 71.3 accuracy, and for WOS-11967 we got 86.3 accuracy.

After obtaining the trained models, we employed standard methods from the SHAP⁴⁾ and LIME⁵⁾ libraries. For SHAP, we utilized the universal *Explainer* method, which autonomously opted for PartitionSHAP, a faster version of KernelSHAP that hierarchically clusters features. Regarding LIME, we configured it to have a maximum of 50 features present in the explanation and a neighborhood size of 500 to learn the linear model. For both methods, we provided the label predicted by the model, not the actual one. Since the output of interpretation modes may contain words with repetitions, we decided to conduct an experiment for a unique output as well, where repeated words are removed from the explanation list.

The results of the experiments using NDCG evaluation are shown on Fig. 2 for GloVe embeddings, Fig. 3 for fastText embeddings, Fig. 4 for unigram_Sentence-Bert embeddings. The *_unique* postfix means that the top $N \in \{1, 3, 5, 10\}$ interpretation output contains only unique words. Tables 3, 4, and 5 present the results, that are averaged across various values of N , where N belongs to the set $\{1, 3, 5, 10\}$.

We can see that the first word in LIME explanations is much semantically closer to the category label for both datasets and both embeddings. LIME NDCG scores are higher at all levels than for SHAP,

⁴⁾<https://github.com/slundberg/shap>

⁵⁾<https://github.com/marcotcr/lime>

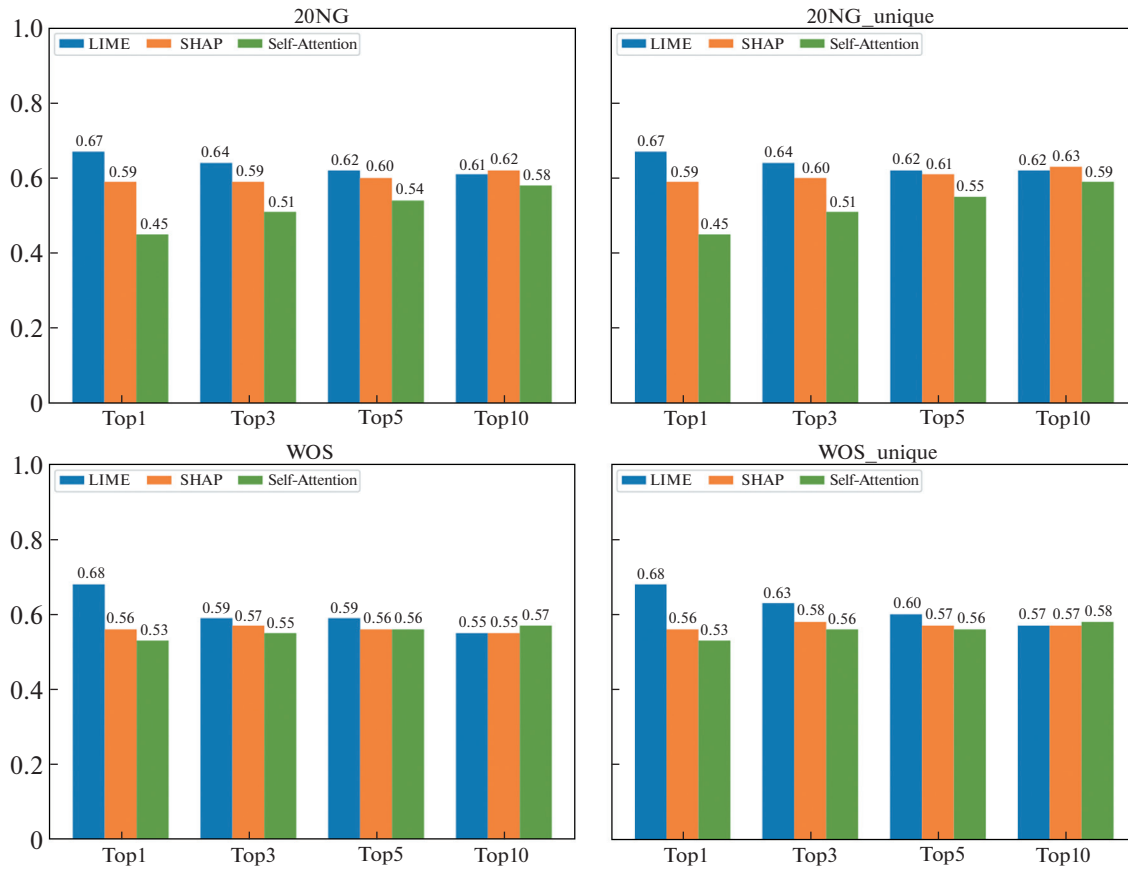


Fig. 2. Interpretation results using GloVe embeddings.

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 findings from [14] based on human evaluation that LIME was the best method in identifying the category of an example based on explanation.

Tables 6 and 7 present the experimental findings of the evaluation conducted on Sentence-BERT. The results are averaged across various values of N , where N belongs to the set $\{1, 3, 5, 10\}$. The suffix “_unique” signifies that the top N interpretation outputs comprise only unique words. Two approaches were explored to combine the interpretation results into sentences: the first involved joining them with spaces, while the second employed commas.

Joining by spaces involves separating words in a sentence with spaces but not adding any additional punctuation. It is a common way to represent text data in its original form for some natural language processing tasks. It is used to preserve the original structure of sentences without introducing additional separation symbols.

Table 3. The mean value of TopN interpretation results using NDCG evaluation (GloVe Results)

	LIME	SHAP	Self-Attention
20NG	0.630	0.600	0.520
20NG_unique	0.637	0.607	0.525
WOS	0.602	0.560	0.552
WOS_unique	0.620	0.570	0.557

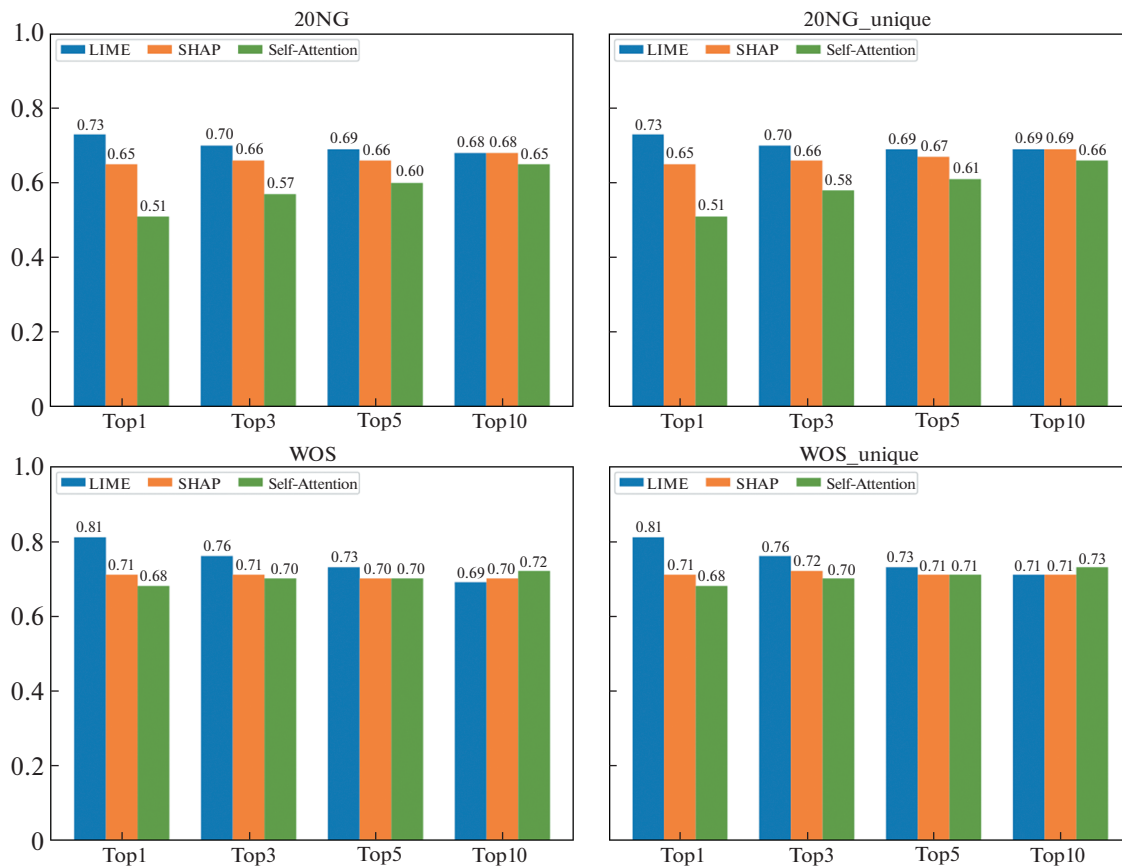


Fig. 3. Interpretation results using fastText embeddings.

Joining words into sentences by commas involves using commas as delimiters between words in a sentence. This method is used for specific text analysis tasks to understand whether the presence of commas is meaningful. Maybe it can help algorithms detect relationships between items in a list or better understand sentence structure.

Table 4. The mean value of TopN interpretation results using NDCG evaluation (fastText Results)

	LIME	SHAP	Self-Attention
20NG	0.700	0.662	0.582
20NG_unique	0.702	0.667	0.590
WOS	0.747	0.705	0.700
WOS_unique	0.752	0.712	0.705

Table 5. The mean value of TopN interpretation results using NDCG evaluation (unigram_Sentence-BERT Results)

	LIME	SHAP	Self-Attention
20NG	0.752	0.717	0.650
20NG_unique	0.755	0.722	0.660
WOS	0.712	0.652	0.667
WOS_unique	0.717	0.662	0.675

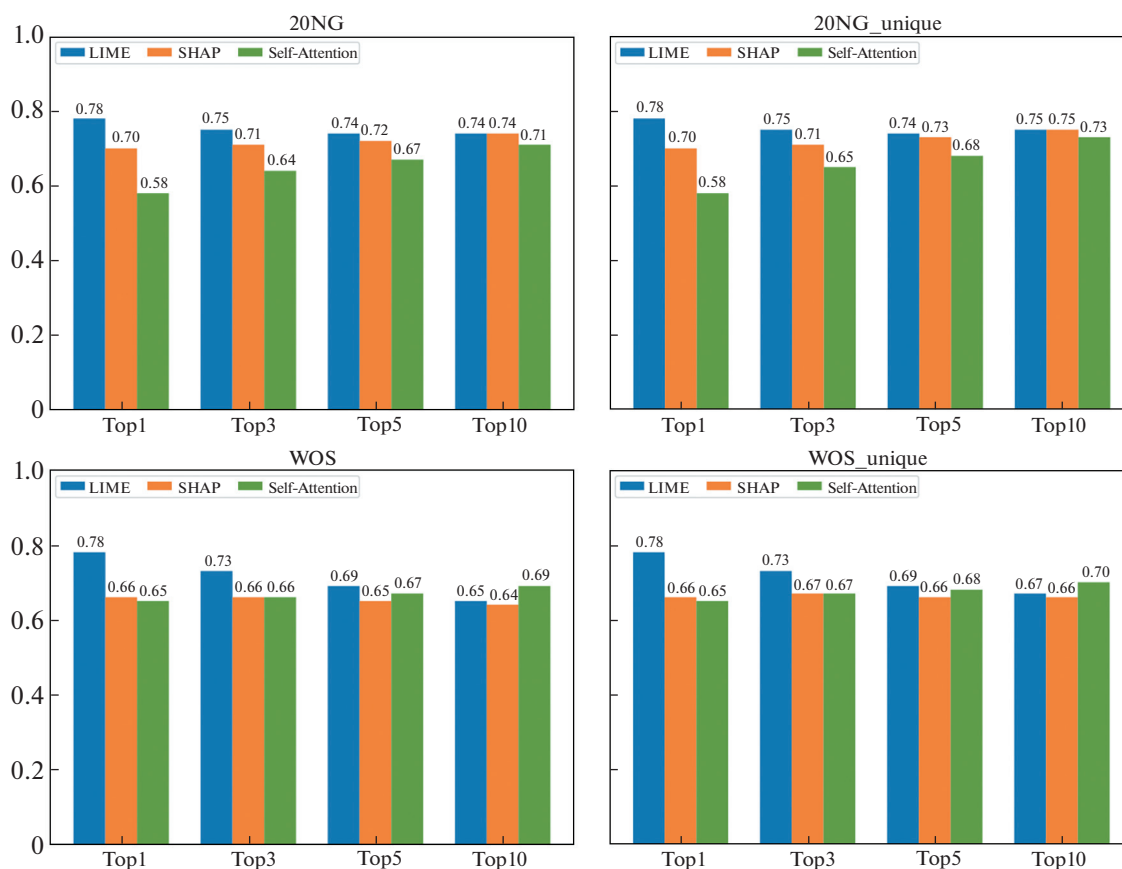


Fig. 4. Interpretation results using unigram_Sentence-BERT embeddings.

Comparing the two approaches revealed that using commas yielded slightly superior outcomes. The scores consistently indicate that LIME outperforms both SHAP and Self-Attention, suggesting a consistent pattern.

Table 6. The mean value of TopN interpretation results using Sentence-BERT evaluation (Joined by spaces)

	LIME	SHAP	Self-Attention
20NG	0.352	0.327	0.277
20NG_unique	0.347	0.327	0.277
WOS	0.457	0.437	0.385
WOS_unique	0.447	0.432	0.372

Table 7. The mean value of TopN interpretation results using Sentence-BERT evaluation (Joined by commas)

	LIME	SHAP	Self-Attention
20NG	0.395	0.375	0.320
20NG_unique	0.395	0.375	0.320
WOS	0.472	0.449	0.412
WOS_unique	0.470	0.447	0.410

7. CONCLUSIONS

In this research, we proposed an automated approach for assessing the interpretability of interpretation methods in text categorization tasks. Our method involves measuring the semantic similarity between explanations and category labels using word embeddings and adapting the NDCG measure from information retrieval. Additionally, we utilized Sentence-BERT embeddings by treating the explanation as a sentence. We conducted experiments on two datasets: 20NewsGroup and the WOS dataset consisting of scientific articles. Three widely recognized interpretation methods, namely LIME, SHAP, and Self-Attention, were compared in our study. We employed GloVe, fastText, and Sentence-BERT embeddings to calculate the semantic similarity.

Our findings indicate that the LIME technique outperforms the other methods in terms of NDCG scores. Moreover, when using Sentence-BERT evaluation, LIME consistently achieves higher scores for semantic similarity on both datasets and across all embeddings. This suggests that LIME provides more effective explanations for accurately capturing the assigned category in specific examples.

In the future, it is planned to continue the study of the interpretability of machine learning models, including trying to adjust the parameters of the LIME and SHAP methods and consider whether these results can be improved.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. X. Li et al., “Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond,” *Knowledge Inform. Syst.* **64**, 3197–3234 (2022).
2. S. Huang et al., “Can large language models explain themselves? A study of LLM-generated self-explanations,” arXiv: 2310.11207 (2023).
3. X. Ye and G. Durrett, “The unreliability of explanations in few-shot prompting for textual reasoning,” *Adv. Neural Inform. Process. Syst.* **35**, 30378–30392 (2022).
4. H. Zhao et al., “Explainability for large language models: A survey,” *ACM Trans. Intell. Syst. Technol.* **15** (2) (2023).
5. A. Madsen, S. Reddy, and S. Chandar, “Post-hoc interpretability for neural nlp: A survey,” *ACM Comput. Surv.* **55** (8), 1–42 (2022).
6. F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” arxiv: 1702.08608 (2017).
7. M. T. Ribeiro et al., “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* (2016), pp. 1135–1144.
8. S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” *Adv. Neural Inform. Process. Syst.* **30** (2017).
9. A. Garcia-Silva and J. M. Gomez-Perez, “Classifying scientific publications with BERT-Is self-attention a feature selection method?,” in *Proceedings of the European Conference on Information Retrieval* (Springer Int., Cham, 2021), pp. 161–175.
10. K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Trans. Inform. Syst.* **20**, 422–446 (2002).
11. N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using Siamese bert-networks,” in *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. 3982–3992.
12. M. O. Yalcin and X. Fan, “On evaluating correctness of explainable AI algorithms: An empirical study on local explanations for classification,” 0-7 (2021).
13. E. Doumard et al., “A quantitative approach for the comparison of additive local explanation methods,” *Inform. Syst.* **114**, 102162 (2023).

14. P. Lertvittayakumjorn and F. Toni, “Human-grounded evaluations of explanation methods for text classification,” in *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.
15. J. De Young et al., “ERASER: A benchmark to evaluate rationalized NLP models,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 4443–4458.
16. Y. Wang et al., “Convolution-enhanced evolving attention networks,” *IEEE Trans. Pattern Anal. Mach. Intell.* **45** (7) (2023).
17. J. Devlin et al., “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019), Vol. 1, pp. 4171–4186.
18. J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing EMNLP* (2014), pp. 1532–1543.
19. Inc Facebook, *fastText*: Library for Fast Text Representation. <https://github.com/facebookresearch/fastText>. Accessed 2024.
20. 20 Newsgroups Dataset. <http://people.csail.mit.edu/jrennie/20Newsgroups/>. Accessed 2024.
21. K. Kowsari et al., “Hdltex: Hierarchical deep learning for text classification,” in *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications ICMLA* (IEEE, 2017), pp. 364–371.

Publisher’s Note. Pleiades Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.