A Hierarchical Explanation Generation Method Based on Feature Interaction Detection

Yiming Ju, Yuanzhe Zhang, Kang Liu, Jun Zhao
[1] The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Science
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
{yiming.ju, yzzhang, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

The opaqueness of deep NLP models has motivated efforts to explain how deep models predict. Recently, work has introduced hierarchical attribution explanations, which calculate attribution scores for compositional text hierarchically to capture compositional semantics. Existing work on hierarchical attributions tends to limit the text groups to a continuous text span, which we call the connecting rule. While easy for humans to read, limiting the attribution unit to a continuous span might lose important long-distance feature interactions for reflecting model predictions. In this work, we introduce a novel strategy for capturing feature interactions and employ it to build hierarchical explanations without the connecting rule. The proposed method can convert ubiquitous non-hierarchical explanations (e.g., LIME) into their corresponding hierarchical versions. Experimental results show the effectiveness of our approach in building high-quality hierarchical explanations.

1 Introduction

The opaqueness of deep natural language processing (NLP) models has increased along with their power (Doshi-Velez and Kim, 2017), which has prompted efforts to explain how these "black-box" models work (Sundararajan et al., 2017; Belinkov and Glass, 2019). This goal is usually approached with attribution method, which assesses the influence of inputs on model predictions (Ribeiro et al., 2016; Sundararajan et al., 2017; Chen et al., 2018)

Prior lines of work on attribution explanations usually calculate attribution scores for predefined text granularity, such as word, phrase, or sentence. Recently, work has introduced the new idea of hierarchical attribution, which calculates attribution scores for compositional text hierarchically to capture more information for reflecting model predictions (Singh et al., 2018; Tsang et al., 2018; Jin et al., 2019; Chen et al., 2020) As shown in Figure 1, hierarchical attribution produces a hierarchical composition of words, and provides attribution scores for every text group. By providing compositional semantics, hierarchical attribution can give users a better understanding of the model decision-making process. (Singh et al., 2018).

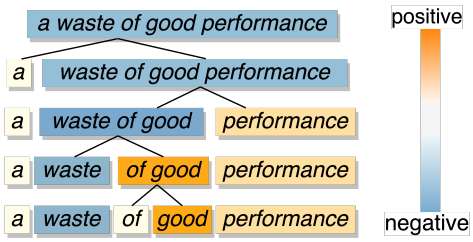


Figure 1: An example of hierarchical attribution from Chen et al. (2020).

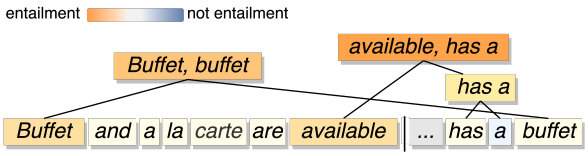
However, as illustrated in Figure 1, recent work (Singh et al., 2018; Jin et al., 2019; Chen et al., 2020) uses continuous text to build hierarchical attributions, which we call **the connecting rule**. While consistent with human reading habits, using the connecting rule as an additional prior might lose important long-distance compositional semantics. The concerns are summarized as follows:

First, modern NLP models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018, 2019) are almost all transformer-based, which use self-attention mechanisms (Vaswani et al., 2017) to capture feature interactions. Since all interactions are calculated in parallel in self-attention mechanism, the connecting rule that only considering neighboring text is incompatible with the basic operation principle of these NLP models.

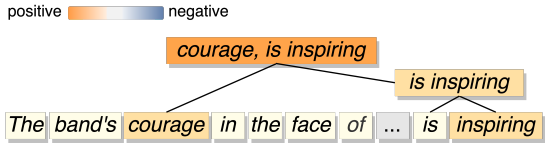
Second, unlike the example in Figure 1, NLP tasks often require joint reasoning of different parts of the input text (Chowdhary, 2020). For example, Figure 2(a) shows an example of natural language interface (NLI) task[1], in which *'has a'* and *'avail-*

[1]NLI is a task requiring the model to predict whether the premise entails the hypothesis, contradicts it, or is neutral.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12600–12611
July 9-14, 2023



(a) Natural Language Inference example.



(b) Sentiment Classification example.

Figure 2: Examples of hierarchical explanations which need long-distance compositional semantics. '...' represent omitted words for clear visualization.

able' are the key compositional semantics to make the prediction: entailment. However, the connecting rule cannot highlight the compositional effect between them because they are not adjacent. Even in relatively simple sentiment classification task, capturing long-distance compositional effect is also necessary. As shown in Figure 2(b), *'courage, is inspiring'* is an important combination but not adjacent.

In this work, we introduce a simple but effective method for generating hierarchical explanations without the connecting rule. Moreover, we introduce a novel strategy for detecting feature interactions in order to capture compositional semantics. Unlike earlier hierarchical attribution approaches, which use specific algorithms to calculate attribution scores, the proposed method can convert ubiquitous non-hierarchical explanations (e.g., LIME) into their corresponding hierarchical versions. We build systems based on two classic non-hierarchical methods: LOO (Lipton, 2018) and LIME (Ribeiro et al., 2016), and the experimental results show that both systems significantly outperform existing methods. Furthermore, the ablation experiment additionally reveals detrimental effects of the connecting rule on the construction of hierarchical explanations. Our implementation and genenerated explanations are available at an anonymous website: https://github.com/juyiming/HE_examples.

2 Method

This section explains the strategy for feature interaction detecting and the algorithm on building hierarchical explanations.

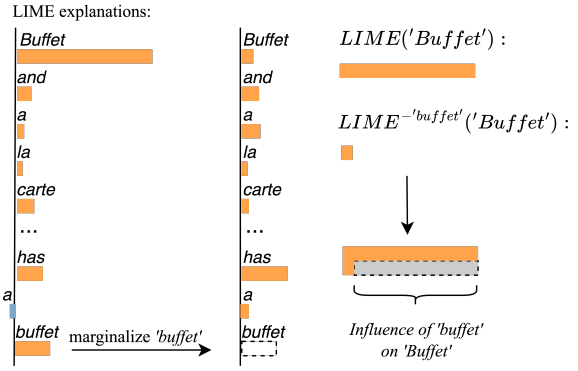


Figure 3: An example of calculating text interaction.

2.1 Detecting Feature Interaction

The structure of hierarchical explanations should be informative enough to capture meaningful feature interactions while displaying a sufficiently small subset of all text groups (Singh et al., 2018). Existing work uses different methods to calculate feature interactions for building hierarchical explanations. For example, Jin et al. (2019) uses multiplicative interactions as feature interaction and Chen et al. (2020) uses Shapley interaction index (Fujimoto et al., 2006).

Unlike previous methods, our approach quantifies feature interaction based on the chosen non-hierarchical method. Specifically, given an attribution algorithm $Algo$, our method measures the influence of one text group on the attribution score of another one. The interaction score between text group g_i and g_j can be calculate as follows:

$$\begin{aligned}\phi_{ij} = &\, abs(Algo(g_i) - Algo^{-g_j}(g_i)) \\ &+ abs(Algo(g_j) - Algo^{-g_i}(g_j)),\end{aligned} \quad (1)$$

where $Algo^{-g_j}(g_i)$ denotes the attribuition score of g_i with g_j be marginalized, abs stands for taking the absolute value.

Figure 3 shows an example of feature interaction detecting. Non-hierarchical method LIME gives the word *'Buffet'* a high attribution score, indicating that it is important for model prediction. This score, however, sharply declines after the word *'buffet'* is marginalized, indicating that *'buffet'* has a strong impact on *'Buffet'* under LIME. Note that in our method, different non-hierarchical attribution methods may lead to different hierarchical structures. Since the calculation principles and even the meaning of scores vary in different attribution methods, this property is more reasonable than building the same hierarchical structures for all attribution methods.

Method/Dataset	SST-2				MNLI				avg
	$AOPC_{pad}$		$AOPC_{del}$		$AOPC_{pad}$		$AOPC_{del}$		
	10%	*20%*	*10%*	*20%*	*10%*	*20%*	*10%*	*20%*	
LOO (Lipton, 2018)	34.8	43.3	34.6	42.0	64.5	65.8	66.5	68.2	52.5
L-Shapley (Chen et al., 2018)	31.9	41.0	38.8	45.6	62.1	67.4	69.2	71.8	53.5
LIME (Ribeiro et al., 2016)	39.3	56.6	40.3	55.8	73.4	79.3	76.6	78.9	62.5
ACD$^{\diamondsuit}$ (Singh et al., 2018)	31.9	38.3	31.1	39.0	60.5	61.4	59.5	61.1	47.9
HEDGE$^{\diamondsuit}$ (Chen et al., 2020)	34.3	46.7	34.0	44.1	68.2	70.9	68.3	70.9	54.7
$HE_{LOO}{}^{\diamondsuit}$	43.9	59.0	42.9	56.3	76.3	78.5	74.7	76.8	63.6
$HE_{LIME}{}^{\diamondsuit}$	42.0	62.4	44.1	61.9	80.1	86.6	83.2	87.3	68.5

Table 1: AOPC(10) and AOPC(20) scores of different attribution methods in on the SST and MNLI datasets. $\diamondsuit$ refers to method with hierarchical structure. *del* and *pad* refer to different modification strategies in AOPC.

Algorithm 1 Generating Hierarchical Structures

Input: sample text X with length n
  Initialization: $G_0 = \{\{x_1\}, \{x_2\}, ..., \{x_n\}\}$
  Initialization: is $H_X = \{G_0\}$
  for $t = 1, ..., n-1$ **do**
    $i, j = argmax\, \phi\ (g_i, g_j \,|\, G_{t-1})$
    $G_t \leftarrow\ (G_{t-1} \setminus \{g_i, g_j\}) \cup \{g_i \cup g_j\}$
    $H_X.add(G_t)$
  end for
Output: H_X

Feature marginalization. The criterion of selecting the feature marginalization approach is to avoid undermining the chosen attribution method. For example, LOO assigns attributions by the probability change on the predicted class after erasing the target text, so we use erasing as the marginalization method. For LIME, which estimates attribution scores by learning a linear approximation, we ignore the sampling points with the target feature during linear fitting.

2.2 Building Hierarchical Explanations

Based on the non-hierarchical attribution algorithm $Algo$, our method builds the hierarchical structure of input text and calculates attribution scores for every text group. Algorithm 1 describes the detail procedure, which recursively chooses two text groups with strongest interaction and merges them into a larger one. $X = (x_1, ..., x_n)$ denotes model input with n words; g denotes a text group containing a set of words in X; G_t denotes the collection of all text groups for the current step t; H_X denotes the hierarchical structure of X. G_0 is initialized with each x as a independent text group and H_X is initialized as $\{G_0\}$. Then, at each step, text groups with the highest interaction score from G_{t-1} are merged as on, and G_t is add into H_X. After $n-1$ steps, all words in X will be merged in one group, and H_X can constitute the final hierarchical structure of the input text.

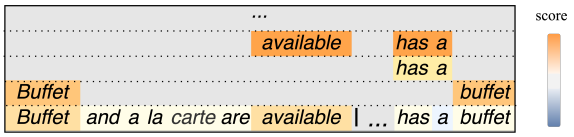


Figure 4: An example of visualization.

2.3 Visualization

Clear visualization is necessary for human readability. Since text groups in our hierarchical explanations are not continuous spans, the generated explanations cannot be visualized as a tree structure as shown in Figure 1. To keep clear and informative, the visualization only shows the newly generated unit and its attribution score at each layer. As shown in Figure 4, the bottom row shows the attribution score with each word as a text group (non-hierarchical attributions); The second row indicates {*'Buffet'*} and {*'buffet'*} are merged togather as one text group: {*'Buffet, buffet'*}; Similarly, the fourth row indicates the {*'has, a'*} and {*'availiable'*} are merged togather as one text group: {*'availiable, has, a'*}.

3 Experiment

We build systems with Leave-one-out (LOO) (Lipton, 2018) and LIME (Ribeiro et al., 2016) as the basic attribution algorithms, denoted as HE_{loo} and HE_{lime}. To reduce processing costs, we limit the

maximum number of the hierarchical layers to ten in HE_{LIME}.

3.1 Datasets and Models.

We adopt two text-classification datasets: binary version of Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) and MNLI tasks of the GLUE benchmark (Wang et al., 2019). We use the dev set on SST-2 and a subset with 1,000 samples on MNLI (the first 500 dev-matched samples and the first 500 dev-mismatched samples) for evaluation. We build target models with BERT_{base} (Devlin et al., 2019) as encoder, achieving 91.7% (SST-2) and 83.9% (MNLI) accuracy.

3.2 Evaluation Metrics.

Following previous work, we use the area over the perturbation curve (AOPC) to perform quantitative evaluation. By modifying the top k% words, AOPC calculates the average change in the prediction probability on the predicted class as follows:

$$AOPC(K) = \frac{1}{N}\sum_{i=1}^{N}\left\{p(\hat{y}|x_i) - p(\hat{y}|\tilde{x}_i^{(k)})\right\},$$

where $p(\hat{y}|)$ is the probability on the predicted class, $\tilde{x}_i^{(k)}$ is modified sample, and N is the number of examples,. Higher AOPCs is better, which means that the words chosen by attribution scores are more important[2].

We evaluate with two modification strategies *del* and *pad*. *del* modifies the words by deleting them from the original text directly while *pad* modifies the words by replacing them with <pad> tokens. For hierarchical explanations, we gradually select words to be modified according to attribution scores. If the word number in a text group exceed the number of remaining words to be modified , this text group will be ignored. The detailed algorithm are described in the appendix.

3.3 Results Compared to Other Methods

As shown in Table 1, we compare our approach with a number of competitive baselines. Except for LIME, none of other baselines (hierarchical or not) shows a obvious improvement over LOO.

[2]Note that because there may be multiple words in a text group in hierarchical explanations, it is impossible to increase the number of perturbed words one at a time until reaching k%. Thus, we directly calculate the change in prediction after perturbing top k% words, which is the same as Chen et al. (2020).

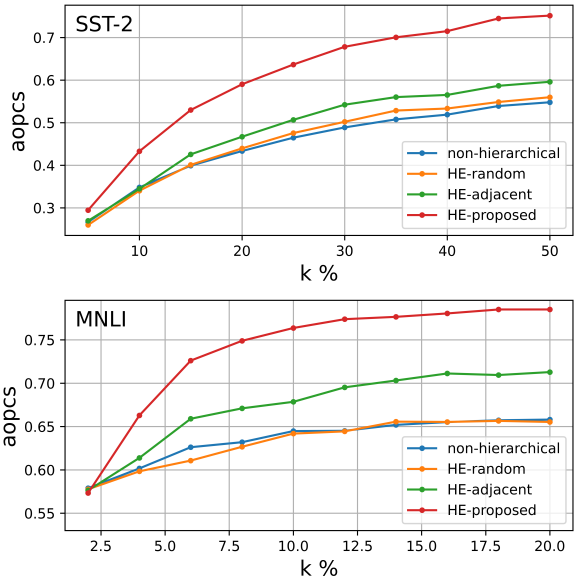


Figure 5: Results of Ablation Experiment.

In contrast, our LOO-based hierarchical explanations outperform LOO on average by more than 11%. Moreover, our LIME-based hierarchical explanations outperform LIME by 6% on average and achieves the best performance. The experimental results in Table 1 demonstrate the high quality of the generated explanations and the effectiveness of our method in converting non-hierarchical explanations to their corresponding versions.

3.4 Results of Ablation Experiment

We conduct an ablation experiment with two special baselines modified from HE_{LOO}: HE-random and HE-adjacent. HE-random merges text groups randomly in each layer; HE-adjacent merges adjacent text groups with the strongest interaction.

As shown in Figure 5, both adjacent and proposed baselines outperform non-hierarchical and random baselines, demonstrating our approach's effectiveness in building hierarchical explanations. Moreover, HE-proposed outperforms HE-adjacent consistently on two datasets, demonstrating the detrimental effects of the connecting rule on generating hierarchical explanations. Note that HE-random on SST-2 slightly outperforms non-hierarchical baseline but has almost no improvement on MNLI. We hypothesize that this is because the input text on SST-2 is relatively short, and thus randomly combined text groups have greater chances of containing meaningful compositional semantics.

4 Conclusion

In this work, we introduce an effective method for generating hierarchical explanations without the connecting rule, in which a novel strategy is used for detecting feature interactions. The proposed method can convert ubiquitous non-hierarchical explanations into their corresponding hierarchical versions. We build systems based on LOO and LIME. The experimental results demonstrate the effectiveness of proposed approach.

Limitation

Since there is currently no standard evaluation metric for evaluating post-hoc explanations, we use AOPC(k) as the quantitative evaluation metric, which is widely used in the research field. However, because different modification strategies might lead to different evaluation results, AOPC(k) is not strictly faithful for evaluation attribution explanations (Ju et al., 2022), Thus, we evaluate with two modification strategies del and pad and we didn't introduce new strategies to get attribution scores, which avoid the risk of unfair comparisons due to customized modification strategies mentioned in Ju et al. (2022). Even so, there is a risk of unfair comparisons because the AOPC(k) tends to give higher scores to erasure-based explanation methods such as LOO. We don't conduct human evaluation because we believe human evaluation needs a very large scale to guarantee objective and stable, of which we can afford the cost. Thus, we post visualizations of all explanations in our experiment to demonstrate the effectiveness of our approach (`https://github.com/juyiming/HE_examples`).

Acknowledgements

This work was supported by the National Key R&D Program of China (2022ZD0160503) and the National Natural Science Foundation of China (No.61976211, No.62276264). This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDA27020100). This research was also supported by Meituan.

References

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5578–5593.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*.

KR1442 Chowdhary. 2020. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Katsushige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. 2006. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55(1):72–99.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*.

Yiming Ju, Yuanzhe Zhang, Zhao Yang, Zhongtao Jiang, Kang Liu, and Jun Zhao. 2022. Logic traps in evaluating attribution scores. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5911–5922, Dublin, Ireland. Association for Computational Linguistics.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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*, pages 1135–1144.

Chandan Singh, W James Murdoch, and Bin Yu. 2018. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Michael Tsang, Youbang Sun, Dongxu Ren, and Yan Liu. 2018. Can i trust you more? model-agnostic hierarchical explanations. *arXiv preprint arXiv:1812.04801*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Appendix

A Experiment Details

Our implementations are based on the Huggingface's transformer model hub[3] and the official code repository of LIME[4] We use the default model architectures in transformer model hub for corresponding tasks. We use the special token <pad> to replace the erased text in LOO. For LIME, we use a kernel width of 25 and sample 2000 points per instance, which is the same as settings of the original paper of LIME. For each dataset, we use one well-trained model for experiments. For methods that require sampling, such as LIME and HEDGE, we conduct experiments three times with different random seeds and report the average results.

Different sampling result will lead to instability in LIME attribution scores. Thus, in HE_{LIME}, when calculating the attribution scores with text group g be marginalized, we will not conduct new sampling, but select samples that does not contain g among the existing sample points. Although this strategy will reduce the sampling points participating in the linear approximation by about half, it ensures the stability of the attribution scores when calculating interaction scores for HE_{LIME}, which is important for

B Experimental Computation Complexity

LOO. For LOO, calculate an interaction score between to text groups is comparable to three forward pass through the network. For the step 1, we need to calculate the interaction score between each two groups. In other step, we need to calculate the interaction scores between the new generated group and other groups. In total, we need to calculate $C_n^2 + (n-2)+, ..., 1 = O(n^2)$ times, where n refers to the sequence length of the input text. Note that through record the model prediction during every iteration, the computational complexity can be reduced by about half.

LIME. As described in Section A, we will not conduct new sampling for calculating attribution scores after feature marginalization. To quantifying feature interactions in each layer, we need to perform n linear approximations with n input features, where n refers to the sequence length of the input text.

C Evaluation

For hierarchical explanations, we gradually select words to be modified according to attribution scores. As shown in Algorithm 2, we first determine the number of words that need to be modified, denoted as k. The target set S is the word set to be modified and is initialized as an empty set. Text groups in hierarchical explanations G is sorted according to their attribution scores $score$ from high to low. Then, text groups in G is added to S in order until the number of words in S equals k. If the number of words in a text group is larger than the number of needed words (k subtracts the num-

[3] https://github.com/huggingface/transformers

[4] https://github.com/marcotcr/lime

Algorithm 2 Evaluation Algorithm For Hierarchical Explanations

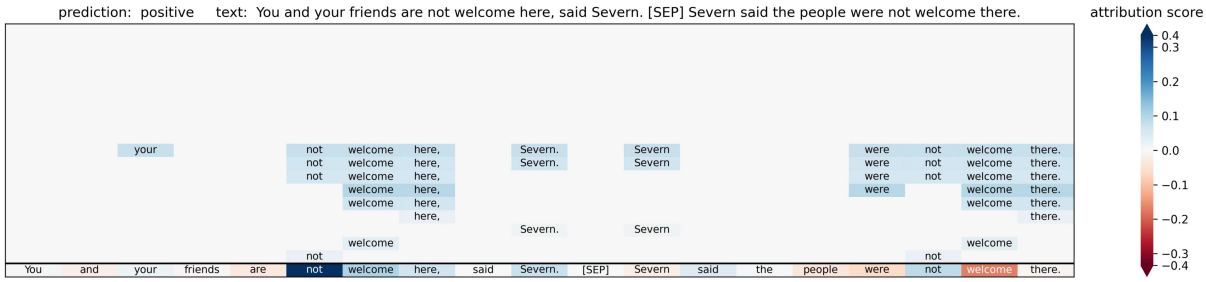
Input: the modified word number k, text groups G, attribution scores $score$
Initialize $S = \{\}$
$Sort(G)$ according to $score$
for each text group $g \in G$ **do**
  if $size(g) <= k - size(S)$ **then**
    $S = S \bigcup g$
  end if
end for
Output: S

ber of words in S), we abandon this text group to guarantee that the number of words in S does not exceed k. For HE$_{LIME}$, since the attribution scores at different levels come from multiple linear fitting results, the attribution scores at different levels can not be compare to each other. We evaluate the aopc score of each layer separately and take the best result for HE$_{LIME}$. For fair comparison, the best evaluation result of ten times experiments are selected for non-hierarchical LIME.

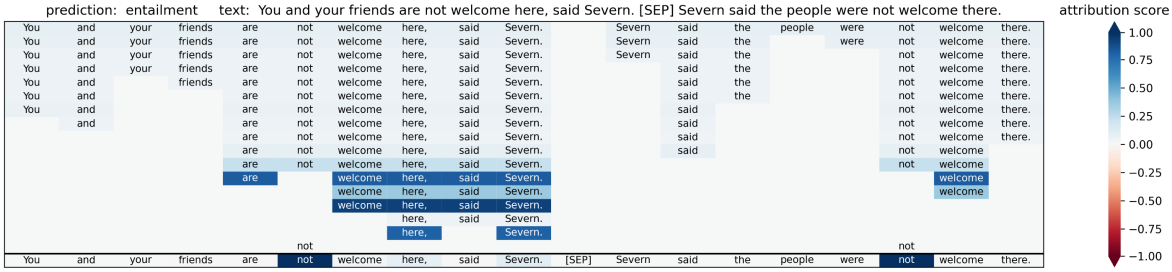
D Visualization

We provide visualizations of all evaluated examples (3,742 samples) at an anonymous website: https://github.com/juyiming/HE_examples.

Note that the maximum number of the hierarchical layers in HE$_{LIME}$ is limited to ten. Moreover, for the convenience of reading, we also select some short-length examples and put them in the appendix, where positive attribution score indicates supporting the model prediction while negative attribution score indicates opposing model prediction. The visualization of hierarchical attributions show that the proposed approach can not only get obvious improvement on quantitative evaluation but also are easy to read for humans.

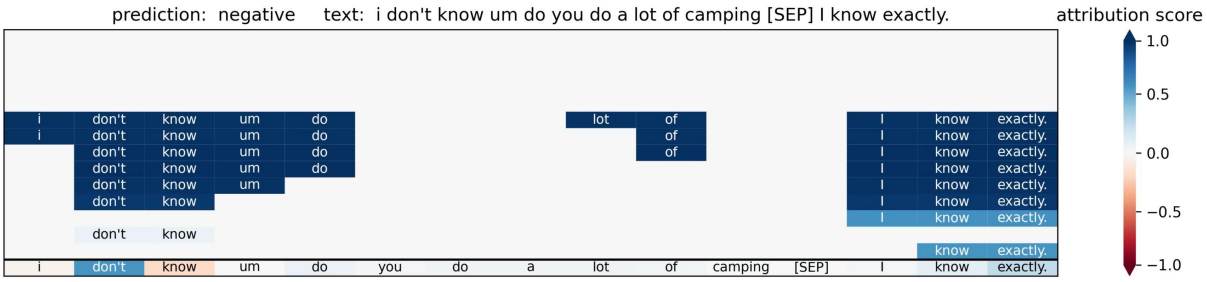


(a) HE$_{LIME}$ example.

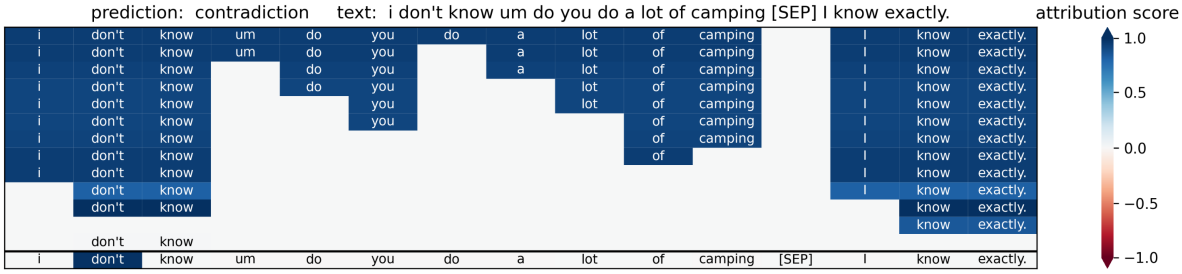


(b) HE$_{LOO}$ example.

Figure 6: An example of visualization

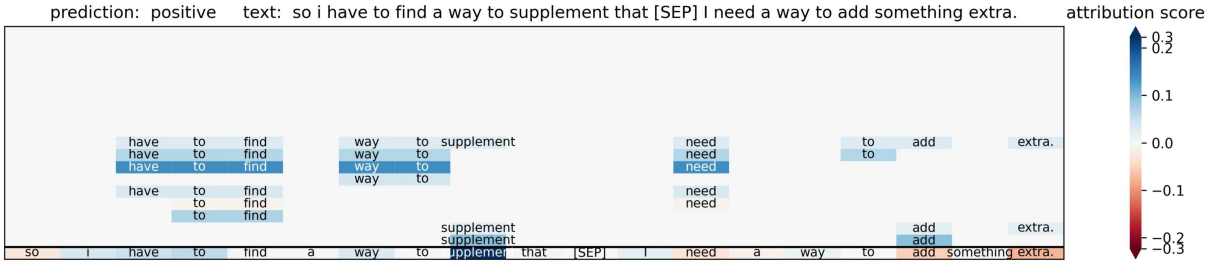


(a) HE_{LIME} example.

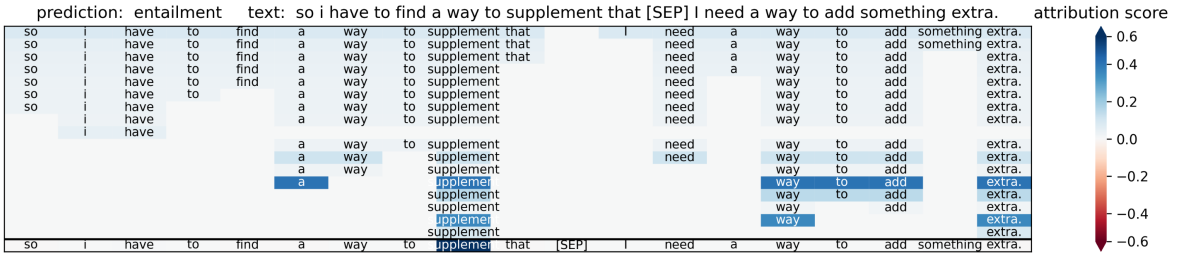


(b) HE_{LOO} example.

Figure 7: An example of visualization

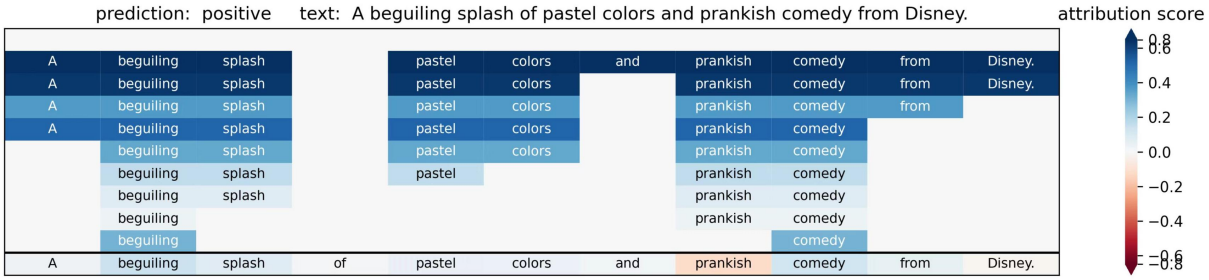


(a) HE_{LIME} example.

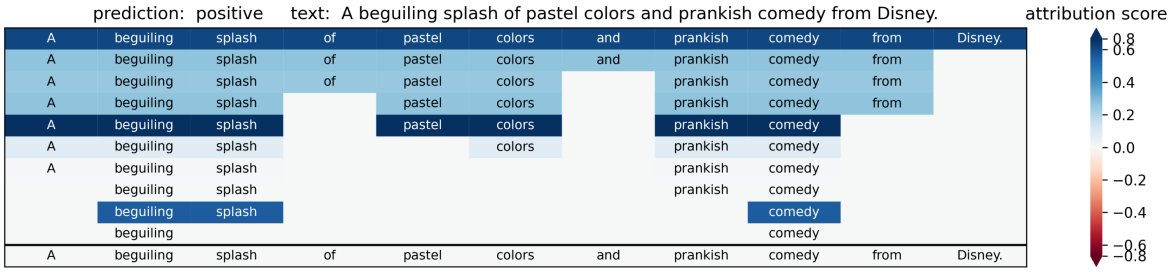


(b) HE_{LOO} example.

Figure 8: An example of visualization

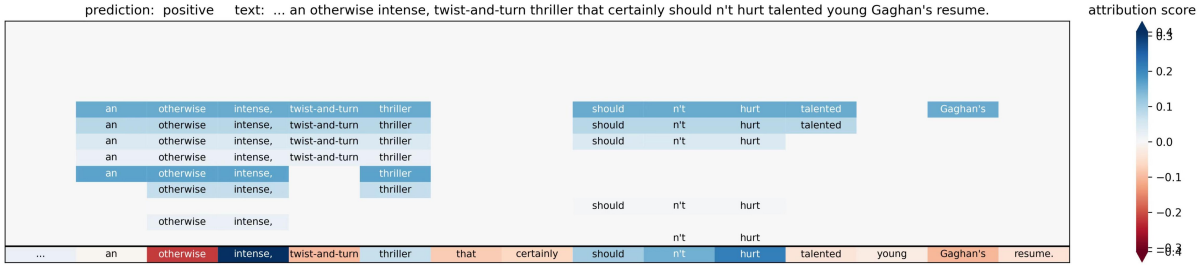


(a) HE_{LIME} example.

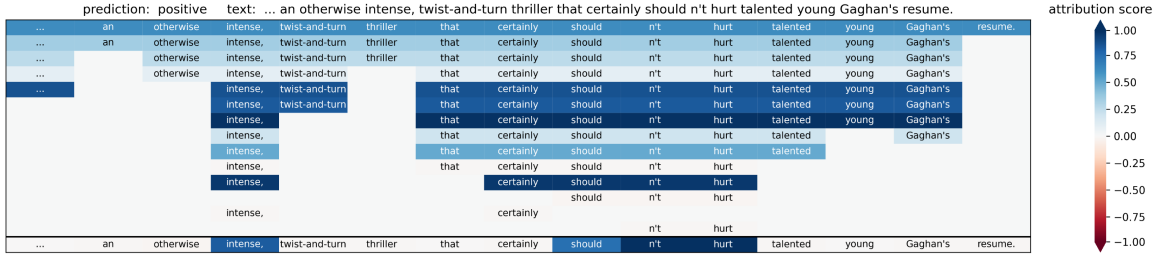


(b) HE_{LOO} example.

Figure 9: An example of visualization

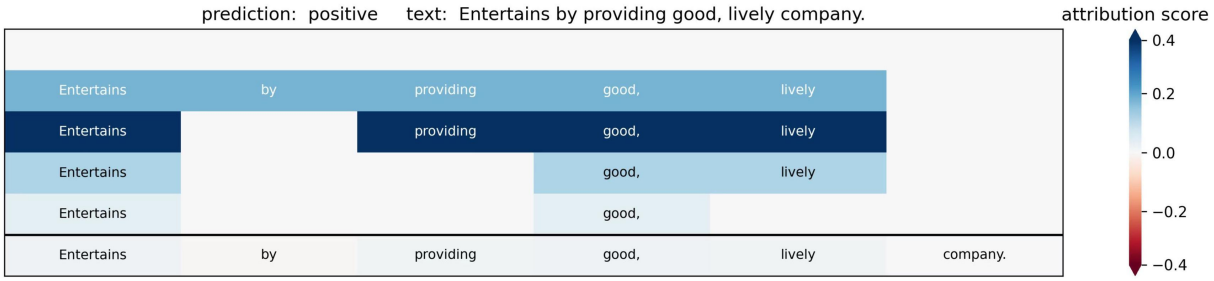


(a) HE_{LIME} example.

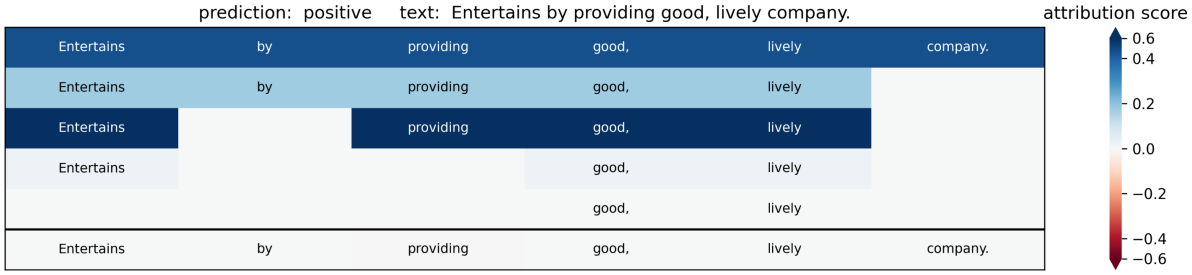


(b) HE_{LOO} example.

Figure 10: An example of visualization



(a) HE_{LIME} example.



(b) HE_{LOO} example.

Figure 11: An example of visualization

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section: Limitations

☒ A2. Did you discuss any potential risks of your work?
This article introduces a method for building post-hoc explanations for deep NLP models, using publicly available datasets and models. We believe that there is no potential risk in this method.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section: Abstract, Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3

☑ B1. Did you cite the creators of artifacts you used?
Section 3, Section: Experiment Details

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The artifacts used are well-known and publicly available, such as bert-base.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The artifacts used are well-known and the consistency between our work and their intended use is obvious.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The used datasets SST-2 and MNLI are well-known and have been widely used for many years. Using them will not bring the mentioned risks.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The artifacts used are well-known and publicly available, such as bert-base.

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
The artifacts used are well-known and publicly available, such as bert-base.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section: Experiment

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section: Experiment Details, Experimental Computation Complexity

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section: Experiment Details

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section: Experiment Details,

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.