Trigger-Argument based Explanation for Event Detection

Yong Guan¹, Jiaoyan Chen², Freddy Lecue³, Jeff Z. Pan⁴; Juanzi Li¹; Ru Li⁵

¹Department of Computer Science and Technology, Tsinghua University, Beijing, China
²Department of Computer Science, The University of Manchester, UK
³INRIA, France
⁴School of Informatics, The University of Edinburgh, UK
⁵School of Computer and Information Technology, Shanxi University, Taiyuan, China

gy2022@mail.tsinghua.edu.cn, j.z.pan@ed.ac.uk, lijuanzi@tsinghua.edu.cn

Abstract

A critical task for constructing event knowledge graphs is event detection (ED), which aims to identify events of certain types in plain text. Neural models have achieved great success on ED, thus coming with a desire for higher interpretability. Existing works mainly exploit words or phrases of the input text to explain models’ inner mechanisms. However, for ED, the event structure, comprising of an event trigger and a set of arguments, provides more enlightening clues to explain model behaviors. To this end, we propose a Trigger-Argument based Explanation method (TAE), which can utilize event structure knowledge to uncover a faithful interpretation for the existing ED models at neuron level. Specifically, we design group, sparsity, support mechanisms to construct the event structure from structuralization, compactness, and faithfulness perspectives. We evaluate our model on the large-scale MAVEN and the widely-used ACE 2005 datasets, and observe that TAE is able to reveal the process by which the model predicts. Experimental results also demonstrate that TAE can not only improve the interpretability on standard evaluation metrics, but also effectively facilitate the human understanding.

1 Introduction

Event Detection (ED) aims at identifying event triggers with specific event types, which is the first and fundamental step for extracting semantic and structural knowledge from plain text (Ahn, 2006; Nguyen and Grishman, 2015). For instance, event mention “The train driver was beaten over the head by a thug.” in Figure 1 comprises an event trigger “beaten” and a set of arguments such as “the train driver”, “the head” and “a thug”. An ideal ED system is expected to detect “beaten” as an event trigger of the type Bodily_harm. Recently, with the growth of open source annotated datasets (Walker et al., 2006; Wang et al., 2020) and the development of deep learning technologies, deep approaches have become popular for tackling the ED problem (Nguyen et al., 2016; Wang et al., 2021). Despite their great performance, they are still opaque for people to comprehend the inner mechanisms.

Although there exist many works that focus on explaining the model behavior on natural language processing (NLP) problems, such as text classification (Lei et al., 2016), text matching (Jiang et al., 2021) and machine reading comprehension (Ju et al., 2021), very little progress has been made to interpret ED models. We identify two major limitations that prevent the existing explanation methods from being applied to ED models.

Neglecting event structured knowledge. Existing methods mainly focus on assessing the contributions of individual input unit (e.g., word or phrase) to generate explanations for neural networks (Li...
et al., 2016; Jiang et al., 2021). As shown in Figure 1(a) and (b), both explanations provide insights of which words (e.g., “beaten”) or phrases (e.g., “beaten over the head”) contribute to the prediction. However, neither of them is suitable to explain ED models as an event is represented as a structure comprising an event trigger and a set of arguments. Thus, the trigger-argument structures are more sensible clues to explain ED systems. In Figure 1(c), “beaten” is an ambiguous word that may evoke completely dissimilar events such as Bodily_harm and Competition. In this case, trigger word “beaten” and its arguments (e.g., “the head” and “a thug” which refer to Body_part and Agent) work together for the prediction Bodily_harm. Thus, how to take advantage of the event structure knowledge for ED model explanation is a non-trivial task.

Explanations cannot reflect the decision-making process. Models usually provide important features which are words or phrases selected from an input text as explanations, but they do not further elaborate the function of these features, i.e., why models produce the prediction according to these features. It poses challenges to interpret an explanation and connect it to model prediction. For example, in Figure 1(a) and (b), models may assign high relevance score to “train driver” or “thug”, but it is still confused why these features can lead to the prediction Bodily_harm. In fact, “train driver” and “thug” serve as Victim and Agent, which compose together for the Bodily_harm event in which “An Agent injures a Victim” in Figure 1(c). Furthermore, Figure 1(d) provides an example that wrongly classifies Competition as Bodily_harm, because models take “Harley” and “John” as Agent and Victim rather than Participant_1 and Participant_2. Thus, exploring explanations that can not only identify important features but also reveal how these features contribute to the prediction are urgently needed.

To address the aforementioned challenges, we propose TAE (Trigger-Argument based Explanation), to construct structure-level explanations for Event Detection (ED) systems. To the best of our knowledge, this is the first exploration to explain ED with structure knowledge.

- We propose a model-agnostic method, called TAE (Trigger-Argument based Explanation), to construct structure-level explanations for Event Detection (ED) systems. To the best of our knowledge, this is the first exploration to explain ED with structure knowledge.
- TAE adopts three strategies, namely, Group Modeling, Sparsity Modeling and Support Modeling to characterize the trigger-argument based explanations from structuralization, compactness, and faithfulness perspectives.
- We utilize FrameNet (Baker et al., 2006), a well-defined knowledge base, to help complete the event structure in MAVEN. The annotated data is released for further research.
- Experimental results on the large-scale MAVEN and widely-used ACE 2005 datasets show that TAE can generate more faithful and human-understandable explanations.

2 Related Work

In this section, we review the related works on Event Detection and Interpretation Methods.

Event Detection. Event detection is a key task for Event Knowledge Graph (Pan et al., 2017) construction. Traditional methods for ED have employed feature based techniques (Ji and Grishman, 2008; Li et al., 2013). These approaches mainly rely on elaborately designed features and NLP tools. Later, advanced deep learning methods have been applied for ED, such as convolutional neural networks (Chen et al., 2015), bidirectional...
recurrent neural networks (Nguyen et al., 2016), which can take advantage of neural networks to learn features automatically. Since pre-trained language models (PLMs) are capable of capturing the meaning of words dynamically by considering their context, they have proven successful on a range of NLP tasks including ED (Tong et al., 2020; Wang et al., 2021, 2020). Although neural networks and PLMs bring incredible performance gains on ED task, they offer little transparency concerning the inner workings.

Interpretation Methods. There has been growing interests in producing explanations for deep learning systems in recent years, enabling humans to understand the intrinsic mechanism. In general, the explanations from these methods can typically be categorized as post-hoc explanations that aim to explain a trained model and reveal how model arrives at prediction (Lipton, 2016). Among them, gradient-based, attention-based and erasure-based methods are three typical methods.

Gradient-based methods are model-aware interpretation methods using gradients to measure feature importance (Shrikumar et al., 2017). Since a token index is not ordinal, methods simply sum up the relevance scores of each representation dimension. Because the score can have a negative or positive sign, the score may become zero even if it does contribute to prediction (Arras et al., 2017).

Attention-based methods attempt to use attention weights as feature importance scores (Vashishth et al., 2019). However, attention is argued to not be an optimal method to identify the attribution for an output as its validity is still controversial (Bastings and Filippova, 2020).

Erasure-based methods are widely-used approaches where a subset of features is considered irrelevant if it can be removed without affecting the model prediction (Feng et al., 2018). A straightforward approach is to erase each token by replacing it with a predefined value such as zero (Li et al., 2016). However, these erasure methods usually generate explanations by calculating the contribution of individual unit to the predictions, which are not suitable for ED as an event is often correctly identified with event structure.

In this paper, we attempt to generate explanations for ED models by considering semantic structured knowledge (Chen et al., 2018) entailed in the input at neuron level, which is complementary to the aforementioned approaches.

3 Preliminaries

3.1 Event Detection
An event refers to “a specific occurrence involving one or more participants” in automatic content extractions. To facilitate the understanding of the ED task, we introduce related terminologies as follows:

Event Trigger: the main word which most clearly expresses an event that happens.

Event Arguments: the entities that are involved in an event that serves as a participant.

Event Mention: a phrase or sentence within which an event is described.

For event mention “The train driver was beaten over the head by a thug”, an event extractor is expected to identify an Bodily_harm event triggered by “beaten” and extract corresponding arguments with different roles such as “the train driver” (Victim) and “a thug” (Agent). In this paper, instead of explaining the overall standard event extraction models, we concentrate only on the ED task. That is, for this example, our goal is to explain why ED models can classify the event as Bodily_harm or not.

3.2 Problem Formulation
ED explanation aims to explain a trained ED model and reveal how the model arrives at the prediction. For an event mention $x = \{x_1, x_2, ..., x_i, ..., x_n\}$ with $n$ words, a given pre-trained neural network (NN) $f$ maps $x$ to the corresponding label $y_j$, where $y_j \in \{y_1, y_2, ..., y_j, ..., y_N\}$ is corresponding event type which has unique trigger-arguments $E_x \in F$.

Assume that the NN model $f = g(h(x))$ can be decomposed into two-stages: (1) utilizes $h(\cdot)$ to map the input $x$ to the intermediate layer $h(x) = \{h_1(x), h_2(x), ..., h_k(x), ..., h_N(x)\}$ with $N$ neurons, such that $h(\cdot) \in \mathbb{R}^{(N-d)}$, and $h_k(x)$ is the $k$-th neuron in $h(x)$; and (2) uses $g(\cdot)$ to map the intermediate layer $h(x)$ to the output $g(h(x))$, which is the probability of input $x$ being predicted as label $y_j$ by NN model, as shown in top part of Figure 2.

To better understand neurons, recent work attempts to identify the closest features to explain its behavior. The correlations between neuron and feature are obtained as follows:

$$Neu(h_k(x)) = \arg \max \rho(h_k(x), F)$$

(1)

where $F$ are features of the input $x$, such as key words, POS, and trigger-arguments. $Neu(h_k(x))$ is the most related feature selected in $x$ to represent $h_k(x)$. $\rho$ is an arbitrary correlation calculation
function, and we use IoU (intersection over union) in this paper. Following the existing work (Ghorbani et al., 2019; Mu and Andreas, 2020), we select the closest layer to the classifier that already learns more abstract information for prediction, to detect the neuron behavior.

4 Method

In this paper, we propose TAE, a trigger-argument based explanation method for event detection, which attempts to utilize event structure knowledge to explain model predictions at neuron level. The overview of our method is shown in Figure 2, which contains three modules: (1) The Group module captures structured knowledge of events; (2) The Sparsity module encourages models to select few but key features in the event structure; (3) The Support module is a fundamental module that guarantees explanations generated by Group and Sparsity consistent with the original prediction.

The loss function of an structured explanation for an event is obtained by an optimization problem:

$$
L = \arg \min \lambda_g L_g + \lambda_s L_s + \lambda_{sd} L_{sd}
$$

(2)

where \((L_g, L_s, L_{sd})\) are from the group, sparsity and support modules, while \(\lambda_g, \lambda_s\) and \(\lambda_{sd}\) are hyper-parameters controlling weights of different losses.

4.1 Group Modeling

The Group module aims to divide neurons into different groups, and each group corresponds to a trigger-argument structure. Some existing works try to aggregate related features according to the knowledge, such as encouraging the highly overlapping regions of the image (Varshneya et al., 2021), or gathering the neighbor words to enhance the current word (De Cao et al., 2020; Jiang et al., 2021). However, these methods might not work, as arguments of event types can be scattered in different positions and usually not adjacent to each other in input texts.

To solve this problem, we propose a group loss objective that constructs event structures by aggregating neurons corresponding to the related arguments. We first use the clustering algorithm k-means (Hartigan and Wong, 1979; Ghorbani et al., 2019) to automatically cluster neurons with the nearest mean into the same group.

$$
G = \text{K-means} \left( \{h_k(x)\} \right)
$$

(3)

where \(G \in \{G_1, G_2, ..., G_L\}\) is the group set and \(L\) is group number.

Then, for individual group \(G_i\), we use the IoU to measure the contribution \(\phi(h_i^l(x))\) of neuron \(h_i^l(x)\) in the group.

$$
\phi(h_i^l(x)) = \frac{2||h_i^l(x) - F_x||_1}{||h_i^l(x)||_1 + ||F_x||_1 + ||F_x - h_i^l(x)||_1}
$$

(4)

where \(F_x\) is the trigger-argument feature of input \(x\), and \(h_i^l(\cdot)\) is the \(i\)-th neuron in group \(G_i\).

Finally, the group objective \(L_g\) is to minimize the intra-cluster sum of the distances from each neuron to the labeled feature in the input (Varshneya et al., 2021), given by the following equation:

$$
L_g = \sum_i L_{gi} = \sum_i \phi(h_i^l(x))
$$

(5)

During train phase, for each batch, we extract the trigger-arguments on the whole batch while calculating the \(L_g\). It means that the neuron can learn the batch data features rather than individual features, which can enhance the generalization ability.

4.2 Sparsity Modeling

The Sparsity module aims to produce compact and human-friendly explanations. This is achieved by removing “dead neurons” (Mu and Andreas, 2020), which are useless for model prediction, while only keeping the key information to explain predictions.

To this end, following the existing work (De Cao et al., 2020), we use the differentiable masking mechanism to filter out the useless neuron features. Specially, for each extracted neuron, a classifier...
with a sigmoid activation function is added to determine whether the neuron should be masked or not. During training phase, we directly use $\ell_1$ norm (Jiang et al., 2021) to minimize the number of the neurons as follows:

$$L_s = \min_k \sum \varphi(h_k(x))$$ (6)

where $\varphi(\cdot)$ is the neuron classifier. The straightforward idea is to minimize the non-zero position.

### 4.3 Support Modeling

The support module aims to ensure the faithfulness of explanations generated by Group and Sparsity. A desirable interpretable event detection model should satisfy that a prediction is directly dependent on the selected features. For an ED model, we choose the neurons in $h(x)$ to generate explanations. Group and Sparsity are utilized to select neuron features $\mu$ containing structured and important information. Thus the goal of Support is to measure whether $\mu$ can depict the true profile of how the model works. Specifically, function $h'(\cdot)$ maps $\mu$ to the new hidden states $h'(\mu)$, and $g(\cdot)$ maps the new hidden states $h'(\mu)$ to the new output $g(h'(\mu))$, as shown in the bottom part of Figure 2.

We introduce an optimization objective to guarantee the support modeling. Different from the existing work matching the current prediction, we directly ask the reconstructed representation to meet the ground truth distribution:

$$L_{sd} = \mathcal{P}(\hat{y}|g(h'(\mu), \theta))$$ (7)

subject to $KL(g(h(x)), g(h'(\mu)))$ (8)

where $\hat{y}$ and $\theta$ are the ground truth labels and trainable parameters respectively. $KL$ represents Kullback–Leibler divergence.

Note $h'(\cdot)$ can be any current popular network architectures, such as LSTM, Transformer and PLMs. In our setting, to maintain the interpretability, we use the simple linear projection and MLP (multilayer perceptron) to build the network, and the computation is much more efficient since we don’t need to optimize the whole backbone (Yeh et al., 2020). In addition, in this way, it mainly focuses on learning the neuron behavior instead of sacrificing the performance of the pre-trained CNN models.

### 5 Experiment

#### 5.1 Datasets

We evaluate our models on MAVEN (Wang et al., 2020) and ACE 2005 dataset (Walker et al., 2006). MAVEN is a manually annotated dataset for event detection task without annotated arguments, which contains 168 event types and 4,480 documents. The event types are manually derived from the frames defined in FrameNet (Baker et al., 1998; Guan et al., 2021b,a). To satisfy our needs, we utilize the automatic frame parser SEMAFOR (Das et al., 2014) to parse the MAVEN data. We select the data that have event type in MAVEN, and regard the corresponding frame elements (Guo et al., 2020) as the event arguments. Finally, we collect 12,649 event mentions, and randomly split them into train/dev/test sets with sizes of 8,469/2,000/2,000.

ACE 2005 is also a manually annotated dataset, which contains 8 event types, 35 argument roles and 599 English documents (Li et al., 2020). We further remove the data without arguments, and finally select 3,014 examples. Since the data size is relatively small, and cannot use it to learn a better NN model. So we directly utilize the models trained on MAVEN to test on ACE 2005, which can also verify the models’ generalization ability.

#### 5.2 ED Models

Our TAE is a model-agnostic method to explain ED models. In this paper, we first select two typical NN models, namely LSTM (Hochreiter and Schmidhuber, 1997) which contains 2 layers with 300 hidden states, LSTM+CNN (Tan et al., 2015) which has 2 layers with 300 hidden states. Moreover, we also select three PLM-based models which

<table>
<thead>
<tr>
<th>Methods</th>
<th>MAVEN</th>
<th>ACE 2005</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P$</td>
<td>$R$</td>
</tr>
<tr>
<td>LSTM</td>
<td>51.3</td>
<td>52.4</td>
</tr>
<tr>
<td>LSTM+CNN</td>
<td>53.5</td>
<td>54.2</td>
</tr>
<tr>
<td>BERT</td>
<td>52.6</td>
<td>63.5</td>
</tr>
<tr>
<td>DeBERTa</td>
<td>53.1</td>
<td>65.2</td>
</tr>
<tr>
<td></td>
<td>58.7</td>
<td>65.6</td>
</tr>
</tbody>
</table>

Table 1: Model performance on MAVEN and ACE. P and R refer to Precision and Recall respectively.
achieve promising performance on ED, including BERT (Devlin et al., 2019) which has 12 layers and 768 hidden states, DMBERT (Wang et al., 2019) which also applied on BERT-base version with 768 hidden states, and DeBERTa (He et al., 2021) which has 24 layers and 1,536 hidden states. Table 1 shows the results (P, R, F1) of different models on both datasets in our experiments, where DeBERTa outperforms the other four models with higher F1 scores.

5.3 Support Evaluation

We adopt three metrics to evaluate support degree (i.e., faithfulness): two metrics from prior explanation work including area over reservation curve (AORC) (DeYoung et al., 2020) and area over the perturbation curve (AOPC) (Nguyen, 2018), and a new defined evaluation metric called support-score (SUPP).

AORC calculates the distance between the original predicted logits and the masked ones by reserving top k% neuron features which are identified by trigger-arguments as follows:

$$AORC = \frac{1}{N} \left[ \sum_{k=0}^{K} \left| \left| \mathcal{P}(\hat{y}|x) - \mathcal{P}(\hat{y}|x)'' \right| \right|_2 \right]$$  \hspace{1cm} (9)

where $\mathcal{P}(\hat{y}|x)''$ means the prediction which reserves the top k% neuron features. Under this metric, lower AORC scores are better.

AOPC score calculates the average change in prediction probability on the predicted class over all test data by deleting the top r% neuron features.

$$AOPC = \frac{1}{N} \sum_{i=1}^{N} \left\{ \mathcal{P}(\hat{y}|x) - \mathcal{P}(r)''(\hat{y}|x) \right\}$$  \hspace{1cm} (10)

where $\mathcal{P}(r)''(\hat{y}|x)$ is the prediction which remove the top r% neuron features. $N$ denotes the number of examples. In our experiment, r is set to 20. Under this metric, the larger scores are better.

We propose SUPP score to verify whether the new prediction $g(h'(\mu))$ is positive to the original ones $g(h(x))$. Under this metric, the larger SUPP scores are better.

$$SUPP = \frac{1}{N} \sum_{i=1}^{N} \left\{ g(h'(\mu)) - g(h(x)) \right\}$$  \hspace{1cm} (11)

We compare TAE with four competitive baselines, namely LEAVE-ONE-OUT (Li et al., 2016), LIME (Ribeiro et al., 2016), BACKSELECT (Carter et al., 2019) and DIFFMASK (De Cao et al., 2020), utilizing AORC, AOPC and SUPP metrics.

Automatic support evaluation results are shown in Table 2, and we have the following three observations:

(1) TAE achieves better performance in most cases across all the three metrics on both datasets. For metric SUPP, all methods achieve positive results, indicating our method can identify important features and make a positive contribution to model predictions.

(2) Compare to LSTM- and CNN-based methods, BERT-based methods achieve significantly better performance. It is perhaps because BERT has
already preserved a large amount of general knowledge by training on large-scale data.

(3) Compared with MAVEN, the results on ACE are equally remarkable. Overall, our model achieves very strong results on different types of data and methods, proving that it is a good model-agnostic approach.

### 5.4 Sparsity Evaluation

For evaluating the sparsity, we directly report the sparsity score, which obtained in Equation 6 just like the explanation work (Jiang et al., 2021), as the metric. In this criterion, the score means the degree of sparsity, and the lower scores are better. The intuition behind this criterion is that a good explanation should be short for understanding.

The results are reported in Table 2. We can see TAE achieves the lowest SPAR values in most cases for the automatic evaluation, for example, the SPAR of our TAE + DeBERTa on ACE and MAVEN are 1.603 and 1.148, which indicates that TAE can effectively discover the useless neurons.

### 5.5 Group Evaluation

In order to verify the effectiveness of the group mechanism, we use two metrics for explanations. First, following the previous explanation work (Bau et al., 2017; Varshneya et al., 2021), for each predefined group, we compute the number of unique trigger-argument in the group as the interpretability score. Second, for trigger-argument structure, we average the IoU score which is computed in Equation 1 to represent its explanation quality score like (Mu and Andreas, 2020).

Figure 3 shows the comparison of the interpretability score with different groups. With the group number increasing, the number of trigger-arguments detected by the model gradually increases, indicating grouping mechanism can improve the model interpretability. However, the number of trigger-arguments remains constant after the group number exceeds 50. A major reason is the uneven distribution of the data, which mainly concentrates on 20% of the event types. Note, the maximum group number is limited to event type number, such as for MAVEN, the maximum group number is 168.

Table 3 shows the IoU score of 8 event types. In this setting, we separate test for each event type. The score increases with the group mechanism on most cases, which can further prove the effectiveness of the group mechanism.

### 5.6 Analysis on Trigger-Argument

To further verify the effectiveness of the trigger-arguments, we introduce features used in Mu and Andreas (2020) as comparison, such as POS (part-of-speech), most common words, and entity. They suggest that neuron cannot be regarded as a simple detector (Bau et al., 2017) but may express the meaning of multiple concepts. So they use composition operations such as OR, AND and NOT to expand the neuron behavior. We use the average IoU score of whole neurons on different formula lengths as one metric: \n\[ SL_{i} = \frac{1}{|h_{x}|} \sum_{j=0}^{\arg \max \text{IoU}(h_{j}(x), F)} \text{IoU(h}_{j}\text{(x), F}) \]  
where SL_{i} is the IoU score of the formula length i, and F is the feature set. h_{j}(\cdot) denotes the j-
The crash sparked a review of helicopter safety.

During the American-led coalition offensive in the Persian Gulf War, American, Canadian, British, and French aircraft and ground forces attacked retreating Iraqi military personnel attempting to leave Kuwait on the night of February 26–27, 1991, resulting in the destruction of hundreds of vehicles and the deaths of many of their occupants.

Figure 5: Examples of deleting important arguments.

Figure 6: An example of TAE explanation.
There might be a few different future directions. Firstly, we might look into the idea of using explanations to further improve ED classification, as well as ED explanations in downstream applications. Secondly, we plan to explore ED under the multi-modal setting. Thirdly, event relation extraction is still challenging and deserves some further investigation. From the practical aspect of event knowledge graphs (EKGs), it is worth investigating high-quality yet efficient methods for constructing EKGs and making use of EKGs to predict future events (Lecue and Pan, 2013; Deng et al., 2020). Furthermore, it might be an idea to integrate commonsense knowledge (Speer et al., 2016; Romero et al., 2019; Malaviya et al., 2020) into event knowledge graphs.

**Limitations**

In this section, we discuss the limitations of TAE. First, as our method depends on event structure information which is obtained through automatic parser, if the parser is not good enough, then it will impact the performance. Second, since we focus on leveraging structural information, we restrict the experiments on text-based event explanation. Future work will explore multi-modal event detection explanations and evaluate models on other NLP tasks.

**Acknowledgments**

This work is supported by the Institute for Guo Qiang, Tsinghua University (2019GQB0003), the NSFC Youth Project (62006136), and the Chang Jiang Scholars Program (J2019032).

**References**


of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4443–4458.


The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.
C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5

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 5

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 5

D X 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.?
Left blank.

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)?
Left blank.

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?
Left blank.

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.