XCrowd

Combining Explainability and Crowdsourcing to Diagnose Models in Relation Extraction

Smirnova, Alisa; Yang, Jie; Cudre-Mauroux, Philippe

DOI
10.1145/3627673.3679777

Publication date
2024

Document Version
Final published version

Published in
CIKM '24

XCrowd: Combining Explainability and Crowdsourcing to Diagnose Models in Relation Extraction

Alisa Smirnova
University of Fribourg
Fribourg, Switzerland
alisa.smirnova@unifr.ch

Jie Yang
Delft University of Technology
Delft, Netherlands
j.yang-3@tudelft.nl

Philippe Cudre-Mauroux
University of Fribourg
Fribourg, Switzerland
philippe.cudre-mauroux@unifr.ch

Abstract

Relation extraction methods are currently dominated by deep neural models, which capture complex statistical patterns while being brittle and vulnerable to perturbations in data and distribution. Explainability techniques offer a means for understanding such vulnerabilities, and thus represent an opportunity to mitigate future errors; yet, existing methods are limited to describing what the model 'knows', while totally failing at explaining what the model does *not* know. This paper presents a new method for diagnosing model predictions and detecting potential inaccuracies. Our approach involves breaking down the problem into two components: (i) determining the necessary knowledge the model should possess for accurate prediction, through human annotations, and (ii) assessing the actual knowledge possessed by the model, using explainable AI methods (XAI). We apply our method to several relation extraction tasks and conduct an empirical study leveraging human specifications of what a model should know and does not know. Results show that human workers are capable of accurately specifying the model should-knows, despite variations in the specification, that the alignment between what a model really knows and what it should know is indeed indicative of model accuracy, and that the unknowns identified through our methods allow to foresee future errors that may or may not have been observed otherwise.

CCS Concepts

• **Computing methodologies** → **Model verification and validation**; **Information extraction**; **Causal reasoning and diagnostics**.

Keywords

error analysis, relation extraction, model interpretation, human computation

ACM Reference Format:
Alisa Smirnova, Jie Yang, and Philippe Cudre-Mauroux. 2024. XCrowd: Combining Explainability and Crowdsourcing to Diagnose Models in Relation Extraction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679777

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679777

1 Introduction

Relation extraction is a key step in populating structured data and knowledge graphs, with application to a wide range of tasks such as question answering and information retrieval; it is also receiving renewed interest in machine learning for developing neuro-symbolic AI systems [12, 19, 36]. Current relation extraction techniques are dominated by deep learning architectures such as LSTM and BERT [18, 30], which have demonstrated remarkable performance; these models however suffer from an inherent issue in terms of robustness: they are brittle and vulnerable to perturbations in data and distributions [39].

Tackling the robustness issue requires an understanding of the reasons or decision mechanisms underlying model failures—it thus can be treated as a problem of *diagnosis*. Traditional diagnosis has been mainly studied under the white-box setting where the decision mechanisms are represented in human-understandable language or logic [6], a property not held by those intrinsically opaque deep learning architectures. The problem is therefore closely connected with explainable AI (XAI), on which a growing body of work can be found (e.g., for explanation-based debugging [22]). Yet, unlike explanations that aim to answer the 'why' question—why a certain decision is made by a model, diagnosis seeks the answer to the 'why not' question—why the correct decision is not made, also referred to as the contrastive-why question [4, 43]. Answering the 'why not' requires knowledge on what the model has *not* learned, in contrast to what the model has learned. Existing explanation-based debugging work addresses this problem in a heuristic manner, relying on the knowledge and spontaneous reaction of developers in diagnosis practice. Such an approach is not only limited but offers incomplete or even dubious insight into the weaknesses of deep learning models.

In this paper, we propose XCrowd, a hybrid human-XAI approach for diagnosing relation extraction models combining both crowdsourced knowledge elicitation and explainable AI. We decompose the diagnosis problem into two problems: eliciting what a model *should know* and what it *really knows*, henceforth referred to as *should-knows* and *really-knows*, respectively. To elicit the should-knows, we design a crowdsourcing task that collects from crowd workers a specification of should-knows, in the form of highlighted tokens that represent workers' rationales in relation classification. Through a comparison with really-knows generated by a configurable XAI component, our approach produces a characterization of model unknowns to explain model decisions. Our approach can not only explain problems in model decision mechanisms that lead to erroneous decision, but also allows to proactively expose such problems even when no errors are observed in model output (i.e.,

when the model is right but for the wrong reasons), thus allowing for proactive diagnosis and treatment of model weaknesses.

We apply XCrowd to several relation extraction tasks and conduct an empirical study over human specification of model should-knows and unknowns. Results show that crowd workers are capable of accurately specifying the model should-knows, despite variations in the specification, that the alignment between really-knows and should-knows is indeed indicative of model accuracy, and that the unknowns produced through XCrowd allow us to automatically predict model errors with high recall. Specifically, the crowdsourced annotations score 88% F1 compared to expert annotations and are highly aligned with true model reasoning with an F1 score of 78% for BERT-based model and 95% for LSTM-based model. On error prediction tasks, we are able to achieve 98% and 82% recall levels for BERT-based and LSTM-based models, respectively.

2 Related Work

Relation Extraction & Robustness. Relation extraction (RE) is a well-established NLP task, part of information extraction. Initially leveraging rule-based and statistical methods, RE methods are currently mostly based on deep neural networks ranging from CNNs, RNNs, LSTMs [29, 30, 46] to transformers-based architectures, such as BERT [44] and SpanBERT [18]. More recently, generative AI became significant for NLP tasks, including relation extraction. The authors of [5] reframe relation extraction as a seq2seq task and propose a seq2seq model based on BART [23]. A very recent survey [41] compares the performance of the state-of-the-art generative models for relation extraction. The authors compare state-of-the-art supervised RE models with FLAN-T5 and GPT-3 in fine-tuned and few-shot settings with a chain of thought. The authors note that providing explanations in the few shot setting leads to more standardized outputs simplifying evaluation but notably, does not improve the performance significantly.

Robustness refers to the insensitivity of a model's performance to miscalculations of its parameters [39]. Two main kinds of robustness have been identified, 1) adversarial robustness, considering the sensitivity to adversarial attacks and perturbations [37]; and 2) natural robustness, relating to the ability of a model to preserve performance under naturally induced data corruptions or alternations [11]. Various methods have been proposed to identify or improve a model's adversarial robustness: data augmentation through adversarial training [7, 13], adding noise [17], or employing robust neural layers such as the spiking architecture [35]. Natural robustness is generally associated to data outside the training set due to distribution shift. This type of robustness is more challenging as out-of-distribution data cannot be generated automatically. A closely related notion is model unknowns [9, 34], describing from a knowledge perspective missing or incorrect knowledge in a model. A recent study investigates language models in terms of known unknowns for question answering [2]. Little attention is devoted to relation extraction in this context, however, which we address in this work by leveraging human computation and explainability.

Explanation in Relation Extraction. Prior work has explored the relationship between explanation and relation extraction performance. Wang et al. [42] and Tang and Surdeanu [38] propose approaches that leverage explanations to improve relation classification. Similarly, the authors of [42] propose a framework to utilize natural language explanations in a low-resource setting (that is, the number of NL explanations is very low compared to the dataset size). They train a module to match a data instance to a logical form and give a similarity score that indicates how likely the instance matches a given form. The authors collected free-form explanations on Amazon Mechanical Turk towards that goal. In [38], researchers build an explainability classifier into the relation extraction model. They demonstrate that the explanations extracted during joint training are closer to grammars written by experts than the ones produced by automated post-hoc methods. The authors of [15] study under which circumstances explanations of individual data instances can improve modeling performance (that is, which explanations at prediction time will lead to better results). They formulate the properties of the datasets when explanations can be useful. Specifically, they report that explanation retrieval does not improve model performance for relation extraction on TACRED, which is also empirically confirmed in [38]: the ablation study demonstrates that explainability does not influence model accuracy. The authors of [14] evaluated several explainability methods on textual and tabular data. In their experiments, they asked humans to predict model outputs on the test set based on model outputs on the validation set. They compared two settings: when humans are given only data instance vs. humans are given data instances along with explanations. The experiments showed limited impact of explanation on human predictions.

In our work, we not only focus on explaining correct predictions, but we also aim at analyzing the model behavior and predicting the incorrect predictions and reasoning of the model on unseen data. We believe that the scenario we consider is more useful and realistic: in practice, one might already have an existing model with known performance and its architecture cannot be changed. Thus, its performance might be improved through model interpretability, as knowing when things go wrong helps improve model performance in a post-hoc manner (e.g., involving human-in-the-loop to fix model errors).

Human Computation. Human computation and crowdsourcing aim to bring together human and artificial intelligence to solve computational problems that are beyond the scope of AI [21, 40]. The computational roles of humans in NLP have been mainly considered in data labeling using crowdsourcing in various NLP tasks, e.g., sentiment and opinion mining [28] and question answering [16]. Recent work has looked into human involvement for cleaning label noises (e.g., in relation extraction [45]), and on the model side, for explaining model behavior with human concepts or evaluating model explanations [32]. Work can also be found on characterizing model unknowns, dating back to the seminal work of Attenberg et al. [3] who propose to ask humans to gather publicly accessible instances that are potentially difficult for the model to handle. The approach has been recently extended by enabling human access to more information related to unknowns for more efficient detection: the authors of [20] introduce a data partitioning technique that first organizes the test data into multiple partitions based on feature similarity, and then uses an explore-exploit strategy to search for unknown instances across these partitions; the authors of [24] propose to use human intelligence to detect unknowns to train an

expansion classifier to identify additional unknowns from existing data. In this work, we aim to bring to the fore the idea of not only detecting, but more importantly, *characterizing* unknowns to gain a deeper understanding of why the model fails, which will allow us to explain erroneous decisions and proactively foresee future errors. On the methodological level, we contribute to the state of the art by introducing a systematic model diagnosis approach that elicits human knowledge for specification of the model's should-knows.

3 Methodology

3.1 Characterizing Knowns and Unknowns

For diagnosing model errors, we create a collection of model reasoning, namely, **model knowns** and **model unknowns**. Figure 1 gives an overview of our approach. Given a model and a dataset, we aim to answer two questions: (i) what the model should know to make a correct prediction, and (ii) what the model really knows. In order to answer the first question, we employ crowd workers to classify what words in the sentence are actually useful for making a judgment on a specific data instance. In order to answer the second question, we apply explainability AI methods (XAI).

3.1.1 Formal definition. Given a data instance $\mathbf{x} = (x_1, x_2, \dots x_n)$ where n is a sentence length and x_i denotes a token, human annotations $\mathbf{ha} = (ha_1, ha_2, \dots ha_n)$ are defined as follows:

$$ha_i = \begin{cases} 1, & \text{if } x_i \text{ annotated as important,} \\ 0, & \text{otherwise.} \end{cases}$$

Automatic explanation of a data instance is represented as a vector $\mathbf{xai} = (xai_1, \dots, xai_n)$ where each element corresponds to the weight of word x_i. The bigger absolute value of xai_i corresponds to the higher word importance. Explainability methods produce explanations for several classes that have the highest probabilities. For collecting model knowns and model unknowns, we only take into account explanations made for the class predicted by the model.

3.1.2 Model knowns and unknowns. Given both vectors **ha** and **xai**, and a threshold t (see below), we define **model knowns** as a set of tokens $x_i : ha_i > 0 \wedge xai_i \geq t$, which basically correspond to true positives: those are the tokens that both the humans and XAI marked as important. We distinguish two types of **model unknowns**: the ones corresponding to the false negatives (the tokens annotated by humans but not by XAI) and the ones corresponding to false positives (the tokens deemed important by XAI but not by humans). They are formally defined as a set of tokens x_i where $x_i : 0 < xai_i < t \wedge ha_i > 0$ and a set of tokens x_i where $x_i : xai_i >= t \wedge ha_i = 0$, respectively. In this paper, we define the threshold t as follows:

$$t = \underset{t}{\operatorname{argmax}}\, F1(ha, xai > t), \tag{1}$$

where F1 denotes the F-measure. The intuition behind this design choice is that we want to maximize the alignment between human annotations and automatic explanations. Maximizing the F-measure means striking a balance between precision and recall, as this is precisely what we want: extracting the most important tokens from the automatic explanations and filtering out the tokens that are less important while keeping the recall high enough.

For each relation class C we collect model knowns and model unknowns from the instances in the validation set where the model predicted class C. We mark them as correct if the model prediction was correct, and as incorrect otherwise. Thus, we have a collection of model reasoning that we can use to diagnose the model from several perspectives: (i) when the model reasoning is incorrect leading to the wrong prediction, and (ii) when the model makes a correct prediction but its reasoning is considered wrong.

3.2 Proactively Predicting Errors

An important usage of model knowns and unknowns is predicting errors in the unseen data. We consider in this context two different methods for constructing the error predictor: decision Tree classifier, and semantic similarity matching.

3.2.1 Decision Tree Classifier. Error Predictor should be a simple and explainable model, such as a decision tree. Our design is as follows. We feed the predictor with all the necessary information that consists of three types of features:

- **Instance encoding** contains information about the classified data instance.
- **Explanation encoding** contains information about model knowns and model unknowns and allows to build connections between right and wrong reasoning. For unseen data instances, the predictor would not have access to the should-knows, thus, it is not straightforward to obtain model knowns and unknowns. To cope with this issue, we treat the should-knows as the fixed feature space and fill the feature vector with appearances of tokens in automatic explanations. That is, we create a vocabulary of should-knows from the collected human annotations of the validation set. Thus, to predict the errors on the unseen data, where should-knows for the specific data instance are unavailable, we fill the feature vector with appearances of tokens in automatic explanation, i.e., really-knows, in this pre-built vocabulary. Our rationale behind this design choice is that more appearances of such tokens in the should-knows vocabulary indicate a higher level of alignment between really-knows and should-knows, and thus can be indicative of model errors.
- **Prediction encoding** connects model reasoning with model predictions.

For instance encoding, we use a traditional set of features that includes words between entities, their part-of-speech tags when available, entity types, etc. Explanation is encoded is a bag of words that have weights $xai_i \geq t_{expl}$, where t_{expl} is a hyperparameter. Prediction is encoded as a binarized model prediction. The classifier is trained to perform binary classification: whether the model made an error or not.

3.2.2 Semantic Similarity Matching. The intuition behind similarity matching is straightforward. For each data instance **x** and the corresponding predicted class r we match the explanation of this instance $x_i : xai_i \geq t$ with all human annotations for the given class r. If we find a match, i.e., if the maximum similarity score is higher than a certain threshold t_{sim}, then the prediction is correct. If there is no match, we classify this prediction as erroneous. We use cosine

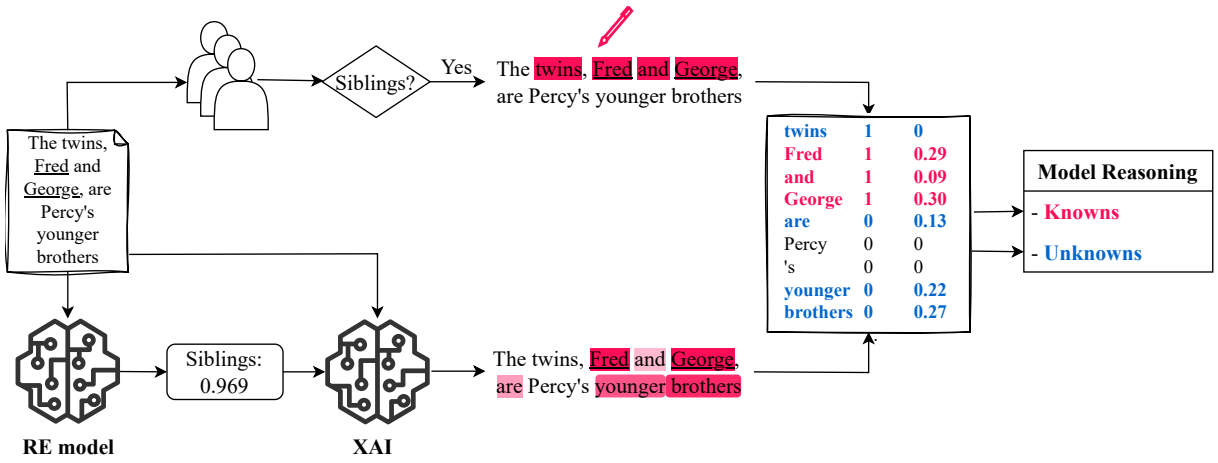


Figure 1: An overview of the proposed approach. Given a data instance and a model we seek to answer two questions: what the model should know about the data and what the model really knows. To answer the former question we design a crowdsourcing task where we ask the crowd to evaluate which words in the sentence are important for making a specific prediction (i.e., relation classification). For the latter question, we employ an explainable AI method (XAI) to assign word weights and highlight the text spans that are important for model predictions. Having both sets of explainability features, we are able to diagnose the model and identify the gap between "true" reasoning and actual model reasoning.

similarity between the corresponding embeddings as a similarity metric.

4 Collecting Human Annotations

In order to answer the question "what should the model know" we employ human annotators on Toloka, a popular crowdsourcing platform[1]. In this section, we explain our approach to design an annotation project in order to obtain truthful annotations.

To ensure the good quality of the annotations, we need to take into account the specific aspects of working with anonymous, online crowd workers, for instance considering:

- the diversity of the crowd in terms of spoken languages and education levels;
- the online nature of our task, where the means to explain the task to workers and motivate them are limited.

4.1 Annotation Task Design

We follow the approach proposed in [27]: instead of asking annotators directly about their rationales, we first ask them to "do" the task of judging the type of relation between two given entities in a sentence. By doing so, the workers are exercising their rationales instead of imagining what could potentially make their rationales annotation more reliable. If on the first step the answer is a positive relation class (i.e., not the *No relation*), we ask workers to identify the most relevant information and highlight the words that were important for their judgment. When the answer is *No relation*, a worker proceeds to the next task. Figure 2 shows an example of our task. In order to simplify our classification task, we split the full set of relation classes into subsets that are more likely to be confused by either a model or a human. Those subsets are mostly

[1]https://platform.toloka.ai

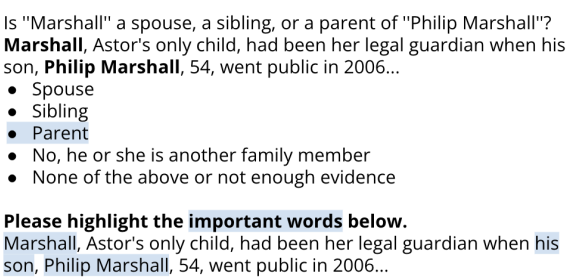


Figure 2: Task design for annotating important words for a subset of relations. The annotators are first asked to answer if the relation is present in the sentence. If the answer is positive, a text annotation field appears where the annotator has to markup the words that they think are important for classification.

defined on common entity types; for instance, there is a subset of relations that most of the time is observed between the entities of type *Organization*.

4.2 Ensuring the Quality of Annotations

4.2.1 Task Design. Crowdsourcing is an extremely powerful paradigm but also intricate in practice. Through crowdsourcing, we are interacting on digital platforms with anonymous human beings who have their own interpretations and routines. That is why properly describing the task at hand is a crucial step on the way to high quality results. The task description needs to convey the logic of the task in an exhaustive while concise form, and to provide illustrative examples. The instruction must have a clear structure and help the crowd workers navigate information easily, e.g., to look up entities

during the annotator's work. It is also essential to keep instructions relatively simple. Having in mind that the annotators might not read the instruction fully, we put the most essential rules first. Our instruction covers the following key points:

- a brief task definition along with definitions of the main concepts;
- the sequence of steps to perform as some workers might be unfamiliar with an annotation interface;
- examples of correct annotations to illustrate various points in our instructions.

Since a relation extraction dataset might consider many classes (in our experiments, we used a dataset with 42 classes), we split the task into several sub-tasks where each sub-task correspond to a group of similar relations (e.g., family relations as shown on Figure 2). We put an example extraction into our supplemental material.

4.2.2 Crowd Identification and Training. Since the task is complex, we want to ensure that the collected annotations are reliable. On the other hand, we want to avoid manual checks as the dataset to handle might be large (e.g., several thousands of instances).

It is important to identify the right crowd for the task at hand in order to achieve good results [10]. The crowdsourcing platform we used allows for a variety of filters as well as mechanisms for training and quality control. For our task, we applied the following filters:

Language. We only selected workers who know English and passed a language test on the platform.

Top Annotators. We used a built-in filter to select the top 10% of annotators based on their overall performance on the platform.

Skill. We only select workers who show that they understood the task (see details below).

Crowd workers that meet the first two requirements are shown the instructions to perform the task. To check that the workers understand our task, we implement *a training task* that is dedicated to teach the workers to perform the task through exercise. Specifically, in addition to input data (a piece of text and a character name), each microtask contains a golden annotation and a hint that is shown to the worker if they make a mistake in that microtask. After the worker finishes their training, *the skill* evaluated as the percentage of correct answers is assigned to the worker. Only the workers who finish the training with a skill level of 25% or more are allowed to complete the rest of the annotation tasks.

We ensure that annotators are fairly paid. We estimated how long it takes to annotate one task by executing the tasks ourselves. Then, we priced the task so that the annotators are paid 15$ per hour.

5 Experiments

In this section we present an empirical analysis on XCrowd for diagnosing relation extraction models. We focus on answering the following questions:

- **RQ1** How reliable are the answers of human annotators in specifying the model's should-knows?
- **RQ2** How much are automatic explanations aligned with human annotations when model predictions are correct or incorrect?
- **RQ3** How effective is our approach at interpreting the model's (erroneous) predictions and in proactively predicting model errors?

Datasets. We evaluate our pipeline on two datasets, **TACREV** [1], a revised version of TACRED [46], and **CoNLL04** [33], both derived from news articles. Each data instance is annotated with entity spans and a relation between these two entities. TACREV has 42 relation types while CoNLL04 has only 6 relation types, including no_relation.

Crowdsourcing setup. Our crowdsourcing task instructions are based on the original TAC KBP 2014 guidelines[2]. We used 94% of the positive instances in the validation set of TACREV for the crowdsourcing task, excluding several classes. Specifically, we excluded (i) `per:charges` because of potentially triggering content, (ii) `org:political_religious_affiliation` because it is very rare, and (iii) `org:website` because in most of the cases only entity names are important for classification. For CoNLL04 we annotated all instances in the validation set.

For TACREV, we divided relation classes into groups as specified in Section 4.2 while for CoNLL04 there was no need to do this because of the small number of relations. For each group, we designed a set of training tasks that consisted of the data instances from the training set and provided the correct answers, including annotations, produced by the first author. An annotator should complete at least 60% of training tasks correctly to get access to the generic tasks. For each set of generic tasks, we used honeypots (tasks with known answers) to measure the annotator's skills. Only the annotators with an accuracy of more than 60% were allowed to keep doing the tasks. Finally, each annotation task was given to 3 annotators.

Relation Extraction Models. We explore two deep learning models for sentence-level relation extraction and assess their explainability.

- Position-aware LSTM model (PA-LSTM) [46]: an LSTM-based model with a position-aware attention mechanism.
- SpanBERT model [18]: an extension of BERT [8] with better span representations, which is crucial for relation extraction.
- RoBERTa-based model [48]: an extension of RoBERTa model [25] for sentence-level relation extraction with an improved entity representation.

For all three models we use the open-source code released by the authors with the default hyperparameters settings leading to the best reported performance.

Though all models are deep learning models, there is a significant difference in their architectures and what kind of knowledge they use for inference. PA-LSTM uses GloVE embeddings [31] as the only source of external knowledge and the rest of the features (e.g., position features and lexical features when available) come from the dataset. In contrast, SpanBERT and RoBERTa are both extensions of pre-trained language models (PLM), that is, they use all the statistical knowledge possessed during the pre-training

[2]https://tac.nist.gov/2014/KBP/ColdStart/guidelines/TAC_KBP_2014_Slot_Descriptions_V1.4.pdf

	P	R	F1	α
		TACREV		
Classification	0.95	0.85	0.90	0.767
Annotation	0.86	0.88	0.85	0.786
		CoNLL04		
Classification	0.91	0.84	0.87	0.83
Annotation	0.94	0.92	0.92	0.832

Table 1: Evaluation of human annotations on the validation set. We calculate the metrics for the classification task automatically since the ground truth is available. We calculate agreement as Alpha Krippendorf. Agreement on annotations is calculated on a per-token basis.

Cuba 's human rights situation has become increasingly tense since the Feb. 23 **death** of **Orlando Zapata Tamayo** after a long **hunger strike** in jail.
Cuba 's human rights situation has become increasingly tense since the Feb. 23 **death of Orlando Zapata Tamayo after** a long **hunger strike** in jail.

Table 2: Example of how different annotators highlight the words for the same task.

phase. Though they do yield better performance than PA-LSTM with RoBERTA outperforming SpanBERT, interpretability of these models can be difficult.

Automatic explainability methods. We compare the following automatic explainability methods:

- All words between subject and object entities: a simple but efficient heuristic for explaining relation extraction.
- LIME [32]: a model-agnostic framework that calculates explanations of data instances by perturbating the tokens. In our setting, we replace up to 50% of the tokens with the unknown (`[UNK]`) token;
- SHAP [26]: a model-agnostic framework that calculates Shapley values where the score of the feature depends on its interactions with other subsets of features.

We chose these two explainability methods because of (i) their popularity in the NLP community, (ii) they are open-source, and (iii) they are easy to adapt to relation extraction task. Both methods are only applicable to the models that are able to produce class probabilities that sum up to 1, which is typically not the case for generative models.

We use the authors' implementations of the relation extraction models. For explainability methods, we use open-source implementations, adapting them to our relation extraction task and data formats. Notably, we mask the full entity name if at least one of its tokens is masked.

5.1 RQ1: Human Annotations Quality

We evaluate the quality of human annotations with respect to both classification and annotations tasks. The evaluation on the classification task gives us a generic understanding of annotation quality. For the classification task, the ground truth is available. We aggregate the annotations by majority vote and report results in Table 1. Evaluating the annotation task is trickier. We proceed as follows:

- We evaluate the agreement between the annotators on each token;
- We sample a random subset of 85 tasks and ask an expert to annotate them. Then, we evaluate standard binary metrics between aggregated crowd annotations and expert annotations. We hired a third-party expert to perform this evaluation.

In total for TACREV, there were 5,586 annotations tasks with 187,467 tokens; 32,110 tokens were annotated as important by at least one annotator and 17,841 tokens (more than a half) were annotated by all 3 annotators. Analogously for CoNLL04, there were 376 annotations tasks with 11,009 tokens; 2,486 tokens were annotated as important by at least one annotator and 1,655 tokens (more than a half) were annotated by all 3 annotators.

Table 1 presents the results of our evaluation. We observe that the humans perform very well on the classification task and show a decent agreement on both classification and annotation. A comparison with annotations produced by our expert also shows that the produced annotations are reliable and of high quality. Table 2 shows examples of the subjectivity of the annotators. Specifically, some annotators tend to be more detail-oriented than others as they annotate not just a word but rather a phrase.

5.2 RQ2: Alignment of Automatic Explanations

In this section, we analyze the automatic explanations. We start by comparing the automatic explanations produced by various explainability methods against human annotations. For each pair (model, explainability method) we take the explanations that correspond to the class predicted by the model and calculate precision and recall scores between explanations and human annotations. That is, given two vectors **ha** and **xai** of human annotations and explainability weights, respectively, we treat **ha**, a binary vector by our definition in Section 3.1, as the ground truth and **xai** as predictions. We calculate the threshold for **xai** as in Eq. 1. Thus, high precision means that the explainability method assigns high weights to the relevant tokens that annotators marked as important and low weights to irrelevant tokens. High recall indicates that the model does not miss important information.

It is worth noting that we use a per-input threshold to calculate the alignment between XAI explanations and human annotations as described in Section 3.1 rather than global threshold. This design choice follows the aim to maximize an alignment between human annotation and explanation produced by the XAI method for each data instance. In contrast to a per-input threshold, using a global threshold would be less sensitive to variability between different data instances such as a high number of relation types

Model	Explainability method	TACREV						CoNLL04					
		P_{corr}	P_{inc}	R_{corr}	R_{inc}	$F1_{corr}$	$F1_{inc}$	P_{corr}	P_{inc}	R_{corr}	R_{inc}	$F1_{corr}$	$F1_{inc}$
PA-LSTM	Words between entities	0.69	0.47	0.97	0.92	0.78	0.57	0.81	0.47	0.96	0.92	0.86	0.57
	LIME	0.95	0.46	0.91	0.79	0.92	0.45	0.95	0.44	0.83	0.69	0.87	0.44
	SHAP	0.92	0.44	0.91	0.79	0.9	0.44	0.93	0.45	0.84	0.69	0.87	0.46
SpanBERT	Words between entities	0.66	0.47	0.96	0.91	0.75	0.58	0.79	0.5	0.96	0.92	0.84	0.59
	LIME	0.78	0.44	0.78	0.76	0.73	0.44	0.9	0.34	0.86	0.77	0.86	0.36
	SHAP	0.74	0.45	0.75	0.72	0.67	0.46	0.83	0.28	0.86	0.84	0.82	0.34
RoBERTa	Words between entities	0.65	0.49	0.96	0.9	0.74	0.59	0.77	0.46	0.96	0.89	0.83	0.55
	LIME	0.77	0.46	0.69	0.73	0.69	0.44	0.83	0.41	0.72	0.78	0.73	0.42
	SHAP	0.76	0.43	0.74	0.78	0.71	0.44	0.75	0.41	0.73	0.75	0.7	0.42

Table 3: Precision, recall, and F1 of automatic explainability methods vs. human annotations for PA-LSTM, SpanBERT, and RoBERTa on the TACREV and CoNLL04 datasets. P_{corr}, R_{corr}, $F1_{corr}$ are precision, recall, and F1 calculated for the instances with the correct predictions of the corresponding model, while P_{inc}, R_{inc}, $F1_{inc}$ are calculated for the erroneous predictions. The metrics are averaged over instances. Human annotations are aggregated by majority vote.

and heterogeneity of natural language. Depending on the scenario, one can choose the threshold that is more specific to relevant data characteristics (i.e., the per-input threshold as we do in our work) or the one that is more robust to data variability and potential data noise (i.e., global threshold). Another potential limitation of the global threshold is that there are instances where all explainability weights are lower than the threshold, thus, it is not possible to calculate precision on these instances, limiting our evaluation and analysis.

We explore two different methods to aggregate the annotations coming from 3 annotators: (i) **At least one**: all tokens annotated by at least one annotator, and (ii) **Majority Vote (MV)**: only tokens annotated by a majority of annotators. We observe very similar results for both aggregation techniques with majority vote yielding slightly higher precision and recall levels. The observation is in line with results from related work that under most circumstances majority voting remains a competitive aggregation method [47]. It is worth noting that applying the quality control techniques described in Section 4.2 and aggregating annotations help mitigate potential negative impact of inexperienced or fraudulent annotators. For the sake of clarity, we report only the results for majority vote.

5.2.1 Relationship Between Model Rationale and Human Annotation. We present the high-level results in Table 3 showing the binary metrics for each model and each explainability methods on two datasets. For both datasets and all models we see the same pattern: precision and F1-score of explanations on instances with incorrect predictions are significantly lower that on instances with correct predictions. This is a clear sign that erroneous predictions are correlated with some wrong reasoning of the model. The higher drop in precision than in recall indicates that the models tend to pay attention to the irrelevant tokens rather than miss the relevant tokens. In particular, the drop in precision for the baseline method, words between entities, means that model errors tend to occur when there is a lot of irrelevant information between the entities (for example, when the entities are far from each other).

Our results show that there is a strong relationship between the alignment of XAI explanations vs. human annotations and the likelihood of the model prediction being correct. Figure 3 highlights specific data instances with different alignment levels. Specifically, rows 1-3 illustrate the "right" cases: high alignment of explanation and correct model prediction, that is, the cases when the model is right for *right reason*. Similarly, rows 6-8 correspond to the cases of misalignment and incorrect model prediction. However, there are counterexamples: (i) the cases when model reasoning significantly deviates from human reasoning but the prediction is nevertheless correct, shown in row 4, and (ii) the cases when model reasoning is accurate but the prediction is wrong, shown in row 5. We discuss the underlying reasons behind those cases in the following subsection.

5.2.2 Impact of Models and Datasets. As can be seen from Table 3, there is a difference in alignment for the two datasets. Specifically, the baseline explainability method, words between entities, has much higher precision on CoNLL04 than on TACREV while the recall is almost the same. This indicates that in TACREV the words between two entities contain irrelevant information more frequently than in CoNLL04. Figure 4 provides insight into how alignment metric F1 is distributed for correct and incorrect predictions across analyzed models and datasets[3]. On TACREV, the difference between PA-LSTM and PLM-based models is significant while on CoNLL04 it is less noticeable.

We hypothesize that the underlying reason is the very different number of relation types in these two datasets (6 in CoNLL04 vs. 42 in TACREV). Moreover, some relation types in TACREV are very similar and hardly distinguishable (e.g., `countries_of_residence` and `origin`, see row 5 in Figure 3) which leads to high explanation alignment but erroneous predictions. Note that for CoNLL04 F1 never equals 1 as there are no such relations that can have very similar explanations of high-level alignment. In addition, a high alignment of PA-LSTM on TACREV is partially attributed to the use of additional lexical features (e.g., part-of-speech tags and named entity tags). Indeed, our ablation study with PA-LSTM trained without these features shows that average F1 of LIME explanations drops from 0.95 to 0.89 while the percentage of instances with correct

[3]We used LIME explanations.

	Row	Alignment	Data Instance
Correct predictions	1	Accurate	He came back to Afghanistan ... and worked in the Foreign Ministry... **Prediction:** `employee_of` **Label:** `employee_of`
	2	Small deviation	He came back to Afghanistan and worked in the Foreign Ministry, specializing in relations with Europe. **Prediction:** `employee_of`
	3	Small deviation	...the death of political prisoner Orlando Zapata, who died 85 days into a hunger strike to protest prison conditions. **Prediction:** `cause_of_death` **Label:** `cause_of_death`
	4	Significant deviation	Mohammed Oudeh, ... and the mastermind of 1972 Munich raid, died of kidney disease on Saturday in Damascus. **Prediction:** `cause_of_death` **Label:** `cause_of_death`
Incorrect predictions	5	Accurate	US actress Patricia Neal, ... died at her home in Massachusetts Sunday at the age of 84... **Prediction:** `countries_of_residence` **Label:** `origin`
	6	Significant deviation	...the death of political prisoner Orlando Zapata, who died 85 days into a hunger strike to protest prison conditions. **Prediction:** `no_relation` **Label:** `cause_of_death`
	7	Significant deviation	...a widespread insider-trading case that has ensnared a number of money managers and company executives, including the Galleon Group co-founder Raj Rajaratnam. **Prediction:** `top_employees` **Label:** `founder`
	8	Significant deviation	At 22, he graduated from West Point... **Prediction:** `cities_of_residence` **Label:** `schools_attended`

Figure 3: Examples of different levels of alignment between XAI and human annotations from TACREV. Subject and object entities are underlined. Model knowns (TP), model unknowns (FN), and model unknowns (FP) are colored accordingly. Accurate automatic explanations correspond to $F1 \geq 0.98$, small deviations correspond to $0.8 \leq F1 < 0.98$, and significant deviations correspond to $F1 < 0.8$.

predictions and highly aligned explanations ($F1 \geq 0.93$) drops from 56% to 46%. In contrast, SpanBERT "looks" at the tokens that often co-occur with the relevant tokens but are not directly relevant for relation prediction: e.g., an instance with relation `cause_of_death` will often mention a date and a place, see row 4 in Table 3 for specific example.

5.2.3 Impact of XAI Methods. The baseline, words between entities, gives a good approximation of the relevant information, also shown in [38] achieving a high precision of 0.8 on CoNLL04 and high recall levels in all considered cases. However, more advanced explainability methods yield higher precision values with the same recall levels, except for both PLM-based models on TACREV. Finally, the difference between the two explainability methods, LIME and SHAP, is negligible.

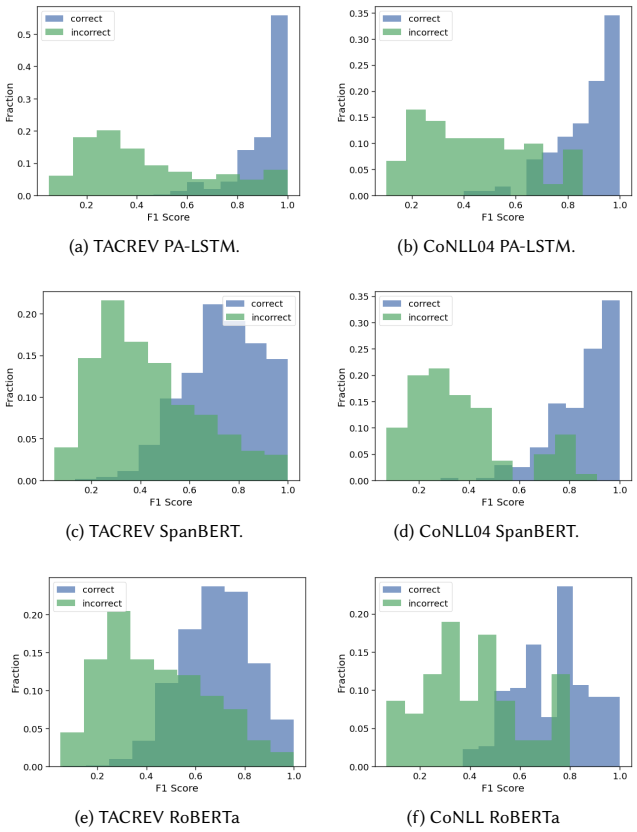
5.3 RQ3: Predicting Model Errors

In the previous subsection, we observed a notable drop in precision for the automatic explanations when the model is wrong. In that sense, explanations might actually be used for predicting model errors on unseen data. In this section, we evaluate the approaches described in Section 3.2 for predicting model errors leveraging model reasoning. Note that we do not try to find erroneous predictions of `no_relation` because the explanations of such predictions are in general not meaningful.

We implement a decision tree predictor as follows. A hyperparameter maximal depth of the decision tree is selected by 5-fold cross-validation on the training data across 100, 150, 200, 250, 300, 350, 400 and set to 350. A hyperparameter t_{expl} is set to 0.28 for TACREV and to 0.27 for CoNLL04. In addition, we present an ablation study with the decision tree predictor without explanation encoding. For the similarity matching approach, we use embeddings produced by `Sentence Transformer`[4]. A hyperparameter t_{sim} is selected by linear search in the range $[0.9, 1]$ with the step 0.01 to maximize $F1$ on the validation set.

Table 4 presents the results of these two approaches for error prediction on TACREV along with the results of a random classifier that predicts 1 (model makes an error) proportionally to the number of errors in the test set. It is worth noting that the task of predicting model errors is imbalanced: naturally, there are much less positive

[4]https://huggingface.co/sentence-transformers



(a) TACREV PA-LSTM. (b) CoNLL04 PA-LSTM.

(c) TACREV SpanBERT. (d) CoNLL04 SpanBERT.

(e) TACREV RoBERTa (f) CoNLL RoBERTa

Figure 4: Distribution of different values of F1 between LIME and human explanations.

examples because the model naturally makes a correct prediction most of the time. We make several observations: (i) Decision Tree achieves significantly higher precision than Similarity Matching due to the heterogeneity of the natural language: patterns in the unseen data are "new" and do not match the gathered collection of the model knowns. Better generalization of model knowns is required to achieve higher precision levels for Similarity Matching, the we leave for future work. (ii) Similarity Matching has very high recall confirming that most of the errors go together with unmatched (not aligned with humans) explanations. (iii) Despite the highest F1 on relation classification task, RoBERTa-based model is the least "fixable". (iv) The task of predicting errors is difficult because language is heterogeneous and, moreover, alignment and thus similarity between explanations produced by XAI and human annotations is not perfect as discussed in Section 5.2. Overall, error prediction with Similarity Matching reaches high recall values making this approach a good chase for the use-cases when missing the errors is costlier than overpredicting the errors. Decision Tree, on the other hand, reaches a balance between precision and recall, showing that model diagnosis represents a very promising approach for error prediction.

Discussion on applications. We see a number of key applications of our method. First, post-hoc interpretations of model predictions

Model	Method	P	R	F1
PA-LSTM	Random	0.296	0.287	0.291
	Similarity Matching	0.267	**0.826**	0.403
	Decision Tree	**0.492**	0.449	**0.470**
	w/o explanations	0.445	0.356	0.395
SpanBERT	Random	0.314	0.313	0.313
	Similarity Matching	0.234	**0.981**	0.377
	Decision Tree	**0.506**	0.432	**0.466**
	w/o explanations	0.479	0.376	0.421
RoBERTa	Random	0.277	0.279	0.278
	Similarity Matching	0.205	**0.899**	0.334
	Decision Tree	**0.408**	0.367	**0.386**
	w/o explanations	0.384	0.352	0.367

Table 4: Performance of the different approaches for error prediction on TACREV. We used LIME explanations for all experiments.

open the door for model improvement, e.g., involving human-in-the-loop to fix model errors. The second key application is improving training data to include enough data for the model to correctly learn to distinguish similar relations. The third application is targeting the interpretability of relation extraction in specific domains, where understanding model decisions is crucial, such as in biomedical or legal fields. Finally, our method can be used to correct errors in datasets. Despite the efforts in [1], our study found a number of inconsistencies in the validation and the test data that are probably also present in the training data.

We make our crowdsourced annotations along with the crowdsourcing task instructions publicly available[5] in order to reproduce our results and help other researchers evaluate the interpretability of their relation extraction models and develop novel methods for leveraging explanations in post-hoc manner.

6 Conclusion and Future Work

In this paper, we presented XCrowd, a hybrid human-XAI approach for diagnosing relation extraction models that combines crowdsourced knowledge elicitation and explainable AI. We conducted an empirical study over a human specification of the model should-knows and unknowns and demonstrated that the unknowns produced through XCrowd allow to foresee future model errors. In terms of future work, we plan to extend XCrowd to improve model performance by fixing model unknowns (e.g., by integrating the missing knowledge into the model via neuro-symbolic approaches).

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 683253/GraphInt). The work is part of the ICAI GENIUS lab of the research program ROBUST (project number KICH3.LTP.20.006), partly funded by the Dutch Research Council (NWO). We thank Toloka for sponsoring the data annotation part of this project.

[5]https://github.com/eXascaleInfolab/xcrowd

References

[1] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of ACL*. https://arxiv.org/abs/2004.14855

[2] Alfonso Amayuelas, Liangming Pan, Wenhu Chen, and William Wang. 2023. Knowledge of Knowledge: Exploring Known-Unknowns Uncertainty with Large Language Models. *arXiv preprint arXiv:2305.13712* (2023).

[3] J Attenberg, PG Ipeirotis, and FJ Provost. 2011. Beat the machine: Challenging workers to find the unknown unknowns, in 'Human Computation', Vol. WS-11-11 of AAAI Workshops, AAAI.

[4] Shreyan Biswas, Lorenzo Corti, Stefan Buijsman, and Jie Yang. 2022. CHIME: Causal Human-in-the-Loop Model Explanations. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 10. 27–39.

[5] Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 2370–2381. https://doi.org/10.18653/v1/2021.findings-emnlp.204

[6] Luca Console, Daniele Theseider Dupré, and Pietro Torasso. 1989. A Theory of Diagnosis for Incomplete Causal Models.. In *IJCAI*. 1311–1317.

[7] Zhun Deng, Linjun Zhang, Amirata Ghorbani, and James Zou. 2021. Improving adversarial robustness via unlabeled out-of-domain data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2845–2853.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Thomas G Dietterich. 2017. Steps toward robust artificial intelligence. *Ai Magazine* 38, 3 (2017), 3–24.

[10] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. 2013. Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *Proceedings of the 22nd international conference on World Wide Web*. 367–374.

[11] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. 2021. A systematic review of robustness in deep learning for computer vision: Mind the gap? *arXiv preprint arXiv:2112.00639* (2021).

[12] Manas Gaur, Keyur Faldu, and Amit Sheth. 2021. Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable? *IEEE Internet Computing* 25, 1 (2021), 51–59.

[13] Sidharth Gupta, Parijat Dube, and Ashish Verma. 2020. Improving the affordability of robustness training for DNNs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 780–781.

[14] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5540–5552.

[15] Peter Hase and Mohit Bansal. 2021. When Can Models Learn From Explanations? A Formal Framework for Understanding the Roles of Explanation Data. *CoRR* abs/2102.02201 (2021). arXiv:2102.02201 https://arxiv.org/abs/2102.02201

[16] Michael Heilman and Noah A Smith. 2010. Rating computer-generated questions with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk*. 35–40.

[17] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. 2015. Robust convolutional neural networks under adversarial noise. *arXiv preprint arXiv:1511.06306* (2015).

[18] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *arXiv preprint arXiv:1907.10529* (2019).

[19] Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramon Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, et al. 2020. Leveraging abstract meaning representation for knowledge base question answering. *arXiv preprint arXiv:2012.01707* (2020).

[20] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. 2017. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[21] Edith Law and Luis Von Ahn. 2011. Human computation. (2011).

[22] Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of nlp models: A survey. *Transactions of the Association for Computational Linguistics* 9 (2021), 1508–1528.

[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[24] Anthony Liu, Santiago Guerra, Isaac Fung, Gabriel Matute, Ece Kamar, and Walter Lasecki. 2020. Towards hybrid human-AI workflows for unknown unknown detection. In *Proceedings of The Web Conference 2020*. 2432–2442.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[26] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[27] Tyler McDonnell, Matthew Lease, Mucahid Kutlu, and Tamer Elsayed. 2016. Why is that relevant? collecting annotator rationales for relevance judgments. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 4. 139–148.

[28] Bart Mellebeek, Francesc Benavent, Jens Grivolla, Joan Codina-Filbá, Marta R Costa-Jussa, and Rafael E Banchs. 2010. Opinion mining of spanish customer comments with non-expert annotations on mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on Creating speech and language data with Amazon's mechanical turk*. 114–121.

[29] Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770* (2016).

[30] Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*. 39–48.

[31] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[32] 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*. 1135–1144.

[33] Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, Hwee Tou Ng and Ellen Riloff (Eds.). ACL, 1–8. https://aclanthology.org/W04-2401/

[34] Shahin Sharifi Noorian, Sihang Qiu, Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2022. What Should You Know? A Human-In-the-Loop Approach to Unknown Unknowns Characterization in Image Recognition. In *Proceedings of the ACM Web Conference 2022*. 882–892.

[35] Saima Sharmin, Nitin Rathi, Priyadarshini Panda, and Kaushik Roy. 2020. Inherent adversarial robustness of deep spiking neural networks: Effects of discrete input encoding and non-linear activations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 399–414.

[36] Alisa Smirnova, Jie Yang, Dingqi Yang, and Philippe Cudré-Mauroux. 2023. Nessy: A Neuro-Symbolic System for Label Noise Reduction. *IEEE Trans. Knowl. Data Eng.* 35, 8 (2023), 8300–8311. https://doi.org/10.1109/TKDE.2022.3199570

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[38] Zheng Tang and Mihai Surdeanu. 2022. It Takes Two Flints to Make a Fire: Multitask Learning of Neural Relation and Explanation Classifiers. *Computational Linguistics* (09 2022), 1–40. https://doi.org/10.1162/coli_a_00463 arXiv:https://direct.mit.edu/coli/article-pdf/doi/10.1162/coli_a_00463/2046371/coli_a_00463.pdf

[39] Andrea Tocchetti, Lorenzo Corti, Agathe Balayn, Mireia Yurrita, Philip Lippmann, Marco Brambilla, and Jie Yang. 2022. AI Robustness: a Human-Centered Perspective on Technological Challenges and Opportunities. *arXiv preprint arXiv:2210.08906* (2022).

[40] Jennifer Wortman Vaughan. 2017. Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. *J. Mach. Learn. Res.* 18, 1 (2017), 7026–7071.

[41] Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting Relation Extraction in the era of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 15566–15589. https://doi.org/10.18653/v1/2023.acl-long.868

[42] Ziqi Wang, Yujia Qin, Wenxuan Zhou, Jun Yan, Qinyuan Ye, Leonardo Neves, Zhiyuan Liu, and Xiang Ren. 2020. Learning from Explanations with Neural Execution Tree. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rJlUt0EYwS

[43] James Woodward. 2005. *Making things happen: A theory of causal explanation*. Oxford university press.
[44] Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2361–2364.
[45] Jie Yang, Alisa Smirnova, Dingqi Yang, Gianluca Demartini, Yuan Lu, and Philippe Cudré-Mauroux. 2019. Scalpel-cd: leveraging crowdsourcing and deep probabilistic modeling for debugging noisy training data. In *The World Wide Web Conference*. 2158–2168.
[46] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. 35–45. https://nlp.stanford.edu/pubs/zhang2017tacred.pdf
[47] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.
[48] Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373* (2021).