It Takes Two Flints to Make a Fire: Multitask Learning of Neural Relation and Explanation Classifiers

Zheng Tang*
University of Arizona

Mihai Surdeanu
University of Arizona

We propose an explainable approach for relation extraction that mitigates the tension between generalization and explainability by jointly training for the two goals. Our approach uses a multi-task learning architecture, which jointly trains a classifier for relation extraction, and a sequence model that labels words in the context of the relation that explain the decisions of the relation classifier. We also convert the model outputs to rules to bring global explanations to this approach. This sequence model is trained using a hybrid strategy: supervised, when supervision from pre-existing patterns is available, and semi-supervised otherwise. In the latter situation, we treat the sequence model's labels as latent variables, and learn the best assignment that maximizes the performance of the relation classifier. We evaluate the proposed approach on the two datasets and show that the sequence model provides labels that serve as accurate explanations for the relation classifier's decisions, and, importantly, that the joint training generally improves the performance of the relation classifier. We also evaluate the performance of the generated rules and show that the new rules are great add-on to the manual rules and bring the rule-based system much closer to the neural models.

1. Introduction

Many domains such as medical, legal, or finance, require that decision making be not only accurate but also trustworthy. Thus, understanding what the underlying model captures is a critical requirement in such applications. To this end, previous efforts addressed this limitation by adding explainability to neural models, which have come to dominate natural language processing (NLP) (Manning 2015). These explanations can be categorized along two main aspects: whether they explain a complete model (*global*) or individual predictions (*local*); and whether they are an integral part of the classification model itself (*self-explaining*) or are generated through a post-processing step (*post-hoc*) (see Related Work for a longer discussion). Most of recent proposed efforts focus on the *local* and *post-hoc* explanations (Ribeiro, Singh, and Guestrin 2016; Shapley 1952; Schwab and Karlen 2019). These directions have a few advantages such as modularity and simplicity. However, they also have two important drawbacks: these types of explanations are not guaranteed to be *faithful* to the original model to be explained, and are not *actionable*, i.e., even if they correctly explain an imperfect classification, there is no clear path towards correcting the underlying model because "changing one thing changes everything" in a neural network (Sculley et al. 2015).

* E-mail: zhengtang@email.arizona.edu.

Our article focuses on addressing the limitations of these local and post-hoc explainability approaches by providing a self-explanatory neural architecture (i.e., explanations are part of classification) that can provide both local and global explanations. In particular, we propose an approach for relation extraction (RE) that *jointly* learns how to explain and predict. Intuitively, our approach trains two classifiers: an explainability classifier (EC), which labels words in the textual context where the relation is expressed as important or not for the relation to be extracted, and a relation classifier (RC), which predicts the relation that holds between two given entities using *only* the words deemed as important. As such, our approach is *self-explanatory* because of inter-dependency between RC and EC, and generates *faithful* explanations that correctly depict how the relation classifier makes a decision (Vafa et al. 2021).

The contributions of this article are the following:

(1) We introduce a hybrid strategy to jointly train the EC and RC. Our method trains the EC as a supervised classifier when information about which words are important for a relation exists. For example, in this article we use a small set of linguistic rules to identify the important words in the relation's context. For example, in the sentence *"John was born in France,"* such a rule may identify the words *born* and *in* as important. Importantly, our approach requires minimal supervision for explanations, e.g., we report results when using an average of 5 rules per relation type on one dataset and fewer on another dataset. For the more common situation where training examples are not associated with such rules, we train using a semi-supervised strategy: we treat EC's labels as latent variables, and learn the best assignment that maximizes the performance of the RC.

(2) We evaluate our approach on two datasets: TACRED (Zhang et al. 2017) and CoNLL04 (Roth and Yih 2004). For (partial) explainability information, we select from the surface rules provided with the dataset (Zhang et al. 2017; Chang and Manning 2014) as well as from a small set of syntactic rules developed in-house using the Odin framework (Valenzuela-Escárcega, Hahn-Powell, and Surdeanu 2016). Our evaluation demonstrates that jointly training for prediction and explainability improves the performance of the relation classifier considerably on CoNLL04, and maintains the same level of performance on TACRED when compared with a state-of-the-art neural relation classifier. Importantly, our method achieves its best performance when using an average of 5 rules per relation type on TACRED and 4 rules per relation type for CoNLL04, which indicates that only minimal guidance from such rules is needed.

(3) More relevant for the goals of this work, we also evaluate our method for explainability using two strategies. The first strategy is automated and focuses on the capacity of our method to identify the same words in the context as the ones identified by rules, to verify that our approach indeed encodes the proper linguistic knowledge. Thus, this evaluation looks at examples associated with rules. In this situation, we measure the overlap between the words identified by the EC as important and the words used by rules using standard precision, recall, and F1 scores. The second strategy relies on *plausability*, i.e., can the machine explanations be understood and interpreted by humans (Wiegreffe and Pinter 2019a; Vafa et al. 2021)? To this end, we compare the tokens identified by the EC against human annotations of the context words marked as important for the relation. In both evaluations, our approach achieves considerably higher overlap with rules/human annotations than other strong baselines such as saliency mapping (Simonyan, Vedaldi, and Zisserman 2013), LIME (Ribeiro, Singh, and Guestrin 2016), SHAP (Lundberg and Lee 2017), CXPlain (Schwab and Karlen 2019), and greedy rationales (Vafa et al. 2021).

(4) We also explore the feasibility of transforming the local explanations into global ones. That is, instead of using the EC to explain individual predictions, we introduce a simple algorithm that converts the tokens marked as important into a set of rules that becomes a new, fully-explainable model that approximates the behavior of the neural RC. We compare the performance of this rule-based model with the performance of the rules written by domain experts, as well as with the neural RC model. The results show that our rule-based model has a considerably higher performance that the manually-written rules, approaching the performance of the neural classifier within a reasonable gap. In some real-world scenarios, this gap may be an acceptable cost, as the generated rule-based model provides *actionable* explainability. That is, when a rule is incorrect, a domain expert can improve it without impacting other parts of the models (Valenzuela-Escárcega et al. 2016).

2. Related Work

Our work lies at the intersection of relation extraction and explainability. We summarize these two research areas next.

2.1 Relation Extraction

Information extraction (IE), i.e., extracting structured information from text such as events and their participants, is one of the fundamental tasks in NLP that was shown to be useful for many end-user applications such as question answering (Srihari and Li 1999, 2000), and summarization (Rau, Jacobs, and Zernik 1989; Zechner 1997). Our work focuses on a subtask of IE: relation extraction (RE), which addresses the extraction of (mostly) binary relations between entities such as `place_of_birth`, which connects a person named entity with a location.

RE has received tremendous attention in the past several decades. We group the works on RE into two categories: before the "deep learning tsunami" (Manning 2015), and after.

2.1.1 Relation extraction before deep learning. The first approaches for RE were rule-based. For example, Hearst (1992) proposed a method to learn hyponymy relations using hand-written patterns. Brin (1998) proposed a dual iterative pattern/relation expansion, which exploited the duality between patterns and relations. Hassan, Awadallah, and Emam (2006) used Hyperlink-Induced Topic Search (HITS) (Kleinberg 1999) to jointly learn patterns and relations in an unsupervised manner. In general, these rule-based methods usually obtain high precision but suffer from low recall.

Statistical methods that followed the above rule-based approaches address the limited generality of rules. In terms of supervision, "traditional" machine learning approaches for RE include fully supervised methods (Zelenko, Aone, and Richardella 2003; Bunescu and Mooney 2005), or methods that rely on distant supervision, where training data is generated automatically by (noisily) aligning existing knowledge bases with texts (Mintz et al. 2009; Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011; Surdeanu et al. 2012). Most of these approaches used explicit features such as lexical, syntactic, and semantic. For example, Kambhatla (2004) proposed a maximum entropy classifier using these features. Zhou et al. (2005) found that additional features such as syntactic chunks further help the classification performance. Jiang and Zhai (2007) evaluate the effectiveness of different feature spaces for RE. Similarly, Chan and Roth

(2011) expanded feature representations to include syntactico-semantic structures that improve RE.

Kernel methods were also a popular direction for relation extraction due to their advantage of avoiding feature engineering. To this end, Miller et al. (2000) introduced a sequence kernel for relation extraction. Several researchers proposed kernels designed around constituent parse trees to capture sentence grammatical structure (Miller et al. 2000; Zelenko, Aone, and Richardella 2003; Moschitti 2006). Bunescu and Mooney (2005); Nguyen, Moschitti, and Riccardi (2009) introduced kernels based on syntactic dependencies, a simpler representation that flattens constituent trees while preserving most syntactic information. To combine the information captured by individual kernels that model different representations, Zhao and Grishman (2005) presented a composite kernel which combines multiple such individual kernels.

2.1.2 Deep learning methods for relation extraction. Deep learning approaches for RE that rely on sequence models range from using CNNs or RNNs (Zeng et al. 2014; Zhang and Wang 2015), to augmenting RNNs with different components (Xu et al. 2015; Zhou et al. 2016), or to combining RNNs and CNNs (Vu et al. 2016; Wang et al. 2016). Other approaches take advantage of graph neural networks (Zhang, Qi, and Manning 2018) or attention mechanisms (Zhang et al. 2017).

More recently, transformer-based (Vaswani et al. 2017) approaches have shown considerable improvements on many natural language tasks including RE. For example, Wu and He (2019) applied BERT (Devlin et al. 2018) to the TACRED RE task. Devlin et al. (2018); Yamada et al. (2020) showed that further improvements are possible with a better representation for the pre-trained language model.

Our approach also fits in this space. We deploy a transformer-based classifier to capture relation mentions, but we also include a novel component dedicated to explainability, which tags the words important for the relation at hand. Importantly, our direction has the relation classifier operate directly on top of the words deemed important for the relation by the explainability classifier, which guarantees that our explanations are *faithful*, i.e., our explanations correctly depict how the relation classifier makes a decision (Vafa et al. 2021). Further, we propose an efficient semi-supervised strategy to jointly train the relation and explainability classifiers using a small amount of linguistic supervision for explainability.

2.2 Explainability

Explainable artificial intelligence (XAI) has recently experienced a resurgence in the context of deep learning (Adadi and Berrada 2018; Gunning and Aha 2019; Arrieta et al. 2020; Danilevsky et al. 2020).

2.2.1 A taxonomy of explanations. Explanations can be categorized along two main aspects: whether they explain a complete model (*global*) or individual predictions (*local*); and whether they are built in the classification model itself (*self-explaining*) or are generated through a post-processing step (*post-hoc*).

Global vs. Local. Rule-based approaches (Hearst 1992; Brin 1998) or decision trees (Béchet, Nasr, and Genet 2000; Boros, Dumitrescu, and Pipa 2017) provide global explainability by constructing transparent models that people can understand. However, these directions were slowly replaced by deep learning, which tends to yield better classifiers (at least with respect to accuracy). Several efforts aimed at bringing back global explainability into deep learning. For example, in the non-NLP context of high-stakes decision-making at population level, Rawal and Lakkaraju (2020) proposed a model-agnostic framework that constructs global counterfactual explanations that provide an interpretable and accurate summary of recourses for an entire population affected by a certain problem such as bad financial credit. Closer to our work, Craven and Shavlik (1996); Frosst and Hinton (2017) both proposed distilling a neural network into a globally-interpretable model such as a decision tree.

However, most recent approaches focus on local model explainability, which preserves the underlying neural classifier and interprets its individual predictions. In this category, Hendricks et al. (2016) produced natural language explanations of individual model outputs. Han, Wallace, and Tsvetkov (2020) used influence-based training-point ranking to study spurious training artifacts in NLP settings. Wachter, Mittelstadt, and Russell (2018); Karimi et al. (2020) used counterfactual explanations to understand model decisions.

Self-explaining vs. Post-hoc. Self-explaining strategies make explanations an integral part of model predictions. For example, Tang, Hahn-Powell, and Surdeanu (2020) proposed an encoder-decoder method for relation extraction, which jointly classifies relations and decodes rules that explain the relation classifier's decisions. Rajani et al. (2019) proposed a framework that provide both answer and explanation for a commonsense QA task. In contrast, post-hoc explanations include an additional component that generates explanations after the main model produces its decisions. In this space, Liu et al. (2018) learn a taxonomy post-hoc to better interpret network embeddings. As mentioned above, Craven and Shavlik (1996); Frosst and Hinton (2017) both proposed post-hoc strategies to distill neural network into decision trees. Li et al. (2016); Fong, Patrick, and Vedaldi (2019); Hoover, Strobelt, and Gehrmann (2020) provided post-hoc visualizations as model explanations. Belinkov et al. (2017); Peters, Ruder, and Smith (2019); Zhao and Bethard (2020); Hewitt et al. (2021) introduced probes, i.e., models trained to predict certain linguistic properties in order to verify that the underlying neural models have learned the desired linguistic knowledge.

With respect to this taxonomy, our approach is self-explaining because our relation extractor has access solely to the context identified as important by the explainability classifier, and local because our core method explains individual predictions. However, in the latter part of this article we propose a simple strategy that converts local explainability into global by converting the entire neural model into a set of rules using the words deemed as important in a dataset by the explainability classifier.

2.2.2 Finding rationales. From a different perspective, our approach can be seen as finding *rationales*, i.e., subsets of context that explain individual model decisions (Vafa et al. 2021). Although these directions fit under local explainability (and mostly post-hoc) we discuss them separately due to their recent popularity and proximity to our work.

Some efforts in this space used gradient-based saliency mapping to determine the importance of tokens in context (Baehrens et al. 2010; Simonyan, Vedaldi, and Zisserman 2013; Devlin et al. 2018; Voita, Sennrich, and Titov 2021). However, gradients can be saturated, i.e., they may be close to zeros and, thus, lose explanatory signal. Ghorbani, Abid, and Zou (2019); Wang et al. (2020) also warn that gradients are fragile and they can be distorted while keeping the same prediction.

As an alternative, some researchers focused instead on attention weights in transformer networks (Wiegreffe and Pinter 2019b; Mohankumar et al. 2020). However, there is also evidence that attention weights may not be good explanations (Jain and Wallace

2019; Brunner et al. 2019; Kobayashi et al. 2020). Other efforts have used adversarial attacks on inputs to identify their importance. For example, HotFlip (Ebrahimi et al. 2017) used word-level substitutions to impact predictions. CXPlain (Schwab and Karlen 2019) calculates feature importance by masking them and comparing differences in output confidences. Feng et al. (2018); Li, Monroe, and Jurafsky (2016) focused on input reduction to identify the importance of input features. Instead of reducing, Vafa et al. (2021) greedily added input information to locate meaningful rationales. However, other research has showed that input perturbation cannot always guarantee a good explanation (Poerner, Roth, and Schütze 2018).

In a different direction, surrogate approaches (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017) generated artificial data in the neighborhood of a prediction to be explained, by randomly hiding features from the instance and learning a surrogate model to explain the predictions. AllenNLP (Wallace et al. 2019) combined adversarial attacks and gradient-based saliency mapping in their toolkit. Lastly, Lei, Barzilay, and Jaakkola (2016); Situ et al. (2021) trained a generator model to produce feature importance.

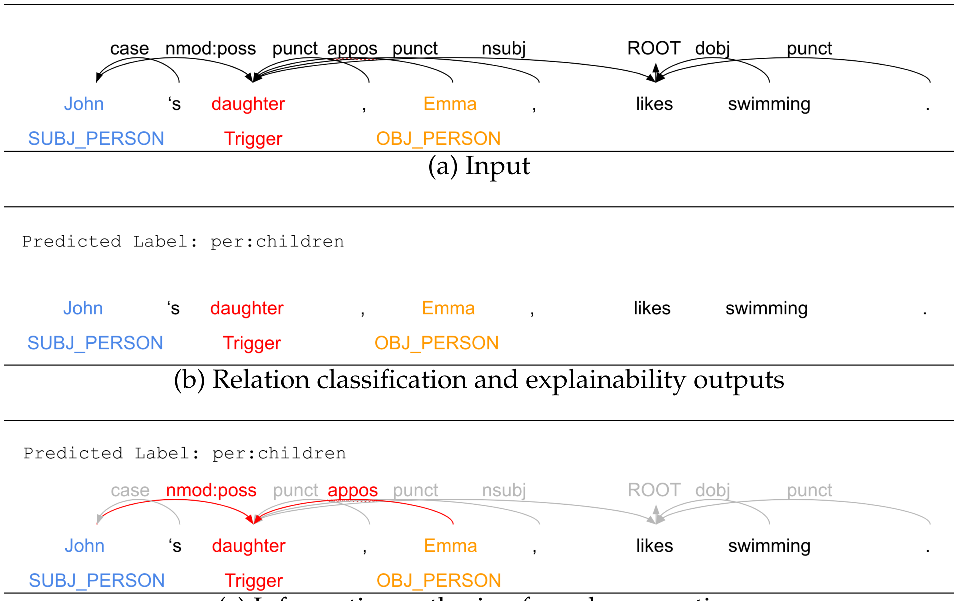
Other than the problems we mentioned above, most of these approaches are either passively reflecting the model behavior or learning rationales in an unsupervised way. Because of this, these methods cannot guarantee faithfulness and plausibility. In contrast, our proposed approach provides local explanations (or rationales) that are designed to be faithful. Further, our empirical evaluation shows that our explanations are also more plausible than other rationale finding methods (see Section 4). Lastly, our approach can generate a globally-explainable rule-based model by aggregating local rationales from a larger dataset and converting them into rules.

3. Approach

At a high level, our approach consists of two main components: a neural relation classifier with an integrated explainability classifier, and a rule generation component, which generates a rule-based model from the explainability information, i.e., context words that explain a relation, provided by the neural model.

3.1 Walkthrough Example

Before getting into the details of our approach, we highlight its key functionality with the walkthrough example shown in Table 1. Consider the sentence "*John's daughter, Emma, likes swimming.*". As shown in Table 1 (a), the task input includes: the raw text in the sentence, the entities participating in the relation (denoted as subject and object) and their types (`PERSON` here), and the syntactic dependency parse tree. Table 1 (b) shows the output of our relation classifier (RC) and explanation classifier (EC): the RC returns the predicted relation `per:children`, while the EC labels the word *daughter* as the trigger of the predicted relation. Step (c) shows the information that is collected for rule generation. This information includes: the two entities, the relation predicted, the tokens identified by the EC as the rationale for the relation, and the shortest syntactic path connecting the two entities with the rationale words. The output rule generated by our approach is shown in step (d). This rule is written in the Odin language (Valenzuela-Escarcega et al. 2015; Valenzuela-Escárcega, Hahn-Powell, and Surdeanu 2016). The rule captures the relation to be predicted (`per:children`), its trigger (*daughter*), the two arguments and their type, e.g., `subject` with the type `SUBJ_Person`, and the syntactic paths between each argument and the trigger phrase, e.g., `nmod:poss` for the subject argument. Note



(a) Input

(b) Relation classification and explainability outputs

(c) Information gathering for rule generation

```
label: per:children
pattern: |
 trigger =
     [word=/daughter/]
  subject:SUBJ_Person  = nmod:poss
  object:OBJ_Person  = appos
```

(d) Generated rule

Table 1: Walkthrough example of our approach. The task input includes information about the entities participating in the relation (denoted as subject and object) and their types (PERSON here). Our neural architecture, which includes both a relation and explanation classifier, predicts the relation that holds between the two entities (per:children here, i.e., the object is the child of the subject), as well as which words best explain the decision (in red). In step (c), the rule generator collects the necessary information from the annotated sentence, i.e., the shortest syntactic dependency path that connects the two entities with the explanation words (in red in the figure). Step (d) shows the generated rule in the Odin language.

that in this simple example, the trigger consists of a single word, but, in general, an Odin rule can take any arbitrary sequence of words as its trigger.

This example shows that our method can be deployed in two ways. First, one can use the joint RC and EC neural classifiers, which predict relations that hold between pairs of entities, as well as local explanations (or rationales) that explain the prediction. Alternatively, a different class of users may use the output of step (d), which, once applied on large text collections, contains a set of rules that describes multiple relation classes. This usage may be preferred in real-world situations that have to mitigate the

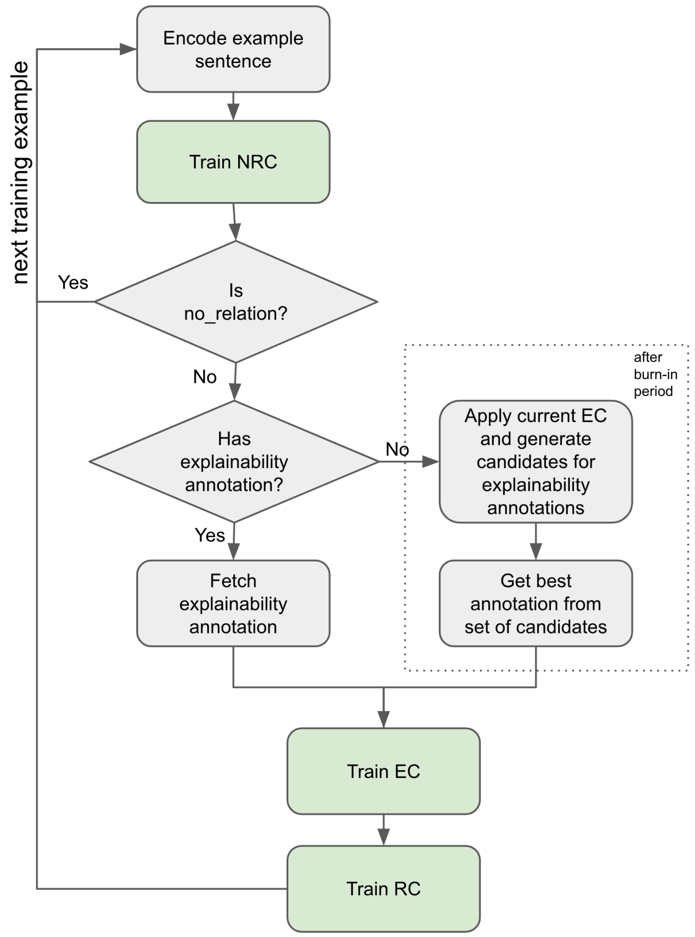

Encode example sentence
Train NRC
next training example
Yes
Is no_relation?
No
after burn-in period
Has explainability annotation?
No
Apply current EC and generate candidates for explainability annotations
Yes
Fetch explainability annotation
Get best annotation from set of candidates
Train EC
Train RC

Figure 1: Flow of our semi-supervised training procedure for an individual training example. All the "Train . . . " blocks (green background) involve parameter updates of the corresponding classifiers. These updates are shown here for an individual training example, but are batched in the actual implementation.

"technical debt" of neural methods, i.e., reduce the cost of maintaining these models over time (Sculley et al. 2015). Although not within the scope of this work, other works have shown that rule-based methods for IE can be improved and maintained at a low cost (Valenzuela-Escárcega et al. 2016).

3.2 Joint Relation and Explainability Classifiers

As mentioned, our approach jointly trains an explainability classifier (EC) and a relation classifier (RC). The RC is a multiclass classifier that distinguishes between actual relation labels seen in training. We couple the RC with a binary classifier that first predicts if the current example contains an actual relation or no relation (marked as `no_relation`). For conciseness, we call this classifier the no relation classifier (NRC). The EC is a binary word-level classifier, which labels words in the sentence that contains the relation with 1, if they are important for the underlying relation, or 0, otherwise.

We start this section with the description of the overall training procedure, and follow with details about the individual classifiers.

3.2.1 Training Procedure. The overall flow of the training procedure is shown in Figure 1. This flow is temporally split in two periods: a burn-in period, which is fully supervised, followed by a period that includes semi-supervised learning (SSL). This distinction is necessary because while all training examples in this task are guaranteed to have RC labels, most examples will *not* have gold explainability annotations. For example, for the sentence *"[CLS] John was born in London."*, the training data contains information that there is a `per:city_of_birth` relation between *John* and *London*, but may not contain information about which words are critical for this relation (*born* and *in*).

Burn-in period. In this stage, shown in the left-hand side of Figure 1, we only use the training examples that are associated with explainability annotations (see Section 3.2.2 for details on how these annotations are generated). Here we train initial versions of the three classifiers: NRC, EC, and RC (see Section 3.2.3 for details on the three classifiers). The purpose of this stage is to initialize the three classifiers such that they can be successfully used to reduce the search space for explainability annotations in the next SSL stage.

After burn-in. In this stage, the training procedure is exposed to all training examples, including those without annotations for explainability. That is, for such training examples, we simply have annotations for the relation labels (or `no_relation`), without knowing which context words explain the underlying relation. In such situations, the right-hand side of the flow in Figure 1 is used, which triggers two additional components: one to generate candidates for explainability annotations, and one to choose the best sequence of word labels (i.e., which words are important and which are not).

For the former component, exhaustively generating all possible label assignments is prohibitively expensive (i.e., $O(2^N)$ for a sequence of length N). To mitigate this cost, we rely on the prediction scores of the EC classifier to reduce the number of candidates. That is, if the score of the binary EC for a given token is higher than a threshold (t_{up}), we directly annotate the corresponding token as important (i.e., assign label 1); if this score is lower than a second threshold (t_{low}), we annotate the token as not important (label 0); and, lastly, if the the score is between the two thresholds, we generate two candidate labels for this token (both 0 and 1). For example, given an input sentence *"[CLS] [SUBJ-PER] was born in [OBJ-CITY] ."*,[1] and these prediction scores from the EC: `[0.12, 0.14, 0.19, 0.86, 0.25, 0.15, 0.01]`, using $t_{up} = 0.8$ and $t_{low} = 0.2$, we produce the following candidate label sequences: `[0, 0, 0, 1, 0, 0, 0]` and `[0, 0, 0, 1, 1, 0, 0]`, because the assignment for the token *in* is ambiguous according to the two thresholds.

Once these candidates are generated, we loop through all the generated sequences of word labels, and pick the sequence $\hat{c}$ that yields the highest score for the correct relation label according to the current RC:

$$\hat{c} = \operatorname*{argmax}_{c} p(R|c) \tag{1}$$

[1] The entities participating in a relation are masked with their named entity labels (see Section 3.2.3).

where R is the gold label of the instance, $p(R|c)$ is the score at the gold label R predicted by the RC for a given annotation candidate c. In the previous example, if the RC scores of the two candidates for the correct relation label `per:city_of_birth` are 0.8 and 0.5, we select the first candidate over the second one.

Then this sequence of labels is used as (pseudo) gold data to train the EC on this training example. This guarantees that each training example has annotations (gold, or generated through the above procedure) for both EC and RC.

Because these two components rely on having reasonable predictions from the EC and RC classifiers, we found it beneficial to include the previous burn-in period, where these classifiers are trained using the (small) amount of supervision available.

3.2.2 Explainability Annotation. As mentioned, a key part of our approach requires that EC annotations be available for a few of the training examples. To this end, rather than relying on manual annotations, which are expensive, we repurpose rules that extract the same relation. The intuition behind our approach is that if a rule exists that extracts the same relation label as the gold label in a training example, then this rule (and, specifically, its lexical elements) can be seen as an explanation of the extraction. In particular, in this article we focus on the TACRED dataset (Zhang et al. 2017), and select explanations from two sets of rules:

(1) Surface rules: The TACRED project generated a set of high-precision rules for the task, implemented in the Tokensregex language (Chang and Manning 2014). For example, the rule `SUBJ-PER was born in * OBJ-CITY`[2] extracts a `per:city_of_birth` relation between a person named entity (the subject) and a city named entity (the object) if the sequence *was born in* occurs somewhere between the two entities. For such rules, we label all tokens contained in the rule (e.g., *was*, *born*, *in*) with the label 1 (i.e., they are important for explainability), and all other tokens in the sentence with 0.

(2) Syntactic rules: In initial experiments, we observed that the TACRED surface rules have high precision but low recall. To improve generalization, we also wrote 38 syntax-based rules using the Odin language (Valenzuela-Escárcega, Hahn-Powell, and Surdeanu 2016).[3] Figure 2 shows an example of such a rule. For these syntactic rules, we marked all their lexical elements (typically the trigger predicates such as *work* or *write* in the figure) as important (label 1), and all other words as not important (label 0).

3.2.3 Classifiers. As mentioned, the building blocks of our approach consist of three classifiers: the no-relation classifier (NRC), the relation classifier (RC), and the explainability classifier (EC). These are jointly trained using the schema previously described in this section. Below we describe their individual details, which are also visualized in Figure 3.

BERT Encoder and NRC:. We follow the entity masking schema from (Zhang et al. 2017) and replace the subject and object entities with their provided named entity (NE) labels, e.g., "*[CLS] [SUBJ-PER] was born in [OBJ-CITY] . . .* ". We feed this input to a BERT-based (Devlin et al. 2018) encoder:

$$[\boldsymbol{h}_0, \ldots, \boldsymbol{h}_n] = \text{Encoder}([w_0, \ldots, w_n]) \tag{2}$$

[2] We simplified the Tokensregex syntax for readability.
[3] All these rules are included in this submission as supplemental material.

```
label: per:employee_of
pattern: |
 trigger =
     [lemma=/work|write|play|consult|serve/]
  subject:SUBJ_Person  = <acl? nsubj
  object:OBJ_Organization  = nmod
```

🟨 Label(s) to assign to a match.

🟦 Lexical constraints on the relation's predicate.

🟥 `argName:ArgType`, where `ArgType` indicates the named-entity category expected for this argument.

Figure 2: An example of a relation extraction rule in the Odin language that extracts the `per:employee_of relation` relation. The rule is driven by verbal triggers such as *work*, *play* or *serve*. The relation's arguments (the subject and object) are identified through both semantic constraints (subject must be `Person`), and syntactic ones (subject must be attached to the trigger through a certain syntactic dependency pattern: an optional (`?`) adnominal clause (`acl`), followed by a nominal subject (`nsubj`). This rule would extract a `per:employee_of` relation from the text *"... Joe is a research scientist working at IBM... "*.

where w_n is the id of the word at position n, and h_n is the hidden representation generated by BERT. We add the special masking tokens for `SUBJ-*` and `OBJ-*` to the vocabulary so that the BERT can handle them properly. We implement the NRC using a feedforward layer with a sigmoid function on top of BERT's `[CLS]` token.

Explainability Classifier (EC):. We implement the EC as a binary token-level classifier, where the positive label indicates that the corresponding token is important for the underlying relation. Section 3.2.2 discusses how these annotations are generated from rules; Section 3.2.1 explains the SSL training procedure when these annotations are not available.

Relation Classifier (RC):. Crucially, the RC relies *only* on words that are marked as important by the EC, or are part of the subject/object entity. This is an important distinction between our approach and other relation extraction methods, which typically rely on the `[CLS]` representation for classification. In the next section, we empirically show that this latter strategy is considerably less explainable than ours. This is because the `[CLS]` representation aggregates information from all tokens in the sentence, whereas our method focuses only on the important ones.

We build the aggregated representation of the important context words, subject and object as follows:

$$\boldsymbol{h}_{final} = f(\boldsymbol{h}_{ctx_1:ctx_n}) \circ f(\boldsymbol{h}_{subj_1:subj_n}) \circ f(\boldsymbol{h}_{obj_1:obj_n}) \tag{3}$$

where $\boldsymbol{h}$ denotes the hidden representations produced by BERT, $f : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$ is the average pooling function that maps from n output vectors into one; and $\circ$ is the

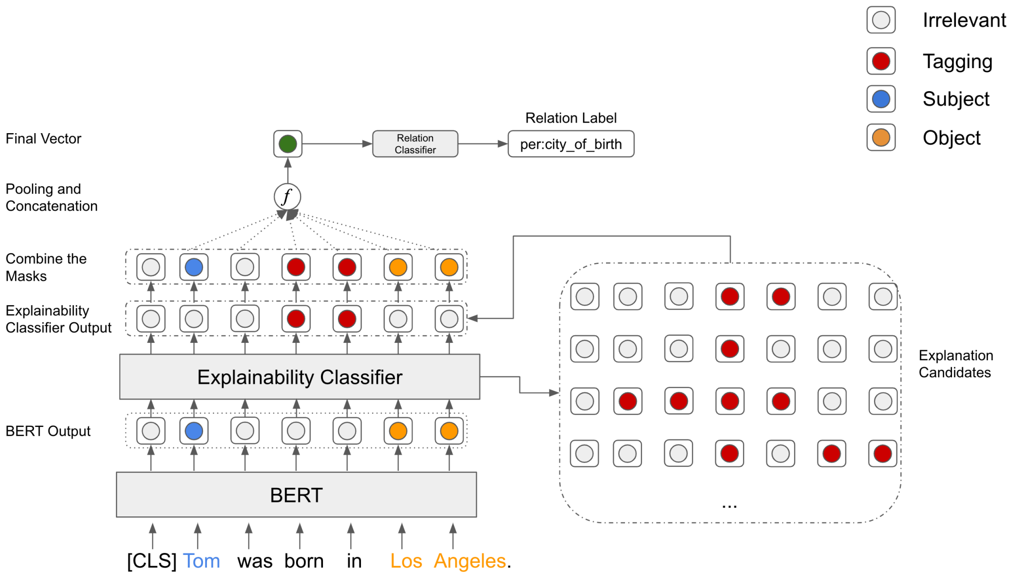


Figure 3: Neural architecture of the proposed multitask learning approach. The entity tokens (subject in blue and object in orange) are masked with their named entity labels, e.g., `SUBJ-Person`, in the actual implementation.

concatenation operator. Importantly, $\boldsymbol{h}_{ctx}$ iterates only over words marked as important by the EC.

The concatenated representation $\boldsymbol{h}_{final}$ is fed to a feedforward layer with a softmax function to produce a probability distribution $\boldsymbol{p}$ over relation types.

The three classifiers are trained using the following joint loss function:

$$loss = loss_{nrc} + loss_{ec} + loss_{rc} \tag{4}$$

$$loss_{nrc} = -(t^n * log(y^n) + (1 - t^n) * log(1 - y^n)) \tag{5}$$

$$loss_{ec} = -(t^e * log(y^e) + (1 - t^e) * log(1 - y^e)) \tag{6}$$

$$loss_{rc} = -log(p(R)) \tag{7}$$

where the losses of the NRC and EC ($loss_{nrc}$ and $loss_{ec}$, respectively) are implemented using binary cross entropy. For both, t indicates the corresponding gold label, and y is the respective sigmoid's activation. The loss of the RC ($loss_{rc}$) is implemented using categorical cross entropy, where $p(R)$ is the likelihood predicted by the model for the correct relation R.

3.3 Aggregating Local Explanations into a Global, Rule-based Model

As mentioned, the last component of our approach aggregates all local RC and EC predictions into a single rule-based model that explains the overall behavior of the RC and EC models. As such, the produced rule-based model brings global explainability to the task. We will show in Section 4 that this transformation comes with a cost in performance, but this cost might be acceptable in scenarios where such RE extraction must be deployed, maintained, and improved over a long period of time.

3.3.1 Rule Generation. As shown in Table 1 (c), our relation and explanation classifiers produce all the information necessary to generate an Odin rule. At a high-level, the Odin rules we employ here follow a predicate (or `trigger` in the Odin language) and argument template, where all arguments are connected to the trigger using a syntactic dependency path. This information is either provided by our classifiers, e.g., we use the rationale tokens identified by the EC as triggers, or can be automatically extracted from the sentence, e.g., we represent the syntactic connections between predicate and arguments using the shortest path that connects them in the syntactic dependency tree. Algorithm 1 describes this entire rule generation process:

Algorithm 1 Rule Generator

```
Input: set of annotated sentences, $\mathbb{S}$
Input: model output $\mathbb{L}$ (from RC) and $\mathbb{T}$ (from EC)
$\mathbb{R} \leftarrow \emptyset$
for every sentence $s$ in $\mathbb{S}$ do
    Get the subject and object entity $e_s$ and $e_o$ from $s$
    Get the predicted relation label $l$ from $\mathbb{L}$ and the rationale words $t$ from $\mathbb{T}$
    if $s$, $l$ and $t$ pass the filter criteria then
        Find the shortest path $p_s$ between $t$ and $e_s$ in the dependency tree
        Find the shortest path $p_o$ between $t$ and $e_o$ in the dependency tree
        $r \leftarrow$ empty Odin rule template
        Assign $l$ to $r$ as the label to match
        Assign $t$ to $r$ as the relation predicate (or trigger)
        Assign $p_s$ and $p_o$ to $r$ as the argument patterns.
        $\mathbb{R} \leftarrow \mathbb{R} \cup \{r\}$
    end if
end for
Output: set of generated rules $\mathbb{R}$
```

3.3.2 The Filtering Process. The neural models do not guarantee that they always provide output that is likely to yield a good rule. To control for this, we employ a set of heuristics as filters that remove RC and EC outputs that are unlikely to be produce a meaningful rule:

1. We remove sentences where too many words are included in the rationale generated by the EC. For example, we have observed that in some extreme cases, the EC included more than half of the words in the sentence in its explanation. Rules generated from such outputs are unlikely to generalize beyond the current sentence, and are thus skipped.

2. We discard sentences in which the subject and object have named entity types incompatible with the relation predicted. For example, a `per:children` relation is only allowed between two `PERSON` entities.
3. We remove sentences in which the explanation words are unlikely to yield good triggers based on their part of speech (POS) tags. That is, most relations are triggered by nominal or verbal predicates (`NN*` or `VB*`), or in some cases such as `per:place_of_birth` by prepositions (`IN`). However, it is unlikely that numbers (`CD`) are correct triggers for any relation. In this work, we only allow the following POS tags: `NN*`, `VB*`, `IN`, `POS`, `DT`, `-LRB-`, `-RRB-`, and `CC`.[4]

To improve coverage, in our experiments we also generate rules for sentences from the training data, for which we know the gold relation label. In these situations, we employ two additional simple filters:

5. We remove sentences for which the RC makes a prediction that disagrees with the gold label.
6. We discard sentences for which the RC does predict the correct label, but the prediction has a low confidence score.

4. Experimental Results

4.1 Data Preparation

We report results on the TACRED dataset (Zhang et al. 2017) and CoNLL04 dataset (Roth and Yih 2004). As discussed in Section 3.2.2, we provided rules for explanation supervision. For the TACRED data, we selected rules from the surface patterns of Angeli et al. (2015), and we combined them with an additional set of 38 syntactic rules in the Odin language (Valenzuela-Escárcega, Hahn-Powell, and Surdeanu 2016) that were manually created by one of the authors from the training data. For CoNLL04 data, we selected from a set of 19 syntactic rules in Odin language, 10 of which are borrowed from the TACRED syntactic rules, since the two datasets shared some overlapping relations.

These rules match 20.7% of positive examples in the TACRED training set and 24.2% of positive examples in the CoNLL04 training set. On average, 7.27 rules are assigned to each TACRED relation, and 3.8 rules are assigned to each CoNLL04 relation.

Importantly, our approach does *not* use rules at evaluation time. However, we take advantage of all existing rules to automatically evaluate the quality of the explanations generated by our method. In the TACRED dataset, the combined set of rules from (Angeli et al. 2015) and our syntactic rules match 23.9% data points in the development set, and 23.9% examples in the test set; in the CoNLL04 dataset, the syntactic rules match 20.1% of examples and 20.9% of examples respectively. We use only these matches for an automated evaluation of explainability (discussed below).

[4] These are all the POS tags allowed for rule triggers during rule generation. We use slightly different subsets of tags for different relations. For example, we allow prepositions (`IN`) for relations such as `per:place_of_birth`, but disable them for other relations, e.g., `per:employee_of`.

4.2 Baselines

4.2.1 Relation Extraction Baselines. For the relation extraction task, we compare our approach with two baselines: an extended version of the rule-based approach of Angeli et al. (2015) , and a neural state-of-the-art RE approach based on SpanBERT (Joshi et al. 2020):

- **Rule-based Extraction.** As mentioned in Section 4.1, we employ two sets of rules. First, we use the tokensregex surface rules from (Angeli et al. 2015), which are executed in Stanford CoreNLP pipeline (Manning et al. 2014). Second, we include the Odin syntactic rules we developed in-house, which are executed in the Odin framework (Valenzuela-Escárcega, Hahn-Powell, and Surdeanu 2016).[5]
- **SpanBERT.** SpanBERT (Joshi et al. 2020) is an extension of the origin BERT. It improves on BERT (Devlin et al. 2018) by (1) masking continuous random spans instead of random tokens, and (2) training the span boundary representations to predict the full content of the masked span without depending on individual token representations within it. And it outperforms BERT in many tasks including the relation extraction tasks. This is the best TACRED BERT-based model available in the HuggingFace transformer library (Wolf et al. 2020) that does not use any external resources, or does not rely on complex hybrid architectures.

4.2.2 Explainability Baselines. For explainability, we compare our approach against seven baselines, detailed below. These are all popular post-hoc explanation approaches published in recent years. Most of them provide a feature importance score for each feature.[6] Here, we labeled the top N positive features identified by the baselines as important.[7] In the first quantitative evaluation of explainability (Section 4.4.2), for all baselines we set N to be equal to the number of words in the gold explanation. Importantly, this means that all baselines have an unfair advantage over our approach, which is non-parametric with respect to N, i.e., it identifies N on the fly for each sentence. In the second, qualitative evaluation of explainability (Section 4.4.3), N is a hyper parameter that we tuned to maximize the baselines' performance.[8]

We detail the seven explainability baselines below:

- **Attention.** Attention weights have been proposed as an explanation mechanism by Bahdanau, Cho, and Bengio (2014). Followup work debated the validity of this strategy (Jain and Wallace 2019; Wiegreffe and Pinter 2019b; Kobayashi et al. 2020). However, because this remains a popular approach, we include attention weights as a baseline in this work. In particular, we use the attention weights from the last layer of a "vanilla" SpanBERT model, i.e., one that is trained on top of the `[CLS]`

[5] The rule set from (Angeli et al. 2015) also included some syntactic rules, but we found out that they only matched the simpler `per:title` relation, so we did not use them.

[6] Except for greedy adding approach which relies on labeling the features to be included in the rationale, similar to what we do.

[7] We ignored tokens part of the subject and object entities for a fair comparison.

[8] We used $N = 3$ for TACRED, and $N = 1$ for CoNLL04.

representation, without an EC. For this baseline, we label as important the top N tokens with the highest `[CLS]` attention weights.

- **Saliency Mapping.** The feature importance score of the token x_i is determined by the highest prediction's accumulated gradients in each dimension of the token in the embedding layer. These scores are obtained through a back-propagation of the highest prediction's probability. Although there are different implementations of the gradient saliency mapping approach (Devlin et al. 2018; Voita, Sennrich, and Titov 2021), we use the simple back-propagation approach from (Simonyan, Vedaldi, and Zisserman 2013).
- **LIME.** Ribeiro, Singh, and Guestrin (2016) proposed the LIME framework, which provides explanations to any black-box classifier. LIME samples the neighbors of the local instance $\boldsymbol{x}$ to be explained, by generating perturbations of the tokens in $\boldsymbol{x}$. Then, it trains a linear separator from these samples to approximate the local behavior of the model. The coefficients of the separator are later used as the feature importance score.
- **SHAP.** The Shapley value (Shapley 1952) is a cooperative game theory concept that calculates the score of feature x_i by taking into account its interactions with all other subsets of features. Similar to what LIME does, Lundberg and Lee (2017) also train a linear model to approximate the local behavior around the sampled neighbors. However, unlike LIME, which uses cosine similarity or L2 distance as its kernel, they propose a SHAP kernel which is determined by the number of permutations of features.
- **CXPlain.** Schwab and Karlen (2019) proposed an approach called CXPlain that explains the decisions of any machine-learning model by measuring the importance of the model's features. To this end, CXPlain masks each token x_i in $\boldsymbol{x}$, and calculates the score of x_i by comparing the output with the masked input $\overline{\boldsymbol{x}}$ against the output that relies on the original input $\boldsymbol{x}$. The difference between the two is calculated using a causal objective.
- **Greedy Adding.** Instead of randomly sampling from perturbations or masking the features, Vafa et al. (2021) proposed a method that greedily adds the features to the input data point. That is, it starts with an empty rationale, and each time it selects and adds the feature that increases the probability of the correct label y_t the most. The process repeats as long as the confidence in predicting y_t keeps increasing.
- **All Words between Subject and Object.** We have observed that most of the important words that determine the relation between the entities occur in the span between the two entities. To capture this intuition, we implemented this simple baseline, which simply includes all the words between subject and object in its rationale.

4.3 Implementation and Evaluation Details

Before introducing our results, we discuss key details about our implementation and evaluation.

To avoid the RC classifier overfitting on the names in the sentence (Suntwal et al. 2019), we mask the subject and object entities by replacing the original tokens in these entities with a special token, i.e., SUBJ-<NE> or OBJ-<NE>, where <NE> is the corresponding name entity type provided in the dataset. We use the pre-trained SpanBERT to encode the input sentence. For the TACRED dataset, which is organized to contain a single relation per sentence, we feed the [CLS] token to the final linear layer for relation classification. However, for the CoNLL04 data, which typically contains more than one relation per sentence, we used the concatenation of the [CLS] hidden state and the average pooling of [SUBJ] and [OBJ] hidden state embeddings. This was necessary to distinguish between the different relations that co-occur in the same sentence. We used the AdamW optimizer (Loshchilov and Hutter 2019) for all training processes. We evaluated all RC classifiers using the standard micro precision, recall, and F1 scores. All neural models were trained using 5 different random seeds; we report the average scores and standard deviation over these seeds for RC.

For explainability, we report two evaluations.[9] For the first, automated evaluation, we use only the data points that are associated with a rule that produces the same relation label as the gold data. For these examples, we consider the lexical artifacts of the rule as gold information for explainability (as explained in §3.2.2). We measure the overlap between the important words produced by the analyzed methods and this data using precision, recall, and F1 scores. We also include a second, qualitative evaluation on the *plausability* of the generated explanations (Vafa et al. 2021), where a more plausible explanation will overlap more with a relation explanation manually generated by domain experts. For this evaluation, we sampled 100 and 60 data points from the test sets of TACRED and CoNLL04, respectively. These sentences where our model predicted a relation, and where there is *no* gold annotation from rule-based method (i.e., no rule matched). We split these data points into two sets: a subset where our method predicted the correct relation, and one where it did not. In other words, in the former set, we investigate the capacity of the explainability methods to explain correct predictions, while in the latter we analyze their capacity to explain why the machine was incorrect. Two domain experts[10] manually annotated rationales for these sentences and the provided relation labels. The annotators were asked to identify the minimal set of tokens that explain the provided relation. Or, in other words, identify the tokens that when replaced with other words change the relation to be predicted. For example, in the sentence *SUBJ-PER was born in OBJ-CITY.*, if we replace the words *born in* with other words (e.g., *moved to*), the relation between the subject and object changes. Importantly, to avoid any potential bias, the two annotators worked completely independently of each other, and had no access to explanations provided by any algorithm.[11] We evaluate the overlap between the machine and human rationales using the same standard precision, recall, and F1 measures.

Lastly, we evaluate the quality of the generated rule-based model. To this end, we evaluated two sets of rules: rules generated from the training sentences,[12] and rules generated over the test set. In the latter scenario, we do not use any gold data. That is, we rely on the predicted relation labels (from the RC) and rationales (from the EC) to generate rules. Thus, the latter setting is akin to transductive learning, i.e., where the model has access to the unlabeled data from the testing partition, but no access to any

[9] We did not include an evaluation of faithfulness, which is typically done by post-hoc explainability approaches (Ribeiro, Singh, and Guestrin 2016; Schwab and Karlen 2019) because our approach is faithful by design, i.e., our RC only relies on the tokens identified by the EC.

[10] These were two of the authors.

[11] To encourage reproducibility, we will release these annotations, if accepted.

[12] We filter our training relations which matched a gold rule, since there is already a rule assigned to them

Approach	Precision	Recall	F1
Baselines			
Rules	**85.82**	24.21	37.77
SpanBERT (Joshi et al. 2020)	69.97±0.58	**70.20**±1.73	**70.07**±0.73
Our Approach			
Supervised	51.06±3.57	48.32±2.33	49.61±2.42
Semi-supervised	73.34±2.08	66.15±1.65	**69.53**±0.76

Table 2: Relation extraction results on the TACRED test partition. Our model trains on the entire training partition using the SSL method discussed in Section 3.2.1. We used the pre-trained SpanBERT-large from HuggingFace.

Approach	Precision	Recall	F1
Baselines			
Rules	81.6	16.82	27.90
SpanBERT (Joshi et al. 2020)	81.30±4.89	71.01±5.11	75.78±4.79
Our Approach			
Supervised	62.71±2.27	53.32±0.95	57.63 ±1.39
Semi-supervised	**83.01**±2.16	**76.30**±3.08	**79.46**±0.92

Table 3: Relation extraction results on the CoNLL04 test partition. Our model trains on the entire training partition using the SSL method discussed in Section 3.2.1. We used the pre-trained SpanBERT-large from HuggingFace.

human annotations. We evaluate the performance of these rule-based models using the same micro precision, recall, and F1 scores as the first RC evaluation.

4.4 Results and Discussion

In this section, we introduce and discuss the results for both relation and explainability classification. We conclude this section with an error analysis that highlights some typical errors in our models.

4.4.1 Relation Extraction. Table 2 and 3 report the RE performance of all methods discussed on the TACRED and CoNLL04 datasets. The results of all statistical approaches are averaged over three random seeds. For all these models we report average performance and standard deviation in the tables. We draw the following observations from these tables:

- First, the SSL variant of our approach improves considerably over the equivalent supervised-only setting (i.e., training just on the data points that have matching rules). The improvement is 19.92% F1 (absolute) on TACRED, and 21.83% (absolute) on CoNLL04. These results highlight the importance of SSL for this task.
- Second, our approach is statistically similar to SpanBERT on TACRED, but yields a statistically-significant improvement of nearly 4% F1 (absolute) on

Approach	Precision	Recall	F1
Attention	30.28	30.28	30.28
Saliency Mapping	30.22	30.22	30.22
LIME	30.45	36.84	32.49
SHAP	31.27	31.27	31.27
CXPlain	53.60	53.60	53.60
Greedy Adding	40.47	50.53	40.81
All words in between SUBJ & OBJ	71.48	86.33	78.21
Our Approach	**98.09**	**99.09**	**98.12**

Table 4: Automated evaluation of explainability on TACRED, in which we compare explainability annotations produced by these methods against the lexical artifacts of rules.

Approach	Precision	Recall	F1
Attention	69.44	69.44	69.44
Saliency Mapping	42.42	42.42	42.42
LIME	62.45	89.39	68.45
SHAP	34.85	34.85	34.85
CXPlain	50.00	50.00	50.00
Greedy Adding	23.24	54.55	29.58
All words in between SUBJ & OBJ	72.99	96.59	77.29
Our Approach	**99.29**	**100**	**99.52**

Table 5: Automated evaluation of explainability on CoNLL04, in which we compare explainability annotations produced by these methods against the lexical artifacts of rules.

CoNLL04.[13] This indicates that jointly training for classification and explainability helps the classification task itself (or, in the worst case, does not hurt relation classification). Table 3 also shows that our approach has the highest RE recall on CoNLL04, higher than the vanilla BERT by 5%. This suggests that explainability also serves as a disambiguator in situations where multiple relations co-occur in the same sentence (the common setting in CoNLL04) by narrowing the text to just the context necessary for the relation at hand. As further evidence that performing RC on top of explanations helps disambiguate the underlying text, the standard deviation of our approach on CoNLL04 is five times smaller than that of SpanBERT.

- Lastly, our approach nearly double the F1 score of the rule-based approach on TACRED, and more than doubles it on CoNLL04. This is caused by large improvements in recall, which highlights the importance of hybrid strategies that combine rules and neural components.

[13] We performed statistical significance analysis using non-parametric bootstrap resampling with 1000 iterations.

Num of Rules	Precision	Recall	F1
Relation Classification			
Up to top 1 (0.98 rules/relation)	69.20	65.89	67.51
Up to top 5 (3.56 rules/relation)	69.96	**69.14**	69.55
Up to top 10 (5.02 rules/relation)	71.62	67.16	69.32
All rules (7.27 rules/relation)	**74.05**	67.10	**70.40**
Explainability Classification			
Up to top 1 (0.98 rules/relation)	91.97	83.75	84.00
Up to top 5 (3.56 rules/relation)	97.03	93.03	93.80
Up to top 10 (5.02 rules/relation)	97.86	94.94	95.18
All rules (7.27 rules/relation)	**98.09**	**99.09**	**98.12**

Table 6: Learning curve of our approach on TACRED based on amount of rules used. In each experiment, we use up to top k rules per relation type; the number in parentheses is the actual average number of rules per type.

4.4.2 Quantitative Evaluation of Explainability. The results of the automated evaluation of explainability in Tables 4 and 5 show that our approach generally improves explainability quality considerably. Post-hoc explanation methods do not provide the same explanation quality compared to our method, which actively models explainability. Note that the high performance of annotating all the words between subject and object is caused by the fact that most data points in this evaluation are associated with surface rules, which prefer shorter contexts that are more likely to contain only significant information. Nevertheless, the 20% F1 gap between this strong baseline and our method indicates that our method successfully learns how to generalize beyond these simple scenarios.

However, we note that these results are not terribly surprising: our method is trained to generate explanations that mimic lexical artifacts of rules, while the other explainability baselines have not been exposed to rules during their training. Thus, this evaluation is necessary (to validate that our approach is learning to do what we intended, which is to mimic the lexical artifacts of rules) but not sufficient. In the next sub-section, we will show that our approach overlaps with human explanations much more than all other explainability baselines.

Table 6 lists a learning curve for our approach on TACRED, as we vary the amount of rules available per relation. That is, for each relation, we use up to top k rules, where k varies from 1 to 10. In the table we include results for both relation and explainability classification using the same measures as the previous tables. The table shows that even in the "up to top 5 rules" configuration (which means an average of 3.6 rules per relation type in practice), our model obtains a close F1 to the SpanBERT model (second row in Table 2) with good explainability. This result indicates that our approach performs well with minimal human supervision for explanation guidance. Note that we do not include the learning curve for CoNLL04 since there are only 19 rules applied to this dataset, which translates into only 3.8 per relation type.

4.4.3 Qualitative Evaluation of Explainability. Tables 7 and 8 lists the results of our evaluation of the plausability of explanations by comparing them against human annotations of explainability. Similar to evaluations of machine translation, we choose the higher scores between the machine methods and any of the two human annotators.

Approach	Precision	Recall	F1
Attention	41.39	20.60	26.50
Saliency Mapping	18.73	35.58	23.41
LIME	14.31	26.03	18.09
SHAP	13.86	22.85	16.79
CXPlain	28.84	55.06	36.48
Greedy Adding	31.59	33.52	30.16
Our Approach	**67.58**	**72.10**	**65.28**

Table 7: TACRED evaluation of the plausability of explanations, which measures the overlap between machine explanations and human annotations. For each method, we pick the higher scores between the two human annotators.

Approach	Precision	Recall	F1
Attention	61.06	30.30	38.94
Saliency Mapping	18.79	39.39	24.43
LIME	22.14	53.33	30.09
SHAP	18.18	36.36	23.27
CXPlain	21.21	44.55	27.82
Greedy Adding	33.33	38.03	32.21
Our Approach	**65.15**	**59.24**	**58.97**

Table 8: CoNLL04 evaluation of the plausability of explanations, which measures the overlap between machine explanations and human annotations. For each method, we pick the higher scores between the two human annotators.

Note that the human annotators had a Kappa agreement (McHugh 2012) of 69.8% on labeling the same tokens as part of an explanation. This is considered moderate (Landis and Koch 1977), which we found encouraging considering the complexity of the task and the fine granularity of the annotations. We investigated the differences between the human annotators and observed that they are caused either by legitimate annotation errors or by the fact that there are multiple valid rationales for a given relation. For example, in the sentence *OBJ-PER is the CEO and president of SUBJ-ORG*, the relation `org:top_membersemployees` can be explained either by the tokens *CEO* or *president*.

The two tables indicate that our approach generates explanations that have considerably higher overlap with human-generated explanations, even though all data points part of this evaluation were chosen to *not* have a matching rule. This suggests that our approach generates high-quality explanations of its predictions regardless of whether it has seen the underlying pattern or not. Moreover, the recall of our approach is much higher than that of the other post-hoc explanations, which have not been exposed to rules during training. This shows that with a small amount of supervision, the generated explanations can be better aligned with human intuitions.

We include several examples of the generated rationales in Figures 4, 5, 6, and 7. These examples indicate that most of the baselines are noisier, i.e., they contain a considerable amount of false positives (words that should not be part of the rationale) and false negatives (words that should be included but are not). In contrast, our method does a better job focusing on the right explanation tokens.

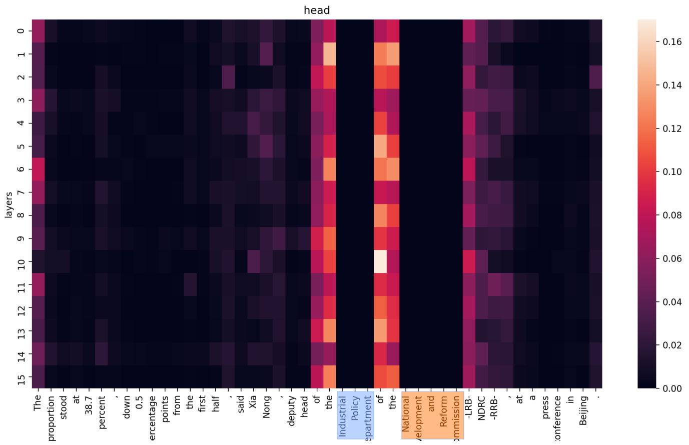
Our Approach	The proportion stood at 38.7 percent , down 0.5 percentage points from the first half , said Xia Nong , deputy head of the Industrial Policy Department **of the** National Development and Reform Commission (NDRC) , at a press conference in Beijing . Gold label: `org:subsidiary`; predicted label: `org:subsidiary`
Saliency	The proportion stood at 38.7 percent , down 0.5 percentage points from the first half , said **Xia Nong** , deputy head of the Industrial Policy Department of the National Development and Reform Commission (NDRC) , at a press conference in **Beijing** . Predicted label: `org:subsidiary`
LIME	The proportion stood at 38.7 percent , **down** 0.5 **percentage** points from the **first** half , said Xia Nong , deputy head of the Industrial Policy Department of the National Development and Reform Commission (NDRC) , at a press conference in Beijing . Predicted label: `org:subsidiary`
SHAP	The proportion stood at 38.7 **percent , down** 0.5 percentage points from the first half , said Xia Nong , deputy head of the Industrial Policy Department of the National Development and Reform Commission (NDRC) , at a press conference in Beijing . Predicted label: `org:subsidiary`
CXPlain	The proportion stood at 38.7 percent , down 0.5 percentage points from the first half , said Xia Nong , deputy head **of** the Industrial Policy Department **of** the National Development and Reform Commission **(** NDRC) , at a press conference in Beijing . Predicted label: `org:subsidiary`
Greedy Adding	The proportion stood at 38.7 percent , down 0.5 percentage points from the first half , said Xia Nong , deputy head of the Industrial Policy Department **of** the National Development and Reform Commission (NDRC) , at a **press conference** in Beijing . Predicted label: `org:subsidiary`
Attention Weights	

Figure 4: Examples of explainability annotations on TACRED for a *correct* RC prediction. The subject and object entities, which are provided in the task input, are highlighted in blue and orange. The important tokens for explainability identified by the various methods are highlighted in red. The bottom of the figure shows the heatmap of $[CLS]$ attention weights for BERT's 16 heads (the lighter the color the higher the weight). We shrink the weight range margin to make the color scale distinguishable. For a fair comparison, we masked the subject and object in the attention weights.

Our Approach	Pauliina Miettinen , **a** defender and goalkeeper **for** Finaland , was named Tuesday as the new coach of Women 's Professional Soccer inaugural champion Sky Blue . Gold label: `per:origin`; predicted label: `per:countries_of_residence`
Saliency	Gonzalez is the Pauliina Miettinen , a **defender** and **goalkeeper** for Finaland , was named Tuesday as the new coach of Women 's Professional **Soccer** inaugural champion Sky Blue . Predicted label: `per:countries_of_residence`
LIME	Pauliina Miettinen , **a** defender and goalkeeper for Finaland , was named **Tuesday** as the new coach of **Women** 's Professional Soccer inaugural champion Sky Blue . Predicted label: `per:countries_of_residence`
SHAP	Pauliina Miettinen , a defender and goalkeeper for Finaland , was **named** Tuesday **as** **the** new coach of Women 's Professional Soccer inaugural champion Sky Blue . Predicted label: `per:countries_of_residence`
CXPlain	Pauliina Miettinen **,** a defender **and goalkeeper** for Finaland , was named Tuesday as the new coach of Women 's Professional Soccer inaugural champion Sky Blue . Predicted label: `per:countries_of_residence`
Greedy Adding	Pauliina Miettinen , a **defender** and goalkeeper for Finaland , was named Tuesday as the new coach of Women 's Professional Soccer inaugural **champion** Sky Blue . Predicted label: `per:countries_of_residence`
Attention Weights	

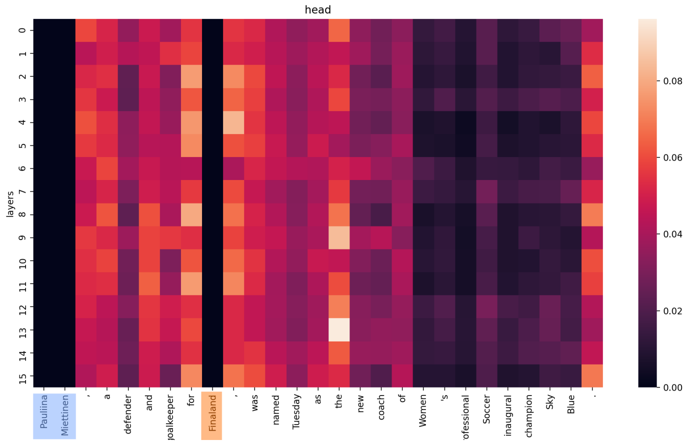


Figure 5: Examples of explainability annotations on TACRED for an *incorrect* RC prediction. This figure follows the same convention as Figure 4.

In the example in Figure 4, both our RC model and vanilla BERT predicted the correct relation. However, our method labels only the preposition *of* and the determiner *the* as its explanation, while other baselines such as LIME and SHAP completely missed them. Greedy adding and CXPlain label more irrelevant words in the context such as *(* and *press conference*. The attention weights do capture the key words, but we can clearly see additional noise surrounding the entities. In the example in Figure 5, both our model and vanilla model predicted the incorrect relation. Our model labels the preposition *for*, which provides a strong hint for its (possibly) incorrect prediction (`per:countries_of_residence`). In contrast, the baselines focus more on the nouns such as *defender* and *champion*. Applying the substitution heuristic indicates that the preposition *for* is necessary for the explanation (e.g., changing it to *against* changes the

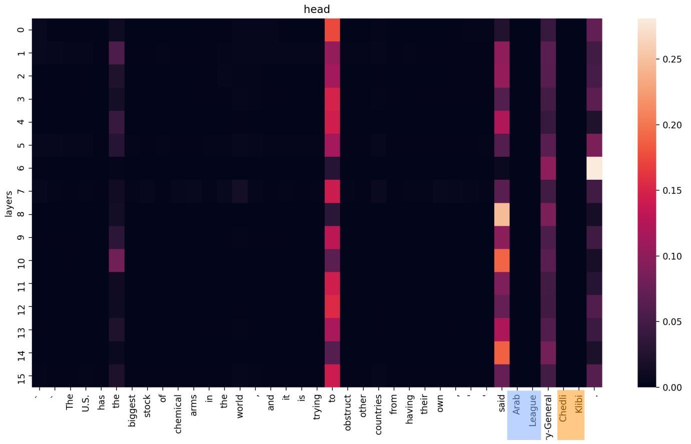
Our Approach	‘ ‘ The U.S. has the biggest stock of chemical arms in the world , and it is trying to obstruct other countries from having their own , ’ ’ said Arab League Secretary-General Chedli Klibi . Gold label: `Work_For`; predicted label: `Work_For`
Saliency	‘ ‘ The U.S. has the biggest stock of chemical arms in the world , and it is trying to obstruct other countries from having their own , ’ ’ said Arab League Secretary-General Chedli Klibi . Predicted label: `Work_For`
LIME	‘ ‘ The U.S. has the biggest stock of chemical arms in the world , and it is trying to obstruct other countries from having their own , ’ ’ said Arab League Secretary-General Chedli Klibi . Predicted label: `Work_For`
SHAP	‘ ‘ The U.S. has the biggest stock of chemical arms in the world , and it is trying to obstruct other countries from having their own , ’ ’ said Arab League Secretary-General Chedli Klibi . Predicted label: `Work_For`
CXPlain	‘ ‘ The U.S. has the biggest stock of chemical arms in the world , and it is trying to obstruct other countries from having their own , ’ ’ said Arab League Secretary-General Chedli Klibi . Predicted label: `Work_For`
Greedy Adding	‘ ‘ The U.S. has the biggest stock of chemical arms in the world , and it is trying to obstruct other countries from having their own , ’ ’ said Arab League Secretary-General Chedli Klibi . Predicted label: `Work_For`
Attention Weights	

Figure 6: Examples of explainability annotations on CoNLL04 for a *correct* RC prediction. This figure follows the same convention as Figure 4.

relation), while the nouns are not relevant. In this example, the attention weights are almost completely noisy.

In Figure 6, both our model and the vanilla SpanBERT model produce the correct prediction. The words *Secretary-General* clearly explain the *Work_For* relation in the explanations generated by our model and greedy adding. The other baselines do not provide meaningful explanations here. In Figure 7, which shows an incorrect prediction, only our model can defend its prediction by its explanation. The baseline approaches

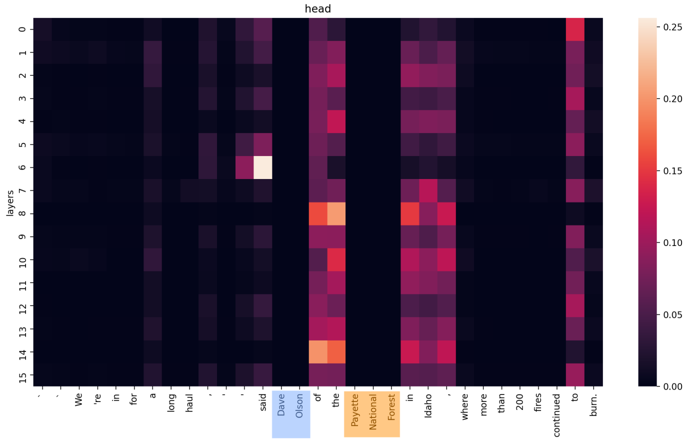
Our Approach	‘ ‘ We ’re in for a long haul , ’ ’ said Dave Olson **of** the Payette National Forest in Idaho , where more than 200 fires continued to burn. Gold label: `no_relation`; predicted label: `Work_For`
Saliency	‘ ‘ We ’re in for a long haul , ’ ’ said Dave Olson of the Payette National Forest in **Idaho** , where more than 200 fires continued to burn. Predicted label: `Work_For`
LIME	‘ ‘ We ’re in for a long haul , ’ ’ **said** Dave Olson of the Payette National Forest in Idaho , where more than 200 fires continued to burn. Predicted label: `Work_For`
SHAP	‘ ‘ We **’re** in for a long haul , ’ ’ said Dave Olson of the Payette National Forest in Idaho , where more than 200 fires continued to burn. Predicted label: `Work_For`
CXPlain	‘ ‘ We ’re in for a long haul , ’ ’ said Dave Olson of the Payette National Forest in **Idaho** , where more than 200 fires continued to burn. Predicted label: `Work_For`
Greedy Adding	‘ ‘ We ’re in for a long haul , **’ ’ said** Dave Olson of the Payette National Forest **in** Idaho , where more than **200** fires **continued** to burn. Predicted label: `Work_For`
Attention Weights	

Figure 7: Examples of explainability annotations on CoNLL04 for an *incorrect* RC prediction. This figure follows the same convention as Figure 4.

cannot provide valid explanations to defend the prediction at all. We also find that with the explanation provided from our model, one can argue that the predicted relation is actually correct, and we should change the gold label instead.

4.4.4 Interpretability: from Local to Global. Lastly, we evaluate the performance of our rule-based model that relies solely on rules, some of which were manually written (see Section 4.1), while some were automatically generated by our approach, as described in Section 3.3. The results are summarized in Tables 9 and 10. We draw two observations from these results:

- Automatically-generated rules can outperform manually-written ones. However, in order to outperform manual rules, our method must be aware

Approach	Precision	Recall	F1
Baseline			
Manual Rules[1]	**85.82**	24.21	37.77
Our Approach			
Rules from Training[2]	78.33	20.00	31.86
Rules from Test[3]	70.25	35.22	46.92
Combination of [2] and [3]	70.64	41.32	52.14
Combination of [1], [2] and [3]	73.88	**55.97**	**63.69**

Table 9: Performance of the rule-based model on the TACRED test partition. [1] is the set of manually-written surface rules of Angeli et al. (2015) coupled with our syntactic rules (see Section 4.1). [2] is the set of rules generated from our explainability classifier's outputs with gold labels on the training partition. [3] is the set of rules from the explainability classifier's outputs with predicted label on the test partition. We also evaluate the performance on combinations of these sets of rules: [2]+[3] contain all rules generated by our approach; [1]+[2]+[3] combine machine-generated rules with the manually-written rules.

Approach	Precision	Recall	F1
Baseline			
Manual Rules[1]	81.82	17.06	28.24
Our Approach			
Rules from Training[2]	**82.68**	24.88	38.25
Rules from Test[3]	67.53	49.29	56.99
Combination of [2] and [3]	67.94	54.73	60.62
Combination of [1], [2] and [3]	70.07	**63.27**	**66.50**

Table 10: Performance of the rule-based model on the CoNLL04 test partition. This table follows the same conventions as Table 9, except, in this case, [1] is the set of manually-written Odin rules we wrote for CoNLL04.

of the distribution of words in the testing partition (setting [2] + [3] in the tables). Importantly, we reiterate that when using the test sentences, our approach does *not* have access to any gold human annotations for RC and EC. That is, the rules generated from test sentences rely only on predicted relation labels and predicted explanations. The fact that rules need to be exposed to more data before they generalize is not extremely surprising: the rule matching engine we currently use relies on exact lexical matching, which means that the actual tokens to be matched must be present in the rule. However, the fact that the knowledge necessary to encode a relation extraction *can* be encoded into rules is exciting. The combination of these observations suggests that a future avenue for research that focuses on "soft rule matching" (Zhou et al. 2020), might be the direction that captures the advantages of both rules and neural methods.

- Interestingly, automatically-generated rules tend to be complementary to the manual ones. The combination of all three rule sets ([1], [2], and [3] in the tables) outperforms considerably both the setting that relies solely on

Rio de Janeiro O GLOBO Gold label: `OrgBased_In`; predicted label: `OrgBased_In`
I had an e-mail exchange with Benjamin Chertoff of Popular Mechanics in the original Loose Change thread that showed that he was not a close relative of Michael Chertoff . Gold label: `no_relation`; predicted label: `per:other_family`
Overview of Arrow Missile Program , Tasks 94AA0008Z Jerusalem THE JERUSALEM POST in English 15 Oct 93 pp 6-8 , 10–FOR OFFICIAL USE ONLY Gold label: `OrgBased_In`; predicted label: `OrgBased_In`
Like al-Shabab , **the** ADF **is primarily a** Muslim **radical** group . Gold label: `org:politicalreligious_affiliation` ; predicted label: `org:politicalreligious_affiliation`

Table 11: Typical errors that our explainability classifier commits. These include errors of under prediction (first two rows), and errors of over prediction (last two rows).

manual rules and the configuration that relies only on automatically-generated ones. The combination of all rule sets outperforms the manually-generated rules by 26% F1 and 38% F1 (absolute) in TACRED and CoNLL04, respectively. Furthermore, the TACRED result of the combined rule set approaches the performance of the neural RC within less than 10% F1. This result suggests that humans and machines can collaborate towards building a fully-explainable model that comes close to the performance of neural classifiers.

4.4.5 Error Analysis. We conclude this section with a brief error analysis of our explainability classifier in the TACRED and CoNLL04 datasets. Table 11 summarizes a few typical errors observed in the two datasets.

The first two rows in the table show examples where the EC generates explanations that rely solely on the subject and object entities, without including any word in the relations' contexts. Note that the example shown in the first row is potentially correct: it is likely that a location name that immediately precedes an organization name indicates the location of that organization. However, the second example is clearly incorrect: the correct explanation to justify the `no_relation` label should minimally include *not* and *relative*. Further, please note that a hypothetical RC that had access to the *unmasked* entities could potentially perform even better. For example, in the first case, one could infer that *O Globo* is based in *Rio de Janeiro* because the former organization name is Portuguese. However, our RC only sees masked subjects and objects. Nevertheless, we believe that our strategy of masking entities participating in relations is a valuable exercise, as it investigates the capacity of neural methods to identify explicit context necessary for relation extraction.

The last two rows in the table show examples where our EC over included words in its explanations. For example, in the last row, the *is* verb should be part of the correct explanation, but all the other words are unnecessary. This happens because the rule lexical triggers in TACRED tend to contain multiple words, which encouraged the EC to learn to include additional words in its explanation. In contrast, in CoNLL04, most triggers are single-word phrases. This encouraged the EC to include one token in its explanation, even though it is unnecessary for the prediction of the relation label.

Nevertheless, as we discussed in the previous sub-sections, our EC makes considerably fewer errors than all other explainability methods. There is no reason to believe that its current errors cannot be fixed with human feedback that would provide a (hopefully small) number of rules to adjust imperfect explanations.

5. Conclusion

We introduced an explainable approach for relation extraction that jointly trains for prediction and explainability. Our approach uses a multi-task learning framework with a shared encoder, and jointly trains a classifier for relation extraction with a second explainability classifier that labels which words in the context of the relation explain the underlying relation. Further, our method is semi-supervised, as annotations for the latter classifier are usually not available.

We evaluated the proposed approach on a relation extraction task in two datasets: TACRED and CoNLL04. Our evaluation showed that, even with minimal supervision for explanation guidance, our method generates explanations for the relation classifier's decisions that are considerably more accurate and plausible than other strong baselines such as LIME, or relying on attention weights (Simonyan, Vedaldi, and Zisserman 2013; Bahdanau, Cho, and Bengio 2014; Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017; Schwab and Karlen 2019; Vafa et al. 2021). Further, our results indicated that jointly training for explainability and prediction improves the prediction task itself, i.e., the relation classifier performs better when it is exposed only to the textual context deemed important by the explainability classifier.

We also showed that it is possible to convert these local explanations into global ones. We converted the outputs of our explainability classifier into a set of rules that globally explains the behavior of the neural relation classifier. Our results showed that our strategy for generating a rule-based model pushes the performance of rule-based approaches closer to that of neural methods.

Longer term, we envision our approach being used in am iterative semi-supervised learning scenario akin to co-training (Blum and Mitchell 1998). That is, the newly generated rules can be converted to executable rules that can be applied over large, unannotated texts to generate new training examples for the relation classifier, and vice versa.

At a higher level, we hope that this work will support meaningful collaborations between NLP researchers and subject matter experts in other domains (e.g., medical, legal), who benefit from the output of NLP systems (e.g., large-scale extraction of biomedical events) but may not understand the intricacies of the neural methods that underlie these NLP approaches.

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the World Modelers program, grant #W911NF1810014

References

Adadi, Amina and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

Angeli, Gabor, Victor Zhong, Danqi Chen, A. Chaganty, J. Bolton, Melvin Jose Johnson Premkumar, Panupong Pasupat, S. Gupta, and Christopher D. Manning. 2015. Bootstrapped self training for knowledge base population. *Theory and Applications of Categories*.

Arrieta, Alejandro Barredo, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Baehrens, David, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Béchet, Frédéric, Alexis Nasr, and Franck Genet. 2000. Tagging unknown proper names using decision trees. In *proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 77–84.

Belinkov, Yonatan, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Association for Computational Linguistics, Vancouver, Canada.

Blum, Avrim and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, ACM.

Boros, Tiberiu, Stefan Daniel Dumitrescu, and Sonia Pipa. 2017. Fast and accurate decision trees for natural language processing tasks. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 103–110, INCOMA Ltd., Varna, Bulgaria.

Brin, Sergey. 1998. Extracting patterns and relations from the world wide web. In *WebDB*.

Brunner, Gino, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2019. On identifiability in transformers. *arXiv preprint arXiv:1908.04211*.

Bunescu, Razvan and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731.

Chan, Yee Seng and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Association for Computational Linguistics, Portland, Oregon, USA.

Chang, Angel X and Christopher D Manning. 2014. Tokensregex: Defining cascaded regular expressions over tokens. *Stanford University Computer Science Technical Reports. CSTR*, 2:2014.

Craven, Mark and Jude W Shavlik. 1996. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pages 24–30.

Danilevsky, Marina, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Association for Computational Linguistics, Suzhou, China.

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

Ebrahimi, Javid, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Feng, Shi, Eric Wallace, Alvin Grissom II au2, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult.

Fong, Ruth, Mandela Patrick, and Andrea Vedaldi. 2019. Understanding deep networks via extremal perturbations and smooth masks.

Frosst, Nicholas and Geoffrey Hinton. 2017. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*.

Ghorbani, Amirata, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688.

Gunning, David and David Aha. 2019. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58.

Han, Xiaochuang, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 5553–5563, Association for Computational Linguistics, Online.

Hassan, Hany, Ahmed Hassan Awadallah, and Ossama Emam. 2006. Unsupervised information extraction approach using graph mutual reinforcement. In *EMNLP*.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Hendricks, Lisa Anne, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19, Springer.

Hewitt, John, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic.

Hoffmann, Raphael, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 541–550.

Hoover, Benjamin, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A visual analysis tool to explore learned representations in Transformer models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Association for Computational Linguistics, Online.

Jain, Sarthak and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Jiang, Jing and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, Association for Computational Linguistics, Rochester, New York.

Joshi, Mandar, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Kambhatla, Nanda. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181.

Karimi, Amir-Hossein, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions.

Kleinberg, Jon M. 1999. Hubs, authorities, and communities. *ACM Comput. Surv.*, 31:5.

Kobayashi, Goro, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Association for Computational Linguistics, Online.

Landis, J Richard and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Lei, Tao, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Association for Computational Linguistics, Austin, Texas.

Li, Jiwei, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, Association for Computational Linguistics, San Diego, California.

Li, Jiwei, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *ArXiv*, abs/1612.08220.

Liu, Ninghao, Xiao Huang, Jundong Li, and Xia Hu. 2018. On interpretation of network embedding via taxonomy induction. Association for Computing Machinery, New York, NY, USA.

Loshchilov, Ilya and Frank Hutter. 2019. Decoupled weight decay regularization.

Lundberg, Scott M and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pages 4765–4774.

Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Association for Computational Linguistics, Baltimore, Maryland.

Manning, Christopher D. 2015. Last words: Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

McHugh, Mary L. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Miller, Scott, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Mintz, Mike, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Mohankumar, Akash Kumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216, Association for Computational Linguistics, Online.

Moschitti, Alessandro. 2006. Making tree kernels practical for natural language learning. In *11th conference of the European Chapter of the Association for Computational Linguistics*.

Nguyen, Truc-Vien T., Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1378–1387, Association for Computational Linguistics, Singapore.

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Association for Computational Linguistics, Florence, Italy.

Poerner, Nina, Benjamin Roth, and Hinrich Schütze. 2018. Evaluating neural network explanation methods using hybrid documents and morphological agreement. *arXiv preprint arXiv:1801.06422*.

Rajani, Nazneen Fatema, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Association for Computational Linguistics, Florence, Italy.

Rau, Lisa F, Paul S Jacobs, and Uri Zernik. 1989. Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing & Management*, 25(4):419–428.

Rawal, Kaivalya and Himabindu Lakkaraju. 2020. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Advances in Neural Information Processing Systems*, 33.

Ribeiro, Marco Tulio, 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, ACM.

Riedel, Sebastian, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Springer.

Roth, Dan 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) at HLT-NAACL 2004*, pages 1–8, Association for Computational Linguistics, Boston, Massachusetts, USA.

Schwab, Patrick and Walter Karlen. 2019. CXPlain: Causal Explanations for Model Interpretation under Uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28.

Shapley, Lloyd S. 1952. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Situ, Xuelin, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5340–5355, Association for Computational Linguistics, Online.

Srihari, Rohini and Wei Li. 1999. Information extraction supported question answering. Technical report, CYMFONY NET INC WILLIAMSVILLE NY.

Srihari, Rohini K and Wei Li. 2000. A question answering system supported by information extraction. In *Sixth Applied Natural Language Processing Conference*, pages 166–172.

Suntwal, Sandeep, Mithun Paul, Rebecca Sharp, and Mihai Surdeanu. 2019. On the importance of delexicalization for fact verification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3413–3418, Association for Computational Linguistics, Hong Kong, China.

Surdeanu, Mihai, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, Association for Computational Linguistics, Jeju Island, Korea.

Tang, Zheng, Gus Hahn-Powell, and Mihai Surdeanu. 2020. Exploring interpretability in event extraction: Multitask learning of a neural event classifier and an explanation decoder. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 169–175, Association for Computational Linguistics, Online.

Vafa, Keyon, Yuntian Deng, David M Blei, and Alexander M Rush. 2021. Rationales for sequential predictions. In *Empirical Methods in Natural Language Processing*.

Valenzuela-Escárcega, Marco A, Gus Hahn-Powell, Dane Bell, and Mihai Surdeanu. 2016. Snaptogrid: From statistical to interpretable models for biomedical information extraction. *arXiv preprint arXiv:1606.09604*.

Valenzuela-Escárcega, Marco A., Gus Hahn-Powell, and Mihai Surdeanu. 2016. Odin's runes: A rule language for information extraction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 322–329, European Language Resources Association (ELRA), Portorož, Slovenia.

Valenzuela-Escarcega, Marco A., Gustave Hahn-Powell, Thomas Hicks, and Mihai Surdeanu. 2015. A domain-independent rule-based framework for event extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Assian Federation of Natural Language Processing: Software Demonstrations (ACL-IJCNLP)*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Voita, Elena, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation.

Vu, Ngoc Thang, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539, Association for Computational Linguistics, San Diego, California.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual explanations without opening the black box: Automated decisions and the gdpr.

Wallace, Eric, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A framework for explaining predictions of NLP models. In *Empirical Methods in Natural Language Processing*.

Wang, Junlin, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of nlp models is manipulable. *arXiv preprint arXiv:2010.05419*.

Wang, Linlin, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Association for Computational Linguistics, Berlin, Germany.

Wiegreffe, Sarah and Yuval Pinter. 2019a. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Association for

Computational Linguistics, Hong Kong, China.

Wiegreffe, Sarah and Yuval Pinter. 2019b. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Association for Computational Linguistics, Online.

Wu, Shanchan 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*, pages 2361–2364.

Xu, Yan, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Association for Computational Linguistics, Lisbon, Portugal.

Yamada, Ikuya, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Zechner, Klaus. 1997. A literature survey on information extraction and text summarization. *Computational Linguistics Program*, 22.

Zelenko, Dmitry, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Zeng, Daojian, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Zhang, Dongxu and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Zhang, Yuhao, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Association for Computational Linguistics, Brussels, Belgium.

Zhang, Yuhao, 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)*, pages 35–45.

Zhao, Shubin and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 419–426, Association for Computational Linguistics, Ann Arbor, Michigan.

Zhao, Yiyun and Steven Bethard. 2020. How does BERT's attention change when you fine-tune? an analysis methodology and a case study in negation scope. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4729–4747, Association for Computational Linguistics, Online.

Zhou, GuoDong, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Association for Computational Linguistics, Ann Arbor, Michigan.

Zhou, Peng, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Association for Computational Linguistics, Berlin, Germany.

Zhou, Wenxuan, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren. 2020. Nero: A neural rule grounding framework for label-efficient relation extraction. In *Proceedings of The Web Conference 2020*, pages 2166–2176.