Best of Both Worlds: A Pliable and Generalizable Neuro-Symbolic Approach for Relation Classification

Robert Vacareanu[♠,♣], Fahmida Alam[♠], Md Asiful Islam[♠], Haris Riaz[♠], and Mihai Surdeanu[♠]

[♠]University of Arizona, Tucson, AZ, USA
[♣]Technical University of Cluj-Napoca, Cluj-Napoca, Romania
{rvacareanu,fahmidaalam,asifulislam,hriaz,msurdeanu}@arizona.edu

Abstract

This paper introduces a novel neuro-symbolic architecture for relation classification (RC) that combines rule-based methods with contemporary deep learning techniques. This approach capitalizes on the strengths of both paradigms: the adaptability of rule-based systems and the generalization power of neural networks. Our architecture consists of two components: a declarative rule-based model for transparent classification and a neural component to enhance rule generalizability through semantic text matching. Notably, our semantic matcher is trained in an unsupervised domain-agnostic way, solely with synthetic data. Further, these components are loosely coupled, allowing for rule modifications without retraining the semantic matcher. In our evaluation, we focused on two few-shot relation classification datasets: Few-Shot TACRED and a Few-Shot version of NYT29. We show that our proposed method outperforms previous state-of-the-art models in three out of four settings, despite not seeing any human-annotated training data. Further, we show that our approach remains modular and pliable, i.e., the corresponding rules can be locally modified to improve the overall model. Human interventions to the rules for the TACRED relation `org:parents` boost the performance on that relation by as much as 26% relative improvement, without negatively impacting the other relations, and without retraining the semantic matching component.[1]

Rule	`[ne=per]+ <nsubj founded >dobj [ne=org]+`
Sentence 1	Bill Gates founded Microsoft
Sentence 2	Bill Gates is the founder of Microsoft
Sentence 3	John moved to New York City

Figure 1: An example of the type of rules we use in our proposed method, together with three sentences. The rule captures the `org:founder` relation with a syntactic pattern anchored by the predicate *founded* that has a person named entity as its subject and an organization as the direct object. By itself, the rule matches the first sentence, but it does not match the other two. When coupled with our semantic matching component, the rule matches the first two sentences.

1 Introduction

After the "deep learning tsunami" (Manning, 2015), neural approaches for information extraction (IE) consistently pushed the boundaries of the state of the art (Yang et al., 2016; Zhang et al., 2017; Guo et al., 2019; Yamada et al., 2020; Zhong and Chen, 2020). However, all these directions come at a cost: (i) low explainability (Danilevsky et al., 2021) and (ii) fragility (Sculley et al., 2015).

Explainability is critical in many domains such as healthcare, law, and finance (Adadi and Berrada, 2018; Goodman and Flaxman, 2016; Tjoa and Guan, 2019). While there have been efforts to incorporate explainability into neural methods (Ribeiro et al., 2016; Lundberg and Lee, 2017; Tang and Surdeanu, 2023, inter alia), most explanations are *local* and *post-hoc*, which has two important drawbacks. First, such explanations are not guaranteed to be *faithful* (Jacovi and Goldberg, 2020). Second, they are not *actionable*. That is, it is not immediately possible to modify the underlying model using insights from the explanations without risking introducing new, unforeseen behavior. In contrast, rule-based[2] methods are explainable and *pliable*,[3] but lack the generalization power of current deep learning systems (Tang and Surdeanu, 2023).

In this paper, we propose a novel neuro-symbolic architecture for relation classification (RC) that preserves the advantages of both directions, i.e., the generalization of neural methods and the pliability of rule-based approaches with a modular ap-

[1] Code available at `https://github.com/clulab/releases/tree/master/naacl2024-softrules`

[2] We refer to syntactic and surface patterns as rules, such as, `[ne=per]+ <nsubj founded >dobj [ne=org]+`.

[3] Term introduced by Dayne Freitag in the panel discussion at the PaN-DL workshop (Chiticariu et al., 2022) to indicate that rules can be modified to improve the corresponding local behavior while minimizing the impact on the rest of the model.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2576–2594
June 16-21, 2024 ©2024 Association for Computational Linguistics

proach, containing two components: a declarative rule-based model and a neural component. The first module implements relation classification with a set of explainable rules. The second increases the generalizability of rules by semantically matching them over text. Figure 1 shows an example of how the two components interact.

Our specific contributions are:

(1) We propose a *modular neuro-symbolic architecture for relation classification* that combines the advantages of symbolic and neural models. The symbolic rule-based component utilizes syntactic or surface rules automatically derived from example sentences, formulated as the shortest syntactic paths between two entities within a sentence. The neural model, which semantically matches these rules over text, is trained *without any human-annotated data*. This training involves a unique process: sentences are randomly selected from a large corpus, and rules are automatically generated between random entities in these sentences. The model is then trained in a contrastive manner to assign a high score to the original (rule, sentence) pair (or a paraphrase of the sentence) and a low score otherwise. The semantic matcher is then combined with the original rule-based model in a two-stage sieve architecture that prioritizes the higher-precision component.

(2) We obtain state-of-the-art performance on three out of four settings in two challenging few-shot RC datasets –Few-Shot TACRED (Zhang et al., 2017; Sabo et al., 2021) and a few-shot version of the NYT29 dataset (Riedel et al., 2010; Takanobu et al., 2019; Alam et al., 2024), without using the background training dataset. For example, on TACRED we observe an improvement of over 12 F1 points over previous state-of-the-art neural-based supervised methods; our overall results on TACRED are 24.19 for 1-shot and 39.38 for 5-shot, *despite never training the model on any annotated examples from this dataset*. Further, the resulting model is relatively *small*, with approximately 350M parameters.

(3) We show that our approach is *pliable* through a user study in which two domain experts manually improved the rules for the `org:parents` relation in TACRED. *Without retraining the semantic-matching neural component*, the performance for this relation increases in all settings for both experts, without impacting negatively the performance for the other relations. To our knowledge, this is the first work that shows that pliability can be preserved in neural directions for IE.

2 Related Work

We overview the three main directions that influenced this work –rule-based approaches, bootstrapping or other seed-based approaches, and explainable deep learning methods– as well as differences between the proposed work and prompting/in-context learning.

2.1 Rule-based Approaches

Rule-based methods were a popular direction for information extraction (IE) before the deep learning era. In the seminal work of Hearst (1992), the author proposed a method to learn pairs of words satisfying the hyponymy relation, starting from a simple hand-written rule. In Riloff (1993), the author introduced *AutoSlog*, a system capable of learning domain specific relations starting from hand-written patterns. The system was subsequently improved in Riloff (1996a) using statistical techniques. Some approaches towards automatically learning the patterns include (Riloff and Jones, 1999; Riloff and Wiebe, 2003; Gupta and Manning, 2014; Vacareanu et al., 2022a); the typical direction is to employ a bootstrapping algorithm, repeatedly alternating between generating rules and generating extractions with the current rules. Such approaches provided the desired explainability and pliability, but, in retrospect, lacked the generalization capabilities of deep learning methods.

2.2 Explainable Deep Learning

Deep learning models have been the preferred approach for the vast majority of NLP tasks including information extraction (IE) in the past years (Hochreiter and Schmidhuber, 1997; Sutskever et al., 2014; Vaswani et al., 2017; Devlin et al., 2018). However, this expressivity came at a cost: numerous articles reported on the fragility of the neural networks (Szegedy et al., 2014; Ilyas et al., 2019; McCoy et al., 2019), and that neural networks can reinforce biases in the data (Bolukbasi et al., 2016; Brunet et al., 2019; Mehrabi et al., 2021). As such, having an explainable system is desirable, as long as it does not come at a high cost with respect to performance. The popular approaches to explaining neural networks are either: (i) feature importance, or (ii) surrogate models (Danilevsky et al., 2021).

Techniques based on feature importance aim to highlight the feature responsible for a given prediction. For example, Sundararajan et al. (2017) uses integrated gradients to assign an importance score to each feature. Other techniques use the attention mechanism as an explanation of the model's prediction (Bahdanau et al., 2015; Xu et al., 2015). Such techniques show that a feature is important, but do not show how it is being used in the model. Moreover, techniques such as interpreting attention scores have been shown to be particularly brittle. For example, Jain and Wallace (2019) has shown that many seemingly different attention patterns can allow for the same end prediction, which raises the question of explanation fidelity. Other improved attention interpretation methods include Kobayashi et al. (2020), which suggest taking the norm of the vectors into consideration as well.

Techniques based on surrogate models train a (typically) smaller and more interpretable model to explain the original one. For example, Ribeiro et al. (2016) train a linear classifier around the point that is to be explained. Lundberg and Lee (2017) uses SHAP values as a unified measure of feature importance. SHAP values are Shapley values (Shapley, 1988) of a conditional expectation function of the original model. The key issue with surrogate models is their potential lack of fidelity with respect to the original model (Danilevsky et al., 2021).

Zhou et al. (2020) proposed an approach in the same space to ours, i.e., they also train a semantic (or "soft") rule matcher (SRM). However, there are multiple critical differences from our work. First, the SRM is used only to augment the training data for a "traditional" opaque deep learning RC model, which is the actual output of the training process. In our approach, the SRM is a critical component of the model used during inference. Second, their SRM module was developed only for surface rules consisting of word constraints, and it is unclear how to expand it to more general patterns.[4] In contrast, the rules we use in our proposed method are closer to real-world application, i.e., they contain syntactic dependency constraints and semantic entity constraints. Furthermore, their proposed approach requires an initial set of labeled data, while we operate solely in a zero-shot fashion.

All in all, while both (i) feature importance and (ii) surrogate models can provide insights into how and why the deep learning model makes a certain prediction, they do not provide any systematic mechanism to make interventions to these systems.

2.3 Seed-Based And Bootstrapping Methods

Bootstrapping (Riloff, 1996b; Lin and Pantel, 2001), is another approach that can be applied to relation extraction. Mausam et al. (2012) constructed a bootstrapping set by starting from a dataset of over 110,000 high-confidence seeds and expanding it through the distant supervision hypothesis and heuristics. Tang and Surdeanu (2023) learn a relation classifier and an explanation classifier jointly, mitigating the tension between the two by bootstrapping from a small set of seeds.

Another approach is that of using a knowledge base and casting the problem as matrix factorization (Riedel et al., 2013; Nimishakavi et al., 2016).

In our work, we do not use the distant supervision approach or any seeds. Instead, we show that a general rule matcher can be learned by just training it on zero-shot rules generated between random entities in a given sentence, without any need of a seed dataset or a knowledge base.

2.4 Prompting and In-context Learning

Lastly, we note that, despite superficial similarities, our work is considerably different from prompting and in-context learning (Brown et al., 2020; Schick and Schütze, 2020). Unlike prompts, our rules are an integral part of the model, both explicitly and through the rule representations learned by our semantic rule matching component. Further, rules offer a higher degree of expressiveness compared to raw text. Rules allow humans to unambiguously compress abstract concepts (e.g., by incorporating syntax and semantics) towards a specific goal. In contrast, with prompting and in-context learning, the level of generalization and abstraction is uncertain (Lu et al., 2021).

These advantages make our method obtain state-of-the-art (SOTA) performance as well as more controllable/pliable behavior (§4). Further, in-context learning tends to perform well only with large language models. In contrast, our neural component uses a much smaller language model containing approximately 350M parameters.

3 Proposed Method

We propose a hybrid model that combines the advantages of rule-based and neural approaches.

[4] For example, their model cannot accommodate more expressive rules that use syntax such as `[ne=per]+ <nsubj founded >dobj [ne=org]+`.

Our approach first attempts to strictly match rules, i.e., all semantic/syntactic/lexical constraints must match in the input sentence for a match to be considered. If no rule matches, we back off to a neural semantic rule matching (SRM) component that semantically aligns rules with text.

A key aspect of our proposed approach is that we do *not* incorporate a `no_relation` classifier in any form, such as a NAV or MNAV (Sabo et al., 2021). This is important as training multiple representation vectors to capture the entire `no_relation` space, as proposed in (Sabo et al., 2021) can be difficult in practice, as reported by the original authors. Instead, our method is simpler: we have rules with associated underlying relations and a single threshold $t \in [0, 1]$ to decide whether the SRM assigned score between a rule and a sentence constitutes a match or not. This threshold is application-specific and can be selected on a development set.

3.1 Strict Rule Matching Component

To implement strict rule matching in our hybrid method we use Odinson (Valenzuela-Escárcega et al., 2020). Odinson is a rule-based IE framework with two key advantages. First, it has the capability to combine surface information with syntactic dependency constraints to create a more expressive rule set. Second, the Odinson runtime engine is optimized for speed, and capable of executing rules consisting of surface and syntactic constraints in near real-time. We provide an example of the rules we use in Figure 1, together with three sentences, one where the rule matches (Sentence 1) and two where it does not (Sentence 2 and 3), according to the strict matching algorithm in Odinson. This example highlights the key limitation of traditional rule engines: even though the second sentence is semantically similar to the first, Odinson does not match it because its syntax does not align with the syntactic constraints in the rule. These are precisely the types of problems we aim to address. Lastly, we emphasize that our proposed method can work with different rule engines.

3.1.1 Rule Generation

In this paper, we use a simple strategy to generate rules for this component: for syntactic rules, we construct rules from the shortest path in the syntactic dependency tree that connects two entities in a training sentence. For surface rules, we simply take the words in-between the entities. Figure 2 shows an example of this process. Because we evaluate in a few-shot setting, the number of rules produced for a given relation label will be small, e.g., 1 or 5.

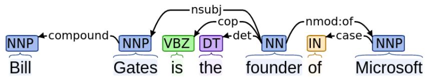
Sentence	
Subject Entity	Bill Gates
Object Entity	Microsoft
Relation	org:founder
Rule	`[ne=per]+ <nsubj founder >nmod_of [ne=org]+`

Figure 2: To create a rule from a sentence, the process involves: (a) parsing the sentence to extract its syntactic dependency tree, (b) identifying the shortest path connecting two entity mentions within this tree, and (c) constructing a rule based on the syntactic dependencies, associated words, and named entity labels found along this path. For example, the rule shown operates as follows: it requires a `per` (person) label connected to the word 'founder' via a nominal subject dependency, and 'founder' in turn linked to a `org` (organization) label through an `nmod_of` dependency.

3.2 Semantic Rule Matching Component

The example in Figure 1 highlights the need for a more nuanced approach to rule-based relation classification, one that allows for degrees of matching to overcome the collapse of every non-match to 0. To this end, we propose a transformer-based architecture (Vaswani et al., 2017; Liu et al., 2019; Radford et al., 2021) that embeds the rule and the sentence; the networks is trained to maximize the cosine similarity between these two embeddings in the case of real matches and minimize it otherwise. We describe the training procedure of our proposed semantic rule matcher below.

3.2.1 Training Dataset

A key question is how to obtain training data for the semantic rule matching component, i.e., data that aligns rules with sentences where they should match. Our method circumvents the need for gold-annotated data, capitalizing on a key insight: for any pair of entities within a sentence, a representative rule can be automatically formulated. Take, for instance, the sentence *John moved to New York City*, featuring entities *John* and *New York City*. From this, we can derive a rule, such as `[ne=per]+ <nsubj moved >nmod_to [ne=loc]+` using the underlying syntactic structure of the sentence. This rule, inherently, is indicative of the relationship between these entities, irrespective of the specific nature of this relationship. By applying this principle, we can train our model to assign a high matching score to the tuple consisting this rule and the original entities within their context, while assigning low scores to any other combinations. This innovative approach allows us to automatically create a training dataset, bypassing the traditional reliance on pre-labeled data.

To encourage the SRM to look beyond syntactic/surface structures, we create paraphrases for the extracted sentences. For example, *John moved to New York City* can be rephrased as *John relocated to New York City* without losing any semantic information. We use this insight to expand the resulting dataset with paraphrases that contain the two entities of interest.[5] We provide more details below.

We start from UMBC, a dataset of English paragraphs, totaling 3 billion words (Han et al., 2013).[6] We pre-process this dataset with standard NLP tools (Manning et al., 2014) for named entity annotations and for dependency parsing. Then, we randomly sample a sentence s_1 containing two random entities of interest (e_1, e_2), and automatically construct a rule r_1 that will match it. The resulting tuple (r_1, s_1) will then be added in the resulting dataset.[7] This process resulted in an initial dataset of approximately 140 million sentence/rule tuples. This dataset is further preprocessed as follows:

(1) We filter the data by removing duplicates and by sub-sampling frequent rules and entities. The underlying motivation is to prevent the model from overfitting to very common rules or entity types. For example, the pair (`ORG, COUNTRY`) is roughly 2 orders of magnitude more common than (`ORG, EMAIL`). At the end of this stage, the resulting dataset has approximately 4 million examples.

(2) We augment the entity types with synonyms, with the goal of encouraging the SRM component to generalize beyond the superficial clues from the entity types. For example, we randomly replace the entity type `per` with `human`, or `individual`. We provide a complete list of the synonyms we used in Appendix C.

(3) We generate paraphrases of the original sentence, while keeping the two entities of interest in the sentence. We use OpenAI's ChatGPT (`gpt-3.5-turbo-1106`) as our paraphraser, using a simple prompt (shown in Appendix D). Out of the paraphrases generated, we keep only those that contain the two entities of interest. We manually analyzed the quality of 50 random paraphrases and found all of them to be of high quality. Details can be found in Appendix I.

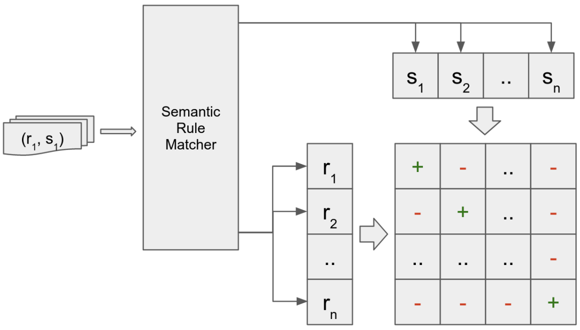


Figure 3: In our training for the Semantic Rule Matcher (SRM), we encode both rules and sentences, followed by calculating cosine similarity between each pair. The goal is to maximize similarity for matching pairs (diagonal of the matrix) and minimize it for non-matching pairs (off-diagonal elements).

In Section 4, we ablate over these three techniques to assess their contribution to the performance of the final model. In total, the resulting dataset has a total of approximately 5.6 million (rule, sentence) pairs, out of which about 1.6 million pairs were generated through paraphrasing. When learning sentence representations, we follow prior works on relation extraction (Zhang et al., 2017; Joshi et al., 2020; Zhou and Chen, 2021) and wrap the entities with special tokens, together with the corresponding named entity. For example, given the entities *Bill Gates* and *Microsoft*, the sentence *Bill Gates founded Microsoft* becomes: *# * per * Bill Gates # founded # * org * Microsoft #.*

3.3 Training the Semantic Rule Matching

We leverage the resulting dataset to train the semantic rule matching component with a CLIP-like objective. Concretely, the dataset consists of examples of the form (r, s), for example: (`[ne=per]+ <nsubj founded >nmod_in [ne=org]+`, *# * per * Bill Gates # founded # * org * Microsoft #*). We train the SRM component to assign a high cosine similarity score between the embedding of r and the embedding of s, and we use the other in-batch examples as negatives (Radford et al., 2021). Importantly, *we do not use any human-annotated data or any domain-specific relation labels* for training. We provide an overview of the training mechanism

[5] We use OpenAI's ChatGPT for this purpose (`gpt-3.5-turbo-1106`)

[6] We remark that using UMBC does not affect the comparability between our proposed method and contemporary methods. For example, the training of RoBERTa involved 160GB of text (Liu et al., 2019), effectively embedding a large part of this text into its weights.

[7] Details on the sampled entities are in Appendix B.

in Figure 3. We use the SRM to encode the rules and the sentences in the current batch. Then, we compute the cosine similarity between every rule and every sentence. Our training objective is then to maximize the similarity scores of matching pairs, found along the diagonal of this matrix. Simultaneously, we minimize the scores of non-matching pairs, which constitute the off-diagonal elements. We include examples of sentences, rules, and their resulting similarities in Appendix A.

4 Experiments

4.1 Experimental Setup

We evaluate our proposed method on Few-Shot TACRED (Sabo et al., 2021), a few-shot variant of the TACRED dataset (Zhang et al., 2017) and on a few-shot variant of the NYT29 dataset (Riedel et al., 2010; Takanobu et al., 2019; Alam et al., 2024). In few-shot settings, the training and testing relation labels are disjoint. We have access to a background training set for tuning the model, but we emphasize that our proposed method does not use it. Each test sentence is accompanied by 1 (1-shot) or 5 (5-shot) support sentences.[8]

We use RoBERTa-large (Liu et al., 2019) for our semantic matching component. Similar to CLIP (Radford et al., 2021), we use one model for encoding the rule and one model for encoding the sentence. We generate rules from the support sentences in each dataset. We use CoreNLP (Manning et al., 2014) to obtain the underlying syntactic structure for rule construction.

At prediction time, we use the proposed method in three ablative configurations: (1) Simply apply the resulting rules in a binary matching fashion, i.e., no SRM (**Hard-matching Rules**); (2) Use the semantic rule matching module to compute a similarity score between each rule and each sentence, interpreting a similarity above a threshold t as a match[9] (**Soft-matching Rules**); (3) A combination of (1) and (2), where we first attempt to apply the rules in a typical binary match/no match way (i.e., "hard" matching), and if no rule matches we fall back to the semantic rule matching component (i.e. "soft" matching). We call this approach **Hybrid**.

4.2 Baselines

We compare our proposed approach with one strong unsupervised baseline and several state-of-the-art supervised approaches from previous work.

Unsupervised Baseline: Similar to the baseline introduced in (Vacareanu et al., 2022b), this baseline utilizes entity types from query and support sentences for classification, defaulting to no_relation if no matching types are found.

Sentence-Pair: Employs a transformer-based model to classify concatenated query and support sentences (Gao et al., 2019). We reimplemented this baseline using sentence transformers (Reimers and Gurevych, 2019).

MNAV (Sabo et al., 2021): A transformer-based relation classifier is trained on a background set to align vector representations for sentences with identical relations, including multiple vectors for the no_relation class. During testing, it calculates similarity scores between the test sentence and both the no_relation vectors and support sentence vectors for each relation. For multiple support sentences of the same relation, it uses an averaged vector representation. The final prediction corresponds to the relation with the highest similarity score.

OdinSynth (Vacareanu et al., 2022b): Utilizes transformer-based rule synthesis from support sentences, predicting the relation with the most rule matches, or no_relation if there are none.

4.3 Main Results

We present our main results in Tables 1 and 2 for the standard 1-shot and 5-shot settings on the two datasets. Additionally, we differentiate between methods using background training datasets from the ones that do not (i.e., are Zero-Shot).[10], [11]

We concentrate our discussion on comparing between contemporary rule-based methods (OdinSynth) and strong neural-based methods (MNAV). We draw the following observations. First, compared to MNAV, the state-of-the-art neural-based method on Few-Shot TACRED, our proposed approach outperforms it in three out of the four settings investigated. For example, in the 1-shot case

[8] We provide additional details of the two datasets we use in Appendix J and details on the hyperparameters and hardware in Appendix F.

[9] We tune the threshold on the development partition of each dataset; we do not train on any data from the datasets.

[10] By zero-shot we mean methods that do not use human-annotated examples for training.

[11] An early iteration of the proposed method was included in (Alam et al., 2024). The results in this work are higher due to minor changes in the surface rules. In particular, in this work we represent lexical information using directly the string, where in the previous one we used a more verbose rule syntax such as word=string.

Model	5-way 1-shot			5-way 5-shot			Uses Bacgkround Data
	P	R	F1	P	R	F1	
Unsupervised Baseline	5.70 ± 0.10	91.02 ± 0.65	10.73 ± 0.18	5.65 ± 0.11	95.56 ± 0.70	10.67 ± 0.20	No
Sentence-Pair (not fine-tuned)	3.9 ± 0.21	5.21 ± 0.31	4.45 ± 0.24	2.76 ± 0.16	8.79 ± 0.58	4.2 ± 0.25	No
Sentence-Pair (fine-tuned)	6.89 ± 0.33	28.56 ± 1.67	11.10 ± 0.55	14.94 ± 0.26	24.03 ± 0.32	18.42 ± 0.16	Yes
MNAV (reported)	-	-	12.39 ± 1.01	-	-	30.04 ± 1.92	Yes
MNAV (re-run by us)	15.11 ± 0.46	8.47 ± 0.31	10.85 ± 0.29	24.48 ± 1.02	32.00 ± 1.07	27.73 ± 0.94	Yes
Odinsynth	23.48 ± 1.46	11.46 ± 1.02	15.40 ± 1.21	29.77 ± 0.83	20.34 ± 0.53	24.16 ± 0.44	No
Hard-matching Rules (ours)	51.35 ± 6.53	2.94 ± 0.48	5.56 ± 0.90	45.94 ± 5.31	10.81 ± 1.23	17.50 ± 1.98	No
Soft-matching Rules (ours)	37.22 ± 1.04	18.21 ± 0.62	**24.45** ± **0.72**	47.73 ± 2.23	35.52 ± 1.88	**40.71** ± **1.83**	No
Hybrid (ours)	35.91 ± 0.97	18.24 ± 0.62	24.19 ± 0.73	42.77 ± 1.88	36.53 ± 1.83	39.38 ± 1.57	No

Table 1: The results for the 5-way 1-shot and 5-way 5-shot settings on the test partition of the Few-Shot TACRED dataset. We split the table into 4 blocks as follows: (1) a strong unsupervised baseline where the classification is performed based on the types of the entities, (2) state-of-the-art neural methods, (3) rule synthesis using transformer networks, and (4) our proposed method. Our proposed method outperforms previous state-of-the-art methods on both 1-shot and 5-shot splits.

of Few-Shot TACRED, our proposed method improves upon MNAV by over 12 F1 points (approximately 100% relative improvement), despite not being trained with any human-annotated data or with any TACRED-specific data. We remark that MNAV outperforms our proposed approach in the 1-shot case on few-shot NYT29. NYT29 was annotated using distant supervision, which often results in shallow, context-free patterns. Our analysis indicates that MNAV, due to its training approach, may be effectively capturing these simple entity patterns. For example, for a sentence such as "Barack Obama was born in Honolulu .", we hypothesize that MNAV might superficially link (Barack Obama, Honolulu) to the relation "born in", irrespective of the context. Consequently, MNAV could mistakenly assign the same relation to a contextually different sentence like "Barack Obama went to high school in Honolulu", where the entities remain the same but the relation differs. We manually checked the top ten most popular entities from the support sentences and from the test sentences and observed that all have corresponding Wikipedia pages (i.e., they are very frequent), further supporting our hypothesis.

Second, our hybrid method largely surpasses Odinsynth, the leading rule-based approach on Few-Shot TACRED, in both 1-shot and 5-shot scenarios. This validates the hypothesis that combining a neural network with traditional rule-based approaches outperforms rule-only methods. The improved performance of our method does not sacrifice precision; it significantly surpasses Odinsynth in both precision and recall. This conclusion also applies to the few-shot variant of NYT29.

All in all, our proposed method obtains state-of-the-art performance *despite not being trained on any of the human-annotated examples from the respective training datasets*.

4.4 Results on the Full Testing Partition

We show the results of our proposed method on the complete test partition of the original TACRED dataset in Table 3. We compare against the method of Sainz et al. (2021), which casts the relation classification task as an entailment problem, resulting in a zero-shot relation classifier. We observe that our proposed method is either close in performance or outperforming the method proposed by Sainz et al. (2021). The results showcase that rules, when paired with neural networks, are competitive with purely neural network approaches, maintaining the high precision of the former and the high expressivity of the latter. Interestingly, the hybrid model has stable performance with or without threshold tuning.

4.5 Ablation Analysis

Next, we analyze the contributions of each key component in our proposed method. We show the results of the ablation study in Table 4. The three components that we analyze are:

(i) The pre-processing of our training dataset, where we filter out duplicates and sub-sample very frequent rules and entities.

(ii) The data augmentation, where we randomly replace the entities in the rule and in the sentence with synonyms. For example, a rule such as `[ne=per]+ <nsubj founded >nmod_in [ne=org]+` becomes `[ne=human]+ <nsubj founded >nmod_in [ne=company]+`. Similar augmentation are performed to sentences as well, where the named

Model	5-way 1-shot			5-way 5-shot			Uses Background Data
	P	R	F1	P	R	F1	
Unsupervised Baseline	11.60 ± 0.18	40.34 ± 0.54	18.03 ± 0.26	11.70 ± 0.25	40.65 ± 0.45	18.17 ± 0.34	No
Sentence-Pair (not fine-tuned)	10.61 ± 0.32	12.39 ± 0.41	11.43 ± 0.35	15.81 ± 0.94	5.41 ± 0.25	8.06 ± 0.39	No
Sentence-Pair (fine-tuned)	38.09 ± 2.42	7.4 ± 0.42	12.4 ± 0.71	36.48 ± 1.37	16.02 ± 0.41	22.26 ± 0.62	Yes
MNAV	25.08 ± 0.73	34.37 ± 0.87	**29.00** ± **0.80**	33.24 ± 1.06	15.47 ± 0.38	21.12 ± 0.55	Yes
OdinSynth	30.07 ± 0.93	9.42 ± 0.31	14.34 ± 0.46	21.61 ± 0.61	17.98 ± 0.45	19.63 ± 0.51	No
Hard-matching Rules (ours)	77.47 ± 1.53	1.53 ± 0.13	3.01 ± 0.25	80.49 ± 1.73	3.40 ± 0.12	6.52 ± 0.23	No
Soft-matching Rules (ours)	20.80 ± 0.38	12.27 ± 0.39	15.44 ± 0.40	24.50 ± 0.83	16.67 ± 0.49	19.84 ± 0.59	No
Hybrid (ours)	22.23 ± 0.47	13.45 ± 0.38	16.76 ± 0.41	27.29 ± 0.77	19.52 ± 0.49	**22.76** ± **0.56**	No

Table 2: The results for the 5-way 1-shot and 5-way 5-shot settings on the test partition of the Few-Shot NYT29 dataset. We split the table into 4 blocks as follows: (1) a strong unsupervised baseline where the classification is performed based on the types of the entities, (2) state-of-the-art neural methods, (3) rule synthesis using transformer networks, and (4) our proposed method. Our proposed method obtains the best performance in the 5-shot case, outperforming neural-based methods trained on the background training data.

	P	R	F1
Sainz et al. (2021)	58.5	53.1	55.6
Soft-Matching Rules (Ours)	70.2	39.0	50.1
Hybrid (Ours)	70.5	45.3	55.1
Sainz et al. (2021)	32.8	75.5	45.7
Soft Matching Rules (ours)	59.4	37.9	46.3
Hybrid (ours)	63.4	49.6	55.7

Table 3: Results on the full testing partition of TACRED. We compare our proposed approach and that of Sainz et al. (2021), which casts the relation classification problem as an NLI problem. We split the results into two blocks. Top: the threshold was tuned on 1% of the development partition; Bottom: the threshold was set to 0.5 without tuning.

entity in the marker (Zhou and Chen, 2021) is changed with its synonyms.

(iii) The inclusion of paraphrases. For example, a sentence such as *Bill Gates founded Microsoft* can be automatically paraphrased into *Bill Gates is the founder of Microsoft* using an LLM without losing any semantic information.

The analysis in Table 4 indicates that all three components contribute to the final performance, to varying degrees. First, our findings suggest that the data pre-processing contributes the most to the final performance, suggesting that the quality and structure of the input data play a crucial role in preparing the model to accurately handle the complexities of relation classification tasks. Second, the decline observed in the "No paraphrases" setting suggests that the inclusion of paraphrases encourages the model to learn less obvious semantic variations. Third, the rule and sentence augmentation appear to have the lowest impact. We argue that this is because both datasets that we use, the few-shot variants of TACRED and NYT29, contain the same common named entities, such as `person` and `organization`. These entities were seen during training, due to their prevalence. We hypothesize that this augmentation shines when the named entities used in the rules are not seen during training. We leave this exploration to future work. We include the corresponding results on Few-Shot NYT29 in Appendix H.

4.6 Are Soft Matching Rules still Pliable?

One key advantage of rules is that they are pliable (see Footnote 3) and modular. This means that a domain expert is able to modify the model effectively without risking introducing unknown and undesirable behavior (Sculley et al., 2015).

We analyze the degree to which interventions on the resulting rules can improve the final performance. We choose the relation `org:parents` from the development set, as it is a relation relatively well represented in the dataset and one where our model obtains a lower F1 score. We design the following experiment: two experts have access to the syntactic rules associated with the support sentences from the development partition of the Few-Shot TACRED. They have up to two hours to improve the rule set and the following operations:

ADD Rule: Adds a new rule, available to every episode. This operation simulates the practical example where practitioners aim to incorporate new knowledge to the model for use during inference.

DELETE Rule: For a given support sentence with the relation `org:parents` in a given episode, the model will not have access to the rule generated on that support sentence.

MODIFY Rule: This operation modifies a given rule. This modification will only be visible in the

		5-way 1-shot			5-way 5-shots		
		P	R	F1	P	R	F1
Model Type	Ablation						
Hybrid	Original	55.67 ± 3.75	32.19 ± 1.26	40.78 ± 1.99	55.04 ± 1.47	50.93 ± 1.94	52.90 ± 1.67
	No Paraphrases	42.88 ± 3.70	27.53 ± 1.38	33.52 ± 2.10	43.84 ± 2.14	51.28 ± 2.63	47.25 ± 2.21
	No data pre-processing	43.00 ± 3.21	22.38 ± 1.82	29.43 ± 2.25	48.16 ± 2.09	44.44 ± 2.89	46.22 ± 2.49
	No Rule/Sentence Augmentation	49.13 ± 3.55	32.77 ± 1.37	39.31 ± 2.11	47.63 ± 1.85	53.36 ± 2.16	50.33 ± 1.96
SoftRules	Original	56.81 ± 3.94	31.70 ± 1.43	40.68 ± 2.17	58.94 ± 1.79	49.60 ± 2.08	53.87 ± 1.96
	No Paraphrases	43.39 ± 3.96	27.10 ± 1.53	33.35 ± 2.29	45.93 ± 2.18	50.59 ± 2.88	48.14 ± 2.40
	No data pre-processing	43.50 ± 3.77	21.92 ± 2.06	29.14 ± 2.62	51.20 ± 1.85	43.15 ± 2.72	46.83 ± 2.37
	No Rule/Sentence Augmentation	49.95 ± 3.71	32.34 ± 1.54	39.25 ± 2.27	50.14 ± 1.94	51.98 ± 2.42	51.04 ± 2.16

Table 4: Ablation results for the 5-way 1-shot and 5-way 5-shot on TACRED's few-shot development partition. Each ablation condition is tested independently, with only one modification applied compared to the Original model.

episodes for which this particular rule appears.

We show examples of the operations and statistics in Appendix E. We show our results in Table 5. We detail two sets of results, showcasing the adaptability and effectiveness of our proposed method in relation classification. The first set is based on expert rule modifications without altering the classification threshold. The second set, in contrast, involves an increase in the threshold specifically for the `org:parents` rules, motivated by the greater average similarity seen with more general rules (created by the human annotators) compared to the lower alignment of highly specific rules (generated automatically from support sentences). For instance, rules synthesized from support sentences often yield highly specific constructs, such as `[ne=org]+ <nmod_from taken >conj_and operating >nmod_under brandname >compound [ne=org]+`. Such rules typically align poorly with the majority of sentences, attracting lower similarity scores. In contrast, the introduction of more general rules, e.g.: `[ne=org]+ >appos subsidiary >nmod_of [ne=org]+`, enhances rule-to-sentence similarity. This observed increase in average similarity was not accounted for with the original, unchanged classification threshold. To address this, we conducted a second set of experiments where the threshold was selectively increased by 0.1, but only for the `org:parents` relation.

We observe a consistent performance increase across both expert interventions and both threshold scenarios. With the classification threshold held constant, expert modifications led to an improvement of approximately 4 F1 points, a relative increase of about 25%. When the threshold for the `org:parents` relation was raised, the performance gains were even more pronounced, exceeding 15 F1 points and representing a relative increase of around 100%. Notably, these enhancements did not adversely affect the performance on other relations.

Model	Original threshold	Stricter threshold
Original	15.57 ± 1.39	15.57 ± 1.39
Expert 1	19.42 ± 0.65	31.78 ± 2.18
Expert 2	19.77 ± 1.08	34.03 ± 1.91

Table 5: F1 scores for the `org:parents` relation after two domain experts individually modified the corresponding rules. We compare scores before and after these changes, in two settings: (i) same threshold, (ii) stricter threshold.

5 Conclusion

We introduced a novel neuro-symbolic approach for relation extraction that combines the better generalization of neural networks with the explainability and pliability of rules. Our method first attempts to match the rule in a typical binary match/no match way. When a rule does not match, our approach then semantically matches it over text using a semantic matching component, which is contrastively trained without any human-annotated training data, akin to an LLM for rules.

We evaluated our model on two challenging few-shot datasets: Few-Shot TACRED (Sabo et al., 2021) and a few-shot variant of NYT29 (Alam et al., 2024). We showed that our method achieves strong performance, outperforming state-of-the-art supervised methods in three out of the four settings we investigated. Moreover, we empirically validated that our proposed method retains the pliability of rule-based methods, i.e., where humans can refine the underlying classification rules to noticeably increase the final performance. Notably, the resulting model is relatively small, i.e., it consists of an encoder of approximately 350M parameters, which makes it considerably more efficient than a decoder-based LLM.

Acknowledgments

This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under the ASKEM program. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

Limitations

We evaluate our proposed approach only for the English language, where high-quality syntactic parsers are available, and relation classification, where most relations to be learned can be well covered by syntactic patterns. Nevertheless, thanks to efforts such as Universal Dependencies (Nivre et al., 2020), high-quality parsing data is available to a large number of languages.

In general, rules seem to perform best for closed-world scenarios common to information extraction tasks. It is not immediately obvious how well rules (even with the proposed "soft" match) would port to more open-ended tasks such as question answering.

Ethics Statement

Our approach uses pre-trained language models as the backbone of our soft matching component. Therefore this work shares many of the same ethical issues such as social biases or perpetuating stereotypes (Weidinger et al., 2021). Our work attempts to improve upon these by using a sieve architecture, where the first step is to apply the rule as in a typical rule-based model. This step is completely transparent to the practitioner, as they can add, modify, or delete rules. In the second step, we use a transformer-based model to semantically match the rules with sentences where an exact match is not possible. Our pliability experiment showed that our approach retains the benefits of typical rule-based models, as the experts are able to intervene on the rules, and, thus, correct any potential biases that may exist. However, we acknowledge that more work is necessary to better understand the transparency of the semantic-matching component. In our work, the rule acquisition strategy was applied over patterns that hold between two entities, where both appear as contiguous spans of text. We did not explore how our rule acquisition strategy could be expanded to handle more complex semantic relationships, such as n-ary relations, discontinuous entities, or overlapping entities.

References

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

Fahmida Alam, Md Asiful Islam, Robert Vacareanu, and Mihai Surdeanu. 2024. Towards realistic few-shot relation extraction: A new meta dataset and evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Torino, Italy. European Language Resources Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NIPS*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard S. Zemel. 2019. Understanding the origins of bias in word embeddings. In *ICML*.

Laura Chiticariu, Yoav Goldberg, Gus Hahn-Powell, Clayton T. Morrison, Aakanksha Naik, Rebecca Sharp, Mihai Surdeanu, Marco Valenzuela-Escárcega, and Enrique Noriega-Atala, editors. 2022. *Proceedings of the First Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*. International Conference on Computational Linguistics, Gyeongju, Republic of Korea.

Marina Danilevsky, Shipi Dhanorkar, Yunyao Li, Lucian Popa, Kun Qian, and Anbang Xu. 2021. Explainability for natural language processing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp; Data Mining*, KDD '21, page 4033–4034, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. In *EMNLP/IJCNLP*.

Bryce Goodman and Seth Flaxman. 2016. European union regulations on algorithmic decision-making and a "right to explanation". *AI Mag.*, 38:50–57.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

S. Gupta and Christopher D. Manning. 2014. Improved pattern learning for bootstrapped entity extraction. In *CoNLL*.

Lushan Han, Abhay Lokesh Kashyap, Timothy W. Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In **SEMEVAL*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. *ArXiv*, abs/1905.02175.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *ArXiv*, abs/2004.03685.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. *CoRR*, abs/1902.10186.

Mandar Joshi, 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.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *EMNLP*.

Dekang Lin and Patrick Pantel. 2001. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Scott M Lundberg 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*, pages 4765–4774. Curran Associates, Inc.

Christopher D. Manning. 2015. Computational linguistics and deep learning. *Computational Linguistics*, 41:701–707.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Annual Meeting of the Association for Computational Linguistics*.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Ani Saxena, Kristina Lerman, and A. G. Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54:1 – 35.

Madhav Nimishakavi, Uday Singh Saini, and Partha Talukdar. 2016. Relation schema induction using tensor factorization with side information. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 414–423, Austin, Texas. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Sebastian Riedel, 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.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Ellen Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In *AAAI*.

Ellen Riloff. 1996a. Automatically generating extraction patterns from untagged text. In *AAAI/IAAI, Vol. 2*.

Ellen Riloff. 1996b. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Ellen Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP*.

Ofer Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. Revisiting few-shot relation classification: Evaluation data and classification schemes. *Transactions of the Association for Computational Linguistics*, 9:691–706.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. In *NIPS*.

Lloyd S. Shapley. 1988. A value for n-person games.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *ArXiv*, abs/1703.01365.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7072–7079.

Zheng Tang and Mihai Surdeanu. 2023. It takes two flints to make a fire: Multitask learning of neural relation and explanation classifiers. *Computational Linguistics*. Accepted on 2022.

Erico Tjoa and Cuntai Guan. 2019. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4793–4813.

Robert Vacareanu, Dane Bell, and Mihai Surdeanu. 2022a. Patternrank: Jointly ranking patterns and extractions for relation extraction using graph-based algorithms. In *Proceedings of the First Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 1–10.

Robert Vacareanu, Marco A. Valenzuela-Escarcega, George C. G. Barbosa, Rebecca Sharp, and Mihai Surdeanu. 2022b. From examples to rules: Neural guided rule synthesis for information extraction.

Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, and Dane Bell. 2020. Odinson: A fast rule-based information extraction framework. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2183–2191, Marseille, France. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Laura Weidinger, John F. J. Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zachary Kenton, Sande Minnich Brown, William T. Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *ArXiv*, abs/2112.04359.

Ke Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Conference on Empirical Methods in Natural Language Processing*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *North American Chapter of the Association for Computational Linguistics*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for entity and relation extraction. *ArXiv*, abs/2010.12812.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373*.

Wenxuan Zhou, 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. *The Web Conference '20 in arXiv: Computation and Language*.

A Qualitative Examples

We provide qualitative examples for the behavior of our proposed semantic rule matcher (SRM) in Table 6.

We split the examples into 7 distinct blocks to facilitate the highlight of different behaviors.

(1) In the first block we highlight how the SRM is able to overlook superficial differences (i.e. *daughter* in text, *son* in rule) and assign a high similarity score. We want to emphasize that a traditional rule-based engine will not be able to match the rule on the given sentence.

(2) Similar to block **(1)**, the SRM is capable of understanding that *graduated from* is similar to *got his degree from*.

(3, 4) We use these blocks to highlight to give a similarity reference for the behavior we want to highlight next. Here, the SRM assigns a high score, as expected. We want to highlight that this rule, in this form, is generic enough to match relations such as `neighborhood of`, `city in country`, among others.

(5, 6, 7, 8) In these blocks we highlight a behavior that is present in the resulting model, despite never being trained for it. Here, we replace the typical named entities with their most fine-grained version: lexicalized entities. The underlying idea is to overcome the lack of expressiveness from the NER model and provide an additional source of signal, from the underlying entities. In block **(5)** we replace the `location` entity types with *Wynwood* and *Miami*.[12] We want to highlight that this rule correctly obtains a higher similarity with the sentence in block **(5)** than with the sentence in block **(6)**, where the entities in the sentence are *Athens* and *Greece*. We remark that the underlying relation in **(5)** is, in the most specific form, *neighborhood of*, while in **(6)** it is *city in country*. Similarly, we provide the alternative rule and the corresponding similarities in blocks **(7, 8)**. We emphasize that the SRM component has not been explicitly trained for this behavior. We leverage this behavior during evaluation for the cases where both entity types were identical (e.g., `[ne=location]+ <appos [ne=location]+`)

B Entity Types in the Training Dataset

We used the following entity type pairs when constructing our dataset consisting of rule and sentence pairs: `[(ORGANIZATION, ORGANIZATION), (ORGANIZATION, PERSON), (ORGANIZATION, COUNTRY), (ORGANIZATION, CITY), (ORGANIZATION, STATE_OR_PROVINCE), (ORGANIZATION, IDEOLOGY), (ORGANIZATION, LOCATION), (ORGANIZATION, URL), (ORGANIZATION, EMAIL), (PERSON, ORGANIZATION), (PERSON, CAUSE_OF_DEATH), (PERSON, NATIONALITY), (PERSON, COUNTRY), (PERSON, LOCATION), (PERSON, CITY), (PERSON, STATE_OR_PROVINCE), (PERSON, IDEOLOGY), (PERSON, CRIMINAL_CHARGE),`

[12] *Wynwood* is a neighborhood in Miami.

1	**Sentence**	Sofia Coppola , daughter of Francis Ford Coppola , is one of the few to succeed in doing so : her film" Lost in Translation" won her a screenplay Oscar
	Rule	[ne=person]+ >appos son >nmod_of [ne=person]+
	Similarity	0.83
2	**Sentence**	John got his degree from Oxford .
	Rule	[ne=person]+ graduated from [ne=organization]+
	Similarity	0.82
3	**Sentence**	John moved to SoHo , Manhattan .
	Rule	[ne=location]+ <appos [ne=location]+
	Similarity	0.68
4	**Sentence**	John moved to Athens , Greece .
	Rule	[ne=location]+ <appos [ne=location]+
	Similarity	0.69
5	**Sentence**	John moved to SoHo , Manhattan .
	Rule	[ne=Wynwood]+ <appos [ne=Miami]+
	Similarity	0.29
6	**Sentence**	John moved to Athens , Greece .
	Rule	[ne=Wynwood]+ <appos [ne=Miami]+
	Similarity	0.21
7	**Sentence**	John moved to SoHo , Manhattan .
	Rule	[ne=Berlin]+ <appos [ne=Germany]+
	Similarity	0.24
8	**Sentence**	John moved to Athens , Greece .
	Rule	[ne=Berlin]+ <appos [ne=Germany]+
	Similarity	0.37

Table 6: Qualitative examples of our semantic rule matcher, split into 7 blocks to highlight different behaviors.

`(PERSON, RELIGION), (PERSON, EMAIL), (PERSON, MONEY), (TITLE, PERSON), (CITY, ORGANIZATION), (CITY, STATE_OR_PROVINCE), (PERSON, PERSON), (PERSON, TITLE), (PERSON, NUMBER), (COUNTRY, ORGANIZATION), (ORGANIZATION, COUNTRY), (NATIONALITY, PERSON), (PERSON, DATE), (COUNTRY, PERSON), (CITY, PERSON), (STATE_OR_PROVINCE, PERSON), (ORGANIZATION, DATE), (NUMBER, PERSON), (DATE, PERSON), (ORGANIZATION, NUMBER), (CAUSE_OF_DEATH, PERSON), (DATE, ORGANIZATION), (LOCATION, ORGANIZATION)]`.

C Entity Types Synonyms

In the training phase of the proposed Semantic Rule Matcher, we randomly replaced the entity types in the rules and in the sentences with synonyms, to encourage generalization beyond superficial clues from the entity types. We present the synonyms we used in Table 7.

D Paraphrasing Prompt

We show the prompt we used to generate paraphrases below. We dynamically set the number of paraphrases to generate based on the text length, ranging from 2 to 5. The intuition is that short sentences admit a lower number of paraphrases. We only keep the paraphrases where the entities of interest are preserved. Additionally, if the entities of interest appear more than one time in the paraphrase, we discard the resulting paraphrase. Following this process, we keep over 80% of the paraphrases that are generated.

> Please generate a number of {how many} paraphrases for the following sentence. Please ensure the meaning and the message stays the same and these two entities are preserved in your generations: "{entity 1}", "{entity 2}".
> Please be concise.
> ‘‘‘
> {text}
> ‘‘‘
> 1.

E Pliability Experiment

We show the number of operations employed by each Expert in Table 8.

We provide examples of each operation below.

(i) ADD: This operation adds a new rule which will be available to every episode. This simulates the practical example where practitioners aim to incorporate new knowledge to the model to be used at inference time.

For example, one annotator added the following rule `[ne=org]+ >appos subsidiary >nmod_of [ne=org]+`. This rule will match sentences like: “Google, a subsidiary of Alphabet, announced a new acquisition.”.

(ii) MODIFY: This operation modifies a given rule. This modification will only be visible in the episodes for which this particular rule appears. This simulates the scenario where the resulting rule has slight inaccuracies.

For example, one annotator changed from `[ne=org]+ <nsubj said >ccomp buy >nmod_for [ne=org]+` to `[ne=org]+ <nsubj said >ccomp buy >dobj [ne=org]+`. This changed rule will match sentences like: “Google said it will buy YouTube.”.

(iii) DELETE: This operation removes the given rule, such that the model will not have access to it. This simulates the scenario where the resulting rule is too noisy to be useful.

For example, one annotator removed the following rule: `[ne=org]+ <nsubj sought >conj_but opted >nmod_for batteries >nmod_from [ne=org]+`

F Hyperparameters

We experiment with multiple settings where we vary the learning rate, the projection dimensions, and the weight decay. This search involved under 20 runs. We show our hyperparameters in Table 9. We use the development partition of Few-Shot TACRED for early stopping.

We ran all our experiments on a system with A100 80 GB GPUs. We used approximately 3 days worth of a single A100 GPU time.

G Rule Augmentation

In the following, we detail how a rule augmentation looks like. We augment rules by replacing the original entities with their synonyms. Our motivation for this is to encourage the rule matcher to look beyond lexical similarities and to judge, instead, the semantic similarity of the two entities (i.e., per should be close to human and different from company). We ablate this choice in Table 3, empirically

Entity	Synonyms
organization	org, company, firm, corporation, enterprise
date	a specific date
person	per, human, human being, individual
number	digits
title	designation, formal designation
duration	time period
misc	miscellaneous
country	nation, state, territory
location	place, area, geographic area, loc
cause_of_death	date of demise, cause of death, death cause, mortal cause
city	municipality, town, populated urban area
nationality	citizenship
ordinal	ranking
state_or_province	region, territorial division within a country
percent	percentage
money	currency
set	collection, group of items
ideology	doctrine, system of ideas and ideals
criminal_charge	accusation, formal allegation
time	period, time period
religion	belief, faith, spiritual belief, worshipper
url	web address
email	electronic mail
handle	username, personal identifier

Table 7: Entity type synonyms used to augment the rules and sentences.

	Operations		
	ADD	MODIFY	DELETE
Expert 1	12	6	16
Expert 2	12	3	28

Table 8: The number of operations performed by each expert during the intervention experiment.

Rule Encoder LR	3e-5
Sentence Encoder LR	1e-5
Projections LR	1e-4
Logit Scale LR	3e-4
Train Batch Size	512
Gradient Clip Val	5.0
Dropout	0.1
Projection Dims	384
Weight Decay	0.001

Table 9: The hyperparameters we used for training the Semantic Rule Matcher.

finding that this brings the largest improvement (i.e., an increase of over 11 F1 points, from 28.91 to 40.50). We included an example in Table 10.

H Ablation Study (extended)

We complement the ablation study from Table 4, which was done on Few-Shot TACRED with an ablation study over the few-shot variant of NYT29. We show our results in Table 11. We remark that the same conclusions hold on both datasets.

I Paraphrase Quality

In the following, we analyze the quality of the paraphrases generated by the large language model. Specifically, we used ChatGPT together with the prompt described in Appendix D. We conducted a manual analysis of over 50 randomly sampled sentences. We observed that all paraphrases correctly preserved the underlying relation. We will release this dataset. We added two examples in Table 12.

J Dataset Details

We provide additional details on the two datasets we used: Few-Shot TACRED (Sabo et al., 2021) and Few-Shot NYT29 (Alam et al., 2024).

TACRED has 42 classes (41 relations, 1 `no_relation` class) distributed across 100,000 examples. The class `no_relation` has the most number of examples, accounting for approximately 80% of the total data. The number of examples per relation follows an exponential distribution, ranging from approximately 4000 for the relation `per:title` to 33 for the relation `org:dissolved`.

NYT29 has 29 relations, distributed across 90,000 examples. This dataset does not have a strict `no_relation` class. The number of examples per relation follows an exponential distribution, ranging from approximately 32,000 for the relation `/location/location/contains` to 10 for /business/company_advisor/companies_advised.

There is no strict overlap between any relations from TACRED and NYT29 either from the dev partition or from the test partition. Nevertheless, we remark that there are similar relations. For example, the relation `per:city_of_death` appears in the test partition of few-shot TACRED and `/people/deceased_person/place_of_death` appears in the test partition of few-shot NYT29.

K Per-Relation Performance

We present per-relation performance metrics for the Few-Shot TACRED dataset, with results for $K = 1$ in Table 13 and for $K = 5$ in Table 14.

Original Rule	`[ne=per]+ <nsubj founder >nmod_of [ne=org]+`
Modified Rule	`[ne=human]+ <nsubj founder >nmod_of [ne=company]+`

Table 10: Illustration of rule augmentation by substituting entity types: converting `per` to `human` and `org` to `company`.

		5-way 1-shot			5-way 5-shots		
Model Type	Ablation	P	R	F1	P	R	F1
Hybrid	Original	9.40 ± 0.55	31.48 ± 1.50	14.48 ± 0.77	10.47 ± 0.56	72.38 ± 1.99	18.29 ± 0.91
	No Paraphrases	8.59 ± 0.77	25.16 ± 1.89	12.80 ± 1.08	9.53 ± 0.53	61.77 ± 2.16	16.51 ± 0.86
	No data pre-processing	7.29 ± 0.53	15.20 ± 1.22	9.85 ± 0.73	9.29 ± 0.60	38.79 ± 1.74	14.99 ± 0.90
	No Rule/Sentence Augmentation	11.10 ± 0.60	26.38 ± 1.60	15.62 ± 0.78	12.49 ± 0.77	60.31 ± 1.97	20.70 ± 1.16
SoftRules	Original	9.40 ± 0.55	31.48 ± 1.50	14.48 ± 0.77	10.47 ± 0.56	72.38 ± 1.99	18.30 ± 0.91
	No Paraphrases	8.59 ± 0.77	25.16 ± 1.89	12.80 ± 1.08	9.53 ± 0.53	61.77 ± 2.16	16.52 ± 0.87
	No data pre-processing	7.29 ± 0.53	15.20 ± 1.22	9.85 ± 0.73	9.30 ± 0.61	38.79 ± 1.74	15.01 ± 0.91
	No Rule/Sentence Augmentation	11.10 ± 0.60	26.38 ± 1.60	15.62 ± 0.78	12.50 ± 0.77	60.31 ± 1.97	20.71 ± 1.17

Table 11: Ablation results on the 5-way 1-shot and 5-way 5-shot on the development partition of the few-shot NYT29 dataset. Each ablation condition is tested independently, with only one modification applied compared to the Original model.

Original	One year I served as research assistant to Wendell Bennett , a brilliant young anthropologist and the next year was the research assistant to Tom McCormick , an excellent , but inarticulate statistician
Paraphrase	After assisting anthropologist Wendell Bennett , I worked as a research assistant to Tom McCormick , a talented statistician who was not very articulate .
Original	In April 1915 , Sir John Nixon took command of British forces in Iraq and received orders to draw up plans for an advance on Baghdad .
Paraphrase	In April 1915 , Sir John Nixon was assigned to lead the British military in Iraq and was instructed to make plans for an assault on Baghdad .

Table 12: Two examples of paraphrases. We underline the entities involved.

Relation	P	R	F1
org:country_of_headquarters	45.13 ± 8.54	10.32 ± 1.97	16.80 ± 3.17
org:founded	42.52 ± 4.66	43.04 ± 8.39	42.41 ± 4.95
org:parents	21.37 ± 8.58	9.53 ± 2.63	13.14 ± 4.12
per:age	72.17 ± 3.54	52.82 ± 2.29	60.97 ± 2.37
per:alternate_names	3.45 ± 3.28	1.39 ± 1.29	1.98 ± 1.85
per:stateorprovince_of_death	62.32 ± 10.28	72.04 ± 3.49	66.43 ± 5.70

Table 13: Per-relation scores achieved by our **Hybrid** method on the development partition of the Few-Shot TACRED dataset for $K = 1$.

Relation	P	R	F1
org:country_of_headquarters	57.38 ± 10.44	21.13 ± 5.18	30.81 ± 6.85
org:founded	51.72 ± 8.34	70.69 ± 6.87	59.57 ± 7.37
org:parents	24.62 ± 4.63	19.08 ± 5.76	21.39 ± 5.15
per:age	71.32 ± 1.26	78.47 ± 2.72	74.72 ± 1.85
per:alternate_names	9.29 ± 3.11	9.24 ± 4.54	9.21 ± 3.81
per:stateorprovince_of_death	64.45 ± 7.59	92.30 ± 0.87	75.71 ± 5.21

Table 14: Per-relation scores achieved by our **Hybrid** method on the development partition of the Few-Shot TACRED dataset for $K = 1$.