Learning to Generate Rules for Realistic Few-Shot Relation Classification: An Encoder-Decoder Approach

Mayank Singh and **Eduardo Blanco**
University of Arizona, Tucson, Arizona, USA
{mayanks43, eduardoblanco}@arizona.edu

Abstract

We propose a neuro-symbolic approach for realistic few-shot relation classification via rules. Instead of building neural models to predict relations, we design them to output straightforward rules that can be used to extract relations. The rules are generated using custom T5-style Encoder-Decoder Language Models. Crucially, our rules are fully interpretable and pliable (i.e., humans can easily modify them to boost performance). Through a combination of rules generated by these models along with a very effective, novel baseline, we demonstrate a few-shot relation-classification performance that is comparable to or stronger than the state of the art on the Few-Shot TACRED and NYT29 benchmarks while increasing interpretability and maintaining pliability.

1 Introduction

In recent years, many data-driven approaches have been proposed for relation classification (RC). Most of them (e.g., Park and Kim, 2021; Lyu and Chen, 2021; Baldini Soares et al., 2019) require extensive training data similar to that in the test set. Using this data, these approaches have achieved high effectiveness in RC tasks. For example, Li et al. (2024) score 91.2% F1 on Re-TACRED (Stoica et al., 2021). Consequently, new data-lean formulations for RC have emerged, including few-shot (FS) RC tasks such as FewRel 1.0 (Han et al., 2018) and FewRel 2.0 (Gao et al., 2019), where there are very few training examples available for each relation in the test set.

However, these new datasets are unrealistic with respect to how relation classification is encountered in practice. A few of the unrealistic traits of these datasets include: a) equal distribution of instances for target relations, b) all test instances having a relation associated with them, and c) absence of common nouns and pronouns as entities. To address these issues, Sabo et al. (2021) propose a more realistic approach to few-shot relation classification (FS-RC) which eliminates these unrealistic traits. They also propose a method to convert supervised datasets to a realistic few-shot version and apply it to TACRED (Zhang et al., 2017), resulting in the FS-TACRED dataset. Later, Alam et al. (2024) follow a similar technique on the NYT29 dataset (Nayak and Ng, 2019; Takanobu et al., 2018) to create the FS-NYT29 dataset.

Many deep-learning approaches have been proposed for realistic FS-RC (Sabo et al., 2021; Lv et al., 2022). These approaches aim only to predict the relation that holds between two query entities, and it is often impossible to explain how

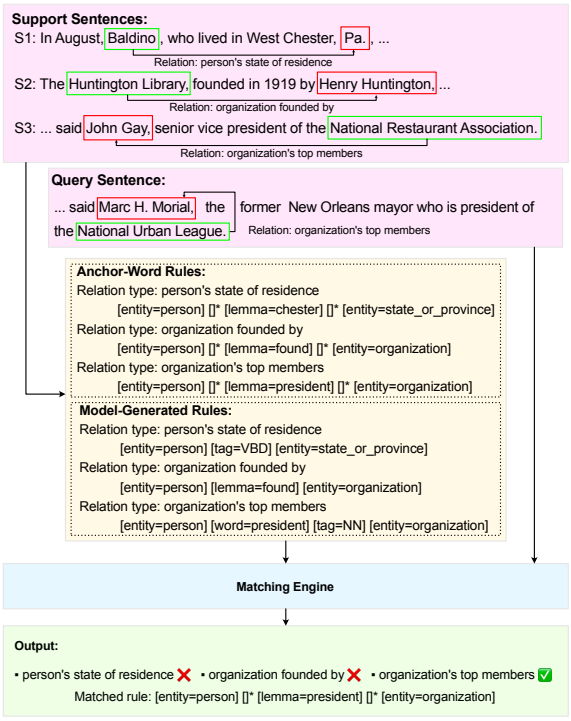

Figure 1: Our approach for few-shot relation classification. We generate two types of rules for each support sentence: Anchor-Word rules and Model-Generated rules. The relation whose corresponding rules match the query sentence is selected as the prediction.

Findings of the Association for Computational Linguistics: EACL 2024, pages 5907–5921
November 12-16, 2024 ©2024 Association for Computational Linguistics

they arrived at these predictions. A new line of work (Vacareanu et al., 2022a,b) aims to solve this task through rules. Rules are, by their very nature, interpretable; it's easy to understand why they make a particular prediction—they either match a given text or they do not. In addition, they are pliable (i.e., easily edited), making it straightforward to correct any erroneously generated rules.

Our work belongs to the latter line of work. We introduce a new approach to generating rules: Encoder-Decoder Language Models—more specifically, T5-style models (Raffel et al., 2019). Such neural models are used in many NLP tasks these days and have been found to be very effective. They also have the benefit of being able to be trained end-to-end. Note that the output of these models may not be interpretable. However, because we output rules instead of relations, using these rules for relation classification allows us to read the matched rules and understand the reasoning behind our predictions. Our models generate rules that outperform all previous work, aside from SoftRules (Vacareanu et al., 2024), in two out of four evaluated scenarios. Notably, our approach is more interpretable because we use exact matching with our rules.

We also propose a novel baseline that generates what we term Anchor-Word rules, which we later demonstrate to be highly effective for FS-TACRED. To create these rules, we first identify an anchor word in a sentence that represents the relation in question. We then complete each rule by connecting the anchor word with the two query entities through appropriate paths.

The main contributions of this paper are:

1. An effective baseline that generates Anchor-Word rules. This method surpasses all previous efforts on FS-TACRED, except for the state of the art.
2. The very first attempt at learning to generate rules for realistic FS-RC using Encoder-Decoder Language Models.
3. By combining Anchor-Word and Model-Generated rules, we achieve results comparable to the state of the art in the 1-shot scenario of FS-TACRED and outperform all prior methods in the 5-shot scenarios of both FS-TACRED and FS-NYT29.

Crucially, our approach is fully interpretable, as rules must match exactly to be considered a match, and pliable, as shown by an experiment where manual rule refinement triples performance.

2 Background

Terminology In relation classification, we are tasked with identifying the relation between two entities in a sentence. A relation instance consists of a subject entity, an object entity, and the relation they are connected by. For example, consider the sentence: "John Riccitiello steps into the role of CEO of Unity Technologies having served on the Board of Directors from November 2013." Here, "John Riccitiello" is the object entity and "Unity Technologies" is the subject entity. The relation described is `org:top_members/employees`, which means an organization's top members or employees.

Realistic FS-RC as proposed by Sabo et al. (2021) follows an N-way K-shot setup where the evaluation protocol consists of many episodes. Each episode consists of N target relations, K support sentences per target relation, and a variable number of query sentences. The query sentences are relation instances that need to be classified into (a) one of the N target relations or (b) the `no_relation` category, which indicates that none of the N target relations hold between the entities in the query sentence. The setup also provides a large number of background relations (and their examples) that are completely disjoint from the set of target relations. These relations can be used to construct a few-shot relation classifier.

FS-TACRED Sabo et al. (2021) propose a conversion logic that can transform any supervised RC dataset into an FS-RC format. When applied to the TACRED (Zhang et al., 2017) dataset—a large English RC dataset composed of sentences from public news articles—this logic generates the FS-TACRED dataset. There are two evaluation scenarios in FS-TACRED: 5-way 1-shot and 5-way 5-shot. As the names suggest, the 5-way 1-shot scenario includes 5 relations with 1 example per relation in an episode, while the 5-way 5-shot scenario contains 5 relations with 5 examples per relation. Both scenarios contain 3 query sentences per episode. We include examples in Appendix A.

The transformation also splits the original 41 relations of the TACRED dataset into train, dev, and test splits such that each relation is unique to its split. Table 1 depicts the total number of unique relations belonging to each split. It should be noted that the depicted numbers are one more than the actual unique relation count because the transformation also includes the `no_relation` category in each split.

	FS-TACRED			FS-NYT29		
	Train	Dev	Test	Train	Dev	Test
Number of relations	26	7	11	16	8	6
Number of relation instances	68,124	22,631	15,509	78,885	5,859	8,759
Number of relation instances (without `no_relation`)	8,163	633	804	56,620	190	2,031
Number of episodes	n/a	10,000	10,000	n/a	10,000	10,000
Percentage of `no_relation` queries	n/a	97.20	94.84	n/a	96.77	76.76
Percentage with at least 1 target relation query	n/a	8.16	14.76	n/a	9.40	54.58
Average number of tokens						
per sentence	34.42	31.79	35.00	38.49	41.93	38.11
between subject and object	6.43	8.45	7.03	9.21	11.64	9.86
in the shortest syntactic path	1.53	1.87	1.65	2.51	3.03	2.73

Table 1: Basic statistics of FS-TACRED and FS-NYT29. An episode includes (a) one or five support examples for five target relations and (b) three query sentences with two entities (subject and object). For many query sentences (∼95% in test set of FS-TACRED), none of the target relations hold between the entities. Syntactic paths are an abstract representation and shorten the distance between subject and object.

The transformation creates 10,000 episodes for each data split it is applied to. The logic can be applied to the train, dev, or test split of TACRED, but we are only concerned with evaluation on the dev or test split in this work. We use the relations and their corresponding instances from the training split as background relation data. To be noted is the percentage of query sentences with the `no_relation` category. As depicted, most query sentences are of this type (i.e., none of the five relations hold).

At the bottom of Table 1, we present the average number of tokens per sentence for each split, including counts between the subject and object entities. We observe similar numbers across splits. We also note the token count along the shortest syntactic path between these entities. These paths are used to create syntax rules, which, as we will show, yield better results due to their shorter lengths.

FS-NYT29 The NYT29 dataset (Nayak and Ng, 2019; Takanobu et al., 2018; Riedel et al., 2010) for relation classification was developed through distant supervision by aligning the New York Times corpus (Sandhaus, 2008) with Freebase (Bollacker et al., 2007) relations. Alam et al. (2024) apply the same FS-RC conversion technique as Sabo et al. (2021) to this dataset to create the FS-NYT29 dataset. The details for this dataset are also depicted in Table 1. Compared to FS-TACRED, FS-NYT29 has a lower percentage of query sentences labeled as `no_relation`.

Rules We use rules for relation classification. These rules are in the Odinson query language (Valenzuela-Escárcega et al., 2020). We use Odinson for two reasons. First, the language supports rules written for both the original token order in a sentence (surface tokens) as well as tokens on the syntactic paths in its dependency tree (syntax tokens). Second, the authors provide an efficient rule-matching engine: it finds matches in around 150 million sentences in under 3 seconds.

Here is a sample rule in this language for the sentence "he eats from the plate": "`[word=he] [tag=VBZ] [tag=IN] [tag=DT] [lemma=plate]`". To represent a word, we enclose a property of the word—its lemma, POS tag, entity type, or word-form—in square brackets. To represent any word, we use "`[]`". Standard regex wildcards are allowed (e.g., with "`*`" we can represent zero or more words satisfying the listed property). Some other example rules to match the above sentence include: "`[tag=PRP] [lemma=eat] [word=from] [lemma=the] [tag=NN]`" and "`[lemma=he] []* [word=plate]`".

As mentioned earlier, these rules can be written for surface tokens (surface rules) as well as syntax tokens (syntax rules). The rules in the previous paragraph are examples of surface rules. Odinson provides keywords for representing dependency edge types to write syntax rules. However, for the sake of simplicity and to avoid dealing with dependency edges of various types, we instead write syntax rules as surface rules over the tokens on the shortest syntactic path between the subject and object entities. For example, for the above sentence, the tokens on the shortest syntactic path between the first and last words can be written as: "he eats plate". Here is a syntax rule, which is formulated as a surface rule, for the running example: "`[word=he] [lemma=eat] [word=plate]`".

3 Related Work

Deep Learning Approaches Sentence-Pair (Gao et al., 2019) concatenates each support sentence with the query sentence sequentially and processes them through a BERT model (Devlin et al., 2019) to predict two values: the first value measures the semantic similarity between the sentences, and the second value quantifies their dissimilarity. The method then selects the relation with support sentences most similar to the query sentence as its prediction. The similarity score for the `no_relation` category is obtained by taking the minimum of the set of dissimilarity scores.

Sabo et al. (2021) propose two similar techniques: NAV and MNAV. In NAV (NOTA As Vectors), they embed query and support sentences into the same latent space. The cosine similarity between the query sentences and support sentences is calculated through their embeddings, and the relation corresponding to the support sentence with the highest similarity is chosen as the output. The core innovation in this technique is that NOTA (none of the above) is also represented as an embedding and treated like a relation. MNAV (or Multiple NOTA As Vectors) is similar to NAV, except multiple embeddings are used to represent NOTA.

CKPT (Lv et al., 2022) utilizes BERT to complete a prompt missing key words that indicate the predicted relation. For example, given "Paris is located in France", the prompt could be "Paris is the [MASK] of France" for the relation "A nation's capital." Additionally, it expands the vocabulary indicating each relation by leveraging external knowledge and outputs relations based on similarity.

Our approach outperforms these methods in three of four scenarios across the evaluated datasets and offers full interpretability and pliability.

Hybrid Approaches: Deep Learning and Rules OdinSynth (Vacareanu et al., 2022a) generates rules for a support relation instance using a branch-and-bound search through the rule space. This process is guided by a specialized BERT model. A target relation is chosen as the prediction for a test instance if it matches a rule for that relation. Although the approach presented here outperforms the OdinSynth rules, we utilize them as a training source for one of our models.

SoftRules (Vacareanu et al., 2024) presents a fuzzy semantic rule matcher—rules do not need to match a sentence exactly to indicate a match. For example, "`[entity=person] [word=founded] [entity=organization]`" will match both "Elon Musk founded Tesla" and "Elon Musk is the founder of Tesla", despite the latter not matching the rule exactly—the authors term this a soft rule and soft match. Matches are determined by a neural model using a specific threshold. This fuzzy matching approach makes these rules less interpretable than ours.

4 Generating Rules for Relation Classification

As mentioned before, in relation classification, we are tasked with finding the relation between two entities (subject and object) in a sentence. A rule that identifies a relation finds a path between the subject and object entities that is peculiar to that particular relation. If the path specified by a rule is found in a given sentence, we expect that such a relation exists in that sentence. In the following sections, we discuss the various ways[1] we generate rules for relation classification.

4.1 An Effective Baseline: Anchor-Word Rules

In this section, we discuss a style of rules which we call Anchor-Word rules. We find these rules very effective at FS-RC. They are based on the intuition that many relations are characterized by certain words—anchor words—around the two entities in question. Anchor words can appear before, after, or between the two entities. To identify the relation, the entities and anchor words have to be connected through specific paths. For example, for the relation `per:city_of_birth` (a person's city of birth), one possible rule could be: "`[entity=person] [lemma=be]? [lemma=bear] [tag=IN] [entity=location]`". In this case, the two entity types are person and location and they are connected by an anchor word that has the lemma 'bear.' In the rule, we have specified one way of going from the subject entity to the anchor word, and then to the object entity. It is possible there are other ways. For simplicity's sake and to prevent having to list all the possible ways, in this work, we allow any word in the path between the anchor words and the entities. Another possible Anchor-Word rule for the previous relation could be: "`[entity=person] []* [lemma=bear] []* [entity=location]`".

Identifying Anchor Words We follow a simple approach to find anchor words. For each word

[1] Code at: https://github.com/mayanks43/anchorT5.

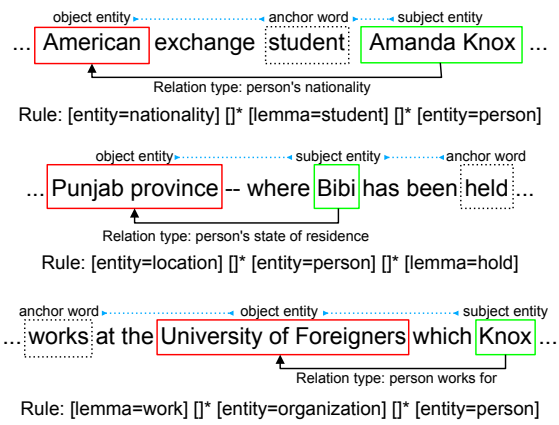

Figure 2: Procedure for identifying two paths forming an Anchor-Word rule depending on the relative position of the anchor word. Paths are defined either by surface (tokens around entities in the sentence) or by syntax (tokens in the shortest syntactic path between entities).

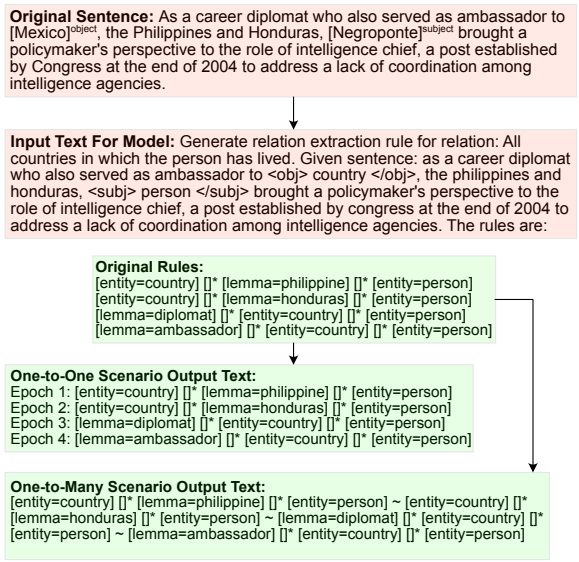

Figure 3: Procedure to prepare data for fine-tuning CodeT5plus. To get the input, we surround entities with tags and replace their contents with entity types. We experiment with two techniques to represent outputs: only one output per input sentence or multiple outputs concatenated together with a delimiter.

in the sentence, we calculate its cosine similarity to a text representing the relation using Sentence-BERT, a sentence similarity model (Reimers and Gurevych, 2019). Words exceeding a similarity threshold are chosen as anchor words. We use descriptions of the relations from the corresponding datasets to represent them in similarity calculations. Additionally, we found that using word definitions, rather than the words themselves, is more effective. We use WordNet (Miller, 1994) to obtain definitions. Finally, for each anchor word, we generate a separate Anchor-Word rule.

Generating Anchor-Word Rules After finding an anchor word, we create the Anchor-Word rule. The construction of the rule follows the order in which the entities and the anchor word are present in the sentence. First, we represent the entities with their entity types and the anchor word with its lemma. We, then, connect these words together into a rule by adding the expression "`[]*`" (which indicates any number of tokens) between them.

We show three types of Anchor-Word rules in Figure 2. Let's consider the first example. The example represents the rule "`[entity=nationality] []* [lemma=student] []* [entity=person]`" and shows part of a sentence "[. . .] American exchange student Amanda Knox [. . .]" that matches this rule. Our procedure first identifies the anchor word 'student.' This word lies between the two entities and thus, to build the rule we place the representation of the anchor word (lemma) between the representations of those entities (entity type). To depict the path between these words, we just use "`[]*`" to indicate any number of tokens. The other examples are similar, but demonstrate different order of words between the entities and the anchor word.

4.2 Learning to Generate Rules

We use an Encoder-Decoder approach to learn to generate rules. More specifically, we fine-tune a T5-style code-generation model, CodeT5plus (Wang et al., 2023), using a dataset that maps relation instances (input) to corresponding rules (output). We use CodeT5plus as it can handle all of the special tokens in the rule language. We fine-tune this model using supervised data from two sources: OdinSynth (Vacareanu et al., 2022a) and Anchor-Word rules. Model training details such as tuned hyperparameters are provided in Appendix D. Rules are generated for the relation instances in the training and development splits of FS-TACRED (or FS-NYT29), and are then used to fine-tune the pre-trained model, CodeT5plus.

Our training methodology is defined by the input and output (i.e., supervision) during training.

Input: The training input for the model consists of a sentence along with the subject and object entities it contains. Encoder-Decoder models expect a full sentence as input. Therefore, to highlight the subject and object entities, we (a) enclose the rel-

	5-way, 1-shot			5-way, 5-shot		
	Precision	Recall	F1	Precision	Recall	F1
Previous work						
MNAV	n/a	n/a	12.39 ± 1.01	n/a	n/a	30.04 ± 1.92
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
CKPT	n/a	n/a	15.14 ± 1.12	n/a	n/a	32.26 ± 2.13
SoftRules	33.46 ± 1.47	19.69 ± 1.14	**24.78 ± 1.22**	51.66 ± 1.85	26.02 ± 1.29	**34.59 ± 1.24**
Anchor-Word rules						
syntax	25.86 ± 0.39	10.73 ± 0.32	15.16 ± 0.35	22.40 ± 0.72	32.02 ± 0.62	26.34 ± 0.46
surface	37.62 ± 3.05	10.48 ± 0.78	16.38 ± 1.22	34.21 ± 1.03	30.83 ± 1.27	32.42 ± 1.10
surface and syntax	27.37 ± 1.20	15.99 ± 0.83	20.19 ± 0.97	28.35 ± 0.93	34.04 ± 1.48	30.92 ± 0.99
Anchor-Word + OdinSynth rules	21.86 ± 0.86	23.10 ± 1.08	22.46 ± 0.94	31.96 ± 0.85	35.49 ± 1.50	33.62 ± 1.01
Model-Generated rules training w/						
Anchor-Word rules						
syntax	21.87 ± 0.60	7.67 ± 0.11	11.35 ± 0.14	18.39 ± 0.59	21.93 ± 0.32	19.99 ± 0.31
surface	27.13 ± 4.00	5.20 ± 0.92	8.73 ± 1.50	22.16 ± 1.54	15.09 ± 1.36	17.94 ± 1.45
surface and syntax	21.92 ± 1.13	10.74 ± 0.78	14.41 ± 0.94	17.67 ± 0.26	29.23 ± 0.91	22.02 ± 0.32
OdinSynth rules						
syntax	19.95 ± 0.94	19.62 ± 1.00	19.79 ± 0.97	20.81 ± 0.86	30.56 ± 1.22	24.75 ± 0.95
surface	31.29 ± 2.30	3.20 ± 0.30	5.81 ± 0.53	35.28 ± 1.18	8.98 ± 0.40	14.31 ± 0.50
surface and syntax	19.82 ± 1.19	20.40 ± 1.30	20.10 ± 1.24	20.97 ± 0.76	31.85 ± 1.21	25.28 ± 0.87
Anchor-Word + OdinSynth rules	19.38 ± 0.74	25.47 ± 1.27	22.01 ± 0.95	20.07 ± 0.58	39.52 ± 1.56	26.62 ± 0.79
+ paraphrasing background	17.60 ± 0.36	28.32 ± 0.85	21.71 ± 0.50	26.41 ± 0.84	27.20 ± 1.38	26.80 ± 1.07
Anchor-Word + Model-Gen. rules	20.31 ± 0.60	28.57 ± 1.19	23.74 ± 0.82	29.67 ± 0.87	36.19 ± 1.59	32.60 ± 1.08
+ paraphrasing support	17.27 ± 0.39	30.06 ± 0.63	21.93 ± 0.38	25.56 ± 0.97	38.92 ± 1.78	30.85 ± 1.17
+ paraphrasing query	19.55 ± 0.63	31.93 ± 1.04	**24.24 ± 0.72**	32.46 ± 0.48	39.92 ± 0.94	**35.80 ± 0.48**
+ paraphrasing support and query	20.80 ± 1.47	21.75 ± 1.61	21.25 ± 1.46	18.56 ± 0.58	54.08 ± 1.55	27.63 ± 0.77

Table 2: Results with the test split of FS-TACRED. Our baseline, Anchor-Word rules, outperforms all previous work except SoftRules even though it disregards the background data (i.e., training data with non-overlapping relations). Anchor-Word and Model-Generated rules result in complementary rules: combining them yields better results. Paraphrasing yields further improvements. All rule combinations (indicated with '+') refer to the best system (i.e., using surface, syntax or both). While we use the background data to train our models and SoftRules does not, our rules and matching mechanism are more interpretable—we know which rule is an *exact* match in the query sentence.

evant words within "<subj> ... </subj>" or "<obj> ... </obj>" tags, and (b) replace the entity words with their respective types. Additionally, we incorporate the relation description in the input as it is crucial in identifying anchor words for Anchor-Word rules. If the target output consists of syntax rules, the syntax path is also included in the input.

Output: Since we have multiple rules available per sentence, the decoder can represent them in multiple ways as output. In this work, we experimented with two such approaches (depicted in Figure 3):

- *One-to-One.* In this approach, we use only one rule as supervision for each sentence. However, across different epochs, we randomly select a different rule for learning. During the validation phase, the loss is zero if the model outputs any one of the possible rules. At prediction time, we use beam search to generate a fixed number of rules. A limitation of this method is that we need to decide the number of rules to be predicted. This can lead to the generation of faulty or repetitive rules when the possible rule set is limited.
- *One-to-Many*. In this approach, we concatenate all possible rules corresponding to a sentence using the delimiter '∼'. During the validation step, the loss is zero only if all the rules are generated correctly. Here, the model determines the number of rules to be generated, allowing for no rule generation if appropriate.

We did not find a clear winner between the two approaches; therefore, we chose the one that achieved the best results for each rule type.

5 Experiments and Results

We apply the above techniques to generate rules for support sentences in the test set, creating multiple rules per target relation for each episode. If a rule matches the query sentence, the score for the corresponding relation increases. The relation with the highest score is the predicted relation for the

	5-way, 1-shot			5-way, 5-shot		
	Precision	Recall	F1	Precision	Recall	F1
Previous work						
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
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
SoftRules	22.23 ± 0.47	13.45 ± 0.38	16.76 ± 0.41	27.29 ± 0.77	19.52 ± 0.49	**22.76 ± 0.56**
Anchor-Word + Model-Gen. rules	32.04 ± 0.43	8.67 ± 0.14	13.64 ± 0.21	27.23 ± 0.78	18.83 ± 0.40	22.26 ± 0.52
+ paraphrasing support	21.92 ± 0.53	12.37 ± 0.29	15.82 ± 0.37	20.38 ± 0.50	22.25 ± 0.45	21.27 ± 0.46
+ paraphrasing query	31.78 ± 0.38	13.25 ± 0.19	18.70 ± 0.25	26.44 ± 0.46	27.27 ± 0.30	**26.85 ± 0.35**
+ paraphrasing support and query	23.75 ± 0.37	21.08 ± 0.39	**22.33 ± 0.38**	20.11 ± 0.40	37.01 ± 0.49	26.06 ± 0.44

Table 3: Results with the test split of FS-NYT29. A more detailed version is available in Table 6. Like FS-TACRED, paraphrasing yields further improvements for FS-NYT29 as well. In the 1-shot scenario, we find that paraphrasing both the query and support sentences helps us attain the second-best performance by a large margin. In the 5-shot scenario, paraphrasing only the query sentences allows us to outperform all previous work.

query. If no rules match, we predict `no_relation`. If there's a tie, a relation is chosen randomly. We compare these predictions with ground truth labels and summarize the Precision, Recall, and F1 scores in Table 2 and 3, including error margins for variability across five randomly seeded runs of FS-TACRED and FS-NYT29. We discuss the specifics of these techniques and examine their individual and combined results in subsequent sections.

5.1 FS-TACRED

Anchor-Word Rules Anchor-Word rules are created in an unsupervised manner, and, therefore, we can directly generate these rules from the support sentences in the test set. The results of applying these rules to directly predict the relations in the test set's query sentences are presented in Table 2, within the block titled "Anchor-Word rules." In that block, we also depict the evaluation metrics for using syntax rules, surface rules, or a combination of both. For the 1-shot scenario, we find that combining syntax and surface rules helps improve the F1 score. However, in the 5-shot scenario, surface rules alone perform the best and mixing them with syntax rules degrades performance. Interestingly, surface rules outperform all previous work in the 1-shot and 5-shot scenarios except SoftRules.

Model-Generated Rules As described in Section 4.2, we fine-tune the encoder-decoder language models using two data sources and train separate models for syntax rules and surface rules. Therefore, we build four models. After training these models, we use them to generate rules for the support sentences in the test set. We report the performance of these generated rules in the third block of Table 2.

As the table indicates, rules generated by models trained on Anchor-Word rules do not perform as well as those directly generated from the support sentences in the test set. In contrast, rules derived from training with OdinSynth rules are more successful. This discrepancy could stem from the difficulty of replicating the logic of the similarity model used for generating Anchor-Word rules with the limited amount of supervised data available (only around 8,000 training data samples). In comparison, the logic behind OdinSynth rules is simpler to replicate. It is also important to note that surface rules generally underperform compared to syntax rules. However, when we combine these rule types, their performance usually exceeds that of each type used independently.

Combining Rules We also experiment with combining all the rule types: Anchor-Word rules and Model-Generated rules (Table 2, last block). Both the 1-shot and 5-shot scenarios benefit from combining the rules. However, the improvement in the 5-shot scenario is minimal compared to the performance of Anchor-Word surface rules.

Paraphrasing We also experiment with paraphrasing the sentences in FS-TACRED. Specifically, we paraphrase three parts of the dataset: a) background relation instances, b) support sentences, and c) query sentences. We generate 5 paraphrases per sentence using ChatGPT 3.5, following the prompt detailed in Appendix C. The experimental results are presented in Table 2 in the 3rd and 4th blocks. We find that paraphrasing query sentences is the only beneficial strategy. In the 1-shot scenario, there is a modest increase of 0.5 F1 points, while the 5-shot scenario sees a more substantial improvement of around 3 F1 points.

Overly general rule
Prevalence: 1-shot (30%) | 5-shot (20%) Gold: `org:top_members/employees` Predicted: `org:founded_by`

Example Query sentence: "... to help create a platform for independent film in China and to strengthen the ties between the Chinese film community and the Tribeca Film Festival," said [Jon Patricof]^{object}, chief operating officer of [Tribeca Enterprises]^{subject}.

Misfiring rule: `[entity=person] [tag=NN] [entity=organization]`

Close-but-not-exact rule
Prevalence: 1-shot (26%) | 5-shot (20%) Gold: `org:founded_by` Predicted: `no_relation`

Example Query sentence: Nielsen said James Finkelstein, who founded [Pluribus]^{subject} this year with George Green and [Matthew Doull]^{object}, will serve as e5's chairman.

Close rule: `[entity=person] []* [lemma=founder] []* [entity=organization]`

No matching rules
Prevalence: 1-shot (14%) | 5-shot (26%) Gold: `per:origin` Predicted: `no_relation`

Example Query sentence: [Graham]^{subject}, a [Southern Tutchone Indian]^{object} from Canada, is charged with first- and second-degree murder in the slaying of Aquash, and could be sent to prison for life if convicted.

Annotation error
Prevalence: 1-shot (8%) | 5-shot (6%) Gold: `no_relation` Predicted: `org:top_members/employees`

Query sentence: Single-sex schools are an "illusionary silver bullet," said [Lisa Maatz]^{object}, director of public policy and government relations for the [American Association of University Women]^{subject}.

Matched rule: `[entity=person] []* [lemma=director] []* [entity=organization]`

Wrong anchor word
Prevalence: 1-shot (4%) | 5-shot (12%) Gold: `no_relation` Predicted: `per:schools_attended`

Example Query sentence: About an hour after landing at Eindhoven, the [foreign ministry]^{object} said "Ruben has arrived safely at [his]^{subject} final destination", which it declined to specify.

Misfiring rule: `[entity=organization] []* [lemma=say] []* [entity=person]`

Table 4: Most common error types discovered after manually analyzing 50 errors made with Anchor-Word and Model-Generated rules for both 5-way 1-shot and 5-way 5-shot scenarios. *Gold* indicates the true relation between the entities (indicated with square brackets) for an example query sentence.

Post-processing We experiment with adding reversed versions of the generated rules as a postprocessing step. For example, if the original rule is "`[entity=person] [lemma=president] [entity=organization]`", we reverse it to "`[entity=organization] [lemma=president] [entity=person]`". We discover empirically that reversing rules is only beneficial for some of the methods in Table 2 and report results accordingly.

5.2 FS-NYT29

We conducted the same experiments and post-processing as FS-TACRED on the FS-NYT29 dataset. Table 3 summarizes the results, with a detailed version in Table 6. Our system outperforms the current SOTA in the 5-shot scenario. In the 1-shot scenario, we achieve the second-best performance. We find paraphrasing to be very effective for this dataset, with F1 scores improved by about 9 points in the 1-shot scenario and about 5 points in the 5-shot scenario.

6 Qualitative Analysis

In this section, we present results from two types of qualitative analyses. First, we perform error analysis on episodes that did not result in the correct answer when using our best method. Thereafter, we investigate the pliability of the generated rules by manually editing them and evaluating the improvements in performance.

6.1 Error Analysis

We analyze 50 errors by our best method for both the 1-shot and 5-shot scenarios (Table 4, 100 total).

Misclassification by overly general rules are common in the 1-shot (30%) and 5-shot scenarios (20%). This type of error occurs when a very general rule was generated, resulting in false positives. An example of a very general rule is "`[entity=person] [tag=NN] [entity=organization]`" for the relation `org:founded_by`.

Close-but-not-exact rules are also common: 26% (1-shot) and 20% (5-shot). This error indicates that one of our rules is very close but fails the exact match requirement. For example, this is the situation when the rule is "`[entity=person] []* [lemma=founder] []* [entity=organization]`" but the query sentence contains the word 'founded' instead. This rule would've matched the sentence with a fuzzy matching approach.

Sometimes, our rules completely miss the relation. This kind of error is much less common in the 1-shot scenario than in the 5-shot scenario (14% vs 26% of examined errors).

Annotation errors are present in any dataset, and they inevitably lead to correct predictions being counted as wrong. In the example depicted in Table 4, the entities in the query sentence are connected by the `org:top_members/employees` relation, and our rules make this prediction. However, the gold truth label indicates `no_relation`, and thus it is counted as an error. Only 6% (1-shot) and 8% (5-shot) of errors belong to this type.

Anchor words are obtained deterministically based on similarity between words in a sentence and relation descriptions. While simple, our approach sometimes identifies anchor words that appear to be unconnected to the relation. In the example rule depicted, the anchor word identifies a word with lemma 'say' as a close word to the `per:date_of_birth` relation, which is incorrect.

6.2 Pliability: Can Humans Quickly Improve Performance?

This section discusses an experiment evaluating the pliability of rules generated by our methods. Similar to Vacareanu et al. (2022a), we refine the rules from our best method by manually editing them over two hours to enhance their performance. Conducted by two experts (from the authors), the exercise involved modifying rules for the `per:date_of_birth` relation by adding, removing, or revising rules. After this exercise, we measured the F1 score of the edited rules for the test sentences of the concerned relation. Post-edit, the average F1 score of these rules on test sentences improved by 227%, demonstrating significant flexibility and pliability. Appendix B provides examples of edits including rules added, modified, and removed.

7 Conclusions

In this work, we describe an approach for realistic few-shot relation classification (FS-RC) using rules generated with Encoder-Decoder Language Models. We also present Anchor-Word rules, an effective baseline to generate rules. A crucial benefit of both approaches is that they generate rules which are inherently interpretable and pliable, allowing users to easily understand and modify them.

Later, we evaluate the rules generated by our methods on two datasets – FS-TACRED and FS-NYT29. We find that Anchor-Word rules are highly effective for FS-TACRED outperforming most previous work, except for the state of the art. Additionally, our Model-Generated rules excel in two of four scenarios across the datasets, surpassing most prior efforts. When combined, these rules match state-of-the-art performance in the 1-shot scenario of FS-TACRED and outperform all previous methods in the 5-shot scenarios of both FS-TACRED and FS-NYT29.

Rules	Precision	Recall	F1
Original	11.68 ± 4.32	50.77 ± 14.14	18.71 ± 6.18
Human 1	62.00 ± 8.94	81.13 ± 12.62	69.69 ± 7.72
Human 2	81.71 ± 16.29	40.72 ± 12.98	52.70 ± 13.17

Table 5: Results on the test split in the 1-shot scenario (`per:date_of_birth` relation) with (a) the best automatically obtained rules (original: Anchor-Word and Model-Generated rules, Table 2) and (b) after humans modify these rules for two hours. Our method is interpretable (exact rules and exact matching) and pliable: F1 improves 272% and 182% respectively.

Limitations

Most machine learning models' outputs are limited by the training data they were shown during the training step. In this work, we used two kinds of rule sources to train our models. There are definitely many more kinds of rules possible and some could be more accurate than the ones we were able to generate. Future work could, therefore, focus on improving the variety and quantity of rule types in the training data and potentially boost the performance of these models.

Syntax rules rely on access to the dependency trees of sentences. These trees may not be available for many low-resource languages. This is a limitation because a part of our technique cannot be applied to these languages (Surface rules will still work). Thankfully, there exists a significant effort in the form of Universal Dependencies (de Marneffe et al., 2021) that aims to create comprehensive dependency tree annotations for all of world's languages (they have already created dependency treebanks for around 100 languages) and this limitation should vanish over time.

Ethics

Model Biases Our work employs T5-style Encoder-Decoder Language Models as the foun-

dation for all our models, inheriting the typical ethical and social risks (Bender et al., 2021) associated with most language models. While our models output rules that users can potentially adjust to correct any biases, there is a risk that biased rules could be produced without user intervention.

Data Sources To build FS-TACRED, we require the TACRED dataset which we obtain from the Linguistic Data Consortium (LDC) under a non-commercial license. We use it solely for research purposes, as intended.

Acknowledgements

This work was supported by NSF grant #2006583. The views, results, conclusions, and suggestions expressed in this material are entirely those of the authors and do not necessarily reflect the views of the NSF. We accessed vital computing resources (GPUs) for this work via the Chameleon Cloud platform (Keahey et al., 2020) and the RunPod platform (`https://runpod.io`). We utilized the Azure OpenAI service for querying ChatGPT with credits from Microsoft's Accelerating Foundation Models Research program.

We are also very grateful to Robert Vacareanu for providing resources related to OdinSynth, which served as a training source for one of our models, to Fahmida Alam and Md Asiful Islam for providing access to FS-NYT29, to Mihai Surdeanu, Gus Hahn-Powell, and Enrique Noriega-Atala for introducing us to this problem, and to the anonymous reviewers for their insightful feedback.

References

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 (LREC-COLING 2024)*, pages 16592–16606, Torino, Italia. ELRA and ICCL.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery.

Kurt D. Bollacker, Robert Cook, and Patrick Tufts. 2007. Freebase: A shared database of structured general human knowledge. In *AAAI Conference on Artificial Intelligence*.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

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 *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 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons learned from the chameleon testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association.

Yang Li, Canran Xu, Guodong Long, Tao Shen, Chongyang Tao, and Jing Jiang. 2024. CCPrefix: Counterfactual contrastive prefix-tuning for many-class classification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2977–2988, St. Julian's, Malta. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Bo Lv, Li Jin, Yanan Zhang, Hao Wang, Xiaoyu Li, and Zhi Guo. 2022. Commonsense knowledge-aware prompt tuning for few-shot nota relation classification. *Applied Sciences*, 12(4).

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. *CoRR*, abs/2105.08393.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Tapas Nayak and Hwee Tou Ng. 2019. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *AAAI Conference on Artificial Intelligence*.

Seongsik Park and Harksoo Kim. 2021. Improving sentence-level relation extraction through curriculum learning. *CoRR*, abs/2107.09332.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

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.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

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.

Evan Sandhaus. 2008. The New York Times Annotated Corpus.

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13843–13850.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI Conference on Artificial Intelligence*.

Robert Vacareanu, Fahmida Alam, Md Asiful Islam, Haris Riaz, and Mihai Surdeanu. 2024. Best of both worlds: A pliable and generalizable neuro-symbolic approach for relation classification. *Preprint*, arXiv:2403.03305.

Robert Vacareanu, George C.G. Barbosa, Enrique Noriega-Atala, Gus Hahn-Powell, Rebecca Sharp, Marco A. Valenzuela-Escárcega, and Mihai Surdeanu. 2022a. A human-machine interface for few-shot rule synthesis for information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 64–70, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Robert Vacareanu, Dane Bell, and Mihai Surdeanu. 2022b. 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, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

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.

Yue Wang, Hung Le, Akhilesh Gotmare, Nghi Bui, Junnan Li, and Steven Hoi. 2023. CodeT5+: Open code large language models for code understanding and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1069–1088, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

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*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

	5-way, 1-shot			5-way, 5-shot		
	Precision	Recall	F1	Precision	Recall	F1
Previous work						
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
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
SoftRules	22.23 ± 0.47	13.45 ± 0.38	16.76 ± 0.41	27.29 ± 0.77	19.52 ± 0.49	**22.76 ± 0.56**
Anchor-Word rules						
syntax	32.82 ± 0.37	4.95 ± 0.10	8.60 ± 0.16	31.45 ± 0.36	12.19 ± 0.29	17.57 ± 0.33
surface	26.81 ± 2.30	1.06 ± 0.12	2.05 ± 0.22	23.22 ± 1.56	3.96 ± 0.27	6.76 ± 0.45
surface and syntax	32.11 ± 0.51	5.21 ± 0.12	8.96 ± 0.19	30.28 ± 0.43	13.32 ± 0.24	18.50 ± 0.29
Anchor-Word + OdinSynth rules	32.17 ± 0.56	8.51 ± 0.18	13.46 ± 0.27	27.55 ± 0.64	18.40 ± 0.37	22.06 ± 0.45
Model-Generated rules training w/						
Anchor-Word rules						
syntax	32.49 ± 0.76	4.87 ± 0.15	8.48 ± 0.25	30.89 ± 0.33	11.92 ± 0.25	17.20 ± 0.27
surface	25.40 ± 1.67	0.96 ± 0.09	1.85 ± 0.17	21.93 ± 1.68	3.56 ± 0.28	6.12 ± 0.48
surface and syntax	32.04 ± 0.32	5.22 ± 0.09	8.98 ± 0.14	29.77 ± 0.62	13.11 ± 0.30	18.20 ± 0.38
OdinSynth rules						
syntax	31.37 ± 0.59	5.43 ± 0.17	9.26 ± 0.27	28.69 ± 0.63	13.28 ± 0.41	18.16 ± 0.49
surface	39.99 ± 0.91	6.62 ± 0.24	11.36 ± 0.38	38.91 ± 0.64	12.57 ± 0.44	19.00 ± 0.57
surface and syntax	33.13 ± 0.73	7.84 ± 0.21	12.68 ± 0.32	28.52 ± 0.57	15.55 ± 0.38	20.12 ± 0.43
Anchor-Word + OdinSynth rules	32.00 ± 0.40	8.46 ± 0.15	13.38 ± 0.22	27.36 ± 0.78	18.19 ± 0.45	21.85 ± 0.55
+ paraphrasing background	31.95 ± 0.52	8.37 ± 0.18	13.27 ± 0.27	27.21 ± 0.69	17.92 ± 0.42	21.61 ± 0.50
Anchor-Word + Model-Gen. rules	32.04 ± 0.43	8.67 ± 0.14	13.64 ± 0.21	27.23 ± 0.78	18.83 ± 0.40	22.26 ± 0.52
+ paraphrasing support	21.92 ± 0.53	12.37 ± 0.29	15.82 ± 0.37	20.38 ± 0.50	22.25 ± 0.45	21.27 ± 0.46
+ paraphrasing query	31.78 ± 0.38	13.25 ± 0.19	18.70 ± 0.25	26.44 ± 0.46	27.27 ± 0.30	**26.85 ± 0.35**
+ paraphrasing support and query	23.75 ± 0.37	21.08 ± 0.39	**22.33 ± 0.38**	20.11 ± 0.40	37.01 ± 0.49	26.06 ± 0.44

Table 6: Results on the test split of FS-NYT29. The scores in the "Previous Work" block are from Vacareanu et al. (2024). Anchor-Word rules are not as effective as FS-TACRED on this dataset. However, their combined performance with Model-Generated rules is on par with previous methods, surpassing MNAV and OdinSynth in the 5-shot scenario. Paraphrasing further improves performance, and our system outperforms all previous work in the 5-shot scenario.

Support for `per:stateorprovinces_of_residence`:
- [The leader of the group of Americans][subject] charged on Thursday with abducting children in Haiti is an [Idaho][object] businesswoman with a complicated financial history that involves complaints from employees over unpaid wages, state liens on a company bank account and lawsuits in small claims court.

Support for `per:origin`:
- There was just one problem: No mention was made of [Alan P. Gross][subject], an [American][object] from Potomac, Md., who passed the holiday in a Cuban military facility, where he has been imprisoned for a year without trial because he tried to help Cuba's Jews.

Support for `org:founded_by`:
- "The consumer is just tired" of all the bad news, said [Bill Martin][object], co-founder of [ShopperTrak][subject], based in Chicago.

Support for `org:top_members/employees`:
- [Patrick Graham][object], president of the local [Urban League][subject], a civil rights group, estimated that black unemployment in the area was 2 1/2 times the overall rate.

Support for `org:member_of`:
- The [White Rose Coalition][object] includes members of the Los Angeles National Impeachment Center (LANIC), CODEPINK, Troops Out Now Coalition, World Can't Wait, ANSWER, [Progressive Democrats of America][subject], the Green Party, Veterans for Peace, United for Peace and Justice, and others.

Query sentence:
- Despite a paralyzing blizzard in Washington, Obama brought together Al Sharpton, founder of the National Action Network; NAACP President [Benjamin Jealous][object]; and Marc Morial, president of the [National Urban League][subject], for a conversation that lasted nearly an hour.

Expected output: `no_relation`, as none of the relations corresponding to the support sentences hold between the two named entities in the query sentence (indicated with square brackets).

Figure 4: Example of a 5-way, 1-shot episode from FS-TACRED. The problem is to identify the relation between the entities in the query sentence (between square brackets) out of the five relations in the support sentences. Each relation is exemplified with one example. Even though we only depict one query sentence in this figure, there are three query sentences per episode in FS-TACRED.

A Examples of FS-TACRED 5-way episode: 1-shot and 5-shot

In Figures 4 and 5, we depict examples from FS-TACRED of a 5-way 1-shot episode and a 5-way 5-shot episode, respectively. In both scenarios, the marked entities (surrounded by square brackets) in the query sentence could belong to one of 5 target relations (5-way) or the `no_relation` category. In the 5-way 1-shot scenario, we are provided with 1 example for each target relation. In contrast, in the 5-way 5-shot scenario, we are provided with 5 examples per target relation. In these examples, we only show one query sentence per episode but we are actually provided with three query sentences per episode in FS-TACRED.

B Pliability Exercise: Original and Modified Rules

We show some example rule changes from the pliability exercise discussed in Section 6.2 in Figure 6. All the rules were related to the relation `per:date_of_birth` which translates to "a person's date of birth." In the rest of the paragraph, we discuss the changes made. We don't modify the rule "`[entity=person] [word=born] [entity=date]`" because it perfectly captures most statements regarding a person's date of birth. Note that this rule is actually a syntax rule and therefore, it only attempts to match a syntactic path. We added a surface rule "`[entity=person] [lemma=be]? [lemma=bear] [tag=IN]? [entity=date]`" similar to the syntax rule just discussed. We removed many rules that were either nonsensical—"`[entity=person] []* [entity=date] []* [lemma=represent]`"—or overly broad—"`[entity=person] [tag=VBN] [entity=date]`". Finally, we modified an anchor-word rule by adding an extra lemma: "`[entity=person] []* [lemma=bear | lemma=birth] []* [entity=date]`".

C Paraphrasing Procedure

We use ChatGPT 3.5 to generate paraphrases for sentences in the FS-TACRED dataset. Below, we describe this technique. For querying the ChatGPT API, we utilize Azure OpenAI Services and use the following prompt (inspired by Vacareanu et al. (2024)):

`Please generate 5 paraphrases for the following sentence. Please ensure the meaning and the message stays the same. Also, ensure that these`

Support for `per:children`:
- Knox's father, Curt Knox, said [his][subject] daughter looked "confident in what [she][object] wants to say."
- "We definitely see it as a victory," said [Kunstler][object], the daughter of [William Kunstler][subject], the colorful crusading civil rights lawyer who died in 1995.
- [Bibi][subject]'s 18-year-old daughter, [Sidra][object], said she followed the crowd to the mosque and witnessed people hitting and insulting her mother.
- She was in [her][object] early teens when her mom told her dad he couldn't see his daughters if [he][subject] continued taking drugs.
- [Andrew E. Lange][object] was born in Urbana, Ill., on July 23, 1957, the oldest son of [Joan Lange][subject], a school librarian, and Albert Lange, an architect, and grew up in Easton, Conn.

Support for `per:city_of_death`:
- Grace Burgess, a spokeswoman for the [New York City][object] medical examiner's office, said the office on Tuesday ruled [Cerniglia][subject]'s death a suicide.
- A chef once featured on Gordon Ramsay's "Kitchen Nightmares" show has jumped to [his][subject] death from the George Washington Bridge that connects [New York][object] and New Jersey.
- The [New York][object] City medical examiner on Tuesday ruled the death of 39-year-old Joseph Cerniglia a suicide and confirmed that [Cerniglia][subject] jumped from the bridge.
- Police say [Samudio][subject] was kidnapped early June in Rio de Janeiro, driven to [Belo Horizonte][object] and killed at a suburban house.
- Dr. [Frank Baldino Jr.][subject] who founded the pharmaceutical company Cephalon, best known for the drug Provigil, which is used to increase alertness, died Thursday in [Philadelphia][object].

Support for `per:schools_attended`:
- [Piedra][subject] testified he struggled to get his career going after graduating in 1998 from [Tufts University School of Dental Medicine][object].
- He attended Princeton University and then the [University of California][object], Berkeley, where [he][subject] received a Ph.D. in 1987 and was promptly hired as a professor.
- [Her][subject] accusers, however, see a dark side to the [University of Washington][object] student standing trial along with Italian Raffaele Sollecito, the engineering student who became her lover just a week before the murder.
- [His][subject] former student Mark Devlin of the [University of Pennsylvania][object] was co-leader of the other, known as the Microwave Anisotropy Telescope.
- Prosecutors had accused [Amanda Knox][subject], 22, then a student at the [University of Washington][object], and her boyfriend, Raffaele Sollecito, 25, of killing her housemate, Meredith Kercher, 21, of Surrey, England, in November 2007 after a scuffle escalated into their coercing her into a sex game.

Support for `per:date_of_birth`:
- Her birth name was Barbara Jean Davis, and [her][subject] birth date was [Jan 31, 1949][object].
- [Baldino][subject] was born [May 13, 1953][object], and grew up in New Jersey and Pennsylvania.
- [Lange][subject] was born [July 23, 1957][object], in Illinois.
- [Ble Goude][subject] was born in [1972][object] in Gbagbo's centre west home region, Guiberoua, and rose to become secretary general of the powerful and aggressive Students' Federation of Ivory Coast (FESCI).
- By the time Emily (born in 1978) and [Sarah][subject] (born in [1976][object]) were kids, their father had become better known for representing accused Mafia don John Gotti and, in a mock trial staged for Fox TV's "The Reporters," a cat named Tyrone.

Support for `org:top_members/employees`:
- The general assembly of the Organisation of Asia-Pacific News Agencies ([OANA][subject]) is seeking to boost the quality of the 40 news agencies across 33 countries that comprise it, said incoming OANA head and chief of Indonesia's state-run Antara news agency [Ahmad Mukhlis Yusuf][object].
- [Robert Holden][object], deputy director at [the National Congress of American Indians][subject], said the Washington, DC-based group is hopeful the use of secured cards could be expanded to allow tribal members to travel abroad.
- The country's installed wind power capacity will reach 20 gigawatts this year, said [Shi Lishan][object], vice director of the [National Energy Administration][subject]'s New Energy Department, the Xinhua news agency said Wednesday.
- [National Taiwan Symphony Orchestra][subject] (NTSO) leader [Liu Suan-yung][object] said Chang, who has played the violin since he was five years old and was now one of the orchestra's violinists, was the top prize winner.
- China's primary energy consumption will be kept to between 4 to 42 billion tonnes of standard coal by 2015, [Jiang Bing][object], director of the development and planning department of the [National Energy Administration][subject] (NEA), said on Saturday.

Query sentence:
- Survivors include [his][subject] wife, [Sandra][object]; four sons, Jeff, James, Douglas and Harris; a daughter, Leslie; his mother, Sally; and two brothers, Guy and Paul.

Expected output: `no_relation`, as none of the relations corresponding to the support sentences hold between the two named entities in the query sentence (indicated with square brackets).

Figure 5: Example of a 5-way, 5-shot episode from FS-TACRED. The problem is to identify the relation between the entities in the query sentence (between square brackets) out of the five relations in the support sentences. Each relation is exemplified with five examples. Even though we only depict one query sentence in this figure, there are three query sentences per episode in FS-TACRED.

Original rules	
Anchor-Word rules	r_1: `[entity=person] []* [lemma=bear] []* [entity=date]` r_2: `[lemma=bear] []* [entity=person] []* [entity=date]` r_3: `[entity=person] []* [entity=date] []* [lemma=represent]`
Model-Generated rules	r_4: `[entity=person] [tag=VBN] [entity=date]` r_5: `[entity=person] [tag=NN] [tag=NN]? [tag=VBD] [entity=date]` r_6: `[entity=person] [word=born] [entity=date]`
Rules after human modifications	
Unmodified	r_6: `[entity=person] [word=born] [entity=date]`
Added	r_7: `[entity=person] [lemma=be]? [lemma=bear] [tag=IN]? [entity=date]`
Removed	r_2: `[lemma=bear] []* [entity=person] []* [entity=date]` r_3: `[entity=person] []* [entity=date] []* [lemma=represent]` r_4: `[entity=person] [tag=VBN] [entity=date]` r_5: `[entity=person] [tag=NN] [tag=NN]? [tag=VBD] [entity=date]`
Modified	r_1': `[entity=person] []* [lemma=bear | lemma=birth] []* [entity=date]`

Figure 6: Examples of the modifications made to the original rules (Anchor-Word and Model-Generated rules) after two hours of human analysis. From the original six rules, 1 rule is not modified; 1 rule is added, 4 rules are removed, and 1 rule is modified.

`two entities are preserved in the paraphrases: <subject entity> , <object entity>. Output in JSON. JSON should be of the format: { "paraphrases": ["..", "..", "..", "..", ".."] }. Sentence: <input text>.`

In this prompt, variable texts such as entities and input text are enclosed in angular brackets. We query the model named `gpt-3.5-turbo`, version `0613`.

D Implementation and Model Training Details

We fine-tuned the CodeT5Plus (Wang et al., 2023) models using Nvidia RTX 4090, RTX 6000, V100, and A100 GPUs with the HuggingFace Transformers Library, version 4.41.2. (Wolf et al., 2019). Our experiments required approximately 10 GPU-days. We used the base version of CodeT5Plus (model name `codet5p-220m`), which contains 220 million parameters, and fine-tuned it using Python with Pytorch version 2.3.1 (Paszke et al., 2019) and Pytorch Lightning version 2.2.5 (Falcon and The PyTorch Lightning team, 2019). For rule matching, we employed the Odinson package version 0.3.1 (Valenzuela-Escárcega et al., 2020). To identify anchor words, we utilized the `all-MiniLM-L12-v2` model from the sentence-transformers package version 3.0.0 (Reimers and Gurevych, 2019).

Hyperparameters were tuned on the development sets of the FS-TACRED and FS-NYT29 datasets. We implemented early stopping based on development set results, maintaining the same batch size of 8 for both training and validation, while using AdamW (Loshchilov and Hutter, 2017) as the optimizer with a learning rate of 0.0001. For generating rules with beam search in the multi-output scenario, we produced a total of 5 sequences per input with a beam size of 6. In the single-output scenario, we maintained the beam size of 6 but generated only one sequence per input.