GLiREL - Generalist Model for Zero-Shot Relation Extraction

Jack Boylan, Chris Hokamp, Demian Gholipour Ghalandari
Quantexa
<firstname><lastname>@quantexa.com

Abstract

We introduce GLiREL (**G**eneralist **L**ightweight model for zero-shot **Rel**ation Extraction), an efficient architecture and training paradigm for zero-shot relation classification. Inspired by recent advancements in zero-shot named entity recognition, this work presents an approach to efficiently and accurately predict zero-shot relationship labels between multiple entities in a single forward pass. Experiments using the FewRel and WikiZSL benchmarks demonstrate that our approach achieves state-of-the-art results on the zero-shot relation classification task. In addition, we contribute a protocol for synthetically-generating datasets with diverse relation labels.

1 Introduction

Recent advances in zero-shot NLP models for entity recognition have been enabled by large-scale synthetic training data generation using state-of-the-art (SoTA) Large Language Models (LLMs) (Zhou et al., 2024). An ongoing line of work achieves drastic improvements in accuracy and usability over previous approaches by using efficient architectures targeted at various NLP tasks (Bogdanov et al., 2024; Stepanov and Shtopko, 2024; Zaratiana et al., 2023). Zero-shot named entity recognition (NER) models such as GLiNER (Zaratiana et al., 2023) do not operate on a fixed label set, only requiring textual labels to be specified at inference time, and can directly perform span classification using labels that are not observed during training.

In contrast to generative models, targeted architectures for zero-shot span classification jointly predict all labels simultaneously, making them much more efficient than auto-regressive models (Zaratiana et al., 2023). Existing SoTA zero-shot relation classification[1] (ZSRC) models achieve strong performance, but are inefficient because every entity pair and candidate label combination is treated as a separate input. Existing methods do not scale to real-world use cases, where a large number of entity pairs are extracted from text, each of which must be classified against many candidate labels. GLiREL takes inspiration from recent successes in zero-shot NER and text classification, adapting these approaches to enable ZSRC that is both efficient and accurate.

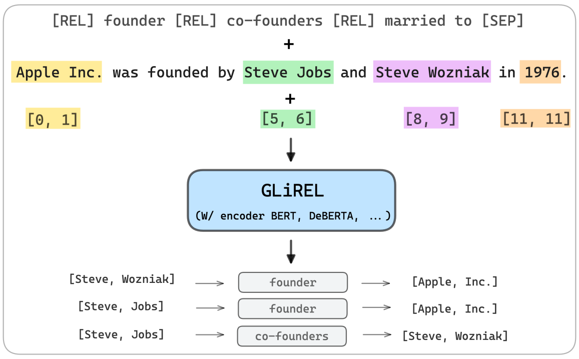

[REL] founder [REL] co-founders [REL] married to [SEP]
+
Apple Inc. was founded by Steve Jobs and Steve Wozniak in 1976.
+
[0, 1]
[5, 6]
[8, 9]
[11, 11]
GLiREL
(W/ encoder BERT, DeBERTA, ...)
[Steve, Wozniak]
founder
[Apple, Inc.]
[Steve, Jobs]
founder
[Apple, Inc.]
[Steve, Jobs]
co-founders
[Steve, Wozniak]

Figure 1: Example inputs and outputs for GLiREL.

While SoTA LLMs excel at information extraction (IE) tasks (Li et al., 2024a; Zhou et al., 2024), there are major limitations to their scale and deployment patterns, including:

- Auto-regressive decoding is unable to take advantage of task-specific parallelism,
- Specific, expensive hardware requirements,
- Output is not sufficiently constrained unless guided by heuristic decoding methods,
- Unpredictable behavior, for example when asked to identify *all* relationships between entities in a document of arbitrary length.

The ability of LLMs to perform zero-shot inference with unconstrained output makes them very flexible, but for many tasks, their auto-regressive

[1] The terms *relation classification* and *relation extraction* are used interchangeably throughout the literature.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8230–8245
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

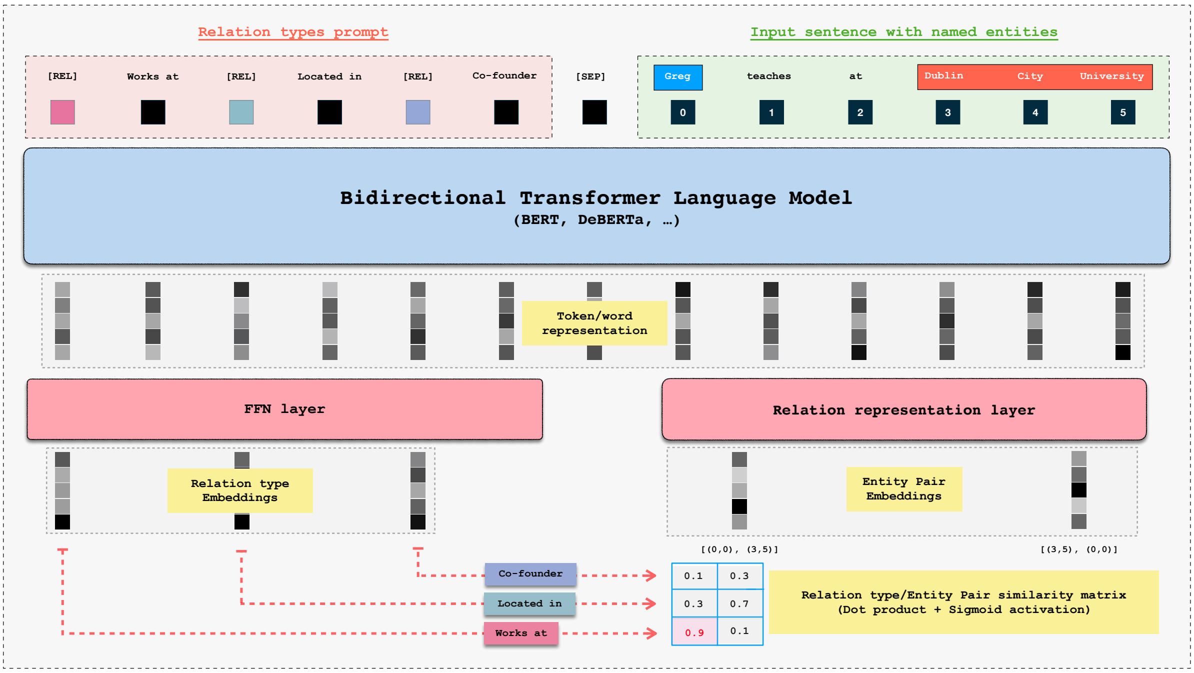


Figure 2: Our proposed approach to zero-shot relation extraction. Firstly, m relation labels and n entities are encoded using a bidirectional transformer. The n entity embeddings will be concatenated to form n^2 pairs in the relation representation layer. The relation labels are fed through a feed-forward network to obtain relation type embeddings. A scoring layer then computes the similarity between every label and every entity pair. Diagram adapted from Zaratiana et al. (2023).

factorization raises issues around reliability. For information extraction tasks in particular, the assignment of relationships between all entities found in a text is a requirement that cannot be reliably achieved with LLMs without auxiliary models (Li et al., 2024b) and/or at-runtime data augmentation mechanisms (Jiang et al., 2024; Ma et al., 2023b).

In contrast to the disadvantages above, LLMs excel at *unconstrained* classification and labeling tasks, and can be effectively utilized to produce large-scale, diverse datasets for training downstream models with task-specific inductive biases. Synthetic dataset generation using general-purpose LLMs is a critical component in the recent success of zero-shot NLP models (Zaratiana et al., 2023; Bogdanov et al., 2024). This paper includes a protocol for generating a large-scale dataset for relationship classification which enables training zero-shot models. The key contributions of our work are:

- A novel zero-shot relation extraction architecture, GLiREL.
- A training dataset construction policy that results in a high-quality synthetic dataset for training zero-shot relation classification models.
- A training paradigm producing SoTA results on zero-shot relation extraction benchmarks.

The paper is organized as follows: section 2 discusses important related work, section 3 discusses the GLiREL model architecture, section 4 presents our experimental results, and sections 5, 6 and 7 provide discussion, analysis, and examinations of the limitations of this work. All code is publicly available.[2]

2 Background

Joint vs Independent NER and Relation Classification Joint entity and relationship classification (Eberts and Ulges, 2019; Zaratiana et al., 2024) can enhance performance through task transfer and global optimization, but increases decoding complexity and reduces flexibility, often requiring bespoke architectures that may not generalize to other tasks. In contrast, traditional IE pipelines use independent models (e.g., spaCy (Honnibal et al., 2020)), offering flexibility but making relationship classification dependent on static upstream NER components. Our work assumes entities are provided by an upstream component and focuses on

[2] https://github.com/jackboyla/GLiREL

detecting relationships between these entities using zero-shot relation labels, thus maintaining pipeline flexibility and allowing classification between any number of entities from diverse sources.

Zero-Shot Relation Extraction Zero-shot relation extraction is an appealing avenue of research because of the flexibility and simplicity of the inference and training paradigms. MC-BERT (Lan et al., 2023) can use previously-unseen relation type labels to classify entity pairs by treating the task as a multiple-choice problem. TMC-BERT (Möller and Usbeck, 2024) extends upon this by incorporating entity types and relation label descriptions. MC-BERT and TMC-BERT construct a template for each entity pair for each candidate label in an instance. This results in a large number of inputs from a relatively small sample size, making it unsuitable for scaling. RelationPrompt (Chia et al., 2022) generates synthetic training examples at inference time using GPT2, requiring a large number ($N = 250$) of examples per label, which is resource-intensive. DSP uses a discriminative prompting strategy to classify both entities and relations in a zero-shot setting. ZS-SKA (Gong and Eldardiry, 2024) performs ZSRC by using templates to augment data and incorporating an external knowledge graph. ZSRE (Tran et al., 2023) encodes text and relation labels separately, computing semantic correlation for each entity pair and label combination, leading to inefficiency in real-world scenarios where many entities are present.

In contrast, GLiREL supports any relation labels at inference time without lengthy descriptions or entity type information (see Figure 1). Multiple entity pairs can be classified in a single input, making our approach more efficient than models that require multiple rounds of inference or at-runtime data generation. Additionally, GLiREL processes relation labels and input text simultaneously, capturing interactions between all labels and entity pairs.

LLMs for Relation Classification LLMs have been leveraged for relation classification, achieving strong zero- and few-shot performance using meta in-context learning and synthetic data generation (Li et al., 2024a; Xu et al., 2023). Some approaches reformulate zero-shot relation classification as a question-answering task (Li et al., 2023). In document-level RE (DocRE), finetuning LLaMA2 with LoRA shows significant improvements, especially when a pretrained language model first classifies whether an entity pair expresses a relationship before passing it to the LLM (Li et al., 2024b). GenRDK (Sun et al., 2024) uses chain-of-retrieval prompts with ChatGPT to generate synthetic data for finetuning LLaMA2. Alternatively, Xue et al. (2024) finetune an LLM to propose head and tail entities given a document and relation label, outperforming other LLM-based baselines.

Zero-Shot Learning and Synthetic Training Data Generation Zaratiana et al. (2023) and Bogdanov et al. (2024) showed that a straightforward and efficient model architecture can achieve excellent performance on the zero-shot NER task, given high-quality, large scale training data. Open-source LLMs have enabled the creation of this kind of training data, through simple and scalable protocols which prompt a model to label the entities in a short text with any type label (Zhou et al., 2024). Importantly, labels are not constrained to a particular taxonomy, and the generative model is free to assign any representative label to entities in the text. In the case of GLiNER (Zaratiana et al., 2023), training on the Pile-NER dataset (created by Zhou et al. (2024)) enabled a new SoTA in zero-shot NER.

3 Method

The GLiREL architecture has three main components:

- A pre-trained bidirectional langage model used as the **text encoder**, which jointly processes candidate relations and input texts.
- An **entity pair representation module** which extracts vector representations for all entities in the text and creates a representation for every pair of entities.
- A **scorer module** to compute the similarity between entity pair representations and relation label representations.

The architecture encodes relation labels and entity pair embeddings in the same latent space to compute their similarity. The overall architecture is illustrated in Figure 2. We choose DeBERTa V3-large as the encoder model due to its excellent performance on downstream tasks (He et al., 2023).

3.1 Input

The model input sequence is comprised of an ordered list of elements, where an element is a string containing one or more tokens. Concretely, inputs are built from:

- a list of M zero-shot labels denoted as t_m, each separated by a special [REL] token: t_0, [REL], t_1, [REL], ..., t_{M-1}. Both t_m and [REL] are treated as elements,
- a special [SEP] token to indicate the end of the labels prompt. [SEP] is treated as a single element,
- the input text, denoted as a list of N tokens $x_0, x_1, ..., x_{N-1}$, where each token is an element.

The GLiREL model additionally expects indices for E known entities in the text, represented as pairs of start and end positions. The input structure is illustrated in Figure 3.

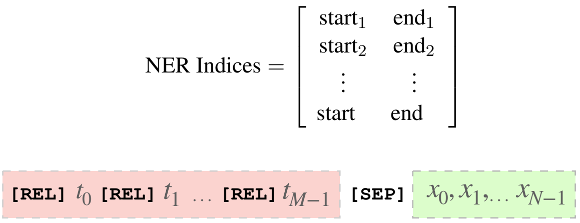


Figure 3: GLiREL input includes relation types $t_0, ..., t_{M-1}$, text tokens $x_0, ..., x_{N-1}$, and the start and end indices of all entities within the text.

Tokenization The special [REL] and [SEP] tokens are added to the encoder's tokenizer vocabulary. The input sequence from Figure 3 is tokenized accordingly, ensuring that relation type labels and special tokens are properly handled. For this study, we follow the pooling strategy described in Zaratiana et al. (2022) by taking the first subtoken representation of each element. Details of the tokenization process are provided in Appendix A.2.

3.2 Token Representation

The token encoder processes the input sequence to compute interactions between all tokens (from both the relationship labels and from the input text), producing contextualized representations. Let $\mathbf{p} = \{\mathbf{p}_t\}_0^{M-1} \in \mathbb{R}^{M \times D}$ represent the encoder's output for each relation type, corresponding to the first subtoken representation of each relation type label. Similarly, $\mathbf{h} = \{\mathbf{h}_i\}_0^{N-1} \in \mathbb{R}^{N \times D}$ denotes the representation of each word in the input text. As already mentioned, for words tokenized into multiple subwords we use the representation of the first sub-word.

3.3 Label and Entity Pair Representation

We aim to encode relationship labels and entity pair embeddings into a unified latent space. We follow the methodology of GLiNER (Zaratiana et al., 2023), with additional steps for entity pair representation and refinement layers.

Relation Label Representation: After pooling, each relationship label in the input sequence is represented by a vector $\mathbf{p}_i$, Relation label representations are additionally transformed by a two-layer feed-forward network (FFN) as shown in equation 1:

$$\mathbf{q} = \text{FFN}(\mathbf{p}) = \{\mathbf{q}_t\}_{t=0}^{M-1} \in \mathbb{R}^{M \times D}, \qquad (1)$$

where M is the total number of relationship labels, and D is the dimensionality of the model's hidden layers. $\mathbf{q}_t$ thus represents the transformed vector for the t^{th} relationship label.

Entity Representation: The entity indices given as input to the model (see Figure 3) are used to extract entity representations from the word representations $\mathbf{h}$. The representation of an entity starting at position i and ending at position j in the input text, $\mathbf{e}_{ij} \in \mathbb{R}^D$, is computed as

$$\mathbf{e}_{ij} = \text{FFN}(\mathbf{h}_i \otimes \mathbf{h}_j). \qquad (2)$$

In equation 2, FFN denotes a two-layer feed-forward network, and $\otimes$ represents the concatenation operation.

Entity Pair Representation: Let $\mathbf{e}_u = \mathbf{e}_{ij}$ represent the u^{th} entity representation computed in Equation 2 using its start and end positions i and j. For any distinct entity pair (u, v), where $u \neq v$, the pair representation $\boldsymbol{\kappa}_{uv}$ is computed as:

$$\boldsymbol{\kappa}_{uv} = \text{FFN}(\mathbf{e}_u \otimes \mathbf{e}_v), \quad \forall u \neq v \qquad (3)$$

where $\otimes$ denotes the concatenation operation, and self-pairs are explicitly excluded. The concatenated entity pair representations are passed through a FFN for projection into the model's latent space. The resulting representations $\boldsymbol{\kappa}_{uv} \in \mathbb{R}^D$ are either further refined (Section 3.4), or used directly for scoring (Section 3.5).

3.4 Refinement Layer

The refinement layer is used to further process both the relation type representations $\mathbf{q}$ and the entity pair representations $\boldsymbol{\kappa}_{uv}$. Inspired by the filter and refine module in joint entity and relation extraction

m	Model	Wiki-ZSL			FewRel		
		P	R	F1	P	R	F1
5	RelationPrompt (Chia et al., 2022)	70.66	83.75	76.63	90.15	88.50	89.30
	DSP-ZRSC (Lv et al., 2023)	94.10	77.10	84.80	93.40	92.50	92.90
	ZSRE (Tran et al., 2023)	**94.50**	**96.48**	**95.46**	96.36	**96.68**	**96.51**
	MC-BERT (Lan et al., 2023)	80.28	84.03	82.11	90.82	91.30	90.47
	TMC-BERT (Möller and Usbeck, 2024)	90.11	87.89	88.92	93.94	93.30	93.62
	GPT-4o†	91.24	72.07	80.03	96.75	83.05	89.20
	GLiREL†	69.88	65.82	62.80	94.56	89.17	81.21
	GLiREL (+ synthetic pretraining)†	89.41	80.67	83.28	**96.84**	93.41	94.20
10	RelationPrompt (Chia et al., 2022)	68.51	74.76	71.50	80.33	79.62	79.96
	DSP-ZRSC (Lv et al., 2023)	80.00	74.00	76.90	80.70	88.00	84.20
	ZSRE (Tran et al., 2023)	85.43	**88.14**	**86.74**	81.13	82.24	81.68
	MC-BERT (Lan et al., 2023)	72.81	73.96	73.38	86.57	85.27	85.92
	TMC-BERT (Möller and Usbeck, 2024)	81.21	81.27	81.23	84.42	84.99	85.68
	GPT-4o†	77.62	66.14	68.35	84.07	58.00	66.20
	GLiREL†	76.45	71.80	68.89	85.40	78.29	80.14
	GLiREL (+ synthetic pretraining)†	**89.87**	81.56	83.67	**91.09**	**87.42**	**87.60**
15	RelationPrompt NG (Chia et al., 2022)	54.45	29.43	37.45	66.49	40.05	49.38
	DSP-ZRSC (Lv et al., 2023)	77.50	64.40	70.40	82.90	78.10	80.40
	ZSRE (Tran et al., 2023)	64.68	65.01	65.30	66.44	69.29	67.82
	MC-BERT (Lan et al., 2023)	65.71	67.11	66.40	80.71	79.84	80.27
	TMC-BERT (Möller and Usbeck, 2024)	73.62	74.07	73.77	82.11	79.93	81.00
	GPT-4o†	**81.04**	32.06	41.57	84.42	65.76	70.70
	GLiREL†	66.14	65.40	60.91	75.76	71.34	70.40
	GLiREL (+ synthetic pretraining)†	79.44	**74.81**	**73.91**	**88.14**	**84.69**	**84.48**

Table 1: Performance comparison of models on Wiki-ZSL and FewRel datasets for various values of unseen relations m. Metrics are averaged at the macro level. Values in **bold** are the best metrics for the given dataset and value of m. The dagger (†) denotes our reported results; the remaining results are taken from their original papers. An extended table comparing more models can be found in appendix Table 6.

Dataset	# Instances	# Rel Types	# Triples
Wiki-ZSL	94,383	113	183,269
FewRel	56,000	80	56,000
Re-DocRED	4,053	96	120,664

Table 2: Dataset statistics.

work from Zaratiana et al. (2024), the refinement layer can be applied to:

- Refine the entity pair representation with respect to the text,
- Refine the relation label representations with respect to the entity pairs, or
- Refine both.

The refinement process is composed of two main stages: (1) a cross-attention mechanism and (2) a feed-forward network (FFN) applied iteratively for a number of layers.

Given the entity pair representations $\boldsymbol{\kappa}_{uv}$ and the relation type representations $\mathbf{q}_t$, the refinement process can be written as follows:

Cross-Attention: The representations are refined using cross-attention, where the entity pair representations attend to the relation type representations, and vice versa. For the entity pair refinement, we compute:

$$\boldsymbol{\kappa}'_{uv} = \boldsymbol{\kappa}_{uv} + \text{CrossAtt}(\boldsymbol{\kappa}_{uv}, \mathbf{q}_t) \tag{4}$$

where $\text{CrossAtt}(a, b)$ represents the cross-attention (also called encoder-decoder attention) mechanism as used by Vaswani et al. (2023), which allows information exchange between a and b.

Self-Attention: The refined entity pair representation undergoes further refinement using a self-attention mechanism to capture intra-pair interactions:

$$\boldsymbol{\kappa}''_{uv} = \boldsymbol{\kappa}'_{uv} + \text{SelfAtt}(\boldsymbol{\kappa}'_{uv}) \tag{5}$$

This process can be repeated for a number of refinement layers – in practice we use a maximum of two refinement layers for efficiency. After the attention mechanism, a feed-forward network (FFN) is applied to further transform the representations:

$$\boldsymbol{\kappa}^{\text{final}}_{uv} = \text{FFN}(\boldsymbol{\kappa}''_{uv}) \tag{6}$$

The same procedure applies when refining relation type representations $\mathbf{q}_t$ to get $\mathbf{q}_t^{\text{final}}$.

The final refined representations $\boldsymbol{\kappa}^{\text{final}}_{uv}$ and $\mathbf{q}_t^{\text{final}}$ are then used for scoring the relation between the entity pair (u, v) and the relation type t, as described in Section 3.5.

3.5 Scoring Layer

To evaluate whether the relationship between entity pair (u, v) corresponds to any relation type t in the set of given relation types T, we calculate the following matching scores:

$$\Phi(u, v) = \{\phi(u, v, t) \mid t \in T\}, \tag{7}$$

where

$$\phi(u, v, t) = \sigma(\boldsymbol{\kappa}^{T}_{uv}\mathbf{q}_t) \in \mathbb{R} \tag{8}$$

In Equation 8, σ denotes a sigmoid activation function. $\boldsymbol{\kappa}_{uv}$ is the representation of the entity pair representation for entities u and v. $\mathbf{q}_t$ is the relation type representation vector type t. As we train with binary cross-entropy loss, $\phi(u, v, t)$ can be interpreted as the probability of the entity pair (u, v) being of type t.

3.6 Training Dataset Generation

As discussed in Section 2, synthetic data generation has been a key enabler for recent improvements in efficient zero-shot NLP models. Due to the difficulty of large-scale manual annotation for relationship classification in particular, synthetic data generation offers a significant improvement in the effectiveness of GLiREL. Our synthetic annotation protocol generates training data for relation classification using an LLM. The goal is to create a flexible relation classification model capable of identifying a broad range of relationship types across various domains. Thus, it is crucial that our training dataset captures diverse relation types.

We follow a methodology similar to Bogdanov et al. (2024), who utilized the C4 dataset (Raffel et al., 2020) to create a rich NER dataset. Bogdanov et al. (2024) sampled from C4, an English web crawl dataset widely used for pretraining LLMs, and employed gpt-3.5-turbo to annotate the entity types. Notably, they did not predefine the entity labels, allowing the LLM to extract a diverse set of entities. Similarly, in our work, we employ an LLM to generate a wide variety of relation labels without imposing predefined relation types, thus ensuring a rich and varied set of annotations. To this end, we use a random sample of the Fineweb dataset (Penedo et al., 2024), another English web crawl dataset chosen for its high quality and diverse material.

We use Mistral 7B-Instruct-v0.2 (Jiang et al., 2023) to annotate every entity pair in every text. The prompt used is shown in the Appendix (Figure 8). Our synthetic dataset contains 63,493 texts with 25,619,624 annotated relations, the majority of which are labeled `NO RELATION`. For our experiments, we discard those labels that intersect with benchmark labels, in order to strictly maintain the zero-shot paradigm. The dataset is available for public use.[3]

3.7 Extending to Coreference Resolution and Document-Level Relation Classification

Co-referential reasoning has been demonstrated to significantly enhance the performance of downstream tasks such as extractive question answering, fact verification, and relation extraction (Ye et al., 2020). Motivated by this insight, we also investigate GLiREL's performance on document-level relation classification, which requires co-reference resolution to aggregate cluster-level relations, projecting relations to the document level. The results of our experiments are presented in Appendix Section A.6.

4 Experiments

4.1 Relation Classification Datasets

We evaluate GLiREL using the Wiki-ZSL and FewRel benchmarks. Chen and Li (2021) derived Wiki-ZSL as a subset of Wiki-KB (Sorokin and Gurevych, 2017), generated through distant supervision. Entities are extracted from complete Wikipedia articles and linked to the Wikidata

[3] `https://huggingface.co/datasets/jackboyla/ZeroRel`

knowledge base to obtain their relations. FewRel (Han et al., 2018) was compiled in a similar manner but underwent additional filtering by crowd workers to enhance data quality and class balance. Although originally designed for few-shot learning, FewRel can be used to benchmark zero-shot relation classification provided that the relation labels in the training and testing sets are disjoint.

Corpus statistics of the Wiki-ZSL and FewRel datasets are summarized in Table 2. Our main results can be seen in Table 1, with an extended table in the appendix (Table 6).

4.2 Zero-Shot Relation Classification Settings

For each dataset, we randomly select m relations as unseen relations ($m = |Y_u|$) and split the data into training and testing sets, ensuring that these m relations do not appear in the training data so that $Y_s \cap Y_u = \emptyset$. We evaluate using macro precision, recall and F1 score. Experiments are repeated five times with different random selections of unseen relations and train-test splits, and the mean metrics are reported. We vary m to examine its impact on performance and to compare against other models.

Training Details For each experiment, we train one model from scratch on the given dataset, and another model is trained following pretraining on our synthetically-annotated dataset. We limit the number of relation type labels prepended to each training instance to 25. For instances where there are less than 25 relation labels, we sample distinct negative labels from the training set. Following Sainz et al. (2023), we introduce regularization by shuffling relation type labels and randomly dropping labels for each instance. We ablate this regularization in Section 5.2. Further training details are provided in the Appendix Section A.5.

Baselines We include the results from works described in Section 2. We also include the results of OpenAI's GPT-4o model (version `gpt-4o-2024-08-06`) as a baseline for LLM performance on the zero shot relation classification task. In our experiments, we use the prompt as shown in Figure 7 to acquire a prediction for each entity pair in each instance.

4.3 Results

GLiREL demonstrates impressive capacity for the zero-shot relation classification task, achieving SoTA performance on both Wiki-ZSL and FewRel. Pretraining on the synthetically annotated dataset shows significant improvement over training from scratch. Additionally, GLiREL is the most successful model in terms of maintaining performance as the number of unseen labels m increases. GLiREL outperforms GPT-4o at every value of m for each dataset. At $m = 15$, GLiREL is marginally better than the current leading model TMC-BERT. It should be noted that both MC-BERT and TMC-BERT require additional data (entity types and descriptions for each relation type label), as well as one forward pass for each entity pair and label, to achieve their result. GLiREL uses only the given relation type labels, and can classify all entity pair relations in a single forward pass.

5 Analysis

5.1 Inference Speed

We compare the inference speeds of GLiREL against some of the highest-performing ZS relation classification models; TMC-BERT (Möller and Usbeck, 2024) and RelationPrompt (Chia et al., 2022). We run inference for each model on both GPU (one Tesla T4) and CPU. We use WikiZSL and FewRel datasets with number of unseen label $m = 10$ and batch size of 32. Each instance in FewRel contains exactly one entity pair, whereas WikiZSL instances have an average of two (up to a maximum 12). Our performance metric is sentences processed per second. The results can be seen in Table 3.

At inference time, the best RelationPrompt model generates N synthetic training examples per unseen label. In our experiments, we set number of synthetic examples $N = 25$, although $N = 250$ is recommended. RelationPrompt NG does not use the generator component to create synthetic training examples. This provides a speed up but at the expense of significant performance deterioration.

RelationPrompt consists of a training data generator (GPT2 with 124M parameters) and an extractor (140M parameters). TMC-BERT has 109M parameters. GLiREL has 467M parameters in total.

Result GLiREL maximizes performance while maintaining efficiency for FewRel on CPU, showing a relatively small decrease in throughput when presented with more entity pairs in WikiZSL. This deterioration is far more pronounced for TMC-BERT, which requires a forward pass for every entity pair, for every candidate label. For Relation-Prompt, the generation component poses a significant bottleneck. On GPU, the margin becomes much more apparent, with GLiREL processing 20x more sentences than RelationPrompt and TMC-BERT on Wiki-ZSL.

Model	Wiki-ZSL		FewRel	
	F_1	Speed (sent/s)	F_1	Speed (sent/s)
CPU				
RelationPrompt NG	43.80	16.41	55.61	9.27
RelationPrompt	71.5	1.96	80.0	1.59
TMC-BERT	81.23	0.12	85.68	1.91
GLiREL	83.67	**4.63**	87.60	**4.93**
GPU				
RelationPrompt NG	43.80	63.1	55.61	59.8
RelationPrompt	71.5	2.06	80.0	1.72
TMC-BERT	81.23	1.41	85.68	27.81
GLiREL	83.67	**47.60**	87.60	**41.07**

Table 3: Under the number of unseen label $m = 10$ on the zero-shot relation classification task, the comparison between RelationPrompt, TMC-BERT, and GLiREL in F1-score and speed. GLiREL speed is shown in **bold**.

5.2 Ablation Study

Relation Type Random Dropping We employed a strategy of randomly dropping relation labels in order to vary the number of relation labels during training. This approach aims to increase model robustness to different numbers of labels at inference time.

Result By ablating this component, we see that GLiREL benefits from random dropping on Wiki-ZSL but actually deteriorates on FewRel. This may be caused by the higher number of entity pairs and relation labels in Wiki-ZSL demanding greater generalization of the model, while the single entity pair in FewRel instances provides a cleaner signal and random dropping is unhelpful noise.

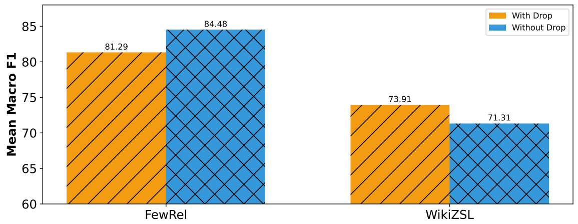


Figure 4: **Addition of random drop:** The effect of randomly dropping relation labels during training on the FewRel and WikiZSL datasets. Using $m = 15$.

Refinement Layers The refinement layers as described in Section 3.4 aim to enhance the representations of both the entity pair representations and the relation label representations respectively.

Result For the FewRel dataset, we observe benefits from having both prompt and entity pair (relation) refinement layers. Conversely, the model performs best on Wiki-ZSL when no refinement layers are used. As with the random drop ablation, this contrast can be attributed to the difference in entity pairs between the two datasets. The FewRel model benefits from additional depth, as it is modelling only one entity pair interaction per instance. With multiple entity pairs in Wiki-ZSL instances, the cross-attention mechanism may introduce unhelpful interactions between irrelevant pairs and relations, amplifying signal noise.

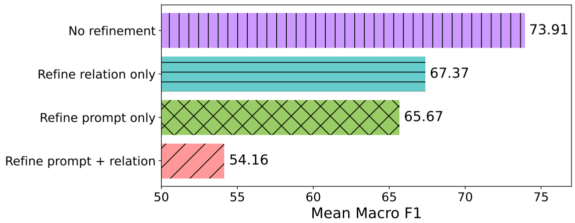


Figure 5: **Addition of refinement layers:** The effect of adding refine layers for entity pair and relation labels representations. From the WikiZSL dataset, using $m =$ 15

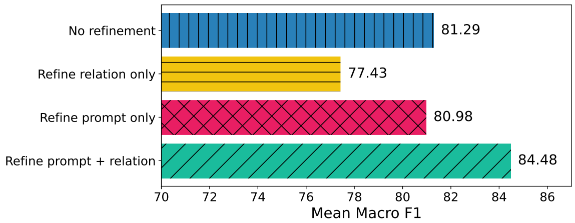


Figure 6: **Addition of refinement layers:** The effect of adding refine layers for entity pair and relation labels representations. From the FewRel dataset, using $m =$ 15

6 Conclusion

We have shown that GliREL is a flexible and highly performant approach to zero-shot relation classification (ZSRC), which achieves SoTA results on challenging benchmarks. Unlike other high-performing ZSRC models, GLiREL can classify multiple entity pairs and relation labels in a single input, making it more efficient. Additionally, we have presented a paradigm for generating high-quality, large-scale synthetic datasets for zero-shot relation classification, as well as an effective training protocol. We hope these methods inspire future work in the area of relation classification.

7 Limitations

As both labels and text are processed in a single forward pass, the number of labels that can be passed in an instance is limited by the model's max sequence length – in DeBERTa's case this is 512 tokens. It is possible to extend DeBERTa's max positional encoding length to larger values, but that has not been studied here. An avenue for future work that may solve this issue has already been implemented for the GLiNER library, through the use of bi- and poly-encoders.[4] Such architectures enable the use of arbitrary amounts of labels. The embeddings of these labels can be precomputed, which may bring additional efficiency benefits.

The joint encoding of labels and input sequence allows the model to condition label and entity pair representations with respect to one another. This is advantageous in cases where it is important for the model to be aware of all possible labels that can be predicted. However, a limitation of this approach is that the model's performance on one label can be influenced by the order and number of other provided labels.

One benchmark-related issue observed by the authors is that texts in which two entities appear may not provide sufficient evidence for the imputed relationship. For example, an instance in Wiki-ZSL (Chen and Li, 2021) imputes the relation label `P20` (`place of death`) between "Jim Dickinson" and "Memphis" for the following text:

> "The Pengwins recorded with Rick Derringer at Bearsville Studios in New York and in Memphis with producer Jim Dickinson, and by Columbia and Polygram."

This is due to the distant annotation of Wiki-ZSL, which uses Wikidata to assign relationships between identified entities. To make a correct prediction, a model would require access to an external knowledge base. This is not implemented in GLiREL or the majority of the methods benchmarked in Table 1. In light of this, the authors suggest moving away from Wiki-ZSL as one of the primary benchmarks for ZSRC, towards benchmarks that assess a model's ability to extract relationships based solely on the text provided (as seen in FewRel).

References

Heike Adel, Benjamin Roth, and Hinrich Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838, San Diego, California. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. NuNER: Entity recognition encoder pre-training via LLM-annotated data. *Preprint*, arXiv:2402.15343.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Razvan Bunescu and Raymond Mooney. 2007. Learning to Extract Relations from the Web using Minimal Supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, Prague, Czech Republic. Association for Computational Linguistics.

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. *Preprint*, arXiv:2104.04697.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Lifted rule injection for relation embeddings. *Preprint*, arXiv:1606.08359.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training. In *European Conference on Artificial Intelligence*.

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.

Jiaying Gong and Hoda Eldardiry. 2024. Prompt-based zero-shot relation extraction with semantic knowledge augmentation. *Preprint*, arXiv:2112.04539.

[4] `https://blog.knowledgator.com/meet-the-new-zero-shot-ner-architecture-30ffc2cb1ee0`

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. *Preprint*, arXiv:1810.10147.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yizhi Jiang, Jinlong Li, and Huanhuan Chen. 2024. Relation classification via bidirectional prompt learning with data augmentation by large language model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13885–13897, Torino, Italia. ELRA and ICCL.

Yuquan Lan, Dongxu Li, Yunqi Zhang, Hui Zhao, and Gang Zhao. 2023. Modeling zero-shot relation classification as a multiple-choice problem. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *Preprint*, arXiv:1706.04115.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting large language models as zero-shot relation extractors. *Preprint*, arXiv:2310.05028.

Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024a. Meta in-context learning makes large language models better zero and few-shot relation extractors. *Preprint*, arXiv:2404.17807.

Xingzuo Li, Kehai Chen, Yunfei Long, and Min Zhang. 2024b. LLM with relation classifier for document-level relation extraction. *Preprint*, arXiv:2408.13889.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Bo Lv, Xin Liu, Shaojie Dai, Nayu Liu, Fan Yang, Ping Luo, and Yue Yu. 2023. DSP: Discriminative soft prompts for zero-shot entity and relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5491–5505, Toronto, Canada. Association for Computational Linguistics.

Youmi Ma, An Wang, and Naoaki Okazaki. 2023a. DREEAM: Guiding attention with evidence for improving document-level relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1971–1983, Dubrovnik, Croatia. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023b. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

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

Cedric Möller and Ricardo Usbeck. 2024. Incorporating type information into zero-shot relation extraction. In *Proceedings of the Extended Semantic Web Conference (ESWC)*, Hersonissos, Greece. Part of the Text2KG Workshop, co-located with the Extended Semantic Web Conference (ESWC).

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Preprint*, arXiv:2406.17557.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015a. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015b. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado. Association for Computational Linguistics.

Ofer Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. Revisiting few-shot relation classification: Evaluation data and classification schemes. *Preprint*, arXiv:2104.08481.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier López de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *ArXiv*, abs/2310.03668.

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.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789, Copenhagen, Denmark. Association for Computational Linguistics.

Ihor Stepanov and Mykhailo Shtopko. 2024. GLiNER multi-task: Generalist lightweight model for various information extraction tasks. *Preprint*, arXiv:2406.12925.

Qi Sun, Kun Huang, Xiaocui Yang, Rong Tong, Kun Zhang, and Soujanya Poria. 2024. Consistency guided knowledge retrieval and denoising in llms for zero-shot document-level relation triplet extraction. *Preprint*, arXiv:2401.13598.

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022. Document-level relation extraction with adaptive focal loss and knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1672–1681, Dublin, Ireland. Association for Computational Linguistics.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2023. Revisiting DocRED – addressing the false negative problem in relation extraction. *Preprint*, arXiv:2205.12696.

Van-Hien Tran, Hiroki Ouchi, Hiroyuki Shindo, Yuji Matsumoto, and Taro Watanabe. 2023. Enhancing semantic correlation between instances and relations for zero-shot relation extraction. *Journal of Natural Language Processing*, 30(2):304–329.

Van-Hien Tran, Hiroki Ouchi, Taro Watanabe, and Yuji Matsumoto. 2022. Improving discriminative learning for zero-shot relation extraction. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 1–6, Dublin, Ireland and Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to unleash the power of large language models for few-shot relation extraction? *Preprint*, arXiv:2305.01555.

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autore: Document-level relation extraction with large language models. *Preprint*, arXiv:2403.14888.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Yann Dauxais, Pierre Holat, and Thierry Charnois. 2024. EnriCo: Enriched representation and globally constrained inference for entity and relation extraction. *Preprint*, arXiv:2404.12493.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2022. Named entity recognition as structured span prediction. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 1–10, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. GLiNER: Generalist model for named entity recognition using bidirectional transformer. *Preprint*, arXiv:2311.08526.

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.

Jun Zhao, WenYu Zhan, Xin Zhao, Qi Zhang, Tao Gui, Zhongyu Wei, Junzhe Wang, Minlong Peng, and Mingming Sun. 2023. RE-matching: A fine-grained semantic matching method for zero-shot relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6680–6691, Toronto, Canada. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. Universalner: Targeted distillation from large language models for open named entity recognition. *Preprint*, arXiv:2308.03279.

A Appendix

A.1 Extended Related Work

Existing Approaches Many systems have addressed relation extraction with varying degrees of success. Earlier works saw CNNs employed in the task of slot filling (Adel et al., 2016), a similar task to relation extraction.

Wang et al. (2022) introduce Deepstruct to improve the structural understanding abilities of language models by pretraining them to generate structures from text on a collection of task-agnostic corpora, enabling zero-shot transfer of knowledge about structure-related tasks. Deepstruct uses this method to achieve state-of-the-art performance on a variety of structured prediction tasks, including RC.

Riedel et al. (2013) use matrix factorization and universal schemas to extract relations by leveraging the shared structure between different relations to improve performance.

Rocktäschel et al. (2015b) and Demeester et al. (2016) focus on injecting logical background knowledge into embeddings for relation extraction by mapping entity-tuple embeddings into an approximately Boolean space, This method improves generalization and leads to significant performance gains over a matrix factorization baseline.

Zhang et al. (2017) have explored the combination of LSTM sequence models with entity position-aware attention to enhance relation extraction. This approach, when coupled with large supervised datasets, has resulted in significant performance improvements in slot-filling tasks.

Question-Answering and Textual Entailment Sainz et al. (2021) and Obamuyide and Vlachos (2018) reformulate relation extraction as an entailment task by using simple verbalizations of relation labels and descriptions. These systems allow for the use of existing textual entailment models and datasets to achieve strong performance in zero-shot and few-shot settings.

Levy et al. (2017) reduced relation extraction to answering reading comprehension questions by associating natural-language questions with each relation slot. This approach enables the use of neural reading comprehension techniques and supports zero-shot learning by facilitating the extraction of new relation types.

Distant Supervision for Relation Extraction Dataset Construction Hand-annotating relation classification (RC) datasets at scale is intractable both because of the size and domain-specificity of relation taxonomies, and especially because of the quadratic number of potential relations in a given text, as a function of the number of named entities in the text. Foundational research leverages distant supervision to bootstrap training datasets for RC, utilizing open source knowledge bases such as Wikidata (Vrandečić and Krötzsch, 2014), Freebase (Bollacker et al., 2008) and DBPedia (Auer et al., 2007) to obtain high-quality relations between entities, and then mining data sources such as Wikipedia for texts mentioning both head and tail entities to construct training datasets (Bunescu and Mooney, 2007; Mintz et al., 2009).

Distant supervision enables the creation of large scale datasets; however, historical work is still constrained to specific pre-defined label sets, and training data is noisy because inputs are not specifically annotated for particular relations.

Real-world Evaluation of Relation Extraction Models Sabo et al. (2021) critique existing few-shot learning (FSL) datasets for RC, highlighting their unrealistic data distributions, and propose a novel method to create more realistic few-shot test data using the TACRED dataset (Zhang et al., 2017), resulting in a new benchmark. Furthermore, they analyze classification schemes in embedding-based nearest-neighbor FSL approaches, proposing a novel scheme that treats the "none-of-the-above" (NOTA) category as learned vectors, improving performance.

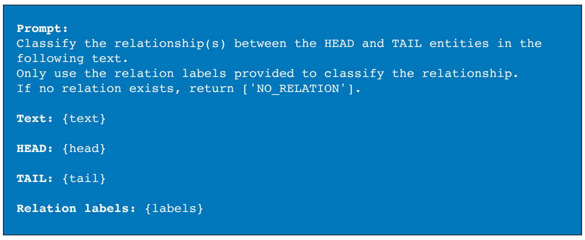
Gao et al. (2019) present FewRel 2.0 by adding a new, dissimilar domain test set and a NOTA option to the existing FewRel (Han et al., 2018) dataset. The authors' experiments reveal that current state-of-the-art models and techniques struggle with these additional challenges that more accurately mirror real-world application of relation extraction models.

A.2 Tokenization Details

The special [REL] and [SEP] tokens are added as special tokens to the encoder's tokenizer vocabulary. The input sequence from Figure 3 is passed to the tokenizer, which joins all elements by whitespace before creating the appropriate encoder-specific subword tokens and input IDs. For example, the label "participation in" would be tokenized into the subword tokens: "particip", "##ation" and "## in". A mapping from the original input elements in Figure 3 to the input IDs is maintained in order to perform subword token pooling of the encoder output. We follow Zaratiana et al. (2022), and perform pooling by taking the vector representation of the first subword token. In the above example, this would correspond to the vector representation for "particip".

With the treatment of one or more tokens as elements, relation type labels can be text of any length. Because the subword tokens of each label are subject to the aforementioned pooling operation, we denote each label using a single index t_m.

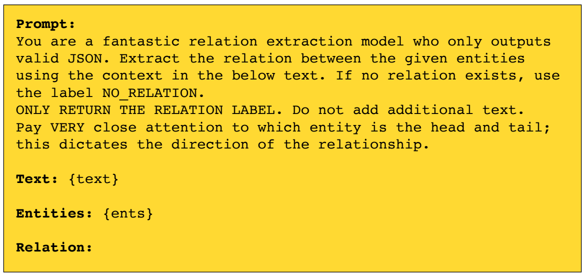
A.3 GPT-4o Baseline Prompt



```
Prompt:
Classify the relationship(s) between the HEAD and TAIL entities in the
following text.
Only use the relation labels provided to classify the relationship.
If no relation exists, return ['NO_RELATION'].

Text: {text}

HEAD: {head}

TAIL: {tail}

Relation labels: {labels}
```

Figure 7: Prompt used to measure GPT-4o performance on the zero-shot relation classification task.

A.4 Synthetic Data Generation Details



```
Prompt:
You are a fantastic relation extraction model who only outputs
valid JSON. Extract the relation between the given entities
using the context in the below text. If no relation exists, use
the label NO_RELATION.
ONLY RETURN THE RELATION LABEL. Do not add additional text.
Pay VERY close attention to which entity is the head and tail;
this dictates the direction of the relationship.

Text: {text}

Entities: {ents}

Relation:
```

Figure 8: Prompt used for synthetic dataset generation.

A.5 Training Setup Details

For the hyperparameters and configuration of our model, refer to Table 4. We used the AdamW optimizer (Loshchilov and Hutter, 2017) with an initial learning rate of 1×10^{-5} for the pretrained encoder parameters and 1×10^{-4} for the remaining parameters involved in span representation, relation representation and scoring layers. A warmup ratio of 10% was used with a cosine scheduler. The hidden layer size for all non-encoder layers was set to 768. The batch size was 8, with total number of steps set to 20,000. All experiments were carried out using one NVIDIA Tesla T4 GPU.

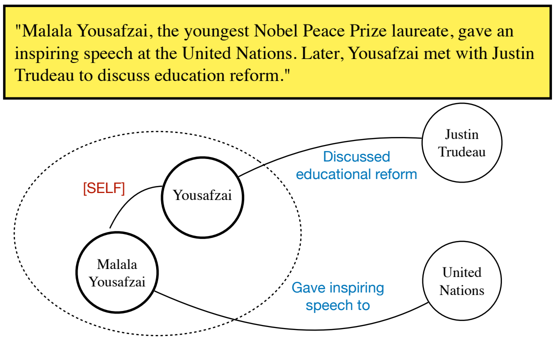
A.6 Coreference Resolution and Document-level Relation Classification

We conceptualize coreference resolution as a specific case of relation classification, where the coreference relation between two mentions referring to the same entity is represented by a special SELF label.

To add coreference resolution ability to GLiREL, we simply include the SELF relation type in the label set during training and inference.

In our experiments (Section 4), we evaluate the performance of this approache using the Re-DocRED dataset (Tan et al., 2023).

Document-Level Relation Classification When coreference information is available, document-level relation extraction (DocRE) can be achieved by propagating local relations across coreference clusters. To implement this, we employ a post-processing step that clusters mentions based on the SELF relation, akin to the connected components algorithm. The outgoing (non-SELF) edges from each mention within a cluster are then interpreted as document-level relations between the resolved entity cluster and other entity clusters in the text. Figure 9 provides an illustration of this concept.



"Malala Yousafzai, the youngest Nobel Peace Prize laureate, gave an inspiring speech at the United Nations. Later, Yousafzai met with Justin Trudeau to discuss education reform."

Figure 9: An example of merging entities into clusters and aggregating their relations.

Number of entity pairs bottleneck One bottleneck of the initial GLiREL archictecture is the fact that the number of entity pairs in an instance scales

Table 4: Training Setup

Hyperparameter/Configuration	**Value**
Optimizer	AdamW (Loshchilov and Hutter, 2017)
Initial Learning Rate (Encoder)	1×10^{-5}
Initial Learning Rate (Other Parameters)	1×10^{-4}
Warmup Ratio	10%
Scheduler	Cosine
Hidden Layer Size (Non-encoder Layers)	768
Batch Size	8
Total Training Steps	20,000
GPU	NVIDIA Tesla T4

almost quadratically with the number of entities ($N^2 - N$, excluding self-pairs). This becomes a significant memory issue when extending GLiREL to document-level relation classification. To alleviate this issue, we can incorporate a naive windowing method, which only pairs entity within a configured token distance window. With the addition of coreference clusters, the relations predicted within each window can then be aggregated across the document.

We further assess the model's effectiveness on the established DocRE dataset Re-DocRED (Tan et al., 2023).

To assess GLiREL's performance on the Document-level relation extraction task, we use the Re-DocRED dataset. In one experiment, the model is trained only to predict relations between entities with no coreference SELF label. The given (gold) coreference clusters are used to aggregate relations across each document. This is the typical setting for the Re-DocRED benchmark. Additionally, we investigate GLiREL's ability to perform coreference resolution by using the prediction of SELF relations between entities to perform coreference.

Baselines We compare GLiREL to the strongest models on this benchmark. KD-RoBERTa (Tan et al., 2022) achieves SoTA results using a RoBERTa-based model, with the addition of an axial attention module to capture interdependencies among entity pairs, and a knowledge distillation framework to make use of large-scale distantly supervised data. Ma et al. (2023a) perform strongly on Re-DocRED by introducing DREEAM, a method that integrates evidence retrieval (ER) to help the model focus on relevant parts of the document. We also compare LLM-based models – LMRC (Li et al., 2024a), AutoRE (Xue et al., 2024), GenRDK and CoR (Sun et al., 2024) – which were introduced in the Background (Section 2).

Results The results of these approaches are shown in Table 5. GLiREL achieves competitive performance against finetuned LLMs with over x15 more parameters. However, GLiREL is surpassed by the more specialised framework LMRC, while both BERT-based methods KD-RoBERTa and DREEAM remain significantly better at this benchmark.

Relying on predicted SELF relations to perform coreference upon proves to be unreliable, showing poor performance on the benchmark without annotated coreference clusters.

Method	Ign F_1	F_1
BERT-based		
KD-RoBERTa$_{large}$ (Tan et al., 2022)	77.60	78.28
DREEAM (Ma et al., 2023a)	**79.66**	**80.73**
LLM-based		
CoR (Sun et al., 2024)	-	37.1 ± 9.2
GenRDK (Sun et al., 2024)	-	41.3 ± 8.9
AutoRE (Xue et al., 2024)	-	51.91
LoRA FT LLaMA2-7B-Chat (Li et al., 2024b)	52.74	53.02
LoRA FT LLaMA2-13B-Chat (Li et al., 2024b)	52.15	52.45
LMRC-LLaMA2-7B-Chat (Li et al., 2024b)	72.33	72.92
LMRC-LLaMA2-13B-Chat (Li et al., 2024b)	74.08	74.63
GLiREL		
GLiREL (+ gold coref clusters)	53.24	54.13
GLiREL (+ predicted coref clusters)	25.97	25.08

Table 5: Results on the test set of Re-DocRED. Best metrics are shown in **bold**

A.7 Full Zero-Shot Relation Classification Results

m	Model	Wiki-ZSL			FewRel		
		P	R	F1	P	R	F1
5	CIM (Rocktäschel et al., 2015a)	49.63	48.81	49.22	58.05	61.92	59.92
	ZS-BERT (Chen and Li, 2021)	71.54	72.39	71.96	76.96	78.86	77.90
	MICRE w/Llama (Li et al., 2024a)	76.46	78.53	77.48	89.34	91.88	90.59
	Tran et al. (2022)	87.48	77.50	82.19	87.11	86.29	86.69
	RelationPrompt NG (Chia et al., 2022)	51.78	46.76	48.93	72.36	58.61	64.57
	RelationPrompt (Chia et al., 2022)	70.66	83.75	76.63	90.15	88.50	89.30
	RE-Matching (Zhao et al., 2023)	78.19	78.41	78.30	92.82	92.34	92.58
	DSP-ZRSC (Lv et al., 2023)	94.1	77.1	84.8	93.4	92.5	92.9
	ZSRE (Tran et al., 2023)	94.50	96.48	95.46	96.36	96.68	96.51
	MC-BERT (Lan et al., 2023)	80.28	84.03	82.11	90.82	91.30	90.47
	TMC-BERT (Möller and Usbeck, 2024)	90.11	87.89	88.92	93.94	93.30	93.62
	GPT-4o	91.24	72.07	80.03	96.75	83.05	89.20
	GLiREL	69.88	65.82	62.80	94.56	89.17	81.21
	GLiREL (+ synthetic pretraining)	89.41	80.67	83.28	96.84	93.41	94.20
10	CIM (Rocktäschel et al., 2015a)	46.54	47.90	45.57	47.39	49.11	48.23
	ZS-BERT (Chen and Li, 2021)	60.51	60.98	60.74	56.92	57.59	57.25
	MICRE w/Llama (Li et al., 2024a)	72.36	74.88	73.60	80.67	82.31	81.48
	Tran et al. (2022)	71.59	64.69	67.94	64.41	62.61	63.50
	RelationPrompt NG (Chia et al., 2022)	54.87	36.52	43.80	66.47	48.28	55.61
	RelationPrompt (Chia et al., 2022)	68.51	74.76	71.50	80.33	79.62	79.96
	RE-Matching (Zhao et al., 2023)	74.39	73.54	73.96	83.21	82.64	82.93
	DSP-ZRSC (Lv et al., 2023)	80.0	74.0	76.9	80.7	88.0	84.2
	ZSRE (Tran et al., 2023)	85.43	88.14	86.74	81.13	82.24	81.68
	MC-BERT (Lan et al., 2023)	72.81	73.96	73.38	86.57	85.27	85.92
	TMC-BERT (Möller and Usbeck, 2024)	81.21	81.27	81.23	84.42	84.99	85.68
	GPT-4o	77.62	66.14	68.35	84.07	58.00	66.20
	GLiREL	76.45	71.80	68.89	85.40	78.29	80.14
	GLiREL (+ synthetic pretraining)	89.87	81.56	83.67	91.09	87.42	87.60
15	CIM (Rocktäschel et al., 2015a)	29.17	30.58	29.86	31.83	33.06	32.43
	ZS-BERT (Chen and Li, 2021)	34.12	34.38	34.25	35.54	38.19	36.82
	MICRE w/Llama (Li et al., 2024a)	67.14	68.87	67.99	73.74	75.83	74.77
	Tran et al. (2022)	38.37	36.05	37.17	43.96	39.11	41.36
	RelationPrompt NG (Chia et al., 2022)	54.45	29.43	37.45	66.49	40.05	49.38
	RelationPrompt (Chia et al., 2022)	63.69	67.93	65.74	74.33	72.51	73.40
	RE-Matching (Zhao et al., 2023)	67.31	67.33	67.32	73.80	73.52	73.66
	DSP-ZRSC (Lv et al., 2023)	77.5	64.4	70.4	82.9	78.1	80.4
	ZSRE (Tran et al., 2023)	64.68	65.01	65.30	66.44	69.29	67.82
	MC-BERT (Lan et al., 2023)	65.71	67.11	66.40	80.71	79.84	80.27
	TMC-BERT (Möller and Usbeck, 2024)	73.62	74.07	73.77	82.11	79.93	81.00
	GPT-4o	81.04	32.06	41.57	84.42	65.76	70.70
	GLiREL	66.14	65.40	60.91	75.76	71.34	70.40
	GLiREL (+ synthetic pretraining)	79.44	74.81	73.91	88.14	84.69	84.48

Table 6: Full performance comparison of models on Wiki-ZSL and FewRel datasets for various values of unseen relations m. All metrics are averaged on the macro (class) level.