GLiNER-Relex: A Unified Framework for Joint Named Entity Recognition and Relation Extraction

Ihor Stepanov[1] Oleksandr Lukashov[1] Mykhailo Shtopko[1] Vivek Kalyanarangan [2]
[1]Knowledgator Engineering, Kyiv, Ukraine
[2]Baldor Technologies Pvt. Ltd. (IDfy), Mumbai, India
ingvarstep@knowledgator.com

Abstract

Joint named entity recognition (NER) and relation extraction (RE) is a fundamental task in natural language processing for constructing knowledge graphs from unstructured text. While recent approaches treat NER and RE as separate tasks requiring distinct models, we introduce GLiNER-Relex, a unified architecture that extends the GLiNER framework to perform both entity recognition and relation extraction in a single model. Our approach leverages a shared bidirectional transformer encoder to jointly represent text, entity type labels, and relation type labels, enabling zero-shot extraction of arbitrary entity and relation types specified at inference time. GLiNER-Relex constructs entity pair representations from recognized spans and scores them against relation type embeddings using a dedicated relation scoring module. We evaluate our model on four standard relation extraction benchmarks: CoNLL04, DocRED, FewRel, and CrossRE, and demonstrate competitive performance against both specialized relation extraction models and large language models, while maintaining the computational efficiency characteristic of the GLiNER family. The model is released as an open-source Python package with a simple inference API that allows users to specify arbitrary entity and relation type labels at inference time and obtain both entities and relation triplets in a single call. All models and code are publicly available.

1 Introduction

Information extraction (IE) from unstructured text is a foundational task in natural language processing (NLP), with broad applications in knowledge graph construction, question answering, document understanding, and retrieval-augmented generation (RAG) systems. Two of the most critical sub-tasks within IE are named entity recognition (NER)—the identification and classification of entities such as persons, organizations, and locations—and relation extraction (RE)—the detection and categorization of semantic relationships between identified entities.

Traditionally, NER and RE have been treated as separate tasks, addressed by pipeline approaches where entities are first extracted and then passed to a relation classifier [Zelenko et al., 2002, Roth and Yih, 2004]. While such pipelines are modular, they suffer from error propagation: mistakes in the NER stage cascade into the RE stage. This observation has motivated a substantial body of work on joint entity and relation extraction, where both tasks are modeled simultaneously to capture their interdependencies [Miwa and Bansal, 2016, Zheng et al., 2017, Fu et al., 2019].

Recent advances in zero-shot NER, particularly the GLiNER framework [Zaratiana et al., 2024c], have demonstrated that compact encoder-based models can achieve competitive performance on recognizing arbitrary entity types. GLiNER represents entity type labels and text tokens within a

shared encoder, enabling flexible extraction of user-defined entity types at inference time. Subsequent work has extended this paradigm to multi-task information extraction [Stepanov and Shtopko, 2024], bi-encoder architectures for scalability [Stepanov et al., 2026], and biomedical adaptation [Yazdani et al., 2025].

However, the relation extraction component of the GLiNER ecosystem has received comparatively less attention. Existing approaches either rely on separate models, such as GLiREL [Boylan et al., 2025], which require pre-identified entities as input, or encode relations implicitly through label concatenation in the multi-task GLiNER formulation [Stepanov and Shtopko, 2024]. Neither approach provides a truly unified model that jointly identifies entities and extracts relations in a single forward pass with shared representations.

In this paper, we introduce **GLiNER-Relex**, a unified architecture for joint NER and RE that extends the GLiNER framework with a dedicated relation extraction module. Our key contributions are:

- **Unified architecture:** We propose a joint model that simultaneously performs entity recognition and relation extraction within a single encoder.
- **Zero-shot relation extraction:** GLiNER-Relex supports arbitrary entity and relation types specified through natural language labels.
- **Relation scoring mechanism:** We introduce a relation scoring module that was inspired by various knowledge graph embedding approaches.
- **Comprehensive evaluation:** We benchmark GLiNER-Relex on four standard RE datasets—CoNLL04, DocRED, FewRel, and CrossRE—comparing against GLiREL, GLiNER2, and GPT-5-mini.
- **Open-source release with simple API:** We release GLiNER-Relex as an open-source model with a straightforward Python API via the GLiNER package.

2 Related Work

2.1 Named Entity Recognition

Named entity recognition has evolved through several paradigms. Early rule-based systems [Appelt et al., 1993] relied on hand-crafted patterns, while statistical methods such as conditional random fields (CRFs) [Lafferty et al., 2001] introduced probabilistic sequence labeling. The advent of deep learning brought BiLSTM-CRF architectures [Lample et al., 2016], which combined learned representations with structured prediction and became the dominant approach prior to the rise of pre-trained transformers. BERT-based models [Devlin et al., 2019] subsequently achieved state-of-the-art supervised NER by fine-tuning on task-specific labeled data.

However, all supervised NER models are limited to a fixed set of entity types defined during training. Zero-shot NER addresses this limitation by enabling the recognition of unseen entity types at inference time. InstructionNER [Wang et al., 2023] reformulated NER as a sequence generation task conditioned on natural language instructions, enabling LLMs to extract entities of specified types. UniversalNER [Zhou et al., 2024] distilled ChatGPT annotations into a smaller model capable of recognizing diverse entity types across domains. GNER [Ding et al., 2024] further advanced generative NER by training on a large-scale dataset spanning diverse entity types.

GLiNER [Zaratiana et al., 2024c] took a different approach by defining NER as a matching problem between text spans and descriptions of the type of natural language entity within a shared

bidirectional encoder, achieving competitive zero-shot performance at a fraction of the computational cost of LLM. Extensions of this framework include multi-task support for NER, RE, QA, and summarization [Stepanov and Shtopko, 2024]; bi-encoder architectures for scaling to thousands of entity types [Stepanov et al., 2026]; schema-driven extraction with GLiNER2 [Zaratiana et al., 2025]; synthetic data augmentation with NuNER [Bogdanov et al., 2024]; and biomedical adaptation with GLiNER-BioMed [Yazdani et al., 2025].

2.2 Relation Extraction

Relation extraction approaches can be broadly categorized into pipeline, joint, and zero-shot methods.

Pipeline approaches first identify entities and then classify relations between entity pairs. Early work used kernel methods [Zelenko et al., 2002] and feature engineering. PURE [Zhong and Chen, 2021] demonstrated that pipeline approaches with distinct contextual representations for entities and relations can achieve strong performance. However, pipeline methods suffer from error propagation between stages.

Joint approaches model entity recognition and relation extraction simultaneously using diverse paradigms. *Sequence labeling methods* include Bi-LSTM with Tree-LSTM for relation prediction [Miwa and Bansal, 2016], multi-tagging formulations [Zheng et al., 2017], and position-attentive labeling for overlapping relations [Dai et al., 2019]. *Decomposition-based methods* divide joint extraction into interdependent subtasks: CasRel [Wei et al., 2020] maps head entities to tail entities via cascade binary tagging, Yu et al. [2020] decompose extraction into head and tail entity stages, and PRGC [Zheng et al., 2021] uses relation judgment, entity extraction, and subject–object alignment. *Table-filling methods* such as UniRE [Wang et al., 2021] and TPLinker [Wang et al., 2020] treat extraction as filling word-pair tables with entity and relation labels. *Set prediction methods* like SPN4RE [Sui et al., 2023] and OneRel [Shang et al., 2022] formulate extraction as direct set prediction, avoiding sequential decoding errors. *Span-based methods* like SpERT [Eberts and Ulges, 2019] enumerate candidate spans and classify entity–relation combinations. *Graph-based approaches* such as GraphRel [Fu et al., 2019] and GraphER [Zaratiana et al., 2024a] formulate IE as graph structure learning, while the autoregressive text-to-graph framework [Zaratiana et al., 2024b] takes a generative approach producing linearized graphs.

2.3 Zero-Shot Relation Extraction

Zero-shot relation extraction has attracted substantial attention as a means to overcome reliance on predefined relation taxonomies.

Entailment and reading comprehension approaches reformulate RE as other well-studied tasks. Levy et al. [2017] reduced relation extraction to answering reading comprehension questions by associating natural-language questions with each relation slot. Obamuyide and Vlachos [2018] and Sainz et al. [2021] reformulated relation extraction as a textual entailment task, using simple verbalizations of relation labels to leverage existing entailment models for zero-shot and few-shot settings.

Attribute and embedding learning approaches project relations into semantic spaces. ZS-BERT [Chen and Li, 2021] performs zero-shot relation classification by learning attribute representations for relation types, projecting both instances and unseen relation labels into a shared embedding space. ZSRE [Tran et al., 2023] encodes text and relation labels separately, computing semantic correlations for each entity-label pair, achieving strong results but at limited efficiency. RE-Matching [Zhao et al., 2023] proposes a fine-grained semantic matching method that decomposes relation

representations into multiple components for more precise zero-shot matching.

Multiple-choice and template-based approaches treat zero-shot RE as a classification problem. MC-BERT [Lan et al., 2023] models zero-shot relation classification as a multiple-choice problem, classifying entity pairs using previously unseen relation type labels. TMC-BERT [Möller and Usbeck, 2024] extends this approach by incorporating entity type information and relation label descriptions for improved performance. However, both MC-BERT and TMC-BERT require constructing a separate input template for each entity pair and candidate label, which limits scalability.

Prompt-based and generative approaches leverage language models for synthetic data and classification. RelationPrompt [Chia et al., 2022] generates synthetic training examples at inference time using GPT-2, though it requires a large number of examples per label, making it resource-intensive. DSP [Lv et al., 2023] employs discriminative soft prompts to jointly extract entities and relations in a zero-shot setting. ZS-SKA [Gong and Eldardiry, 2024] performs zero-shot RE by using templates for data augmentation and incorporating an external knowledge graph.

LLM-based approaches leverage large language models directly for relation extraction. Li et al. [2024a] demonstrated that meta in-context learning enables LLMs to achieve strong zero-shot and few-shot RE performance. For document-level RE, Li et al. [2024b] showed that combining a pretrained classifier with LLaMA2 fine-tuned via LoRA yields significant improvements. GenRDK [Sun et al., 2024] uses chain-of-retrieval prompts with ChatGPT to generate synthetic data for fine-tuning.

Efficient encoder-based approaches target both accuracy and scalability. GLiREL [Boylan et al., 2025] adapted the GLiNER approach to relation classification, encoding relation labels alongside text in a shared bidirectional transformer and scoring entity-pair representations against relation-type embeddings. GLiREL achieved state-of-the-art results on Wiki-ZSL and FewRel while being significantly more efficient than template-based methods. However, GLiREL operates as a standalone relation classifier that requires pre-identified entities from an external NER model. GLiDRE [Armingaud and Besançon, 2025] extends the GLiNER approach to document-level relation extraction, achieving strong results on Re-DocRED. GLiNER2 treats relation extraction as a head-and-tail matching task after learning groups of relation representations. While it works without extracted entities, it can't be limited to selected entity types, making it an open-relation extraction approach.

2.4 Joint Entity and Relation Extraction with Encoder Models

The intersection of efficient encoder models and joint extraction remains underexplored. While GLiNER multi-task [Stepanov and Shtopko, 2024] supports relation extraction by concatenating source entity and relation as a label (e.g., "Bill Gates | founded"), this formulation reduces RE to a span extraction problem and does not explicitly model entity pairs. GraphER [Zaratiana et al., 2024a] provides true joint extraction but operates in a supervised setting with fixed entity and relation types. Our work, GLiNER-Relex, bridges this gap by providing zero-shot joint NER and RE within a single efficient encoder model.

3 Method

3.1 Overview

GLiNER-Relex extends the GLiNER architecture to jointly perform named entity recognition and relation extraction. The model takes as input a text sequence along with user-specified entity type labels and relation type labels, and produces both entity spans with their types and relation triplets connecting entity pairs. The architecture consists of five main components: (1) a shared encoder

that jointly processes text, entity labels, and relation labels; (2) a span representation layer for entity extraction; (3) an entity pair construction module with optional adjacency-guided pair selection; (4) a relation scoring layer; and (5) a multi-task training objective that jointly optimizes entity, adjacency, and relation losses.

Figure 1: Overview of the GLiNER-Relex architecture.

3.2 Input Representation

Given a text sequence $T = (t_0, t_1, \ldots, t_N)$, a set of entity type labels $\mathcal{Y}_E = \{e_1, e_2, \ldots, e_K\}$, and a set of relation type labels $\mathcal{Y}_R = \{r_1, r_2, \ldots, r_M\}$, we construct a unified input sequence by concatenating

three prompted segments:

$$X = [\underbrace{\texttt{[ENT]}\ e_1\ \texttt{[ENT]}\ e_2\ \cdots\ \texttt{[ENT]}\ e_K}_{\text{entity types prompt}},\ \underbrace{\texttt{[REL]}\ r_1\ \texttt{[REL]}\ r_2\ \cdots\ \texttt{[REL]}\ r_M}_{\text{relation labels prompt}},\ \underbrace{\texttt{[SEP]}\ t_0\ t_1\ \cdots\ t_N}_{\text{input sentence}}] \quad (1)$$

Each entity type label is preceded by a special `[ENT]` delimiter token, and each relation type label is preceded by a special `[REL]` delimiter token. This layout places entity type labels and relation type labels into a shared context window with the input text, enabling cross-attention between all three components within the transformer encoder. The `[ENT]` and `[REL]` tokens' hidden representations after encoding serve as the entity type and relation type embeddings used for downstream scoring.

Throughout the paper, we use $\mathcal{Y}_E$ and $\mathcal{Y}_R$ to denote the sets of entity and relation *type labels* provided at inference time, and $\mathcal{E}$ to denote the set of *recognized entity spans* produced by the model (introduced in Section 3.5). These are distinct objects and are used consistently throughout.

3.3 Shared Encoder

The unified input sequence is processed by a bidirectional transformer encoder f_{enc} (DeBERTa-v3 in our implementation):

$$H = f_{\text{enc}}(X) \quad (2)$$

From the contextualized hidden states H, we extract three sets of representations:

- **Word embeddings** $H_T = \{h_{t_i}\}_{i=0}^{N}$: Representations for each word in the input text, obtained by aggregating subword token embeddings.
- **Entity type embeddings** $H_E = \{h_{e_k}\}_{k=1}^{K}$: Representations for each entity type label, extracted at the positions of the corresponding `[ENT]` delimiter tokens.
- **Relation type embeddings** $H_R = \{h_{r_m}\}_{m=1}^{M}$: Representations for each relation type label, extracted at the positions of the corresponding `[REL]` delimiter tokens.

The word embeddings are optionally passed through a bidirectional LSTM layer for additional sequence modeling:

$$H'_T = \text{BiLSTM}(H_T) \quad (3)$$

3.4 Entity Extraction

Following the standard GLiNER span-based approach, we construct span representations for all candidate spans up to a maximum width W:

$$s_{i,j} = \text{SpanRep}(H'_T[i:j]) \quad (4)$$

where SpanRep combines start and end token representations with learned width embeddings. Entity type embeddings are projected through a dedicated layer:

$$\hat{h}_{e_k} = \text{Proj}_E(h_{e_k}) \quad (5)$$

Entity scores are computed via dot-product similarity between span and entity type representations:

$$\text{score}_{\text{ent}}(i, j, k) = s_{i,j} \cdot \hat{h}_{e_k} \tag{6}$$

Entities are decoded using greedy span selection with a confidence threshold τ_E.

3.5 Entity Pair Construction

After entity extraction, the model must determine which pairs of recognized entities to evaluate for relations. Let $\mathcal{E}$ denote the set of recognized entities. GLiNER-Relex supports two entity pair construction strategies, selected via configuration.

All-pairs enumeration. The simplest approach enumerates all ordered pairs of recognized entities. Given $|\mathcal{E}|$ entities, this produces $|\mathcal{E}| \times (|\mathcal{E}| - 1)$ candidate pairs. While exhaustive, this strategy scales quadratically and is best suited for sentences with a moderate number of entities. This is the strategy used in the released GLiNER-Relex checkpoint (Section 3.9).

Adjacency-guided selection. To reduce the number of candidate pairs, the framework optionally includes a *RelationsRepLayer* that predicts a soft adjacency matrix $\hat{A} \in [0, 1]^{|\mathcal{E}| \times |\mathcal{E}|}$ over entity span representations. A pair mask zeros out entries involving padded entities: $\hat{A}_{a,b} \leftarrow \hat{A}_{a,b} \cdot m_a \cdot m_b$. The layer supports six interchangeable decoder architectures:

- *Dot-product:* $\hat{A}_{a,b} = \sigma(s_a^\top s_b)$, optionally with L_2-normalization (cosine similarity). Parameter-free baseline.
- *Bilinear:* Projects entities via $W_P \in \mathbb{R}^{D \times d_L}$ and scores $\hat{A}_{a,b} = \sigma(z_a^\top z_b)$ where $z = W_P^\top s$, decoupling adjacency from span representations.
- *MLP:* Concatenates pairs and applies a two-layer MLP, $\hat{A}_{a,b} = \sigma(\text{MLP}([s_a; s_b]))$, enabling asymmetric and nonlinear interactions.
- *Attention:* Multi-head self-attention over entities, with attention weights averaged across heads to form $\hat{A}$.
- *GCN:* Computes an initial dot-product adjacency, applies a graph convolutional layer with symmetric normalization ($\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$) to refine representations via message-passing, then predicts the final adjacency from the updated features.
- *GAT:* Multi-head attention updates entity representations, which are then projected and scored bilinearly, combining contextual refinement with a learnable output space.

Entity pairs with $\hat{A}_{a,b}$ above a threshold τ_A are retained for relation classification, effectively pruning unlikely pairs before the more expensive relation scoring step. During training with ground-truth adjacency labels, this component is supervised with a dedicated adjacency loss (Section 3.7).

The six decoders described above are framework-level options supported by the GLiNER-Relex codebase. The released checkpoint uses the all-pairs enumeration strategy and does not activate any adjacency decoder; a systematic ablation of decoder architectures is left to future work (Section 5.4).

For each selected entity pair (a, b), the head and tail span representations s_a and s_b are extracted from the entity representations for downstream relation scoring.

3.6 Relation Scoring

Given selected entity pairs with head representations s_a and tail representations s_b, along with relation type embeddings $H_R = \{h_{r_m}\}_{m=1}^{M}$ from the shared encoder, the model scores each entity pair against each candidate relation type. The GLiNER-Relex framework implements two families of relation scoring mechanisms.

Pair representation layer. In this approach, head and tail entity representations are concatenated and projected through an MLP layer to produce a unified pair representation:

$$p_{a,b} = \text{MLP}([s_a; s_b]) \tag{7}$$

where $[s_a; s_b] \in \mathbb{R}^{2D}$ is the concatenation and $\text{MLP} : \mathbb{R}^{2D} \to \mathbb{R}^{D}$ projects the pair back to the shared embedding dimension via a linear layer with dropout. The relation score is then computed as a dot product between the pair representation and the relation type embedding:

$$\text{score}_{\text{rel}}(a, b, m) = p_{a,b} \cdot h_{r_m} \tag{8}$$

This formulation encourages the model to learn a shared semantic space in which entity pair representations are close to the embeddings of their corresponding relation types. Since both entity pairs and relation labels are encoded jointly by the shared transformer, the dot-product scoring enables zero-shot generalization to unseen relation types specified through natural language descriptions at inference time.

Knowledge graph–inspired triple scoring layers. As an alternative, the framework supports a family of triple scoring functions $f(s_a, h_{r_m}, s_b) \to \mathbb{R}$ drawn from the knowledge graph embedding literature [Bordes et al., 2013, Yang et al., 2015, Trouillon et al., 2016]. Each scoring function models the interaction between head entity, relation, and tail entity representations using a distinct geometric or algebraic assumption. The implemented variants include:

All triple scoring variants operate on the same entity and relation representations produced by the shared encoder. Each scoring function receives the head, relation, and tail embeddings and produces a scalar compatibility score, computed over all entity pair–relation type combinations in a single batched operation.

In our experiments, we found that the pair representation layer with MLP projection achieves the best balance of accuracy and efficiency, and it is used in the released model. The knowledge graph–inspired layers offer a richer space of inductive biases that may benefit specialized applications, such as settings where relation symmetry, transitivity, or compositionality are important structural priors.

3.7 Training Objective

The model is trained with a multi-task objective that combines up to three loss components:

$$\mathcal{L} = \lambda_E \mathcal{L}_{\text{ent}} + \lambda_A \mathcal{L}_{\text{adj}} + \lambda_R \mathcal{L}_{\text{rel}} \tag{9}$$

where $\mathcal{L}_{\text{ent}}$ is the entity extraction loss, $\mathcal{L}_{\text{adj}}$ is the optional adjacency matrix loss (present only when the adjacency-guided pair selection is used), and $\mathcal{L}_{\text{rel}}$ is the relation classification loss. The coefficients λ_E, λ_A, and λ_R control the relative contribution of each component. The specific values used for the released checkpoint are reported in Section 3.9.

All losses use focal loss [Lin et al., 2017] with optional negative sampling to handle the severe class imbalance inherent in both tasks:

$$\mathcal{L}_{\text{focal}}(p, y) = -\alpha(1 - p_t)^{\gamma} \log(p_t) \tag{10}$$

where $p_t = p$ if $y = 1$ and $p_t = 1 - p$ otherwise, α is a balancing factor, and γ is the focusing parameter. When $\gamma = 0$ the focal loss reduces to α-balanced binary cross-entropy; we retain the focal-loss interface so that γ can be increased in future training runs without changes to the optimization pipeline.

3.8 Training Data

Following the synthetic data generation paradigm established in prior GLiNER work [Stepanov and Shtopko, 2024], we construct training data through a two-stage annotation pipeline using LLMs applied to text sampled from FineWeb [Penedo et al., 2024].

Stage 1: Large-scale pre-training data. We sampled a large corpus of text from FineWeb, segmented it into individual sentences, and annotated approximately 1 million sentences using Qwen3-32B [Yang et al., 2025]. In addition, we annotated 50,000 full-length texts (without sentence splitting) to expose the model to document-level context. The sentence-level and document-level annotations were then mixed into a unified training set. Each training instance contains tokenized text, entity spans with type labels, and relation triplets linking entity pairs with relation type labels, aligned with the GLiNER input format.

Stage 2: High-quality fine-tuning data. To further improve model quality, we curated a smaller, high-quality dataset of approximately 3,000 examples using a multi-step annotation pipeline. First, we sampled texts from FineWeb and applied a general-purpose NER model to extract coarse entity types (e.g., person, organization, location). We filtered for entity-rich passages, retaining only texts with a sufficient density of recognized entities. Next, the selected texts were annotated using Gemini [Google, 2025] in two passes: (1) an extraction pass, where the model was prompted to identify entities and relations, with relations drawn from 12 pre-defined semantic groups (e.g., affiliation, spatial, causal) that serve as high-level categories without strictly limiting the verbalized relation labels; and (2) a correction pass, where the model reviewed and refined the extracted entities and relations to improve annotation consistency and accuracy. This high-quality dataset was used for final fine-tuning of the model.

3.9 Implementation and Training Details

The released GLiNER-Relex checkpoint uses a DeBERTa-v3-large shared encoder followed by a bidirectional LSTM with hidden size 1024. The model supports input sequences of up to 2048 words and candidate spans of up to $W = 12$ words, which covers the vast majority of entity mentions in the training data.

For entity-pair construction (Section 3.5), the released model uses the *all-pairs enumeration* strategy; the adjacency-guided selection path and the six decoder architectures described in Section 3.5 are supported by the framework but are not active in the released checkpoint. Accordingly, the adjacency loss is omitted from the training objective ($\lambda_A = 0$) and the effective objective reduces to $\mathcal{L} = \lambda_E \mathcal{L}_{\text{ent}} + \lambda_R \mathcal{L}_{\text{rel}}$ with $\lambda_E = \lambda_R = 1.0$. Both component losses use the focal-loss formulation of Section 3.7 with $\alpha = 0.75$ and $\gamma = 0$, which reduces to α-balanced binary cross-entropy.

Training follows the two-stage pipeline described in Section 3.8. **Stage 1** (large-scale pre-training on the approximately one million synthetically annotated sentences plus 50,000 document-level examples) runs for a single epoch with AdamW, a warmup ratio of 0.05, batch size 8, and differential learning rates of $1{\times}10^{-5}$ for the encoder and $5{\times}10^{-5}$ for the task-specific layers (span representation, pair representation, and relation projection). **Stage 2** (fine-tuning on the approximately 3,000

high-quality Gemini-annotated examples) runs for 5 epochs with the same optimizer, warmup ratio, and batch size, but with reduced learning rates of 3×10^{-6} for the encoder and 5×10^{-6} for the task-specific layers. The lower Stage-2 learning rates are chosen to preserve the broad zero-shot capabilities learned in Stage 1 while refining the model on the higher-quality annotations.

At inference, we use an entity confidence threshold $\tau_E = 0.3$ and a relation confidence threshold $\tau_R = 0.5$, consistent with the default values exposed in the public API (Section 5.2). Table 1 summarizes all hyperparameters for reproducibility.

Component	**Hyperparameter**	**Value**
Encoder	Backbone	DeBERTa-v3-large
	Max sequence length	2048 words
	BiLSTM hidden size	1024
Span encoder	Max span width W	12 words
Pair construction	Strategy	all-pairs enumeration
	Adjacency decoder	none
Loss	Focal α	0.75
	Focal γ	0
	λ_E (entity)	1.0
	λ_A (adjacency)	0 (disabled)
	λ_R (relation)	1.0
Stage 1 (pre-training)	Optimizer	AdamW
	Encoder learning rate	1×10^{-5}
	Task-specific layers learning rate	5×10^{-5}
	Warmup ratio	0.05
	Batch size	8
	Epochs	1
Stage 2 (fine-tuning)	Optimizer	AdamW
	Encoder learning rate	3×10^{-6}
	Task layer learning rate	5×10^{-6}
	Warmup ratio	0.05
	Batch size	8
	Epochs	5
Inference	Entity threshold τ_E	0.3
	Relation threshold τ_R	0.5

Table 1: Hyperparameters for the released GLiNER-Relex checkpoint.

4 Experiments

4.1 Benchmarks

We evaluate GLiNER-Relex on four standard relation extraction benchmarks:

CoNLL04 [Carreras and Màrquez, 2004]: A dataset from news articles annotated with entity types (Person, Organization, Location, Other) and relation types (Located-In, Work-For, OrgBased-In, Live-In, Kill). It is commonly used for evaluating joint entity and relation extraction models.

DocRED [Yao et al., 2019]: A large-scale document-level relation extraction dataset constructed from Wikipedia. It contains 96 relation types and requires reasoning across multiple sentences

within a document to identify relations, making it substantially more challenging than sentence-level benchmarks.

FewRel [Han et al., 2018]: A few-shot relation classification benchmark with 100 relation types derived from Wikidata. We evaluate in the zero-shot setting where the model must classify relations for types not seen during training.

CrossRE [Bassignana et al., 2022]: A cross-domain relation extraction dataset spanning multiple domains (AI, literature, music, politics, science, news). It is designed to evaluate the transferability of RE models across different textual domains.

4.2 Baselines

We compare GLiNER-Relex against:

- **GLiREL** [Boylan et al., 2025]: A GLiNER-family model specialized for zero-shot relation classification. GLiREL operates on pre-identified entities rather than performing end-to-end extraction; we evaluate it by supplying ground-truth entity spans and types from each benchmark, so the model predicts only which relation (if any) holds over each entity pair. This gives GLiREL an upper-bound-style advantage relative to joint systems, and we include it to characterize the performance ceiling of specialized relation classifiers when NER errors are fully removed.
- **GLiNER2** [Zaratiana et al., 2025]: The multi-task GLiNER model from Fastino Labs, which supports NER, text classification, and structured extraction. We evaluate its relation extraction capability through the schema-driven interface.
- **GPT-5-mini**: OpenAI's compact large language model, evaluated in a zero-shot prompting setup with structured output specifications for relation extraction.

4.3 Evaluation Protocol

All models are evaluated in a zero-shot setting where no training examples from the target datasets are provided. We report Micro-F1 scores for relation extraction, measuring the overall precision-recall balance across all relation types. For GLiNER-Relex, inputs exceeding the model's maximum sequence length of 2048 words (relevant primarily for a small number of long DocRED documents) are truncated at the document end; no sliding-window aggregation is performed, so relations spanning beyond the truncation point are not recoverable.

Entity handling differs across systems and is made explicit in Table 2. GLiNER-Relex and GLiNER2 perform end-to-end extraction from raw text. GPT-5-mini receives the list of candidate entity types in the prompt and produces entities and relations jointly in its structured output. GLiREL, as a dedicated relation classifier, is evaluated with ground-truth entity spans and types supplied as input; its scores therefore reflect relation-classification performance conditional on perfect NER rather than end-to-end extraction.

4.4 Results

Table 2 presents the zero-shot relation extraction results across all four benchmarks.

Model	Entities	CoNLL04	DocRED	FewRel	CrossRE	Avg.
GLiREL†	gold	4.5	2.4	**24.0**	1.4	8.1
GLiNER2	predicted	32.9	11.7	20.8	6.0	17.8
GPT-5-mini	prompted	**42.4**	18.6	15.0	12.4	22.1
GLiNER-Relex (ours)	predicted	40.4	**31.3**	12.5	**18.1**	**25.6**

Table 2: Zero-shot relation extraction performance (Micro-F1 %) on four benchmarks. Bold indicates the best performance per dataset across end-to-end systems. The "Entities" column indicates how each model obtains entity spans at inference time: *predicted* (joint extraction from raw text), *prompted* (LLM extracts entities and relations jointly from natural-language prompt), or *gold* (ground-truth entities supplied as input). The "Avg." column reports the unweighted mean Micro-F1 across the four benchmarks. †GLiREL is a dedicated relation classifier evaluated with ground-truth entity spans and types, and is not directly comparable to the end-to-end systems; we include it as an upper-bound reference for specialized relation classification.

GLiNER-Relex demonstrates strong performance across the evaluated benchmarks, achieving the best Micro-F1 on two out of four datasets among end-to-end systems and competitive results on the remaining two.

On **CoNLL04**, GLiNER-Relex achieves 40.4% Micro-F1, closely approaching GPT-5-mini (42.4%) and substantially outperforming GLiNER2 (34.1%). This demonstrates that our unified architecture can effectively capture sentence-level relations while maintaining the efficiency advantages of encoder-based models.

On **DocRED**, GLiNER-Relex achieves the highest performance at 31.3%, outperforming GPT-5-mini (18.6%) by a large margin of 12.7 percentage points and GLiNER2 (12.4%) by 18.9 points. This result is particularly significant because DocRED requires document-level reasoning across multiple sentences, suggesting that the shared encoder representations in GLiNER-Relex effectively capture long-range dependencies between entities.

On **FewRel**, GLiNER-Relex achieves 12.5%, trailing both GPT-5-mini (15.0%) and GLiNER2 (16.8%). GLiREL, which is evaluated with gold entities and was designed specifically for the sentence-level, entity-pair-aligned format that FewRel exemplifies, achieves the highest score at 23.9%. The relatively lower performance of end-to-end systems here reflects two factors: FewRel's 100 fine-grained Wikidata relation types challenge the zero-shot generalization of the relation-type embeddings, and the benchmark's design favors models with access to pre-identified entity pairs—an advantage that GLiREL receives by construction.

On **CrossRE**, GLiNER-Relex achieves the best performance at 18.1%, substantially outperforming GPT-5-mini (12.4%), GLiNER2 (4.9%), and GLiREL (1.4%). This result highlights the model's strong cross-domain transferability, a direct benefit of the zero-shot formulation where relation types are specified through natural language descriptions.

A consistent pattern emerges from the GLiREL results: despite the advantage of receiving ground-truth entities, GLiREL's performance drops sharply outside of FewRel's narrow sentence-level format—to 4.5% on CoNLL04, 2.2% on DocRED, and 1.4% on CrossRE. This suggests that the performance gap between specialized and unified relation extractors is not primarily driven by NER errors but by the distributional and structural assumptions baked into each architecture. Joint modeling with broad zero-shot pre-training, as in GLiNER-Relex, transfers more reliably across domains and granularities.

Restricting the comparison to end-to-end systems for apples-to-apples averaging, GLiNER-Relex's mean Micro-F1 across the four benchmarks is 25.6%, compared to 22.1% for GPT-5-mini

and 17.1% for GLiNER2. The model particularly excels on benchmarks requiring document-level reasoning (DocRED) and cross-domain generalization (CrossRE).

5 Discussion

5.1 Key Benefits

GLiNER-Relex unifies capabilities that prior approaches in the GLiNER ecosystem offer only in isolation. Where GraphER [Zaratiana et al., 2024a] requires supervised training on fixed types, GLiREL [Boylan et al., 2025] depends on an external NER pipeline, and GLiNER multi-task [Stepanov and Shtopko, 2024] reduces RE to span extraction via label concatenation, GLiNER-Relex jointly extracts entities and relations in a single forward pass with explicit entity-pair modeling and zero-shot generalization to arbitrary types.

This design yields three practical benefits. First, the unified architecture eliminates error propagation between separate NER and RE stages, as both tasks share representations within the same encoder. Second, the zero-shot formulation allows users to adapt the model to new domains by simply specifying entity and relation type labels as natural language strings, without any retraining. Third, the encoder-based architecture is orders of magnitude faster than autoregressive LLM-based extraction, making it well suited for latency-sensitive and resource-constrained deployments. The benchmark results confirm these advantages: GLiNER-Relex achieves the highest average Micro-F1 across all four datasets (25.6% vs. 22.1% for GPT-5-mini and 17.1% for GLiNER2), with particularly strong performance on document-level (DocRED) and cross-domain (CrossRE) tasks.

To quantify the efficiency advantage of the encoder-based design, we benchmarked GLiNER-Relex against GPT-5-mini on a held-out set of 50 documents sampled from FineWeb (mean length 288 words, approximately 360 sub-word tokens), annotated with a DocRED-style schema comprising 6 entity types and 50 relation labels. GLiNER-Relex was executed on a single NVIDIA L4 GPU with batch size one; GPT-5-mini was accessed through the OpenAI API with default reasoning effort and structured JSON output, and per-request timings include network latency. Under these conditions, GLiNER-Relex achieved a mean per-document latency of 0.9 s (throughput 1.11 docs/sec), while GPT-5-mini averaged 64 s per document (throughput 0.016 docs/sec)—a throughput advantage of approximately 70× in favor of the encoder-based model. This gap reflects both architectural differences and the reasoning-token overhead inherent to GPT-5-mini; reducing the reasoning effort would narrow the margin at some cost to extraction quality. The same workload incurs a non-trivial per-token cost on the hosted API, while GLiNER-Relex runs entirely on local infrastructure. These measurements substantiate the efficiency claim quantitatively and make GLiNER-Relex well-suited to high-volume pipelines such as knowledge-graph construction over corpora, where per-document extraction cost compounds rapidly across millions of documents.

5.2 Usage

GLiNER-Relex is released as an open-source model with a simple Python API that enables joint entity and relation extraction with user-defined types:

```
from gliner import GLiNER

model = GLiNER.from_pretrained(
    "knowledgator/gliner-relex-large-v1.0"
)
```

```
entity_labels = ["location", "person", "date", "structure"]
relation_labels = ["located in", "designed by", "completed in"]

text = ("The Eiffel Tower, located in Paris, France, "
        "was designed by engineer Gustave Eiffel "
        "and completed in 1889.")

entities, relations = model.inference(
    texts=[text],
    labels=entity_labels,
    relations=relation_labels,
    threshold=0.3,
    relation_threshold=0.5,
    return_relations=True,
    flat_ner=False
)
```

The `inference` method accepts entity and relation type labels as plain strings, with separate confidence thresholds for entity recognition (`threshold`) and relation extraction (`relation_threshold`). Setting `return_relations=True` enables joint extraction, returning both entity spans with types and relation triplets linking entity pairs. The `flat_ner` parameter controls whether overlapping entity spans are permitted.

5.3 Applications to GraphRAG

A particularly promising application is in Graph-based Retrieval-Augmented Generation (GraphRAG) pipelines [Edge et al., 2025]. GraphRAG constructs a knowledge graph from a document corpus and leverages graph structure—community detection, summarization, multi-hop traversal—to answer complex queries that require synthesizing information across passages. Current implementations typically rely on LLMs for extraction, which introduces significant computational cost at scale.

GLiNER-Relex offers an attractive alternative in this setting: as an encoder-based model, it is substantially faster than autoregressive LLMs (quantified in Section 5.1), which could enable knowledge graph construction over large corpora within tighter time and cost budgets. Its zero-shot capabilities allow domain-specific extraction without fine-tuning. We view GLiNER-Relex as a promising extraction backbone for GraphRAG systems in latency-sensitive, resource-constrained, or privacy-sensitive deployment scenarios, and the same model can be reused at query time to parse user questions into entity mentions for graph traversal. A systematic comparison of GraphRAG pipelines built on GLiNER-Relex versus LLM-based extractors—measuring end-to-end answer quality, graph coverage, and total cost—is an interesting direction that we leave to future work.

5.4 Limitations and Future Directions

GLiNER-Relex has several limitations. The zero-shot performance does not yet match fully supervised models or frontier LLMs on all benchmarks. Our experiments indicate room for improvement, particularly with large numbers of fine-grained relation types where the embedding space may require hierarchical or prototype-based representations.

Precision degrades in entity-dense passages: all-pairs enumeration produces quadratic candidate pairs, increasing spurious predictions. Although adjacency-guided selection mitigates this, dense entity graphs remain challenging for domains such as biomedicine, legal text, and financial reports.

Long document extraction is limited by two compounding factors. First, many recent encoder models are pre-trained on sequences shorter than 512 tokens, which limits their generalization to longer inputs. Second, as document length grows, so does the number of recognized entities, and the number of candidate entity pairs increases quadratically. This places increasing pressure on the model's fixed-dimensional embedding space, which must encode rich semantic and relational information for a rapidly growing set of combinations.

These limitations suggest several future directions: (1) extending the bi-encoder architecture [Stepanov et al., 2026] to decouple the type vocabulary from the input context; (2) hierarchical encoding or cross-chunk attention for long documents; (3) integration with entity linking through GLinker for end-to-end knowledge graph construction; (4) a systematic ablation of the six adjacency-decoder architectures described in Section 3.5, which are supported by the framework but inactive in the current released checkpoint; and (5) end-to-end evaluation of GraphRAG pipelines built on GLiNER-Relex versus LLM-based extractors.

6 Conclusion

We introduced GLiNER-Relex, a unified framework for joint named entity recognition and relation extraction that extends the GLiNER architecture with a dedicated relation extraction module. Our model achieves competitive zero-shot performance on standard relation extraction benchmarks, outperforming both GLiNER2 and GPT-5-mini on document-level and cross-domain benchmarks, while maintaining the computational efficiency characteristic of encoder-based models. The model's simple inference API allows users to specify arbitrary entity and relation types through natural language labels, enabling zero-shot extraction across diverse domains without retraining. GLiNER-Relex provides an efficient and accessible solution for extracting structured knowledge from unstructured text, with applications spanning knowledge graph construction, document understanding, and information extraction across domains including biomedicine, law, enterprise knowledge management, and financial intelligence. Being much more efficient than LLMs and having competitive zero-shot capabilities make GLiNER-Relex a promising choice for building knowledge graphs that power RAG systems, enabling multi-hop question answering.

References

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. FASTUS: A finite-state processor for information extraction from real-world text. In *Proceedings of IJCAI*, 1993.

R. Armingaud and R. Besançon. GLiDRE: Generalist lightweight model for document-level relation extraction. *arXiv preprint arXiv:2508.00757*, 2025.

E. Bassignana, V. Basile, and M. Polignano. CrossRE: A cross-domain dataset for relation extraction. In *Findings of EMNLP*, 2022.

G. Bogdanov et al. NuNER: Entity recognition encoder pre-training via LLM-annotated data. *arXiv preprint arXiv:2402.15343*, 2024.

A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NeurIPS*, pages 2787–2795, 2013.

J. Boylan, C. Hokamp, and D. Gholipour Ghalandari. GLiREL – generalist model for zero-shot relation extraction. In *Proceedings of NAACL*, pages 8230–8245, 2025.

X. Carreras and L. Màrquez. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL*, 2004.

C.-Y. Chen and C.-T. Li. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of NAACL*, 2021.

Y.K. Chia, L. Bing, S. Poria, and L. Si. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of ACL*, 2022.

D. Dai, X. Xiao, Y. Lyu, S. Dou, Q. She, and H. Wang. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of AAAI*, volume 33, pages 6300–6308, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, 2019.

S. Ding, P. Xu, Z. Liu, and D. Barbosa. GNER: A generative model for named entity recognition. *arXiv preprint arXiv:2406.01085*, 2024.

M. Eberts and A. Ulges. Span-based joint entity and relation extraction with transformer pre-training. 2019.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization, 2025. URL `https://arxiv.org/abs/2404.16130`.

T.-J. Fu, P.-H. Li, and W.-Y. Ma. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of ACL*, pages 1409–1418, 2019.

J. Gong and H. Eldardiry. Prompt-based zero-shot relation extraction with semantic knowledge augmentation. *arXiv preprint arXiv:2112.04539*, 2024.

Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2025.

X. Han et al. FewRel: A large-scale supervised few-shot relation classification dataset. In *Proceedings of EMNLP*, 2018.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, 2001.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *Proceedings of NAACL*, pages 260–270, 2016.

Y. Lan, D. Li, Y. Zhang, H. Zhao, and G. Zhao. Modeling zero-shot relation classification as a multiple-choice problem. In *Proceedings of IJCNN*, pages 1–8, 2023.

O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of CoNLL*, pages 333–342, 2017.

G. Li, P. Wang, J. Liu, Y. Guo, K. Ji, Z. Shang, and Z. Xu. Meta in-context learning makes large language models better zero and few-shot relation extractors. *arXiv preprint arXiv:2404.17807*, 2024a.

X. Li, K. Chen, Y. Long, and M. Zhang. LLM with relation classifier for document-level relation extraction. *arXiv preprint arXiv:2408.13889*, 2024b.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

B. Lv, X. Liu, S. Dai, N. Liu, F. Yang, P. Luo, and Y. Yu. DSP: Discriminative soft prompts for zero-shot entity and relation extraction. In *Findings of ACL*, pages 5491–5505, 2023.

M. Miwa and M. Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of ACL*, 2016.

C. Möller and R. Usbeck. Incorporating type information into zero-shot relation extraction. In *Proceedings of the Text2KG Workshop at ESWC*, 2024.

A. Obamuyide and A. Vlachos. Zero-shot relation classification as textual entailment. In *Proceedings of the FEVER Workshop*, pages 72–78, 2018.

G. Penedo et al. The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

D. Roth and W.-t. Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL*, 2004.

O. Sainz, O. Lopez de Lacalle, G. Labaka, A. Barrena, and E. Agirre. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of EMNLP*, pages 1199–1212, 2021.

Y. Shang, H. Huang, and X. Mao. OneRel: Joint entity and relation extraction with one module in one step. In *Proceedings of AAAI*, pages 11285–11293, 2022.

I. Stepanov and M. Shtopko. GLiNER multi-task: Generalist lightweight model for various information extraction tasks. *arXiv preprint arXiv:2406.12925*, 2024.

I. Stepanov, M. Shtopko, D. Vodianytskyi, and O. Lukashov. The million-label NER: Breaking scale barriers with GLiNER bi-encoder. *arXiv preprint arXiv:2602.18487*, 2026.

D. Sui, X. Zeng, Y. Chen, K. Liu, and J. Zhao. Joint entity and relation extraction with set prediction networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):12438–12450, 2023.

Q. Sun, K. Huang, X. Yang, R. Tong, K. Zhang, and S. Poria. Consistency guided knowledge retrieval and denoising in LLMs for zero-shot document-level relation triplet extraction. *arXiv preprint arXiv:2401.13598*, 2024.

V.-H. Tran, H. Ouchi, H. Shindo, Y. Matsumoto, and T. Watanabe. Enhancing semantic correlation between instances and relations for zero-shot relation extraction. *Journal of Natural Language Processing*, 30(2):304–329, 2023.

T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *Proceedings of ICML*, pages 2071–2080, 2016.

J. Wang, Y. Shen, and L. Chen. InstructionNER: A multi-task instruction-based generative framework for few-shot NER. *arXiv preprint arXiv:2203.03903*, 2023.

J. Wang et al. UniRE: A unified label space for entity relation extraction. In *Proceedings of ACL*, 2021.

Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of COLING*, pages 1572–1582, 2020.

Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of ACL*, pages 1476–1488, 2020.

A. Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*, 2015.

Y. Yao et al. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of ACL*, 2019.

A. Yazdani, I. Stepanov, and D. Teodoro. GLiNER-BioMed: A suite of efficient models for open biomedical named entity recognition. *arXiv preprint arXiv:2504.00676*, 2025.

B. Yu, Z. Zhang, X. Shu, Y. Wang, T. Liu, B. Wang, and S. Li. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proceedings of ECAI*, pages 2282–2289, 2020.

U. Zaratiana, N. Tomeh, N. El Khbir, P. Holat, and T. Charnois. GraphER: A structure-aware text-to-graph model for entity and relation extraction. *arXiv preprint arXiv:2404.12491*, 2024a.

U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois. An autoregressive text-to-graph framework for joint entity and relation extraction. In *Proceedings of AAAI*, volume 38, pages 19477–19487, 2024b.

U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of NAACL*, pages 5364–5376, 2024c.

U. Zaratiana, G. Pasternak, O. Boyd, G. Hurn-Maloney, and A. Lewis. GLiNER2: Schema-driven multi-task learning for structured information extraction. In *Proceedings of EMNLP: System Demonstrations*, pages 130–140, 2025.

D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. In *Proceedings of EMNLP*, pages 71–78, 2002.

J. Zhao, W. Zhan, X. Zhao, Q. Zhang, T. Gui, Z. Wei, J. Wang, M. Peng, and M. Sun. RE-Matching: A fine-grained semantic matching method for zero-shot relation extraction. In *Proceedings of ACL*, pages 6680–6691, 2023.

H. Zheng, R. Wen, X. Chen, Y. Yang, Y. Zhang, Z. Zhang, N. Zhang, B. Qin, M. Xu, and Y. Zheng. PRGC: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of ACL*, pages 6225–6235, 2021.

S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of ACL*, pages 1227–1236, 2017.

Z. Zhong and D. Chen. A frustratingly easy approach for entity and relation extraction. In *Proceedings of NAACL*, 2021.

W. Zhou, S. Zhang, Y. Gu, M. Chen, and H. Poon. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *Proceedings of ICLR*, 2024.