Testing Extensions of GLiNER for Relation Extraction, Event Extraction, and Entity Descriptions in Historical and Fictional Narratives

Yassine Machta

Abstract

This project explores the extension of the GLiNER framework to support relation extraction, event extraction, and entity description generation, with a focus on historical and fictional narratives. We investigate how these tasks can be integrated into a unified pipeline, leveraging recent advances in large language models and neural entity recognition. Our ambition extends beyond traditional NER and relation extraction: we aim to extract and organize events, draft both physical and moral descriptions of entities, and ultimately structure texts into interconnected "hyper storylines" that capture the evolution of characters and events across a narrative. Some of our results show that current non-LLM based methods lack flexibility and require significant guidance, especially for relation extraction, but there is promising potential for more interpretative, LLM-driven approaches.

1 Introduction

Understanding complex narratives, such as those found in historical accounts and fiction, requires not only identifying entities but also extracting the relationships between them, recognizing key events, and generating meaningful descriptions. While Named Entity Recognition (NER) and relation extraction have seen significant progress, the integration of event extraction and entity description generation remains less explored.

This project aims to bridge that gap by developing a unified pipeline capable of extracting entities, their relations, and events, as well as drafting rich descriptions—both physical and moral—of characters. A central ambition is to move beyond isolated information extraction and toward modeling the narrative structure itself, organizing extracted information into "hyper storylines" that reflect the interplay of characters, events, and evolving relationships throughout a text. Such a framework opens new possibilities for computational literary analysis, digital humanities, and the study of complex narrative dynamics in both historical and fictional domains.

2 Background and State-of-the-Art Review

2.1 GLiNER & GLiNER Multi-task

2.1.1 Methodology and Ambitions

GLiNER (Generalist Lightweight Model for Named Entity Recognition) is designed to extract arbitrary entities using a compact encoder-based architecture, specifically a bidirectional transformer. Unlike traditional NER models limited to predefined entity types, GLiNER offers flexibility by allowing entity extraction based on user-defined instructions. This approach enables parallel entity extraction, enhancing efficiency compared to the sequential token generation in large language models (LLMs) [4].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

GLiNER multi-task extends this capability by addressing multiple information extraction tasks, including:

- Named Entity Recognition (NER)
- Open NER
- Relation Extraction
- Summarization
- Question-Answering
- Open Information Extraction

The model is trained using a combination of synthetic data generated by prompting LLaMA 3 8B on Wikipedia articles and high-quality manually curated NER datasets. This diverse training enables the model to generalize across various tasks effectively [4].

2.1.2 Results

GLiNER multi-task demonstrates strong performance across several benchmarks:

- **NER:** Achieved an average F1 score of 0.6276 across diverse domains, outperforming other GLiNER variants and NuNER_Zero-span in certain areas.
- **Question-Answering:** On the SQuAD2.0 dataset, it attained an exact match score of 87.72 and an F1 score of 91.99, closely rivaling UTC-DeBERTa-large-v2.
- **Summarization:** Outperformed other models on the CNN/DailyMail dataset with ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.2484, 0.0881, and 0.2279, respectively.
- **Relation Extraction:** Achieved an exact match of 82.5 and an F1 score of 87.36 on the FewRel dataset, surpassing other models like UTC-DeBERTa-large-v2.
- **Self-Training:** Implemented self-learning approaches that improved F1 scores by up to 2% on cross-domain NER benchmarks, particularly enhancing performance on initially weaker domains like AI.

2.2 ITER: Iterative Transformer-based Entity Recognition and Relation Extraction

ITER is an efficient encoder-only transformer model for joint entity recognition and relation extraction. Unlike autoregressive approaches that generate structured outputs sequentially and are computationally expensive, ITER performs the task in three parallelizable steps, greatly accelerating inference while maintaining or surpassing state-of-the-art performance. Traditional pipeline-based methods for relation extraction first identify entities and then classify relations, but suffer from error propagation. Joint models improve performance but are often autoregressive, limiting throughput.

ITER addresses these limitations by using a non-autoregressive, encoder-based architecture such as T5 or DeBERTa that generates structured outputs in three steps: span start detection, span pairing, and relation classification. Each step is parallelizable across the sequence, enabling high throughput. The model uses gated feed-forward networks to process contextualized token representations from the encoder and predicts both entities and relations in a single pass. Compared to span-based and table-filling approaches and autoregressive models, ITER avoids bottlenecks, achieving linear complexity in sequence length for entity detection and quadratic only in the number of entities for relation classification, which is typically much smaller than sequence length. ITER achieves state-of-the-art results on several benchmarks, including 71.9 F1 on ACE05 and 85.6 F1 on ADE for relation extraction, and 91.9 F1 on ACE05 and 92.2 F1 on ADE for named entity recognition. It processes up to 600 or more samples per second on a single consumer GPU, making it dramatically more efficient than autoregressive baselines, and demonstrates that encoder-only architectures can match or exceed the performance of autoregressive models for structured prediction tasks while being much more efficient.

2.3 GLiREL: Generalist Model for Zero-Shot Relation Extraction

GLiREL [3] is a generalist, lightweight model designed for efficient and accurate zero-shot relation extraction (ZSRE). Inspired by recent advances in zero-shot named entity recognition (NER) such as

GLiNER, GLiREL enables the prediction of arbitrary relation labels between multiple entities in a single forward pass, overcoming the inefficiency of previous approaches that required separate inputs for each entity pair and candidate label.

GLiREL leverages a bidirectional transformer encoder (e.g., DeBERTa V3-large) to jointly encode relation labels and input text. Entity representations are extracted and concatenated to form all possible entity pairs, which are then scored against candidate relation labels using a similarity-based approach in a shared latent space. This architecture allows GLiREL to classify all entity pair relations in parallel, making it significantly more efficient than auto-regressive LLMs or template-based methods.

A key innovation in GLiREL is its use of large-scale synthetic training data, generated by prompting LLMs (such as Mistral 7B-Instruct) to annotate diverse relation types over web-scale corpora. This enables robust zero-shot generalization to unseen relation labels at inference time. The model also supports document-level relation extraction and can be extended to coreference resolution tasks.

Empirical results on the Wiki-ZSL and FewRel benchmarks demonstrate that GLiREL achieves state-of-the-art performance in zero-shot relation classification, outperforming strong baselines such as TMC-BERT, MC-BERT, and RelationPrompt, as well as large LLMs like GPT-4o. Notably, GLiREL maintains high accuracy and throughput even as the number of unseen relation labels increases, and is orders of magnitude faster at inference than competing models.

GLiREL's architecture and training paradigm represent a significant step forward for scalable, flexible relation extraction in NLP pipelines, particularly in settings where new relation types must be handled without retraining.

2.4 Mistral LLMs

Mistral AI has developed a series of language models focusing on efficiency and performance. In this project we use a free mistral API key from la plateforme and use mistral large to try and improve the extracted entities, relation extraction and event extraction. Unfortunately, the API has a limited number of uses and experiments are limited by that. but it's quite fast.

3 Methodology

3.1 Named Entity Extraction

We employ the GLiNER model for Named Entity Recognition (NER) on historical and fictional texts. The input text is first split into manageable chunks using a sentence boundary-based splitting algorithm to ensure that each chunk does not exceed the model's maximum input length. Each chunk is processed by the GLiNER model, which predicts entities according to a predefined set of entity types (e.g., person, location, organization, artifact). The results from all chunks are aggregated, and a majority-voting deduplication strategy is applied to assign the most frequent label to each unique entity mention.

3.2 Entity Context Collection

For each unique entity, we collect all sentences from the text in which the entity appears. This mapping of entities to their contextual sentences is used in subsequent post-processing steps.

3.3 Entity Fusion via Substring Matching and LLM-based Equivalence

To address redundancy and ambiguity in the extracted entity list, we identify pairs of entities where one is a substring of the other (e.g., "Columbus" and "Christopher Columbus"). For each such pair, we construct a context by concatenating all sentences mentioning either entity.

We then use a Large Language Model (LLM) in a question-answering setup to determine if the two entities refer to the same real-world entity. Specifically, for each pair, we pose the following yes/no question to the LLM:

> Do the entities "ent1" and "ent2" refer to the same thing in this context? Answer yes or no.

If the LLM answers "yes", the two entities are merged into a single group. This process is repeated for all substring-related pairs, and groups are updated accordingly.

3.4 Canonical Entity Selection

After fusion, each group of equivalent entities is represented by a canonical entity, chosen as the mention with the most words. The most common label among the group's members is assigned as the canonical label. The final output is a deduplicated and disambiguated list of entities, each with its aliases and assigned type.

3.5 Relation Extraction

To extract semantic relations between entities, we employ several state-of-the-art neural models, including GLiNER Multitask, GLiREL, and ITER. Each model is evaluated on both fictional and historical texts.

GLiNER Multitask Relation Extraction We use the `GLiNERRelationExtractor` module, which is initialized with a multitask GLiNER model. For each input text, we specify a set of possible relation types (e.g., *parent*, *ruler of*, *located in*) and entity types (e.g., *person*, *organization*, *location*). The model predicts tuples of the form (head entity, relation, tail entity) for all detected relations in the text. This approach is applied to both fictional narratives and historical documents, with the set of relation types adapted to the domain.

GLiREL (spaCy Integration) We also experiment with the GLiREL model, integrated into a spaCy pipeline. The GLiREL component is configured with a set of allowed relation types and entity type constraints. The pipeline processes the text and outputs a list of predicted relations, each with a confidence score. This allows for flexible and extensible relation extraction, leveraging spaCy's efficient text processing capabilities.

ITER Model For further comparison, we utilize the ITERForRelationExtraction [2] model. The input text is tokenized and encoded, and the model generates both entity and relation predictions. Although the output format is less human-readable, this model provides an additional benchmark for relation extraction performance.

3.6 Summarization

To generate concise summaries of long texts, we use the `GLiNERSummarizer` module. For short texts, the summarizer is applied directly. For longer documents, we split the text into manageable chunks (based on sentence boundaries and a maximum word count), summarize each chunk individually, and concatenate the resulting summaries. This chunked summarization approach mitigates the model's input length limitations and produces a coherent overall summary.

3.7 Implementation Details

All models are run on GPU-enabled hardware for efficiency. Hyperparameters such as entity and relation type lists, confidence thresholds, and chunk sizes are tuned empirically for each task and dataset. The methodology is implemented in Python, leveraging libraries such as `gliner`, `glirel`, `spacy`, and `iter`.

4 Results

4.1 Named Entity Recognition (NER) with Base GLiNER

The base GLiNER model demonstrated fast and efficient entity extraction. However, due to the necessity of splitting long texts into manageable chunks, the model often lost context between chunks.

This led to duplicate entities, especially when the same entity was referenced by different names (e.g., first name in one chunk, last name in another). As a result, the deduplication process was imperfect, and some entities appeared multiple times under different aliases.

4.2 Entity Deduplication with Mistral

To address the duplication issue, we employed the Mistral LLM. For each pair of potentially duplicate entities, we provided the model with both entity names and all sentences in which they appeared as context. Mistral was able to accurately determine whether two mentions referred to the same real-world entity, effectively removing duplicates and improving the quality of the entity list.

4.3 GLiNER Question Answerer and Open Extractor

We also experimented with the GLiNER Question Answerer and Open Information Extractor modules. However, neither approach produced the desired results for entity equivalence or relation extraction in our use case. The models were unable to provide reliable yes/no answers or extract the necessary information for entity fusion.

4.4 Relation Extraction

GLiNER Multitask The GLiNER multitask relation extractor requires explicit specification of relation labels. This constraint makes it difficult to use in a fully automatic or zero-shot setting, as it presupposes knowledge of all possible relation types present in the text. For a more robust pipeline, an additional component would be needed to automatically extract and qualify relation labels before passing them to the extractor.

GLiREL The GLiREL model was able to detect relations, including some not explicitly present in the text. However, the results were inconsistent, and the confidence scores for detected relations were generally low. This limits the practical utility of GLiREL for high-precision relation extraction in our experiments.

ITER The ITER model occasionally produced relation outputs, but these were encoded as numerical identifiers. Unfortunately, documentation on decoding these identifiers into human-readable relation types was lacking, making the results difficult to interpret and use.

4.5 Limitations

Due to API usage limits, we were unable to fully explore the capabilities of the Mistral-based pipeline. This restricted the scale and depth of our experiments with entity deduplication and relation extraction using LLMs.

5 Conclusion and Future Work

Our experiments demonstrate that current neural methods for entity and relation extraction, such as GLiNER, GLiREL, and ITER, are fast and effective for basic tasks but remain quite inflexible. These models require explicit guidance, particularly in the form of predefined entity and relation labels. This specificity limits their ability to generalize or adapt to new domains without significant manual intervention. In contrast, large language models (LLMs) like Mistral offer greater flexibility and interpretative power, enabling them to resolve ambiguities and infer implied meanings that more rigid models cannot. However, LLM outputs are not always easily controlled or reliable, and their responses can vary significantly depending on prompt phrasing and context.

Looking forward, the ambition of this project is to advance the analysis of historical and fictional texts by not only extracting entities and their relations, but also by drafting rich descriptions (both physical and moral) and, crucially, by extracting and organizing events. The ultimate goal is to structure texts into "hyper storylines" [1], modeling the interplay of characters, events, and narrative arcs. This approach promises to open new avenues for computational literary analysis, digital humanities, and knowledge extraction from complex narrative sources.

The code and experiments for this project are available at: https://github.com/MachtaYassine/NLP_GliNER

References

[1] HyperStorylines Project, https://gitlab.inria.fr/ilda/hyperstorylines

[2] Moritz Hennen, Florian Babl, and Michaela Geierhos. Iterative Transformer-based Entity Recognition and Relation Extraction. University of the Bundeswehr Munich, Ludwig Maximilian University of Munich, 2023. https://github.com/fleonce/iter

[3] Jack Boylan, Chris Hokamp, and Demian Gholipour Ghalandari. GLiREL: Generalist Model for Zero-Shot Relation Extraction. Quantexa, 2024. https://github.com/jackboyla/glirel

[4] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. FI Group, LIPN, CNRS UMR 7030, France, 2023. https://github.com/urchade/GLiNER