Experimenting with GLiNER: Football Players Biographies Analyses

Aymeric TIBERGHIEN*
ENSAE Paris
Palaiseau
aymeric.tiberghien@ensae.fr

Abstract

This article present the results of an experiment over a small database of text entries around the same niche (Footballers Biographies) by the model GLiNER with the labels Person, Awards, Date, Competitions and Teams that need to be linked with entities from the entries.

1 Introduction

This article presents the results of an experiment based on the GLiNER model, first presented in *GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer* by **Urchade Zaratiana, Nadi Tomeh, Pierre Holat** and **Thierry Charnois**. The original study's Github repository can be found here. This short article will go through five main parts and then a conclusion. In order, the main parts will be: a brief state of the art on the topic, a short description of the original paper, a description and justification of the experiment, a walkthrough how the database for the experiment was created and finally an analysis of the results obtained by GLiNER on aforementioned database. We will then conclude on the relevance of the experiment.

2 Brief State of the Art

Traditional NER began with rule-based systems that relied on handcrafted patterns and gazetteers, achieving reasonable precision but struggling to generalize across domains or languages (Weischedel et al., 1996; Mikheev et al., 1999; Nadeau et al., 2006; Zamin and Oxley, 2011). To overcome these limitations, statistical sequence-labeling approaches (e.g. CRFs, HMMs) were introduced (Lafferty et al., 2001), later giving way to neural models that framed NER as token-tag prediction using BiLSTMs and transformers (Huang et al., 2015; Lample et al., 2016; Akbik et al., 2018). More recently, span-based methods have treated entity recognition as a classification problem over all text spans (Sarawagi and Cohen, 2004; Fu et al., 2021; Li et al., 2021; Zaratiana et al., 2022a,b,c, 2023), while alternative views have reframed NER as question answering or text-generation tasks (Li et al., 2019; Yan et al., 2021). The advent of large autoregressive LLMs unlocked "open" NER capable of extracting arbitrary entity types via natural-language prompts. InstructUIE fine-tuned FlanT5-11B on IE datasets to achieve strong zero-shot results (Wang et al., 2023), GoLLIE leveraged detailed annotation guidelines and CodeLLaMa for enhanced performance (Sainz et al., 2023), and UniversalNER distilled ChatGPT's capabilities into smaller models (Zhou et al., 2023). USM proposed a unified semantic matching framework for open extraction, further diversifying the landscape of promptable NER systems. While these LLM-based methods offer flexibility, they incur high compute and inference costs. GLiNER departs from autoregressive architectures by

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

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

framing open-type NER as span–type matching with a bidirectional transformer backbone. It embeds entity-type prompts alongside text, computes span representations via a lightweight feed-forward network, and scores all span–type pairs in parallel, yielding CPU-efficient, zero-shot extraction that outperforms both ChatGPT and fine-tuned LLMs across diverse benchmarks.

3 Description of the Article

GLiNER addresses the flexibility limitations of traditional NER systems and the high resource requirements of large autoregressive LLMs by introducing a compact, open-type model trained to recognize arbitrary entity types through span-type matching. Built on a bidirectional transformer encoder, GLiNER embeds natural-language entity-type prompts alongside the input text and computes span representations via a lightweight feed-forward network. At inference, it scores all span–type pairs in parallel, enabling rapid, bidirectional extraction without sequential token generation.

Architecturally, GLiNER consists of three core modules: (i) a pretrained BiLM (e.g., DeBERTa) to produce contextual token embeddings; (ii) a span representation module that applies a two-layer FFN to concatenated start/end token vectors; and (iii) an entity representation module that refines [ENT] token embeddings into type vectors. Matching scores between span and entity embeddings are computed via a dot-product followed by a sigmoid activation, framing NER as a binary classification over span–type pairs.

Training is performed on the diverse Pile-NER dataset, sampled from the Pile corpus and annotated with ChatGPT-generated entity types. GLiNER employs the AdamW optimizer with a learning rate of 1e-5 for backbone layers and 5e-5 for FFN layers, a 10% warmup, cosine decay schedule, and up to 30k steps. To improve robustness, negative span-type examples are introduced by sampling random entities within each batch, and entity order is shuffled with occasional dropout. The largest variant (GLiNER-L, 0.3B parameters) trains in approximately five hours on an A100 GPU.

In zero-shot evaluations, GLiNER-L achieves an average F1 of 55.6 on the Out-of-Domain NER Benchmark—surpassing both ChatGPT and fine-tuned LLMs—and even the smallest 50M-parameter model outperforms general-purpose models on this suite. Across 20 English NER datasets, GLiNER variants obtain an average F1 of 47.8, outpacing ChatGPT (36.5) and UniNER-7B (45.7), and dominate 13 out of 20 tasks. Multilingual zero-shot tests on MultiCoNER reveal that GLiNER-Multi surpasses ChatGPT in most languages, notably those using Latin scripts, despite being trained only on English examples.

By combining promptable flexibility with the efficiency of bidirectional encoders, GLiNER offers a practical alternative for resource-constrained scenarios, achieving state-of-the-art zero-shot NER performance while remaining lightweight and CPU-friendly. Future work will focus on extending support for discontinuous entities and refining evaluation metrics beyond strict span matches.

4 About the Experiment

4.1 Description

The experiment that is the subject of this paper is an extended test of GLiNER's capabilities on a Database made of 30 biographies of football players that were taken from Wikipedia. The labels that could be retrieved from the entities in the texts were **Person, Award, Date, Competitions** and **Teams**. The experiment is then a simple test of GLiNER performances on this database.

4.2 Justification

The justification behind this experiment lies in the usage example given in the original study's GitHub. Indeed, the authors invite us to try the GLiNER library through a little experiment on the Wikipedia biography of football player Cristiano Ronaldo. One thought that extending this test to 29 other biographies could be interesting to study the model's behaviour.

5 Database of the Experiment

The database is made of 30 biographies of football players that were taken from Wikipedia and the labelization was done by hand. Each text entry was studied in order to count how many unique person names, awards, competitions, teams and dates were mentioned in the text. This method is of course flawed but analysis of the results will take this margin of error into account. We then ran with each text entry given the aforementioned labels to see its performance. The lengths of the text entries vary throughout the whole database to see how the model reacts to different text lengths.

6 GLiNER Performances Analysis

This analysis section will be divided in two. First, we will go over some quantitative analyses of the model's performances over the database constructed earlier. Then, in a second qualitative analysis we will go over three text entries from the database and see how well the model performed over them individually and expose some leads of explanation.

6.1 Quantitative Analysis

For each of the five labels, we introduce a metric to measure the model's performance on them. The absolute difference between how many entities the model linked to each label and how many entities were linked to each label during the labellization process. Then, we have five metrics: **Person_Diff, Award_Diff, Date_Diff, Competitions_Diff** and **Teams_Diff**. Finally, we also take into consideration the length of each text entry. To put these results in context we can look at the

Metric	Person_Diff	Award_Diff	Date_Diff	Competitions_Diff	Teams_Diff
MAE	0.47	3	1.9	2.3	1.7
Standard Deviation	1	4	2.2	2	1.5

Table 1: Mean Absolute Error and their Standard Deviation for each Label

average number of entities that were linked to each label during the labellization stage: The main

Label	Person	Award	Date	Competitions	Teams
Average Number	2.4	4.8	7.1	6.9	5.7

Table 2: Average Number of Entities linked to each label

takeaways from these results are that the model is very efficient to recognize the names of people for sure. Then we can emit first hypotheses before the qualitative analysis to understand model behavior over the other four labels:

- Awards: knowledge of accolades about what is considered an award despite not being immediately followed by the word "award" (e.g. UEFA Team of the Season...) and ability to dissociate awards and records (e.g. a club's all-time top-scorer).
- Date : what the model can consider as a date, i.e. does it need more precision about an entity to consider it a date or does a single year suffices?
- Competitions and Teams: Some competitions and teams mentions could very well be mixed up by the model when juxtaposed (e.g. Premier League club Chelsea) while they were correctly separated during the labellization phase.

These hypotheses can be summed up by how the database was built overall: by hand by a single person. However, the following qualitative deep-dive into three entries will allow us to get more detail as to how the model behaves. Still on the quantitative side, we can try and see if there is any relationship between model performance and length of each text entry. The following plots show this relationship:

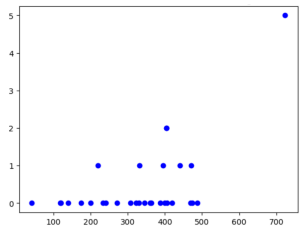

Figure 1: Peformance on Person Label vs Text Length

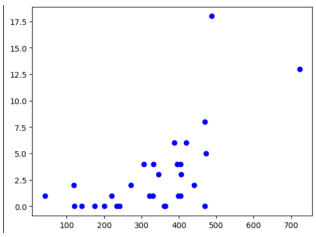

Figure 2: Performance on Awards Label vs Text Length

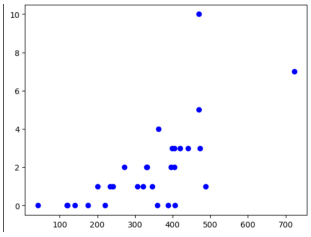

Figure 3: Performance on Date Label vs Text Length

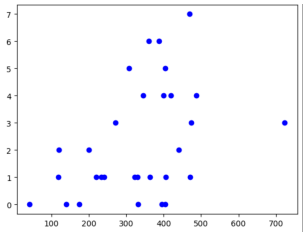

Figure 4: Performance on Competitions Label vs Text Length

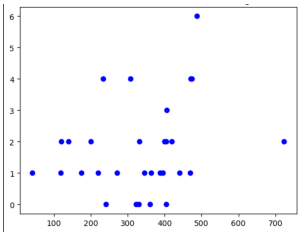

Figure 5: Performance on Teams Label vs Text Length

The main takeaway from these five plots is that there is a clear impoorement of model performance on each label when the texts lenghten. There also seems to be a problem on the singular entry that is above 500 words. We will deep dive into it next.

6.2 Qualitative Analysis

The average text length in the database is 330 words. We will deep dive into three examples to see how the model behaved on a very short entry (42 words), an average entry (331 words) and on the longest entry (723 words). We name each deep dive by the player the entry is about.

- Hugo Lloris: No errors besides in awards label where the model mixes an award and a record.
- Gonzalo Higuain:
  - Person and Teams: No errors.
  - Awards and Competitions: Two-way error, the model confuses a competition (Coppa Italia) with an award.
  - Dates: The analysis revealed an error in the database creation (only 8 dates had been noticed since then corrected to 10) which then causes the model to make two mistakes. No real explanation can be given since these two occurrences are in the same sentence, and said sentence has been correctly treated by the model to get the mention of a Team in the Teams label.
- Robert Lewandowski: The model seems to have stopped in the middle of the text (after 299 words) which leads to a huge difference between the database numbers and its result for every label. No explanation as to why it stopped, especially knowing previous analysis did not stopped despite being 332 words long. If we look at the part that was analyzed by the model, here are the results:
  - Person and Dates: No errors.
  - Awards: A two-way error with the competition Supercopa de España.
  - Competitions: previously mentioned two-way error, as well as one duplicate competition being counted twice and one competition completely overlooked (Bundesliga).
  - Teams: Two duplicate teams (Barcelona and Bayern Munich) being counted twice as well.

What we can say is that our fears about the awards label were confirmed but not for the right reasons since the entries we deep dove in did not comprehend mentions of aforementioned examples (UEFA Team of the Season etc..). Another common mistake made by the model on the awards topic is mixing up some competitions and awards which is a two-way error, penalizing model performance twice. There seem to have not been a huge problem in the Teams label detection in our three examples which contradicts the hypotheses we formulated earlier.

7 Conclusion

To conclude this paper, we can say that this experiment is quite conclusive on what GLiNER can do in a niche test environment. Despite the small sample size offered by the handmade database, we could already see some trends between model performance and text lengths. Experimenting with a bigger and more varied database could confirm these first takeaways.