

LELA: An End-to-end LLM-based Entity Linking Framework with Zero-shot Domain Adaptation

Samy Haffoudhi, Nikola Dobričić, Fabian Suchanek, Nils Holzenberger

Télécom Paris, Institut Polytechnique de Paris

{samy.haffoudhi, nikola.dobricic, fabian.suchanek, nils.holzenberger}@telecom-paris.fr

Abstract

Entity linking is a key component of many downstream NLP systems, yet existing approaches are often tied to the specific target knowledge bases and domains, limiting their real world application. In this paper, we extend LELA, a modular and domain-agnostic LLM-based entity disambiguation method, into a practical Python library that integrates zero-shot Named Entity Recognition (NER) – thereby providing a complete end-to-end pipeline for entity-linking in real-world usage. We provide experimental results validating LELA’s performance and robustness across diverse entity linking settings. In our demo, users can play with the system on their own input texts. All code is publicly available at <https://github.com/dig-team/LELA>, and a video is at <https://www.youtube.com/watch?v=WdupiRjLbR4>.

1 Introduction

Entity linking (EL) is the task of identifying and mapping ambiguous mentions of entities in a natural language text to reference entities in a knowledge base (KB). For example, the text can be:

France hosted the Olympics in Paris.

The KB contains each entity, associated with a short textual description, e.g.:

Paris (city)	Capital city of France
Paris (novel)	Novel by Emile Zola
France	Country in Europe
France Gall	French singer

Entity linking consists of two sub-tasks, *mention detection* (MD) and *entity disambiguation* (ED). Mention detection aims to identify the mentions of entities (here: “France”, “the Olympics”, and “Paris”). Mention detection often boils down to Named Entity Recognition (NER). Entity disambiguation, on the other hand, aims to map the identified mentions to their correct entity in the KB, if they exist (here: “France” to France and “Paris” to

```
from lela import Lela

# Choose each component of LELA
config = {
    "loader": {
        "name": "text" # or: pdf, docx, ...
    },
    "ner": {
        "name": "gliner", # or: regex, spacy
        "params": {"labels": ["person", "location"]},
    },
    "candidate_generator": {"name": "bm25"},
    # or: fuzzy, dense, openai_api_dense
    "reranker": {"name": "llama_server"},
    # or: none, cross_encoder_vllm...
    "disambiguator": {
        "name": "vllm",
        # or: first, openai_api, transformers
        "params": {"model_name": "Qwen/Qwen3-4B"},
    },
    "knowledge_base": {
        "name": "jsonl", "params": {"path": "my_kb.jsonl"}
    },
}

lela = Lela(config)

# Run the pipeline on a document
results = lela.run("docs/file1.txt")
```

Figure 1: LELA is designed for a modular use in Python.

Paris (city), with no mapping for “Olympics”). Entity Linking is an important preprocessing step in tasks such as information extraction [Martinez-Rodriguez *et al.*, 2020], knowledge-based question answering [Welty *et al.*, 2012], and knowledge graph completion [Ji and Grishman, 2011].

Most entity linking approaches focus on linking to general KBs such as Wikidata [Vrandečić, 2012], DBpedia [Auer *et al.*, 2007] or Yago [Suchanek *et al.*, 2024]. However, in real-world applications, the KB is often proprietary or domain-specific, as in the legal or biomedical domain, or inside a company. Entity disambiguation for such KBs has been addressed recently by the LELA system [Haffoudhi *et al.*, 2026], which can disambiguate entities in a true zero-shot fashion without any need for training data or fine-tuning. However, LELA performs only entity disambiguation, not mention detection, i.e., it assumes that entity mentions have already been identified in the text. In real-world scenarios, however, the mentions of the entities have to be detected.

In this demo paper, we extend LELA to an end-to-end EL system that can ingest text and a KB and output a disambiguation – with no need for either training-data, fine-tuning, or the prior identification of the mentions. We contribute a modular and fully open framework, where different named-entity extractors, candidate retrievers, rerankers, and disambiguators can be used interchangeably. Our demo lets users run LELA on their own input texts with different KBs.

2 Related Work

Current research typically bifurcates the problem into two distinct tasks: zero-shot NER, which identifies mentions of unseen entity types [Zaratiana *et al.*, 2024; Bogdanov *et al.*, 2024; Cocchieri *et al.*, 2025], and zero-shot ED, which maps mentions to a KB not encountered during training [Logeswaran *et al.*, 2019a; Wu *et al.*, 2020a; Haffoudhi *et al.*, 2026]. To our knowledge, a unified, end-to-end architecture for zero-shot EL remains unexplored. Consequently, existing frameworks such as spaCy¹, Zshot [Picco *et al.*, 2023], and GLINKER² rely on modular, multi-stage pipelines. However, these are often tightly coupled to specific architectures. LELA differentiates itself by offering a flexible LLM-native architecture and integrating the state-of-the-art MD and ED components, leading to unparalleled robustness in real-world scenarios. Our experiments confirm that the capabilities of LLMs are essential for handling the ambiguity of true zero-shot ED, allowing LELA to significantly outperform existing frameworks on complex, unseen domains.

3 System Design

Our work extends LELA from a disambiguation-only method to a complete end-to-end entity linking framework. Our system is built as a modular pipeline on top of spaCy’s component architecture, where each stage is a pluggable component that can be changed independently. The pipeline works in four stages:

1. Named Entity Recognition. The first stage identifies entity mentions in the input text. We provide several interchangeable NER components: a zero-shot GLiNER model [Zaratiana *et al.*, 2024] (default: NuNER_Zero-span [Bogdanov *et al.*, 2024]), spaCy’s pretrained NER models, and a lightweight regex-based recognizer for rapid prototyping or well-defined mention formats. Long documents are automatically chunked with overlap to respect model context limits.

2. Candidate Generation. For each detected mention, the system retrieves candidates from the KB. Available retrievers include: BM25 (with rank-bm25³), dense retrieval with FAISS⁴ indexing, and fuzzy string matching via RapidFuzz⁵.

3. Reranking (optional). Candidates can be reranked to filter the set passed to the disambiguator using cross-encoder models with top-*k* cutoff. This stage can also be skipped. See [Haffoudhi *et al.*, 2026] for a full study of the use of reranking for entity disambiguation.

4. Disambiguation. The final stage selects the correct entity from the candidates using LLM reasoning, following the LELA methodology [Haffoudhi *et al.*, 2026]. The LLM receives the top-*k* candidates from the reranker along with the input context (with the mention marked in square brackets) and selects the most likely entity by index. LLM-based disambiguation supports linking a detected mention to “NIL” via an explicit “None of the candidates” option. A baseline disambiguator that selects by retrieval rank is also provided. The entity is then displayed when hovering over the highlighted span.

Backends. The components 2-4 can also be disabled for use cases that require only mention detection. For a given component, the framework provides different backend options: vLLM⁶ for fast LLM inference using the offline API, the Transformers⁷ library, SentenceTransformers⁸ for the embedding and reranking models, as well as HTTP requests to OpenAI API compatible endpoints⁹, for interacting with models served using vLLM (online API), llama.cpp¹⁰ (for quantized models, potentially running on CPUs), or proprietary models hosted remotely.

Interfaces. The KB is given as a simple JSONL file, with each line representing an entity with an identifier, a label, and a description. Our system can be used via three interfaces:

- a **Python API** for programmatic use (as in Figure 1),
- a **command-line interface (CLI)** (`lela --config config.json --input doc.txt`) for batch processing of documents in various formats (text, PDF, DOCX, HTML, JSON, JSONL), and
- an interactive **Web application** built with Gradio¹¹ (described in Section 5).

All three interfaces share the same underlying spaCy pipeline and configuration system. The system supports caching, progress tracking, and GPU memory estimation to help users select appropriate model configurations.

Extensibility. New components can be added by registering a spaCy factory (for pipeline stages) or a registry entry (for loaders and knowledge bases). Each component follows a simple protocol: (1) NER components populate `doc.ents`, (2) candidate generators populate a custom `candidates` extension, and (3) disambiguators set a `resolvedentity` extension – making it straightforward to integrate domain-specific models without modifying the core framework.

¹<https://spacy.io/api/large-language-models#el-v1>

²a recently-proposed production-oriented EL framework that is built around GLiNER [Stepanov and Shtopko, 2024; Stepanov *et al.*, 2026]: <https://github.com/Knowledgator/GLinker?tab=readme-ov-file>

³https://github.com/dorianbrown/rank_bm25/tree/master

⁴<https://github.com/facebookresearch/faiss>

⁵<https://github.com/rapidfuzz/RapidFuzz>

⁶<https://github.com/vllm-project/vllm>

⁷<https://github.com/huggingface/transformers>

⁸<https://github.com/huggingface/sentence-transformers>

⁹<https://developers.openai.com/api/reference/overview>

¹⁰<https://github.com/ggml-org/llama.cpp>

¹¹<https://www.gradio.app/>

4 Experimental Validation

Datasets. We evaluate our framework on two different benchmark datasets: Elgold [Islamaj *et al.*, 2021], which evaluates entity linking to Wikipedia across seven domains; and MHERCL [Graciotti *et al.*, 2025] which focuses on Wiki-data long-tail entities and the musical heritage domain.

LELA Configuration. We report results for LELA using the Qwen3-Embedding-4B retriever, the Qwen3-Reranker-4B reranker [Zhang *et al.*, 2025] and the Qwen3-30B-A3B reasoning LLM [Yang *et al.*, 2025]. We retrieve 100 candidates per mention, set the top- k cutoff to 10 candidates, and sample three outputs for self-consistency. We use the NuNER_Zero-span model [Bogdanov *et al.*, 2024] for mention-detection with a fixed NER label set, based on the set of mention types present in the benchmark.

Baselines. We evaluate BLINK¹² [Wu *et al.*, 2020b], as another representative of zero-shot entity disambiguation methods, and the two state-of-the-art end-to-end entity linking methods for Wikipedia: ReFinED [Ayoola *et al.*, 2022] and Relik [Orlando *et al.*, 2024]. On MHERCL, we also compare to the recently-proposed GLINKER framework.

Evaluation Metrics. We measure performance using EL-EVANT [Bast *et al.*, 2022]. On Elgold, we report *InKB* EL F1 score [Röder *et al.*, 2018]. On MHERCL, we follow prior work [Graciotti *et al.*, 2025] and report EL F1.

Results. On the **Elgold** benchmark (Table 1), LELA performs on par with the fully supervised state-of-the-art models, ReFinED and Relik. LELA achieves this competitiveness without the required extensive task-specific training on Wikipedia. This advantage is most visible in the complex domains such as the *Science paper abstracts*, where LELA surpasses ReFinED by nearly 18 percentage points, demonstrating superior robustness. Furthermore, LELA systematically outperforms BLINK, when both rely on the same MD component, confirming the performance gains induced by the reasoning capabilities of our LLM-based ED compared to standard encoder-only baselines.

On the **MHERCL** benchmark (Table 2), which targets long-tail entities in the musical domain, LELA establishes a new state-of-the-art. It outperforms ReFinED and significantly surpasses other zero-shot or modular baselines. These results confirm that while supervised methods struggle to adapt to specialized domains without retraining, LELA handles arbitrary distributions robustly.

5 Demo

Our demo allows users to link entities in arbitrary documents with different KBs using LELA’s Web application. Our interface shows the different steps of the disambiguation process and measures the execution time. Finally, it shows the recognized mentions with their link to the entities in the KB, or to NIL. The Web interface mirrors the full modularity of our approach: Users can experiment with different NER systems, candidate retrievers, rerankers, and reasoning LLMs –

Domain	BLINK	Relik	ReFinED	LELA
1 (News)	68.2 $\pm$ 3.1	76.5 $\pm$ 2.9	78.4$\pm$2.8	74.7 $\pm$ 3.0
2 (Jobs)	43.3 $\pm$ 4.2	67.0 $\pm$ 5.1	72.0$\pm$4.6	60.2 $\pm$ 4.7
3 (Movie)	69.2 $\pm$ 4.8	72.5 $\pm$ 5.0	74.3 $\pm$ 4.8	75.3$\pm$4.7
4 (Auto)	63.7 $\pm$ 5.4	65.8 $\pm$ 6.0	74.9$\pm$5.1	66.5 $\pm$ 5.6
5 (Amazon)	63.4 $\pm$ 4.7	67.8 $\pm$ 4.9	66.9 $\pm$ 4.9	71.5$\pm$4.6
6 (Science)	29.3 $\pm$ 3.3	31.5 $\pm$ 4.0	23.8 $\pm$ 3.8	41.6$\pm$3.8
8 (Historic)	65.7 $\pm$ 7.0	68.2 $\pm$ 7.4	72.2$\pm$6.9	69.9 $\pm$ 7.0
Macro	57.6 $\pm$ 1.8	64.2 $\pm$ 2.0	66.1$\pm$1.8	65.7 $\pm$ 1.8

Table 1: F1 values (InKB, in %) on Elgold across the seven domains, with 95% confidence intervals. Gray results have overlapping confidence intervals with the best results. Unlike competing methods, LELA was not fine-tuned for entity linking on Wikipedia.

Method	EL F1
GLINKER [Stepanov and Shtopko, 2024]	16 $\pm$ 1.5
BLINK [Wu <i>et al.</i> , 2020a]	47 $\pm$ 2.0
Relik [Orlando <i>et al.</i> , 2024]	44 $\pm$ 2.0
ReFinED [Ayoola <i>et al.</i> , 2022]	49 $\pm$ 2.0
LELA (ours)	56$\pm$2.0

Table 2: F1 values (in %) on the MHERCL benchmark, with 95% confidence intervals.

trading off speed against performance. We provide a large range of reference KBs to try out: Users can work with both general-purpose KBs (such as YAGO [Suchanek *et al.*, 2024]) and domain-specific KBs (such as the Crossref Funder Registry¹³ or the 16 different Wikia domains from ZESHEL [Logeswaran *et al.*, 2019b]).

In a twist that goes beyond existing demos, users can come up with their own KBs. For example, a user interested in biology can create a small dictionary with two meanings of the word “culture” (the process of growing cells in the lab vs. the ensemble of arts, customs, and traditions), and ask the system to identify mentions to these entities and determine which meaning is intended in a sentence. Similarly, users from industry can create a dictionary of technical terms that their own approaches struggle with (such as terms that have specific meanings in industry jargon), and see if our approach copes, experimenting with different pipeline configurations.

6 Conclusion

LELA is a Python library for entity linking, handling both mention detection and entity disambiguation. It is true zero-shot, in the sense that it needs neither training data nor fine-tuning, and works on domain-specific text and knowledge bases out-of-the-box. LELA can be used through a Python API, a CLI or a Web Interface, making it suited for real-world applications. It is modular, offering different options for the various stages of the entity linking pipeline, and can be extended with additional components. Future work can integrate additional components, add support for KBs in Turtle format, as well as automate the generation of textual descriptions and NER labels.

¹²Coupled with NuNER_Zero-span for mention detection.

¹³<https://www.crossref.org/services/funder-registry/>

Acknowledgements

The work was partially supported by Agence de l’Innovation de Défense – AID - via Centre Interdisciplinaire d’Etudes pour la Défense et la Sécurité – CIEDS - (project 2024 - KB-LM).

References

- [Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC’07/ASWC’07, pages 722–735, Berlin, Heidelberg, November 2007. Springer-Verlag.
- [Ayoola *et al.*, 2022] Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking. In Anastassia Loukina, Rashmi Gangadharaiah, and Bonan Min, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online, July 2022. Association for Computational Linguistics.
- [Bast *et al.*, 2022] Hannah Bast, Matthias Hertel, and Natalie Prange. ELEVANT: A Fully Automatic Fine-Grained Entity Linking Evaluation and Analysis Tool. In Wanxiang Che and Ekaterina Shutova, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–79, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics.
- [Bogdanov *et al.*, 2024] Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne P Bernard. NuNER: Entity Recognition Encoder Pre-training via LLM-Annotated Data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11829–11841, Miami, Florida, USA, 2024. Association for Computational Linguistics.
- [Cocchieri *et al.*, 2025] Alessio Cocchieri, Marcos Martínez Galindo, Giacomo Frisoni, Gianluca Moro, Claudio Sartori, and Giuseppe Tagliavini. ZeroNER: Fueling Zero-Shot Named Entity Recognition via Entity Type Descriptions. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15594–15616, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [Graciotti *et al.*, 2025] Arianna Graciotti, Nicolas Lazzari, Valentina Presutti, and Rocco Tripodi. Musical heritage historical entity linking. *Artificial Intelligence Review*, 58(5):140, February 2025.
- [Haffoudhi *et al.*, 2026] Samy Haffoudhi, Fabian M. Suchanek, and Nils Holzenberger. LELA: an LLM-based Entity Linking Approach with Zero-Shot Domain Adaptation, January 2026. arXiv:2601.05192 [cs].
- [Islamaj *et al.*, 2021] Rezarta Islamaj, Chih-Hsuan Wei, David Cissel, Nicholas Miliaras, Olga Printseva, Oleg Rodionov, Keiko Sekiya, Janice Ward, and Zhiyong Lu. NLM-Gene, a richly annotated gold standard dataset for gene entities that addresses ambiguity and multi-species gene recognition. *Journal of Biomedical Informatics*, 118:103779, June 2021.
- [Ji and Grishman, 2011] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1148–1158, 2011.
- [Logeswaran *et al.*, 2019a] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-Shot Entity Linking by Reading Entity Descriptions, June 2019.
- [Logeswaran *et al.*, 2019b] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-Shot Entity Linking by Reading Entity Descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, 2019. Association for Computational Linguistics.
- [Martinez-Rodriguez *et al.*, 2020] Jose L. Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. Information extraction meets the Semantic Web: A survey. *Semantic Web*, 11(2):255–335, February 2020.
- [Orlando *et al.*, 2024] Riccardo Orlando, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. ReLiK: Retrieve and LinK, Fast and Accurate Entity Linking and Relation Extraction on an Academic Budget. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14114–14132, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [Picco *et al.*, 2023] Gabriele Picco, Marcos Martinez Galindo, Alberto Purpura, Leopold Fuchs, Vanessa Lopez, and Thanh Lam Hoang. Zshot: An Open-source Framework for Zero-Shot Named Entity Recognition and Relation Extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 357–368, Toronto, Canada, 2023. Association for Computational Linguistics.
- [Röder *et al.*, 2018] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. GERBIL – Benchmarking Named Entity Recognition and Linking consistently. *Semantic Web*, 9(5):605–625, August 2018.
- [Stepanov and Shtopko, 2024] Ihor Stepanov and Mykhailo Shtopko. GLiNER multi-task: Generalist Lightweight Model for Various Information Extraction Tasks, August 2024.
- [Stepanov *et al.*, 2026] Ihor Stepanov, Mykhailo Shtopko, Dmytro Vodianytskyi, and Oleksandr Lukashov. The

Million-Label NER: Breaking Scale Barriers with GLiNER bi-encoder, February 2026.

- [Suchanek *et al.*, 2024] Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 131–140, Washington DC USA, July 2024. ACM.
- [Vrandečić, 2012] Denny Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion*, pages 1063–1064, New York, NY, USA, April 2012. Association for Computing Machinery.
- [Welty *et al.*, 2012] Chris Welty, J. William Murdock, Aditya Kalyanpur, and James Fan. A comparison of hard filters and soft evidence for answer typing in watson. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part II*, volume 7650 of *Lecture Notes in Computer Science*, pages 243–256. Springer, 2012.
- [Wu *et al.*, 2020a] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics.
- [Wu *et al.*, 2020b] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable Zero-shot Entity Linking with Dense Entity Retrieval, September 2020. arXiv:1911.03814 [cs].
- [Yang *et al.*, 2025] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Hao-ran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, May 2025. arXiv:2505.09388 [cs].
- [Zaratiana *et al.*, 2024] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico, 2024. Association for Computational Linguistics.
- [Zhang *et al.*, 2025] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models, June 2025. arXiv:2506.05176 [cs].