Introducing FELA - Flexible Entity Linking Approach

Adam Aron Rynkiewicz[1,2,*], Raul Palma[1] and Paulina Poniatowska-Rynkiewicz[1,3]

[1]Data Analytics and Semantics Division, Poznan Supercomputing and Networking Center, Jana Pawla II 10 61-139 Poznan, Poland
[2]Institute of Computing Science, Poznan University of Technology, Piotrowo 2 60-965 Poznan, Poland
[3]Department of Plant Genomics, Institute of Bioorganic Chemistry PAS, Noskowskiego 12/14 61-704 Poznan

Abstract

With the rapid expansion of digital data, effective mechanisms for transforming raw information into structured knowledge are increasingly essential. End-to-end entity linking presents a promising solution by disambiguating entity mentions and aligning them with knowledge bases. However, most existing approaches are tailored to a single KB, limiting their adaptability and scalability across diverse knowledge resources. To address this limitation, we introduce *FELA* - Flexible Entity Linking Approach - a framework designed for seamless entity linking across multiple knowledge bases. *FELA* leverages fine-tuned Large Language Models, a generic embedding model, and a Large Language Model-based reranking module to enhance entity disambiguation. Our approach achieves state-of-the-art performance on Wikidata entity linking benchmarks, demonstrating its effectiveness and flexibility. Furthermore, we illustrate *FELA*'s extensibility by applying it to Agrovoc, showcasing its capability to generalize beyond Wikidata. This work contributes to the development of more flexible, scalable, and domain-agnostic entity linking solutions, facilitating knowledge extraction across heterogeneous data sources.

Keywords

End-to-end entity linking, Large Language Models, Wikidata

1. Introduction

End-to-end entity linking aims to identify named entities within a text corpus and map them with corresponding entities in a target knowledge base. This process involves two key steps: Named Entity Recognition (NER) and Entity Disambiguation (ED). In the first step, the model detects named entities within the text; for example, in the sentence ”Paris is the capital of France,” the entities ”Paris” and ”France” should be recognized. In the second step, the model assigns each identified entity a unique knowledge base (KB) identifier, such as Wikidata Q90 for Paris and Q142 for France, ensuring accurate linkage.

Entity disambiguation presents significant challenges due to factors such as name variations, gaps in the target knowledge base, evolving information, and inherent entity ambiguity. Context plays a crucial role in resolving these ambiguities. For instance, a search for ”Paris” in Wikidata yields over 120,000 results, including Paris (the mythological son of the King of Troy), Paris (a city in Idaho, USA), and numerous other entities containing the term ”Paris.” Therefore, effective entity linking requires the use of contextual information to accurately determine the intended reference.

1.1. Problem statement

Most existing approaches are designed for a single KB, which limits their adaptability and scalability when applied to diverse knowledge resources [1, 2, 3]. This lack of flexibility presents a significant challenge in real-world applications where multiple KBs, each with different structures, schemas, and

3rd Workshop on Hybrid Artificial Intelligence and Enterprise Modelling, June 16–17, 2025, Vienna, Austria
*Corresponding author.
In: Janis Grabis, Yves Wautelet, Emanuele Laurenzi , Hans-Friedrich Witschel, Peter Haase, Marco Montali, Cristina Cabanillas, Andrea Marrella, Manuel Resinas, Karolin Winter. Selected Papers of HybridAIMS and CAI Workshops. Co-located with CAiSE 2025.
arynkiewicz@man.poznan.pl (A. A. Rynkiewicz)
0000-0002-0528-7544 (A. A. Rynkiewicz); 0000-0003-4289-4922 (R. Palma); 0009-0003-4939-2412 (P. Poniatowska-Rynkiewicz)

domain coverage, must be integrated. Furthermore, these solutions often rely on extensive fine-tuning to accommodate new KBs or incorporate updated information, making them resource-intensive and time-consuming. Effective fine-tuning requires large, high-quality datasets to ensure that the model generalizes well. However, obtaining such datasets can be expensive and labor intensive, creating a bottleneck for scalability and continuous learning. Consequently, there is a need for more adaptable, scalable, and data-efficient approaches that can operate seamlessly across multiple KBs without extensive retraining.

1.2. Research objectives

Currently, Large Language Models (LLMs) are the dominant paradigm in Natural Language Processing (NLP), achieving state-of-the-art performance across various tasks and consistently outperforming traditional models in benchmark evaluations. In addition, LLMs are constantly evolving, with frequent updates and new versions enhancing their capabilities. Their widespread adoption has also led to the development of a rich ecosystem of tools and techniques for inference, fine-tuning, integrating external knowledge, and improving model adaptability. Given their scalability, reasoning ability, and capacity to process vast amounts of information, LLMs present a natural and promising solution to the entity linking problem. This research aims to explore how LLMs can be effectively leveraged for entity linking, addressing challenges such as knowledge integration, adaptability across multiple knowledge bases, and minimizing the need for extensive fine-tuning.

We present a highly flexible, end-to-end entity linking framework comprising a NER module, an ED module, and a reranking module, all leveraging large language models (LLMs). Our framework achieves state-of-the-art (SOTA) performance on standard benchmarks. The key contributions of our work are as follows:

1. We develop *FELA*, a modular and flexible entity linking framework using compact open-source LLMs (with no more than 7 billion parameters). Despite its lightweight design, our approach achieves SOTA performance on benchmark datasets.
2. We publicly release our code to facilitate further research and reproducibility.

The remainder of this paper is organized as follows: In Section 2, we review existing entity linking methods, with a particular focus on NER, ED, and combined approaches. Section 3 provides a detailed description of the *FELA* framework, including the LLMs and other models used, as well as data sources. In Section 4, we present our experimental results and benchmark evaluations. Finally, Section 5 summarizes our findings, discusses key challenges, and outlines directions for future research.

2. Related Work

This section reviews existing methods in entity linking (EL), grouped into Named Entity Recognition (NER), Entity Disambiguation (ED), and end-to-end EL approaches.

2.1. Named Entity Recognition Solutions

Traditional NER approaches rely on predefined entity types, limiting their flexibility. Recent models like *GLiNER* [4] and *UniversalNER* [5] address this by enabling recognition of arbitrary types without retraining.

GLiNER combines a pretrained encoder, span representation module, and entity representation module. It computes token and span embeddings, compares them with entity embeddings, and identifies matches via a matching score. Despite having only 300M parameters, it performs strongly across benchmarks.

UniversalNER (*UniNER*) distills LLM capabilities into a smaller model using a conversational tuning strategy. It reformulates entity recognition as a Q&A task, generating JSON lists of identified entities

per type. A negative sampling strategy improves its instruction-following and generalization across tasks.

2.2. Entity Disambiguation Solutions

ED typically involves: (1) entity representation, (2) candidate retrieval, and (3) disambiguation. *BLINK* [3] uses a bi-encoder to map text and entities into a shared space, retrieves top-k candidates via k-NN, and applies a cross-encoder for final disambiguation.

EPGEL [6] enhances retrieval using detailed BART-based profiles (titles and descriptions), indexed in Elasticsearch. A cross-encoder ranks candidates based on contextual similarity. It outperforms *BLINK* on several benchmarks.

anydef [7] also utilizes LLM-generated profiles but does not include a reranking step. Instead, it applies binary quantization to embeddings for efficient storage and retrieval of relevant entities. Although it involves less extensive fine-tuning, its performance is comparable to that of *EPGEL*.

2.3. End-to-End Entity Linking Solutions

ReFinED [2] performs mention detection, embeds entity spans, and computes typing and description scores to select the best match. It can assign NIL labels when no match exists, improving real-world applicability. However, it requires fine-tuning to generalize beyond Wikipedia and Wikidata.

3. Methods

3.1. Approach Overview

FELA[1] consists of three modular components: NER, ED, and Reranking. The pipeline begins with a chunking step to split text into manageable segments, each processed by the quantized *UniNER-W4A16* model to identify entities. This produces entity-labeled spans for further processing.

In the ED stage, entity spans are contextualized using the *anydef-v2-linear-W4A16* model, which generates entities profiles, which are then passed to the *mxbai-embed-large-v1* [8, 9] embedding model. These 1024-dimensional embeddings are quantized and matched against a *Faiss* [10] vector store to retrieve top-k candidates from each KB.

Finally, the *bge-reranker-v2-gemma* [11, 12] model evaluates the retrieved candidates against the *anydef*-generated profile by computing semantic similarity scores between the definitions of the candidates and the profile. This reranking process enables the integration of multiple knowledge bases by allowing the model to consider heterogeneous candidate definitions across sources and select the one most aligned with the contextualized profile, returning a more accurate and context-aware disambiguation result.

3.2. Models

We did not introduce any modifications to *mxbai-embed-large-v1* and *bge-reranker-v2-gemma*. However, we made minor adjustments to the original *UniNER* and *anydef* models [2].

For *UniNER*, we incorporated a tokenizer into its configuration, eliminating the need for the FastChat conversation template. Furthermore, to reduce memory consumption, we applied quantization using the LLM Compressor, using 512 randomly selected samples from the *UniNER* dataset. This process reduced the precision of the model weights to 4 bits while maintaining the activation precision at 16 bits (W4A16), significantly optimizing the memory footprint while achieving comparable performance.

We also refined the *anydef* model. To mitigate catastrophic forgetting during fine-tuning [13, 14], we merged *anydef* with *Mistral-7B-v0.1* [15] using various techniques, including TIES, linear and task

[1]The *FELA* implementation is available at https://github.com/daisd-ai/FELA
[2]Models are available at https://huggingface.co/daisd-ai

arithmetic from the MergeKit framework [16]. Additionally, we applied quantization using the LLM Compressor [17], 512 randomly selected samples from the *anydef* dataset, and experimenting with different configurations: W4A16 and W8A8 (where both weights and activations were quantized to 8-bit precision). Finally, we evaluated the performance of each variant across entity disambiguation benchmarks. Our results indicated that the best performing configuration - after the original *anydef* - was the linear merging approach combined with the quantization of W4A16. However, the performance differences between the models, in terms of precision, did not exceed two percentage points.

For inference with *UniNER* and *anydef*, we utilize vLLM [18], while for reranking with *bge-reranker-v2-gemma*, we employ FlagEmbedding [19, 20, 21].

3.3. Knowledge Bases

We integrated Wikidata and Agrovoc as KBs. Wikidata is a large-scale, general-purpose KB containing structured data on a wide range of entities across domains, whereas Agrovoc is a domain-specific thesaurus focused on agriculture, food, and related areas. Although both KBs provide hierarchical and multilingual information, Wikidata includes rich interlinked entity relations and metadata, whereas Agrovoc emphasizes standardized terminology and concept hierarchies within the agricultural domain.

For both KBs, we adhered to established protocols for downloading and preprocessing. Wikidata requires substantial preprocessing due to the occurrence of entities classified under Wikimedia categories. Specifically, out of approximately 110 million entities, only 40 million were relevant for entity linking purposes.

Agrovoc also requires preprocessing, as only about 15% of its entities include descriptions. To address this limitation, an LLM was employed to generate missing entity data.

Following the preprocessing stage, the subsequent steps for both KBs were the same: (1) **Entity Profile Creation:** Each entity was represented by a structured profile, including its label, description, and type. (2) **Embedding Generation:** Each profile was encoded as a 1024-dimensional vector using the *mxbai-embed-large-v1* model. (3) **Memory Optimization:** Binary quantization was applied to reduce the memory footprint of embeddings, improving retrieval efficiency. This process significantly decreased the storage requirement from 120 GB (for uncompressed Wikidata embeddings) to 5 GB, with a minor trade-off in retrieval accuracy. (4) **Storage and Indexing:** The quantized embeddings were stored in a Faiss vector database to facilitate efficient retrieval.

During inference, entity linking was performed by retrieving the top-k (or top-k / 2 for improved efficiency) candidates from both KBs. A reranker model was then applied to identify the most suitable candidate. This framework ensures that any KB can be seamlessly integrated into the pipeline, even in cases where entity profiles lack certain required attributes.

4. Results

The results are organized into two main sections: the evaluation of our modified NER and ED models, and the overall performance of the entity linking pipeline in comparison to the *ReFinED* solution. At the end of the section we present example results of end-to-end entity linking using *FELA*.

Table 1 presents a comparison between the original *UniNER* model and its quantized version, *UniNER-W4A16*, in terms of F1 score across the *CrossNER* [22] and *MIT* [23] datasets. As anticipated, the quantized model generally underperforms compared to the original, with an average decrease in the F1 score of less than 2 points. However, in specific cases, such as the *CrossNER-Music* and *MIT-Restaurant* datasets, the quantized model exhibits slightly superior performance, indicating that quantization does not always lead to degradation and may, in some scenarios (restaurant and music related entities), improve model efficiency without significantly compromising accuracy.

Table 2 presents the precision scores for the original *anydef-v2* model and its quantized counterpart, *anydef-v2-linear-W4A16*, evaluated on the *RSS-500* [24], *ISTEX-1000* [24], *Reuters-128* [25], and *TweekiGold* [26] datasets. Consistent with the *UniNER* evaluation, the quantized model demonstrates lower precision on two datasets (*RSS-500* and *ISTEX-1000*), retains equivalent performance on the

Table 1
F1 score of *UniNER-7B-all* and quantized *UniNER-W4A16* models across test datasets.

Dataset	UniNER-7B-all	UniNER-W4A16 (ours)
CrossNER AI	62.21	61.16
CrossNER literature	66.59	66.36
CrossNER music	69.20	69.46
CrossNER politics	66.64	66.35
CrossNER science	70.19	66.33
MIT Movie	59.77	57.82
MIT Restaurant	36.70	36.92

TweekiGold dataset, and, unexpectedly, surpasses the original model on the *Reuters-128* dataset. This suggests that while quantization generally results in a minor decline in precision, it can also lead to improvements in certain situations, in this particular case, economic news.

Table 2
Precision (expressed as a percentage) of *anydef-v2* and *anydef-v2-linear-W4A16* entity disambiguation systems across test datasets.

	Precision [%]	
Dataset	anydef-v2	anydef-v2-linear-W4A16 (ours)
RSS-500	66.89	64.90
ISTEX-1000	85.82	84.33
Reuters-128	64.88	68.28
TweekiGold	75.93	75.93

The precision and retrieval rate of the end-to-end entity linking solutions, *ReFinED* and *FELA*, were evaluated on the custom *RSS-500*, *Reuters-128*, and *TweekiGold* datasets [3]. Table 3 presents the precision scores of both models. *FELA* performs better on two datasets, indicating stronger performance in general and social media contexts, while *ReFinED* is better suited to structured, domain-specific texts, in this case - economic news (*Reuters-128* dataset).

Table 3
Precision (expressed as a percentage) of *ReFinED* and *FELA* processing systems across test datasets.

	Precision [%]	
Dataset	ReFinED	FELA (ours)
RSS-500	72.85	76.82
Reuters-128	59.50	51.17
TweekiGold	65.05	71.18

In the context of end-to-end entity linking, retrieval rate measures the effectiveness of the retrieval component in identifying and ranking the correct entity among the top-k candidate entities. A higher retrieval rate indicates that the correct entity is more frequently included among the top-ranked candidates, thus increasing the likelihood of successful disambiguation in the subsequent processing stage.

Table 4 compares the retrieval rates of *ReFinED* and *FELA*. Consistent with the precision evaluation, *FELA* demonstrates superior retrieval performance on the *RSS-500* and *TweekiGold* datasets but falls behind on the *Reuters-128* dataset. These results highlight *FELA*'s strength in retrieving relevant entity candidates for certain datasets, while *ReFinED* remains more effective for others.

[3]The datasets are available at https://github.com/daisd-ai/FELA

Table 4
Retrieval rate (expressed as a percentage) of *ReFinED* and *FELA* processing systems across test datasets for 25 candidates.

	Retrieval rate [%]	
Dataset	ReFinED	FELA (ours)
RSS-500	89.40	91.39
Reuters-128	70.97	63.60
TweekiGold	82.18	84.72

5. Conclusions

We presented *FELA*, a modular and resource-efficient end-to-end entity linking framework built on compact open-source LLMs. Designed for adaptability across multiple KBs, *FELA* minimizes the need for fine-tuning while achieving state-of-the-art performance on benchmark datasets.

The framework comprises three independent components, NER, ED, and Reranking, which allow flexible updates and integration. This modularity, combined with techniques such as model quantization and efficient retrieval, enables scalable deployment without significant performance trade-offs.

Our experiments show that *FELA* outperforms *ReFinED* on most benchmarks. Nevertheless, there is room for improvement, particularly in refining the reranking module and exploring more generalizable embedding models. Moreover, evaluating *FELA* on domain specific benchmarks could reveal additional routes for improvement.

In summary, *FELA* offers a robust and extensible foundation for multi-KB entity linking. By releasing our code and approach, we aim to support continued research and encourage the development of efficient, generalizable EL systems.

Acknowledgments

This work was supported by the PoliruralPLUS [grant agreement 101136910].

Declaration on Generative AI

During the preparation of this work the authors used Writefull for Overleaf in order to improve language and stylistics of the manuscript. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

[1] M. P. Kannan Ravi, K. Singh, I. O. Mulang', S. Shekarpour, J. Hoffart, J. Lehmann, CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 504–514. URL: https://aclanthology.org/2021.eacl-main.40. doi:10.18653/v1/2021.eacl-main.40.

[2] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, A. Pierleoni, ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking, in: A. Loukina, R. Gangadharaiah, B. Min (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, Association

for Computational Linguistics, Hybrid: Seattle, Washington + Online, 2022, pp. 209–220. URL: https://aclanthology.org/2022.naacl-industry.24. doi:10.18653/v1/2022.naacl-industry.24.

[3] L. Wu, F. Petroni, M. Josifoski, S. Riedel, L. Zettlemoyer, Scalable zero-shot entity linking with dense entity retrieval, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6397–6407. URL: https://aclanthology.org/2020.emnlp-main.519. doi:10.18653/v1/2020.emnlp-main.519.

[4] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, GLiNER: Generalist model for named entity recognition using bidirectional transformer, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 5364–5376. URL: https://aclanthology.org/2024.naacl-long.300/. doi:10.18653/v1/2024.naacl-long.300.

[5] W. Zhou, S. Zhang, Y. Gu, M. Chen, H. Poon, UniversalNER: Targeted distillation from large language models for open named entity recognition, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=r65xfUb76p.

[6] T. Lai, H. Ji, C. Zhai, Improving candidate retrieval with Entity Profile Generation for Wikidata Entity Linking, in: Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 3696–3711.

[7] A. A. Rynkiewicz, P. Formanowicz, R. Palma, Universal entity linking, 2025. Manuscript under review.

[8] S. Lee, A. Shakir, D. Koenig, J. Lipp, Open source strikes bread - new fluffy embeddings model, 2024. URL: https://www.mixedbread.ai/blog/mxbai-embed-large-v1.

[9] X. Li, J. Li, AoE: Angle-optimized embeddings for semantic textual similarity, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 1825–1839. URL: https://aclanthology.org/2024.acl-long.101/. doi:10.18653/v1/2024.acl-long.101.

[10] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, IEEE Transactions on Big Data 7 (2019) 535–547.

[11] C. Li, Z. Liu, S. Xiao, Y. Shao, Making large language models a better foundation for dense retrieval, 2023. arXiv:2312.15503.

[12] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 2318–2335. URL: https://aclanthology.org/2024.findings-acl.137/. doi:10.18653/v1/2024.findings-acl.137.

[13] A. Alexandrov, V. Raychev, M. N. Mueller, C. Zhang, M. Vechev, K. Toutanova, Mitigating catastrophic forgetting in language transfer via model merging, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 17167–17186. URL: https://aclanthology.org/2024.findings-emnlp.1000/. doi:10.18653/v1/2024.findings-emnlp.1000.

[14] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, Y. Zhang, An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL: https://arxiv.org/abs/2308.08747. arXiv:2308.08747.

[15] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[16] C. Goddard, S. Siriwardhana, M. Ehghaghi, L. Meyers, V. Karpukhin, B. Benedict, M. McQuade, J. Solawetz, Arcee's MergeKit: A toolkit for merging large language models, in: F. Dernoncourt, D. Preoţiuc-Pietro, A. Shimorina (Eds.), Proceedings of the 2024 Conference on Empirical Methods

in Natural Language Processing: Industry Track, Association for Computational Linguistics, Miami, Florida, US, 2024, pp. 477–485. URL: https://aclanthology.org/2024.emnlp-industry.36. doi:10.18653/v1/2024.emnlp-industry.36.

[17] vLLM Project, Llm compressor: an easy-to-use library for optimizing models for deployment with vllm, https://github.com/vllm-project/llm-compressor, 2025.

[18] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[19] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, J.-Y. Nie, C-pack: Packed resources for general chinese embeddings, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 641–649. URL: https://doi.org/10.1145/3626772.3657878. doi:10.1145/3626772.3657878.

[20] S. Xiao, Z. Liu, P. Zhang, X. Xing, Lm-cocktail: Resilient tuning of language models via model merging, 2023. arXiv:2311.13534.

[21] P. Zhang, S. Xiao, Z. Liu, Z. Dou, J.-Y. Nie, Retrieve anything to augment large language models, 2023. arXiv:2310.07554.

[22] Z. Liu, Y. Xu, T. Yu, W. Dai, Z. Ji, S. Cahyawijaya, A. Madotto, P. Fung, Crossner: Evaluating cross-domain named entity recognition, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 13452–13460. URL: https://doi.org/10.1609/aaai.v35i15.17587. doi:10.1609/AAAI.V35I15.17587.

[23] J. Liu, P. Pasupat, D. S. Cyphers, J. R. Glass, Asgard: A portable architecture for multilingual dialogue systems, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (2013) 8386–8390. URL: https://api.semanticscholar.org/CorpusID:14903208.

[24] A. Delpeuch, OpenTapioca: Lightweight Entity Linking for Wikidata, 2019. URL: https://hal.science/hal-02098522, journal-article.

[25] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, A. Both, N^3 - a collection of datasets for named entity recognition and disambiguation in the NLP interchange format, in: N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland, 2014, pp. 3529–3533. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/856_Paper.pdf.

[26] B. Harandizadeh, S. Singh, Tweeki: Linking named entities on Twitter to a knowledge graph, in: W. Xu, A. Ritter, T. Baldwin, A. Rahimi (Eds.), Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020), Association for Computational Linguistics, Online, 2020, pp. 222–231. URL: https://aclanthology.org/2020.wnut-1.29. doi:10.18653/v1/2020.wnut-1.29.