Contextual Augmentation for Entity Linking using Large Language Models

Daniel Vollmers, Hamada M. Zahera, Diego Moussallem, Axel-Cyrille Ngonga Ngomo
Data Science Group, Paderborn University, Germany
{daniel.vollmers,hamada.zahera,diego.moussallem,axel.ngonga}@uni-paderborn.de

Abstract

Entity Linking involves detecting and linking entity mentions in natural language texts to a knowledge graph. Traditional methods use a two-step process with separate models for entity recognition and disambiguation, which can be computationally intensive and less effective. We propose a fine-tuned model that jointly integrates entity recognition and disambiguation in a unified framework. Furthermore, our approach leverages large language models to enrich the context of entity mentions, yielding better performance in entity disambiguation. We evaluated our approach on benchmark datasets and compared with several baselines. The evaluation results show that our approach achieves state-of-the-art performance on out-of-domain datasets.

1 Introduction

Entity Linking (EL) in knowledge graphs (KGs) involves identifying and connecting entities within a text to their corresponding entries, enhancing the semantic understanding of the text (Oliveira et al., 2021). This process includes two main steps: (i) Named Entity Recognition (NER), which detects entity spans such as names, dates, and locations; and (ii) Entity Disambiguation (ED), which resolves ambiguities by accurately matching these entities to their corresponding entries in the knowledge graph.

Traditional EL approaches, such as the two-stage architecture (Sevgili et al., 2022), divide the task into candidate generation and entity re-ranking phases. Another approach involves bi-encoder and cross-encoder models, as demonstrated by systems like BLINK (Wu et al., 2020). Bi-encoder models independently encode mentions and entities for efficient retrieval. In contrast, cross-encoder models jointly evaluate mention-entity pairs to enhance accuracy. Additionally, generative models (Wang et al., 2023) consider candidate generation as a text generation task, where models learn to generate unique entity names based on contextual information. However, While these traditional methods excel in identifying and linking common entities, they often struggle with handling long-tail entities, i.e., rare or those with multiple meanings, making it difficult to accurately disambiguate them in different contexts. For example, consider the entity ‘`Jaguar`’. In a general context, ‘`Jaguar`’ could refer to the animal, the car brand, or even a sports team. However, in a domain-specific context, such as a biology research paper, ‘`Jaguar`’ would most likely refer to the animal. Traditional methods may not effectively handle such distinctions, leading to potential errors in Entity Linking. Another example, "`Angelina met her partner Brad and her father Jon in AK`", where entities are identified by their first names, while news articles commonly use surnames.

In this paper, we propose a LLM-based augmentation strategy to enrich the context of entities mentions in short texts such as questions. Our approach expands entities mentions by prompting the LLM to extend them to their likely Wikipedia titles, thereby replacing ambiguous entity spans with more easily linkable ones. For example, the spans “`Angelina`”, “`Brad`” and “`Jon`” should be expanded to “`Angelina Jolie`”, “`Brad Pitt`” and “`Jon Voight`”. Additionally, our strategy replaces abbreviations like “`AK`” commonly used for the state of Alaska, with the full name *“Alaska”*. To link entities in out-of-domain datasets, we use an autoregressive model (De Cao et al., 2021). which generates entity names token-by-token based on context. This approach allows the model to adapt to new or unseen entities by leveraging the surrounding text, thereby improving accuracy of identifying and linking entities that are absent from the training data. We conducted several experiments on different benchmarks to evaluate the performance of our approach against various baselines.

Proceedings of the 31st International Conference on Computational Linguistics, pages 8535–8545
January 19–24, 2025. ©2025 Association for Computational Linguistics

Specifically, we experimented with two types of models: An end-to-end model, which directly links entities and a traditional two-step approach, which first identifies entity spans before applying the disambiguation step. The evaluation results demonstrate that our approach significantly outperforms different baselines by a large performance margin on the benchmarking datasets. We summarize the main contributions in this paper as follows:

- We propose an LLM-based approach for Entity Linking that leverages zero-shot prompting, which achieves state-of-the-art results on most out-of-domain evaluation datasets.
- We evaluated different LLM-based augmentation strategies for Entity Linking, comparing their effectiveness in both the two-step approach and the end-to-end approach.
- We evaluated the performance of a joint model for entity recognition and disambiguation compared to end-to-end models and NER models.
- We make our source code and fine-tuned models publicly available at the GitHub repository.[1]

2 Related Work

Entity Linking is typically involves two phases: *span detection* and *entity disambiguation* (Sevgili et al., 2022). During span detection phase, most existing approaches employ standard named entity recognition to identify relevant spans in text. In the disambiguation phase, candidate entities are generated from a knowledge graph and linked to the most appropriate match using pre-built search indices, where each entry corresponds to a KG entity.

Disambiguation Approaches like MAG (Moussallem et al., 2017) and DoSeR (Zwicklbauer et al., 2016) rely on pre-built indices for effective entity disambiguation. For example, MAG utilizes five distinct indices—*surface forms, personal names, rare references, acronyms*, and *contextual information*—to query entities. Following the index query, MAG applies an additional step to refine the candidate set for final disambiguation. In contrast, DoSeR integrates text-based retrieval with surface forms, leverages a Word2Vec embedding model, and uses a priori probabilities derived from occurrence frequency for candidate generation, employing a similar candidate expansion strategy as MAG. Other approaches such as Mulang' et al. (2020) introduce context information by extracting triples from knowledge graphs and verbalizing them to the input sequence. Meanwhile, Raiman and Raiman (2018) incorporate type information into the disambiguation process to improve accuracy. Moreover, BLINK (Wu et al., 2020) adopts an embedding-centric approach for candidate generation, employing a Bi-Encoder model to generate representations of both candidates and mentions. Entity search within the index is executed through KNN-Search based on context embedding vectors. Other methods, similar to BLINK (Lai et al., 2022), combine text-based retrieval with deep neural embedding models for entity disambiguation.

Alternatively, Parravicini et al. (2019) propose an embedding-based approach where node embeddings, derived using the *word2vec* algorithm, assesses vertex similarity. This method involves evaluating candidates via tuples, where each tuple corresponds to candidate entities linked to mentions in a document. A global similarity score, calculated from these node embeddings, determines the score for each tuple. In contrast, some approaches utilize re-ranker models to calculate embeddings, considering both the mention's context and candidate entities. These models employ a feedforward layer to re-rank candidates (Wu et al., 2020; Lai et al., 2022). Lastly,Xin et al. (2024) introduce the first use of LLMs for context augmentation in entity disambiguation, focusing solely on enhancing the context of entity mentions without addressing NER. This method, however, is computationally intensive due to the need to augment each mention individually.

End-to-end Entity Linking Unlike, the traditional two-step Entity Linking process, recent approaches omit the NER step and directly extracting or annotating entity candidates from the input sequence. These approaches often employ fine-tuned models for autoregressive annotation (Zhang et al., 2021). Early work by Kolitsas et al. (2018) integrated mention and entity embeddings based on Word2Vec and sequence-to-sequence models like LSTM (Hochreiter and Schmidhuber, 1997) to incorporate contextual information. Furthermore, van Hulst et al. (2020) introduced an approach that predicts coherence scores to align en-

[1] https://github.com/dice-group/AugmentedEL

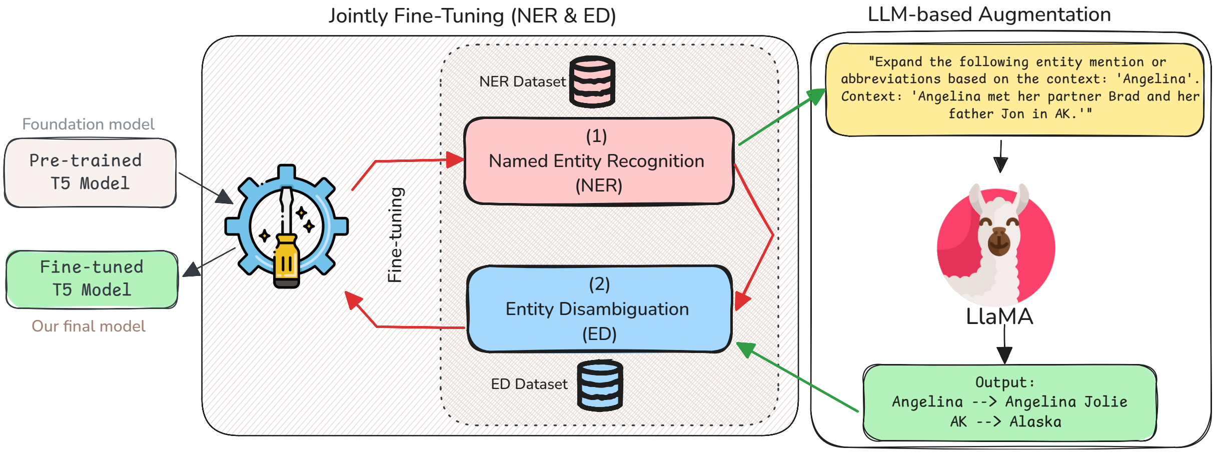
Jointly Fine-Tuning (NER & ED)
LLM-based Augmentation
Foundation model
Pre-trained
T5 Model
Fine-tuned
T5 Model
Our final model
Fine-tuning
NER Dataset
(1)
Named Entity Recognition
(NER)
(2)
Entity Disambiguation
(ED)
ED Dataset
"Expand the following entity mention or abbreviations based on the context: 'Angelina'. Context: 'Angelina met her partner Brad and her father Jon in AK.'"
LlaMA
Output:
Angelina --> Angelina Jolie
AK --> Alaska

Figure 1: The architecture of our approach, including Jointly Fine-tuning and LLM-based Augmentation

tity annotation. Modern Entity Linking methods mainly involve fine-tuning Large LLMs for this task. For instance, GENRE framework (De Cao et al., 2021) leverages a fine-tuned BART model for autoregressive annotation of input sequences, utilizing an offline prefix trie from Wikipedia titles to constrain the decoding process and thus reduce the search space. In contrast, EntQA approach (Zhang et al., 2021) uses a vector-based search index similar to BLINK to identify entities within the input sequence. During the disambiguation phase, it computes candidate spans for each entity and selects the highest-scored candidate for linking. Our work differs from these methods by addressing the entire Entity Linking pipeline, including NER, and introduces an LLM-based augmentation to improve entity disambiguation, particularly on out-of-domain datasets.

3 Approach

This section outlines our approach for Entity Linking using a fine-tuned T5 model and contextual augmentation using LLMs. First, we provide the task definition of Entity Linking, then a description our approach's with the main components, including: *Named Entity Recognition (NER)*, *LLM-based Augmentation*, *Entity Disambiguation (ED)* and the *Joint Fine-tuning of (NER&ED)*. Afterwards, we describe our strategy to mitigate LLMs hallucination and the ablations of our approach to assess the performance of LLM-based augmentation.

3.1 Task Definition

Entity Linking task involves two main steps: (I) Named Entity Recognition, and (II) Entity Disambiguation. Named Entity Recognition identifies and extracts entity mentions (e.g., people, organizations, locations) from text. Given a context $C = (t_1, t_2, \cdots, t_k)$, the goal is to identify a subset of tokens $M = m_1, m_2, \cdots, m_i$ representing entity mentions. The NER function maps C to a list of entity mention spans M (Sevgili et al., 2022).

$$\text{NER} : C \rightarrow M^n \tag{1}$$

After identifying entities, Entity Disambiguation resolves any potential ambiguities by linking each mention m_i to the correct entity e_i in a knowledge graph, using similarity measures and contextual information (Sevgili et al., 2022).

$$\text{ED} : [(m_1, \cdots, m_n), C] \rightarrow (e_1, \cdots, e_n) \tag{2}$$

3.2 Architecture

We employ the T5 model as the *foundation architecture* in our approach and fine-tune it on NER and ED tasks. This joint fine-tuning allows our model to leverage shared knowledge across both tasks, improving its overall performance. The following sections summarize the main components of our approach.

3.2.1 Named Entity Recognition

We use the T5 model with a transformer architecture to captures contextual information from both preceding and succeeding text. Furthermore, the T5 model regards NER task as a text-to-text problem, where both input and output are sequences of text. Initially, the input text is tokenized and processed by the model's encoder, which generates contextual embeddings by considering the surrounding words. These embeddings are then used by the decoder to produce an output sequence with entity tags.

In our approach, we fine-tune the T5 model on annotated datasets with entities, allowing it to learn accurate tagging based on context. For instance, the sentence "Angelina met her partner Brad and her father Jon in AK" is transformed into [BEGIN_ENT] Angelina [END_ENT] met her partner [BEGIN_ENT] Brad [END_ENT] and her father [BEGIN_ENT] Jon [END_ENT] in [BEGIN_ENT] AK [END_ENT]. In this output, "Angelina", "Brad", "Jon", and "AK" are recognized as entities and marked with an annotation tag.

3.2.2 LLM-based Augmentation

To further augment the NER process, we employ the LlaMA3 model to perform additional entity recognition and expand on the entities detected from the previous step. The core idea is to replace ambiguous or incomplete entity mentions with more precise and recognizable forms, such as full names or specific titles. This is achieved by prompting the LlaMA3 model to generate these extended forms based on the given context. Our prompt includes a structured input with the entity mention and its surrounding context. For example, consider the entity mention "Angelina" in the sentence: *"Angelina met her partner Brad and her father Jon in AK."*. We prompt the LlaMA3 model to expand entity mentions as follows:

LLM Prompt

Expand the following entity mention 'Angelina' and abbreviations 'AK' based on the context: .
Context: 'Angelina met her partner Brad and her father Jon in AK.'

In response, the LlaMA3 model expands the entities (e.g., "Angelina"⟶"Angelina Jolie" and for the abbreviation "AK"⟶"Alaska". This approach allows us to replace ambiguous mentions with their more specific counterparts, thereby improving the ability of the model to link entities in the follow-up entity disambiguation step.

3.2.3 Entity Disambiguation

Entity Disambiguation is crucial for resolving ambiguities when multiple entities share the same name. In our approach, the T5 model addresses this by encoding the context around an entity mention and creating a representation that captures both semantic (e.g., word meaning) and syntactic information (e.g., sentence structure). determine the correct entity based on the surrounding text. Then, it decodes this representation to predict the correct entity from a set of candidates, using attention mechanisms to focus on relevant parts of the input text. Formally, given a context C, the model generates an output sequence T with entity mentions $m \in M^n$ and their corresponding URIs $e \in E^n$, represented by their titles in a target knowledge graph. The output structure is defined as:

$$\text{[BEGIN\_ENT]}\ m\ \text{[END\_ENT]}[\text{title}(e)] \qquad (3)$$

where $m \in M^n$ denotes an entity mention and $e \in E^n$ represents the associate URI.

Unlike traditional methods, our approach is unique as it leverages one single T5 model to perform both NER and ED sequentially. First, the T5 model generates an intermediate sequence I for NER:

$$\text{I} = \text{[BEGIN\_ENT]}\ m\ \text{[END\_ENT]}|m \in \text{M}^n \qquad (4)$$

Afterward, it predicts the target output based on the output of the NER step. At inference time, the target output is expanded by our augmentation strategy as presented in section 3.2.2.

3.2.4 Jointly Fine-tuning (NER&ED)

To fine-tune the T5 model for both NER and Entity Disambiguation, we create two different training samples: one for the NER task and another for the Disambiguation task. For the NER task, the input is a text sequence without annotations, appended with the suffix *target_ner*. The target sequence is the same text with annotated entity mentions, as detailed in section 3.2.1. For the Disambiguation task, the input is a text sequence with annotated entities, appended with the suffix *target_el*, and the target sequence includes the corresponding Wikipedia entity labels, as described in section 3.2.3.

We combine the NER and Disambiguation samples into a single dataset to fine-tune the model jointly for both tasks. Additionally, we integrate the NER and Entity Disambiguation tasks within a unified framework (see Figure 1). This integration enhances the robustness and accuracy of our approach in identifying and disambiguating entities, leading to improved performance in Entity Linking.

3.3 Mitigating LLMs Hallucination

During our implementation, we found that LLM hallucination is a critical problem, especially in the augmentation and disambiguation phase. In the disambiguation step, the LLM model occasionally predicts labels, that do not exist in Wikipedia. To avoid this, we generated a dictionary that maps all Wikipedia titles to their URIs. By applying this dictionary, we can omit all annotations, where no exact match in the dictionary exists. In the augmentation step, hallucination occurs when the LLM model provides augmentations for spans that do not exist in the sequence or are not annotated by the NER step. To avoid misleading expansions, we only consider those spans for expansion that precisely match one of the annotated spans from the NER step. Unlike, the augmentation and disambiguation steps, we did not encounter problems with hallucination in the NER step.

4 Ablations

We conducted various ablations of our model to evaluate the impact of augmentation in different setups of EL models, as follows:

End-to-end foundational model Recent studies focuses on developing Entity Linking models that omits mention detection step and directly predicts the entity set (De Cao et al., 2021; Zhang et al., 2021). In the end-to-end (E2E) approach, an expanded set of entities is used to directly compute E^n for the context C.

To setup the end-to-end approach, we trained our T5 model to directly predict the target sequence T from the input context C. This approach aligns with the same experiments setup by De Cao et al. on end-to-end Entity Linking. However, our implementation employs the T5-base model instead of the BART model. We selected the T5-based model due to its superior performance without requiring a prefix trie. Additionally, we used an augmentation strategy with two inference steps with the model: first, the model identifies entity mentions to be expanded, then, performs final entity disambiguation. After the first step, we extract entities mentions from the output sequence, ignoring the predicted titles. Subsequently, we use the same LLM prompt, as in the augmentation Section 3.2.2, to find possible expansions. To integrate these expansions into the sequence, we replace the mentions with their expanded forms, omitting the tags for annotating entity mentions, similar to the foundational model.

NER augmentation To improve the performance of our Entity Linking system, we introduce our LLM prompt to find additional entity spans in the input texts. We use the following instruction to guide the LLM in finding accurate entity spans from the input text:

> **LLM Prompt**
>
> `Please` **`generate one list with all entities`** `from the following text in JSON format, excluding numbers. Do not format the json output:`
> **`Context`**`: 'Angelina met her partner Brad and her father Jon in AK.'`

We then apply a regular expression, similar to the first expansion strategy, to extract the new entity spans. Since we cannot rely on specified indices in the LLM output, we only consider spans that exactly match the original input sequence. To avoid overlapping annotations, we order the newly extracted spans by length in descending order and add expansions only where no surrounding annotation is present.

Alternative NER approaches we conducted experiments using the state-of-the-art framework Flair (Akbik et al., 2019) for the mention detection step. Additionally, we experimented with a hybrid approach that employs an end-to-end foundational model for mention detection and then introduces those entities into the disambiguation step.

5 Experiments

We conducted our experiments to answer the following research questions:

RQ$_1$ How well does our approach compared to state-of-the-art baselines?

RQ$_3$ How does the LLM-based augmentation impact foundational models?

RQ$_3$ Which augmentation strategy works best?

5.1 Experimental Setup

We fine-tuned the T5 model, using it as a foundational model, on the KILT dataset. The fine-tuning was conducted on 4 NVIDIA-A100 GPUs, each with 40GB of memory, for one week. Following this, we further trained the model on the AIDA-train dataset for up to 125 epochs with an early stopping strategy. We used the AIDA-test-A split as the development set. We chose the base version of the T5 model due to its size, which is comparable to the models used in baseline approaches. The T5 implementation was obtained from Hugging Face.[2] Our training setup mirrors that of the baseline approaches; no additional datasets were used for further training. The end-to-end foundational model, as detailed in Section 4, was trained under the same conditions. For inference, we deployed the foundational model and the LLM for the augmentation strategy on a separate machine equipped with two NVIDIA H100 GPUs. This setup facilitated the use of large models like the LLaMA3 with 70 billion parameters.

5.2 Evaluation

We conducted our experiments using the GERBIL framework (Röder et al., 2018), which benchmarks Entity Linking on various datasets. We configured an A2KB experiment and integrated our approach as a web service. We report the InKB micro F1 scores, which is a standard metric in the literature. This metric considers only entities with a corresponding link in the target knowledge graph, thereby excluding *out-of-wiki* entities from the evaluation (Röder et al., 2018). For our baseline experiments, we focused only on the approaches that were evaluated in an end-to-end setup, including NER. This inclusion is critical as the NER output significantly impacts the performance of entity disambiguation.

5.3 Datasets

We used different datasets for the end-to-end Entity Linking task in our experiments (De Cao et al., 2021; Zhang et al., 2021). These datasets are: AIDA-test-B (Hoffart et al., 2011)(AIDA), Derczynski (Derczynski et al., 2015)(DER), KORE 50 (Hoffart et al., 2012)(K50), MSNBC, NS3-Reuters-128, NS3-Reuters-500 (Röder et al., 2014)(R-128,R-500), and the OKE challenge datasets OKE-2015 and OKE-2016 (Nuzzolese et al., 2015). Table 1 provides detailed statistics of these datasets, including number of entities that have a corresponding entry in the knowledge graph (#InKG entities) and number of documents (#Docs). The AIDA-test-B (Hoffart et al., 2011) dataset contains the largest number of entities with corresponding entries in the knowledge graph. Given that our models were trained on the AIDA-train dataset, AIDA-test-B serves as an in-domain dataset. All other datasets are considered out-of-domain. For the knowledge graph, we used the 2019-Wikidata-dump from the KILT dataset, [3] which is commonly used by state-of-the-art Entity Linking systems.

Table 1: Dataset Statistics

Dataset	**#InKB entities**	**#Docs**
AIDA-test-B	4,485	230
Der	201	183
KORE50	139	48
MSNBC	737	20
N3-Reuters-128	626	115
N3-RSS-500	515	425
oke-2015	481	100
oke-2016	221	55

5.4 Experimental Results

5.4.1 Comparison to Baseline Approaches (RQ)1

In this section, we compared the performance of our model against state-of-the-art baselines across various datasets. We obtained the scores of baselines from Zhang et al.. Our evaluation focused on three setups that demonstrated the most stable results: the foundational T5 model with entity span expansion (T5$_{(NER+ED)}$), a combination of the T5 model with the Flair framework for NER (Flair & T5$_{(ED)}$), and an end-to-end T5 ablation model (Section 4) with span expansion (E2E T5).

Table 2 reports the evaluation results on across all datasets. Unlike traditional Entity Linking systems such as (Hoffart et al., 2011) and (van Hulst et al., 2020), which uses separate components for span detection and entity disambiguation, our models do not generate candidate entity sets. Instead, we use the fine-tuned T5 models with LLM-augmentation for entity expansions for contextual information. Our model achieves the best performance on all datasets except AIDA and MSNBC. On the MSNBC dataset, our model's performance is comparable to the state-of-the-art. However, we observed a performance drop on the in-domain AIDA-test-B dataset compared to the baselines. Interestingly, the expansion strategy is less effective on the AIDA and MSNBC datasets, as described in Section 5.4.2. This might be due to our foundational models have been trained the AIDA data, so

[2] https://huggingface.co/docs/transformers/model_doc/t5

[3] https://github.com/facebookresearch/KILT

Table 2: Performance comparison against baseline approaches using InKB micro F1 (**RQ**$_1$). T5$_{(\mathrm{ED})}$ is fine-tuned for Entity Disambiguation, while T5$_{(\mathrm{NER+ED})}$ is jointly fine-tuned for both NER and Disambiguation. The best result is highlighted in bold.

Approach	AIDA	MSNBC	Der	K50	R-128	R-500	OKE 2015	OKE 2016	Avg
Hoffart et al.	72.8	65.1	32.6	55.4	46.4	42.4	63.1	0.00	47.2
Steinmetz and Sack	42.3	30.9	26.5	46.8	18.1	20.5	46.2	46.4	34.7
Moro et al.	48.5	39.7	29.8	55.9	23.0	29.1	41.9	37.7	38.2
Kolitsas et al.	82.4	72.4	34.1	35.2	50.3	38.2	61.9	52.7	53.4
van Hulst et al.	80.5	72.4	41.1	50.7	49.9	35.0	63.1	58.3	56.4
De Cao et al.	83.7	**73.7**	54.1	60.7	46.7	40.3	56.1	50.0	58.2
Zhang et al.	**85.8**	72.1	52.9	64.5	54.1	41.9	61.1	51.3	60.5
Flair & T5$_{(\mathrm{ED})}$	71.4	61.3	51.6	**72.7**	**54.5**	56.3	**66.6**	**61.5**	**62.0**
T5$_{(\mathrm{NER+ED})}$	71.6	69.3	**55.7**	70.6	51.7	**56.6**	59.4	58.5	61.7
E2E T5	69.0	64.2	53.7	64.3	51.9	57.3	61.6	58.4	60.1

the expansion did not provide substantial new contextual information for the input sequences. The MSNBC dataset, which includes 20 news articles, shares similarities with the in-domain AIDA-test-B dataset. The other datasets often contain much shorter texts with fewer entities and are exclusively news-based.

Our findings indicate that the more a dataset differs from our training data, the better our model performs relative to the baselines. Notably, on the KORE 50 dataset, which contains highly ambiguous entities, our model with a joint mention detection and disambiguation improves the F1 score by over 6 points and 8 points when combined with the third-party flair NER framework.

5.4.2 Evaluation of foundational models (RQ)$_2$

To address this research question, we evaluated the performance of various foundational models, specially T5, and investigated how our LLM-based augmentation strategy on affected their performances. We employed the LlaMA3 model for entity expansion (e.g., `"Angelina" to "Angelina Jolie"`) to facilitate entity disambiguation, as described in Section 5.4.3).

Table 3 reports the evaluation results for this experiment. We employed the same models as in the previous section, along with the mixed foundational model (Mixed model) presented in 4. By analyzing the results of both the foundational models and those enhanced with our expansion strategy, we observed significant improvements on most out-of-domain datasets. The augmented version of the traditional setup, which separates mention detection and evaluation– outperforms the augmented end-to-end model on five out of eight datasets. For the other three datasets, the performance difference between the two setups was negligible. However, without augmentation, the end-to-end model generally performs works better than the traditional setup. This finding suggests that our LLM-based entity expansion is particularly effective with a traditional setup for disambiguating and liking entities to knowledge graphs more accurately.

5.4.3 Evaluation of augmentation strategies (RQ)$_3$

Table 4 shows the results for the two augmentation strategies –mention expansion and NER expansion– for both the foundational model (T5$_{(\mathrm{NER+ED})}$) and the combination of Flair and the foundational model (Flair & T5$_{(\mathrm{ED})}$). As previously described, the mention expansion consistently improves the performance of the Entity Linking system. Conversely, the NER expansion strategy shows variable efficacy on some of the datasets. It improves performance on some datasets but leads to worse results on others compared to using only mention expansion. This variation in performance may originate from the differences in annotation strategies among the datasets. Particularly, in the Der dataset, there is a significant variance in performance, where the NER expansion strategy annotates more entities the benchmarks datasets, negatively impacting results relative to the foundational model and other strategies.

To further explore this issue, we conducted an additional experiment using only the LLM expansion strategy for entity extraction, labeled as "LLM-only" in Table 4). The results indicate that the LLM-only strategy performs well with short texts

Table 3: Comparison of different foundational models (**RQ**)$_2$. The first 4 rows present of the approach, when no augmentation strategy is applied. The other rows present results when the entity expansion strategy (Section 3.2.2) is applied. The best result is highlighted in bold.

Model	AIDA	MSNBC	Der	K50	R-128	R-500	OKE 2015	OKE 2016	Avg
Flair & T5$_{\text{(ED)}}$	**73.5**	67.0	48.8	50.0	41.0	55.1	58.9	55.0	56.2
T5$_{\text{(NER+ED)}}$	67.4	61.5	53.9	50.0	47.6	55.2	55.9	56.0	56
E2E T5	68.0	63.0	50.4	49.6	49.1	56.8	56.1	56.5	56.1
Mixed model	66.8	61.9	53.5	48.5	49.2	55.3	56.8	50.0	55.2
Flair & T5$_{\text{(ED)}}$ aug.	71.4	61.3	51.6	**72.7**	**54.5**	56.3	**66.6**	**61.5**	**62.0**
T5$_{\text{(NER+ED)}}$ aug.	71.6	**69.3**	**55.7**	70.6	51.7	56.6	59.4	58.5	61.7
E2E T5 aug.	69.0	64.2	53.7	64.3	51.9	57.3	61.6	58.4	60.1
Mixed model aug.	69.0	63.1	55.0	69.6	52.3	**58.1**	60.6	53.5	60.2

Table 4: Evaluation of different augmentation strategies (**RQ**$_3$). Models with NER-exp indicate the use of both NER and mention expansion, while other models use only mention expansion. The best result is highlighted in bold.

Model	AIDA	MSNBC	Der	K50	R-128	R-500	OKE 2015	OKE 2016	Avg
T5$_{\text{(NER+ED)}}$	**71.6**	69.3	**55.7**	70.6	51.7	**56.6**	59.4	58.5	61.7
T5$_{\text{(NER+ED)}}$, NER-exp	68.0	**70.3**	37.4	69.3	53.0	45.4	44.6	**57.4**	55.7
Flair & T5$_{\text{(ED)}}$	71.4	61.3	51.6	72.7	**54.5**	56.3	**66.6**	61.5	**62.0**
Flair & T5$_{\text{(ED)}}$, NER-exp.	67.4	57.0	37.1	76.1	51.7	46.5	62.8	61.2	57.5
LLM-only	63.7	62.6	36.4	**75.1**	49.3	47.3	61.7	59.6	57.0

in the KORE 50 dataset but underperforms on other datasets when not combined with with the T5 model. We retain the T5 model for predicting final identifiers, since using mention expansion alone led to hallucinations, making many predicted identifiers untraceable in the title dictionary.

6 Conclusion and future work

In this paper, we present our approach for Entity Linking using a jointly fined-tuned model and contextual augmentation with LLMs. In particular, our approach employs a fine-tuned T5 model that integrates NER and disambiguation tasks into a unified framework, reducing resource demands compared to separate models for each task. Although this setup may slightly drop performance, our experiments showed that this performance loss is only marginal, mainly due to unseen entities. Furthermore, our approach leverages the LlaMA-3-70B model to expend entities mentions with contextual augmentation. The evaluation results demonstrate that LLM-based augmentation significantly improves the performance on out-of-domain datasets, achieving state-of-the-art results compare to traditional two-step methods (i.e., entity recognition and disambiguation).

Future work will focus on analyzing the performance of LLM-based disambiguation strategies in predicting rare entities, which require creating a new benchmark datasets. In the appendix, we present an additional experiment that shows larger LLMs performs better than smaller ones, as they return more consistent output with fewer variations.

7 Limitations

Similar to the approaches (De Cao et al., 2021), we use Wikipedia as the main knowledge graph, where unique titles facilitate entity identification. However, this method is not always possible with other knowledge graphs like Wikidata (Vrandečić and Krötzsch, 2014). Existing benchmarks and training datasets are also based on Wikipedia, making text-based features sufficient for efficient Entity Linking. However, in other knowledge graphs, the graph-based structure is crucial for disambiguation. For instance, the German state of Berlin and the city of Berlin share the same label but are distinct entities, making interconnected entities crucial for disambiguation. The current benchmarking datasets used for evaluation are outdated and do not address newer challenges like predicting rare entities. Additionally, it is unclear to what extent these datasets have been included in the training data of current LLMs. Therefore, future research should focus on developing new datasets that also address the limitations of modern LLMs.

Acknowledgement

This work has been supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL under the grant no NW21-059D, by the German Federal Ministry of Education and Research (BMBF) within the project KIAM under the grant no 02L19C115, by the German Federal Ministry of Education and Research (BMBF) within the project COLIDE under the grant no 01IS21005D and the European Union's Horizon Europe research and innovation programme under grant agreement No 101070305.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing &amp; Management*, 51(2):32–49.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. KORE: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 545–554, New York, NY, USA. Association for Computing Machinery.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Tuan Manh Lai, Heng Ji, and ChengXiang Zhai. 2022. Improving candidate retrieval with entity profile generation for wikidata entity linking. *arXiv preprint*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2017. Mag: A multilingual, knowledge-base agnostic and deterministic entity linking approach. In *K-CAP 2017: Knowledge Capture Conference*, page 8. ACM.

Isaiah Onando Mulang', Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. 2020. Evaluating the impact of knowledge graph context on entity disambiguation models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2157–2160. ACM.

Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. 2015. Open knowledge extraction challenge. In *SemWebEval@ESWC*.

Italo L. Oliveira, Renato Fileto, René Speck, Luís P.F. Garcia, Diego Moussallem, and Jens Lehmann. 2021. Towards holistic entity linking: Survey and directions. *Information Systems*, 95:101624.

Alberto Parravicini, Rhicheek Patra, Davide B. Bartolini, and Marco D. Santambrogio. 2019. Fast and accurate entity linking via graph embedding. In *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA'19, New York, NY, USA. Association for Computing Machinery.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: Multilingual entity linking by neural type system evolution. *Preprint*, arXiv:1802.01021.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. 2014. N^3 - a collection of datasets for named entity recognition and disambiguation in the NLP interchange format. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3529–3533, Reykjavik, Iceland. European Language Resources Association (ELRA).

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. GERBIL - benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625.

Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570.

Nadine Steinmetz and Harald Sack. 2013. Semantic multimedia information retrieval based on contextual descriptions. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 382–396. Springer.

Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 2197–2200, New York, NY, USA. Association for Computing Machinery.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Sijia Wang, Alexander Hanbo Li, Henghui Zhu, Sheng Zhang, Pramuditha Perera, Chung-Wei Hang, Jie Ma, William Yang Wang, Zhiguo Wang, Vittorio Castelli, et al. 2023. Benchmarking diverse-modal entity linking with generative models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7841–7857.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Amy Xin, Yunjia Qi, Zijun Yao, Fangwei Zhu, Kaisheng Zeng, Xu Bin, Lei Hou, and Juanzi Li. 2024. LLMAEL: Large language models are good context augmenters for entity linking. *Preprint*, arXiv:2407.04020.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2021. EntQA: Entity linking as question answering. *CoRR*, abs/2110.02369.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. DoSeR - a knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In *The Semantic Web. Latest Advances and New Domains*, pages 182–198, Cham. Springer International Publishing.

A Comparison of different LLM models

As an appendix, we evaluated, which LLM models perform best for our entity span expansion strategy. We experimented with models with different numbers of parameters. The LLama 3 model with 70 billion parameters worked best compared to all other models. Our results show, that the expansion works better the more parameters the model has. The Mistral and the LLama 3 model with 8 billion parameters achieve similar performance. The main reason for the difference in performance is that the larger models produce more stable outputs compared to the smaller models. In the larger models, the JSON output usually has quite similar formatting over all documents. In comparison, in the smaller models, the JSON output was slightly different from document to document, which makes it hard to extract the expansions from the output. Furthermore, the output was not always complete as there were some entities missing especially in larger sequences.

Table 5: Comparison for different models for the expansion. The best result is highlighted in bold.

Model	AIDA	MSNBC	Der	K50	R-128	R-500	OKE 2015	OKE 2016	Avg
LLama3 70B	**71.6**	**69.3**	**55.7**	**70.6**	**51.7**	**56.6**	**59.4**	**58.5**	**61.7**
LlaMA3 8B	69.6	67.7	53.2	53.3	48.1	54.7	53.5	49.1	56.2
LLama 2 70B	70.2	68.1	51.5	57.9	48.2	57.4	57.2	50.5	57.6
LLama 2 7B	70.3	70.0	53.0	51.5	47.7	54.7	55.6	54.3	57.1
Mistral	70.7	67.5	54.2	48.3	47.6	56.1	56.1	56.1	57.1