Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering using only Knowledge Graphs

Cedric Möller [a,*] and Ricardo Usbeck [a,b]

[a] *Semantic Systems Group, Universität Hamburg, Mittelweg 177, 20148 Hamburg, Germany*
E-mail: firstname.lastname@uni-hamburg.de
[b] *HITeC Hamburg e.V., Vogt-Kölln-Straße 30, 22527 Hamburg, Germany*

Abstract. Entity Linking is crucial for numerous downstream tasks, such as question answering, knowledge graph population, and general knowledge extraction. A frequently overlooked aspect of entity linking is the potential encounter with entities not yet present in a target knowledge graph. Although some recent studies have addressed this issue, they primarily utilize full-text knowledge bases or depend on external information. However, these resources are not available in most use cases. In this work, we solely rely on the information within a knowledge graph and assume no external information is accessible.

To investigate the challenge of identifying and disambiguating entities absent from the knowledge graph, we introduce a comprehensive silver-standard benchmark dataset that covers texts from 1999 to 2022. Based on our novel dataset, we develop an approach using pre-trained language models and knowledge graph embeddings without the need for a parallel full-text corpus. Moreover, by assessing the influence of knowledge graph embeddings on the given task, we show that implementing a sequential entity linking approach, which considers the whole sentence, can outperform clustering techniques that handle each mention separately in specific instances.

1. Introduction

Entity Linking (EL) [1] is an essential part of numerous downstream tasks, such as question answering [1], knowledge graph population [2] or relation extraction [3]. Yet, EL is still accompanied by several challenges. The main problem is the **ambiguity of entity mentions**. If multiple entities in a knowledge graph (KG) can be referred to by the same name, deciding on the correct one becomes increasingly difficult. The inclusion of context information in the KG or in the input text is usually employed to solve this.

A second problem, only rarely thoroughly considered is the possibility of out-of-KG entities. These are entities that are referred to in the input text but do not actually exist in the KG yet. Consider for example the news message "The President of the Japan Football Association and deputy Olympic Committee chief Kozo Tashima tests positive for <u>COVID-19</u>. Japan insists the 2020 Summer Olympics will still go ahead as planned.". This message is from the beginning of 2020. The mentioned entity "COVID-19" might not yet have existed in a target KG. Hence, an

*Corresponding author. E-mail: cedric.moeller@uni-hamburg.de.

[1]Note that in literature there is a separation between Entity Linking and Entity Disambiguation. The latter means only the disambiguation part and not the entity mention recognition part. According to this difference, we target Entity Disambiguation in this work, however, refer to it as Entity Linking as used in many computational linguistics communities.

entity linker might link it to a different (likely coronavirus-related) and thus incorrect entity. [2] Most methods ignore this case by assuming that all mentions truly refer to an entity in the KG. However, this would mean that an Entity Recognizer (ER) detecting those entity mentions is able to differentiate between entities in the KG and outside. This is clearly wrong as ERs usually have no access to the KG, in particular in the case of dynamically changing KGs.

We developed an integrated method that can identify and cluster **out-of-KG entities**. While some methods do exist which consider this task jointly, they either rely on external information [4–6] such as crawled webpages or exclusively focus on encyclopedias such as Wikipedia as the underlying knowledge base [7–15]. We want to offer an alternative utilizing purely KGs [16], in our case Wikidata, without using any external data.

Our contributions are as follows:

1. A novel, openly available Entity Linking dataset containing out-of-KG entities;
2. A sequential Entity Linking method supporting
   (a) Detection of out-of-KG entities;
   (b) Clustering of out-of-KG entities;

All our code, dataset and the used KG dump are publicly available at https://github.com/cedricmoeller/ELwithOOKGDetectionClustering for replicability reasons.

2. Method

2.1. Problem definition

Given a document $d = (t_1, t_2, \dots, t_n)$ represented as a sequence of tokens t_i, where each token can be classified as part of a mention m_j or not, the objective is to construct a mapping function $f : M \rightarrow E$ that accurately associates each mention $m_j \in M$ to its corresponding entity $e_k \in E$. Entities can either be present in the Knowledge Graph (KG) or absent from it. For mentions referring to entities not in the KG, the aim is to associate them with one another.

Formally, let $M = \{m_1, m_2, \dots, m_p\}$ be the set of mentions in the document, and $E = E_{\text{in-KG}} \cup E_{\text{out-of-KG}}$ be the set of entities, where $E_{\text{in-KG}}$ is the set of entities in the KG and $E_{\text{out-of-KG}}$ is the set of entities not in the KG. The goal is to find an optimal mapping function f such that:

1. For each mention m_j referring to an entity in the KG, $f(m_j) = e_k \in E_{\text{in-KG}}$.
2. For each mention m_j referring to an entity not in the KG, $f(m_j) = e_k \in E_{\text{out-of-KG}}$, and all mentions referring to the same entity outside the KG are assigned to the same e_k.

2.2. Candidate Generation

As we focus on a KG-only use case, we can not rely on an existing entity mention dictionary as utilized in other EL works [17]. [3] Hence, we can only use information from the KG. This mainly restricts us to labels and aliases existing for each entity. Thus, for candidate generation, we fill an ElasticSearch index with all labels and aliases. We query this index using a combination of TF-IDF and fuzzy search to compensate for less frequent words and possible typos or other small variations. We retrieve a candidate set of size 100. Due to this process, our method does not rely on a parallel text corpus.

[2] We abbreviate entities not in the KG as out-of-KG and entities in the KG as in-KG.

[3] An exception in our work is the evaluation of the AIDA-CoNLL dataset. As many EL works utilize the existing candidate set by Le and Titov [17], we also relied on it to be able to compare and verify the performance in a replicable way.

2.3. *Entity Linker*

The entity linker is a bi-encoder together with an additional ranking model. A bi-encoder was chosen instead of a cross-encoder as it assumed that one needs to link against a large number of candidates. This is necessary to guarantee a large enough recall in the candidate generation. A cross-encoder is deemed too expensive in such a case as one would need to encode the text together with the entity candidate of each mention multiple times. In a bi-encoder, the text and all candidates are encoded separately. It consists of a mention encoder and an entity encoder. Figure 1 depicts the model architecture.

2.3.1. *Mention Encoder*

The mention encoder is based on a pre-trained RoBERTa model.[4] For efficiency, we opt for fine-tuning only bottleneck adapters [19] instead of the whole model. The input to the model is the tokenized input text. All embedded tokens of each entity are averaged and taken as their embedded representation. Furthermore, the embedded representation is scaled via a linear layer to project it to the same space as the entity embedding space. The final embedded vector is defined as e_m.

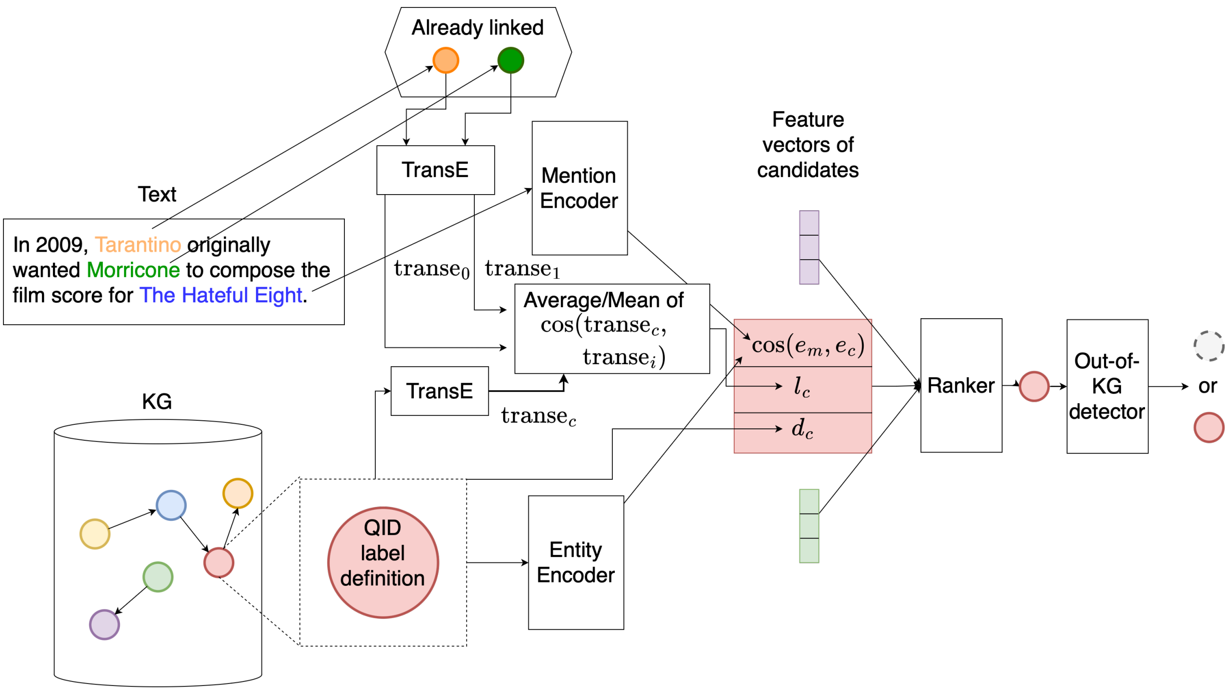

Fig. 1. First stage - Entity linking and out-of-KG detection of the entity mention "The Hateful Eight". The mention is encoded and compared against the entity encoding. The out-degree of the candidate entities are retrieved. Furthermore, the KG embedding of the candidate is compared against already linked entities. All features are fed into a ranker which determines the correct candidate or detects the mention as out-of-KG. The different colors represent different entities.

2.3.2. *Entity Encoder*

The entity encoder creates a latent representation of the KG entity by embedding its definition. We define the definition of each entity as the value of a `schema: description` triple in Wikidata. Note that we use the term *definition* instead of *description* here as entity linking methods often refer to the first paragraph of Wikipedia articles as descriptions. These are much longer than the short descriptions in Wikidata.[5] The definition embedding

[4] We chose RoBERTa-base due to its improved performance over BERT and resource reasons [18].

[5] Of course, the model is compatible with long descriptions but we wanted to focus on the difficulties of only using the information in a KG.

is generated by encoding a concatenation of the entity's main label and definition and feeding it into an adapter-equipped RoBERTa model. The embedded vector of the [CLS] token is taken as the representation of the entity and projected to the same vector space as the mention embedding. The final definition vector of a node c is denoted as e_c.

We consider three additional features for each entity: First, the popularity of a node d_c is measured by the number of outgoing edges of the entity in the KG. The second feature is the TransE-embedded vector transe_c of the node [20]. And the last feature is the type information of an entity.

2.3.3. Ranker

After computing all mention encodings and all assigned candidate entity encodings, as well as their features, it is necessary to combine and rank them. The highest-ranked entity will be the one to which we link.

Linking In-KG candidates The final ranker is a linear layer combining all the aforementioned features. For each candidate-mention-pair, the following inputs are fed into a linear layer.

1. Cosine similarity $\cos(e_c, e_m) = \frac{<e_c, e_m>}{||e_c||_2 ||e_m||_2}$ ($< \cdot, \cdot >$ represents the dot product) between mention embedding e_m and entity definition embedding e_c
2. Node popularity in the form of out-degree d_c
3. Average cosine similarity of the candidate TransE embedding to the TransE embeddings of past linking decisions $l_c = \overline{\cos(\text{transe}_c, \text{transe}_i)} | i \in D$ where D is the set of the past linked entity identifiers

The final logits are calculated as follows (with $\oplus$ denoting vector concatenation):

$$r_c = \text{Linear}(\cos(e_c, e_m) \oplus d_c \oplus l_c)$$

Note that the fourth feature utilizes the TransE embeddings of past linking decisions which introduces *sequentiality*. This implies that each linking decision is influenced by the preceding decision within the same document.

out-of-KG decision The out-of-KG entity detection decision is determined by looking at the maximum-scored entity candidate and deciding whether it is similar enough to the mention. If not, it is an out-of-KG entity. During training, we rely on softmax, to consider all candidates. First, $\sigma(r_c)$ over all candidates is calculated and then multiplied with the feature concatenation $\frac{<e_c, e_m>}{||e_c||_2 ||e_m||_2} \oplus d_c \oplus l_c$ of all candidates to get an accumulated feature vector a. σ stands here for the softmax operation over all candidates. This vector is fed through another single-layer network to get an additional scalar:

$$r_{\text{out-of-KG}} = \text{Linear}(a)$$

By introducing this additional decision, we are able to detect out-of-KG entities directly without relying on a validation dataset to tune a threshold. Also, it is not necessary to train the model with actual out-of-KG entities. During training, the model is trained by randomly including or excluding the true candidate from the candidate list.

2.4. Clustering out-of-KG entities

A detected out-of-KG entity is represented by its mention embedding e_m, and its surrounding linked entities. To identify whether two out-of-KG entities refer to each other, we apply a linear layer to two features: 1) the cosine similarity between both entity embeddings, and 2) the mean cosine distance between the TransE embeddings of the linked entities that surround the mentions. To obtain informative cosine similarities, we further optimize the model to return larger cosine similarities for mentions pointing to the same entity and smaller cosine similarities for mentions pointing to different entities. It is trained via cross-entropy loss where negative mentions are all other mentions in the same batch not referring to the same entity. Using the pairwise scores output by the linear layer, we cluster all out-of-KG detected entities via DBSCAN clustering.[6] The process is depicted in Figure 2.

[6] We also evaluated agglomerative average-linkage, maximum-linkage and single-linkage clustering but achieved worse results.

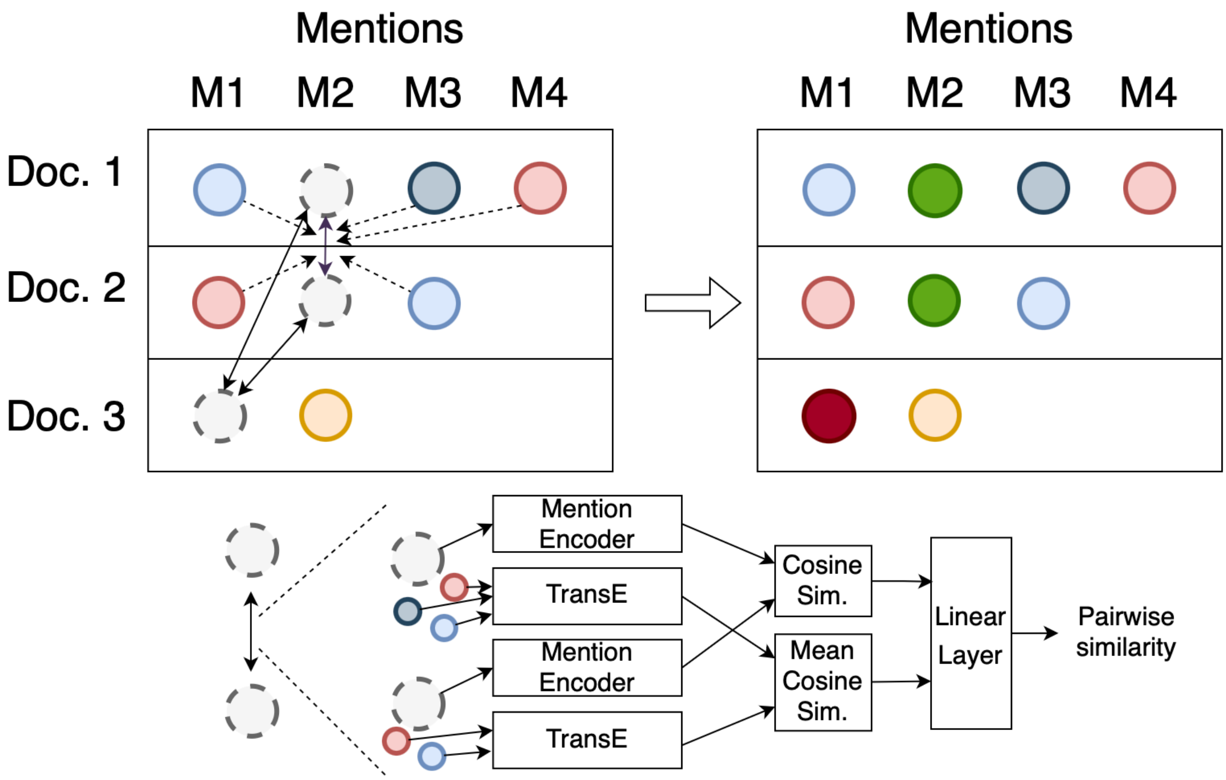

Fig. 2. Second stage - Clustering the out-of-KG detected entities. Left: Mentions before clustering with three ambiguous mentions. Right: Mentions after clustering with one pair grouped together (green) and another being a singleton (red). Circles with dotted borders illustrate out-of-KG entity mentions. Dotted arrows signalize the impact of already linked entities on the similarity measure between mentions. Note that not all dotted arrows are drawn to simplify the figure.

2.5. *Training*

The final loss function consists of multiple aggregated separate loss functions. For entity linking, mention-mention similarity and mention-entity similarity cross-entropy loss was employed. All losses are aggregated equally to return the final loss. Early stopping with a patience of 10, warm-up and a linear learning rate decay are employed. Due to the size of Wikidata, training our own embeddings was intractable. Hence, we opted for the trained set provided by PyTorch-BigGraph [21]. The model was trained for 30 epochs on a single NVIDIA RTX A6000 machine.

2.6. *Inference*

When the TransE embeddings are used as a feature in the ranker, beam search with 10 beams is employed to identify the best sequence of linked entities. We evaluated different window sizes for the included surrounding entities in both, the sequential linking and the out-of-KG entity clustering, and determined a window size of 6 as the best.

3. Experiments

The experiments are split into three parts. First, we evaluated the performance of the chosen entity linking method in regard to the used features. The best-performing model was then used further. Secondly, we evaluated the impact of the out-of-KG detection mechanism. After the best model was determined, the capability of methods to not only detect out-of-KG entities but also cluster them is examined.

	train	dev	test	overall
# examples	63,623	11,205	11,206	86,034
# mentions	185,039	32,652	32,660	250,351
# out-of-KG mentions	0	2,579	2,519	5,098
# unique entities	38,066	9,349	9,386	45,655
# unique out-of-KG entities	0	751	734	1,221
Average of # mentions per example	2.9	2.91	2.91	2.9

Table 1
Statistics of Wikievents dataset

3.1. Methods

We compare our sequential method to three different clustering-based methods which cluster all mentions and entities at once. They are the state-of-the-art NASTyLinker by Heist et al. [22] (denoted NASTyLinker), the top-down clustering approach by Kassner et al. [23] (denoted Edin) and the bottom-up clustering approach by Agarwal et al. [24] (denoted bottom-up).[7] All clustering methods are evaluated by using the trained bi-encoders for computing the similarities between the mentions and the ranker for the mention-candidate similarities.

3.2. Datasets

To examine how to handle out-of-KG entities in the task of EL, we created an entity linking dataset from the current-events page of Wikipedia [8], dubbed `Wikievents`. On the current-events page, short news snippets stating recent events are available. These texts contain hyperlinks to articles in Wikipedia. We crawled the current-events page texts between 1999-12-29 and 2022-10-01. Each hyperlink was identified and taken as an entity mention. The corresponding page title of the Wikipedia article was mapped to the Wikidata QID. Furthermore, all entity mentions retrieved were filtered further by only keeping those which are `instance of` (P31) of some class and were no `subclass of` (P279) of any other class. The data was split into a train, development, and test set according to the ratios $(0.74, 0.13, 0.13)$. The cutoff date for the knowledge graph and the examples of the development and test sets are 2019-01-28. The development and test sets are created by randomly splitting all examples after the cutoff date. The statistics of the dataset can be found in Table 1. Note, the training dataset contains no out-of-KG entities as the included texts all are from before the cutoff date. Also, three examples from the dataset can be seen in Figure 3.

Example 1: <u>**Hoda Muthana**</u>, an **Alabama** woman who joined the **Islamic State**, is banned from entering the **United States**. .

Example 2: <u>**Peter Kaiser**</u> wins the 2019, 1000-mile Iditarod, arriving in **Nome** in 9 days, 12 hours and 39 minutes..

Example 3: In the aftermath of <u>**Cyclone Idai**</u>, those infected by cholera jump to 139 confirmed cases in **Mozambique**..

Fig. 3. Wikievents example sentences. Entities marked in bold with Out-of-KG entities being underlined.

Additionally, we artificially added out-of-KG entities to the well-known `AIDA-CoNLL` dataset [5]. This dataset was chosen as it is a popular dataset in the entity linking domain. The original dataset had only links to Wikipedia which we mapped to Wikidata. out-of-KG entities were added by gathering all occurring entities and randomly

[7]Note that the original NASTyLinker used an additional cross-encoder. While we do not, a cross-encoder is orthogonal to our changes and can be incorporated in our method as well. By NASTyLinker we refer here to the employed clustering method, not the mention-entity scoring mechanism.

[8]https://en.wikipedia.org/wiki/Portal:Current_events

	train	dev	test	all
# examples	946 (946)	216 (216)	231 (231)	1,393 (1,393)
# mentions	34,268 (46,678)	9,558 (11,824)	8,942 (11,206)	52,768 (69,708)
# out-of-KG mentions	0 (9,710)	952 (2,252)	900 (2,262)	1,852 (14,224)
# unique entities	3,935 (4,065)	1,638 (1,641)	1,530 (1,532)	5,562 (5,569)
# unique out-of-KG entities	0 (1)	166 (1)	155 (1)	275 (1)
Average of # mentions per example	36.2 (49.3)	44.2 (54.7)	38.7 (48.5)	37.9 (50.0)

Table 2

Statistics of artificially created out-of-KG-entity enriched AIDA-CoNLL dataset (statistics of the original dataset in brackets)

Dataset	Entity type	1	2	3	4	5	6-10	11-20	21-50	50-
Wikievents train	in-KG	62.6	14.4	6.5	3.4	2.3	5.0	2.7	1.9	1.2
	out-of-KG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wikievents dev	in-KG	65.6	13.9	5.9	3.5	2.0	4.6	2.5	1.3	0.7
	out-of-KG	75.4	12.1	4.9	1.5	0.9	2.5	0.7	1.1	0.9
Wikievents test	in-KG	65.8	13.9	5.8	3.1	2.2	4.7	2.2	1.6	0.6
	out-of-KG	75.7	12.5	2.9	2.5	1.4	1.9	1.1	1.1	1.0
AIDA-CoNLL train	in-KG	45.6	19.1	9.3	7.2	3.5	8.8	3.3	2.4	0.9
	out-of-KG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AIDA-CoNLL testa	in-KG	54.4	21.2	6.5	4.8	2.4	6.7	2.5	1.2	0.3
	out-of-KG	47.6	24.1	7.2	4.8	2.4	10.2	3.0	0.6	0.0
AIDA-CoNLL testb	in-KG	48.4	22.3	9.7	7.1	3.7	5.1	2.0	1.5	0.2
	out-of-KG	48.4	25.8	7.7	3.9	4.5	5.8	2.6	0.6	0.6

Table 3

Percentage of entities occurring in clusters of different sizes

selecting 10% of all entities that occurred and declaring them out-of-KG. We removed each mention of such an entity from the AIDA-CoNLL training set. Also, we removed all already existing out-of-KG entities to only focus on the artificial ones. Note that those already existing were without identifiers and therefore not of use for our setting. The statistics of the AIDA datasets can be found in Figure 2. In brackets, the original statistics can be found. Note that this version of AIDA-CoNLL is already mapped to Wikidata. Additional statistics on the mention clusters in the datasets can be found in Table 3.

3.3. Evaluation metrics

To evaluate the performance of the EL without out-of-KG detection, we report the in-KG linking accuracy. It is calculated by dividing the number of correctly linked mentions by the number of all mentions. EL with out-of-KG detection is evaluated by calculating the Precision, Recall and F-measure. The true positives are entity mentions detected correctly as being in the KG and correctly linked. False positives are entity mentions which were incorrectly linked to entities in the KG. This encompasses entity mentions referring to entities in the KG and entity mentions not existing in the KG yet. Lastly, false negatives are entity mentions which do refer to being in the KG but are detected as being out-of-KG.

Model	Accuracy
Mention Encoder	0.852 ± 0.002
Pop.	0.615 ± 0.000
TransE.	0.643 ± 0.000
Mention Encoder + Pop.	0.850 ± 0.002
Mention Encoder + TransE	0.866 ± 0.003
Mention Encoder. + Pop. + TransE	**0.868 ± 0.004**

Table 4

Comparison of entity linking performance with different features on AIDA-CoNLL (repeated with three different seeds)

Furthermore, to evaluate only the identification, we gather three additional metrics. First, the number of out-of-KG entities correctly detected divided by the number of all out-of-KG entities (denoted as out-of-KG-IA), second, the same for all in-KG entities (denoted as in-KG-IA) and third, the harmonic mean of the other two (denoted as H-IA).

Lastly, to evaluate the clustering performance, we measured the CEAF, MUC and B^3 as commonly employed in coreference resolution [25]. Furthermore, we report the in-KG F-measure, the out-of-KG F-measure (defined by Kassner et al. [23]) and the combination of them $\frac{F_{\text{in-KG}}F_{\text{out-of-KG}}}{2}$. These measures are reported for out-of-KG entity mentions and entity mentions in the KG.

3.4. Results

3.4.1. Entity Linking Performance

In the first step, we evaluated how the different features affected the entity linking performance on the modified AIDA-CoNLL dataset. In this section, our primary goal is to select a suitable entity linking method that will be effective in the later stages of our research, particularly for handling out-of-KG entities. As such, we have opted not to compare our approach with other existing methods at this point, instead concentrating on identifying the specific features and determining the potential advantages of a sequential linking method. As can be seen in Table 4, the mention encoder itself already contributes the most to the overall performance. The popularity contributes only slightly. Nevertheless we see that popularity is still a feature disambiguating nearly 61.5% of all mentions.

3.4.2. out-of-KG detection

In the Tables 5, we show the precision, recall and F-measure of the models trained with different out-of-KG ratios (how likely the true candidate is removed from the candidate set). As can be seen, the recall is the largest if the model can ignore to detect out-of-KG entities. This is the case as no entity mentions are filtered out and all are linked to an entity in the KG. However, the precision is lower as all out-of-KG entities are misdetected as being in the KG and hence automatically also mislinked. The larger the ratio is, the higher the precision becomes. Of course, when only out-of-KG entities occur in the training data, the precision decreases again as the model does not learn to link at all. Interestingly, the maximum F-measure is reached at different points for the different datasets. For AIDA-CoNLL, the largest F-measure is reached at a small out-of-KG ratio of 0.05 while for the Wikievents-test set it is reached at 0.50 (From 0.60 on it decreased for Wikievents). We suspect that this is the case as the candidate sets in both cases have a different candidate gold-candidate recall rate (how often the true entity is in the candidate set). For AIDA-testb, the gold-candidate recall is 0.97, while for Wikievents it is 0.89.[9] Hence in AIDA, a larger ratio leads to more entity mentions detected as out-of-KG while the true entity is in the candidate set. **Nevertheless, it is clear, that incorporating the detection of out-of-KG entities during training leads to a higher F-measure.**

[9]These rates are not in conflict with the previous statement that Wikievents mentions are less ambiguous. Different candidate generation methods were applied as mentioned in 2.2 resulting in different gold-candidate recall rates.

out-of-KG ratio	P	R	F1
0.0	0.795	**0.867**	0.829
0.05	0.860	0.844	**0.852**
0.1	0.860	0.827	0.843
0.2	0.875	0.815	0.844
0.3	0.875	0.804	0.838
0.4	0.872	0.788	0.828
0.5	**0.880**	0.756	0.814
1.0	0.447	0.071	0.122

(a) AIDA-testb

out-of-KG ratio	P	R	F1
0.0	0.771	**0.834**	0.801
0.05	0.819	0.831	0.825
0.1	0.832	0.829	0.830
0.2	0.851	0.824	0.837
0.3	0.861	0.820	0.840
0.4	0.875	0.816	0.845
0.5	**0.887**	0.809	**0.846**
1.0	0.011	0.000	0.001

(b) Wikievents-test

Table 5

Effect of different sample ratios for out-of-KG entities on entity linking performance

3.4.3. *Clustering out-of-KG entities*

For all four methods, the hyperparameters were tuned on the validation set and then used on the test set in regard to the combined F-measure of out-of-KG entity linking and in-KG entity linking. The metrics are presented separately for in-KG entities and out-of-KG entities.[10]

	Sequential	Seq. w/o transe	Bottom-up	Edin	NASTyLinker
$CEAF_{inkg}$	0.929 ± 0.002	0.929 ± 0.002	0.888 ± 0.010	**0.938 ± 0.005**	<u>0.932 ± 0.001</u>
MUC_{inkg}	0.989 ± 0.000	0.989 ± 0.000	0.985 ± 0.001	**0.993 ± 0.000**	<u>0.991 ± 0.001</u>
$B3_{inkg}$	0.938 ± 0.001	0.938 ± 0.001	0.916 ± 0.005	**0.952 ± 0.003**	<u>0.944 ± 0.002</u>
MUC_{ookg}	0.981 ± 0.001	0.981 ± 0.001	0.957 ± 0.002	**0.992 ± 0.001**	<u>0.989 ± 0.001</u>
$B3_{ookg}$	0.866 ± 0.004	0.864 ± 0.006	0.594 ± 0.025	**0.958 ± 0.003**	<u>0.929 ± 0.017</u>
$CEAF_{ookg}$	0.821 ± 0.009	0.819 ± 0.011	0.450 ± 0.022	**0.945 ± 0.004**	<u>0.911 ± 0.021</u>
$F1_{inkg}$	**0.843 ± 0.002**	**0.843 ± 0.002**	0.826 ± 0.001	0.836 ± 0.005	0.831 ± 0.009
$F1_{ookg}$	<u>0.605 ± 0.014</u>	0.603 ± 0.016	0.048 ± 0.004	**0.613 ± 0.021**	0.534 ± 0.047
$F1_{combined}$	<u>0.724 ± 0.007</u>	0.723 ± 0.009	0.437 ± 0.001	**0.725 ± 0.013**	0.682 ± 0.027

(a) AIDA-CoNLL testb

	Sequential	Sequential w/o transe	Bottom-up	Edin	Nasty
$CEAF_{inkg}$	0.894	0.894	0.875	<u>0.906</u>	**0.927**
MUC_{inkg}	0.951	<u>0.951</u>	0.939	0.950	**0.963**
$B3_{inkg}$	0.911	0.910	0.882	<u>0.924</u>	**0.933**
MUC_{ookg}	<u>0.915</u>	0.912	0.900	0.862	**0.931**
$B3_{ookg}$	0.714	<u>0.726</u>	0.680	0.693	**0.758**
$CEAF_{ookg}$	0.621	0.628	<u>0.653</u>	0.652	**0.738**
$F1_{inkg}$	0.840	0.840	0.805	<u>0.844</u>	**0.847**
$F1_{ookg}$	0.299	0.303	0.180	<u>0.303</u>	**0.403**
$F1_{combined}$	0.570	0.572	0.493	<u>0.573</u>	**0.625**

(b) Wikievents-test

Table 6

Clustering performance (Thresholds determined on validation set). Best in bold, second best underlined.

[10] in-KG entities are considered here as well as they can also occur in the out-of-KG entity detection. We also conducted the experiments while assuming a perfect out-of-KG detection result and achieved similar results.

As can be seen in Table 6a for AIDA-CoNLL, the clustering methods often outperform the sequential method on the clustering metrics for the out-of-KG entities. For the in-KG entities the sequential method comes close but does not outperform the best performing clustering methods. In regard to the F-measure, the in-KG entity linking performance of the sequential method outperforms all others. For the out-of-KG entity linking, the best performing model is the Edin clustering method. The sequential method has the second-best performance here. The SOTA NASTyLinker is outperformed by both. The additional KG information does help the entity linking process of the in-KG entities while it does not help the clustering of out-of-KG entities.

For the Wikievents dataset, the sequential method is outperformed by the clustering methods on both, the clustering metrics and the F-measures (see Table 6b). We suspect that this is the case due fewer entities co-occurring in the Wikievents dataset. This delivers less context in the form of already linked entities when using the TransE encodings. This leads to fewer benefits for the in-KG entity linking as well as more noise for clustering the out-of-KG entity mentions. The best performing model is the NASTyLinker. Additionally, we examined the wrong clusters when using the mention encoder and identified that most often COVID-related entities are clustered together. As the model was trained on data before COVID-19 had an impact it struggles to differentiate entities related to this as they are syntactically very close. The clustering performance for all methods is reduced in comparison to the AIDA-CoNLL dataset. **Evidently, the clustering of out-of-KG entities is more challenging on the Wikievents dataset.** This introduces a new research avenue.

In contrast to the positive impact of knowledge graph (KG) information on in-KG entity linking, incorporating it into the clustering of out-of-KG entities has a negligible effect. This is the case for both datasets.

4. Related Work

Entity linking methods can be categorized into two types. First, discriminative methods that are based on the bi-encoder / cross-encoder pairing [26–28]. Both encoders are commonly BERT-like models. The bi-encoder encodes the description of each entity and matches it to the text by using an approximate nearest neighbor search. This is important as the next step, the cross-encoding, is expensive. Here, those neighbors are reranked by applying a cross-encoder to the concatenation of both, the input text and the entity description. The highest-ranked entity is then the final linked one. Another type of entity linker is based on generative models [29, 30]. Here, instead of using some external description of an entity, the whole model memorizes the knowledge graph (KG) during training. The linked entity is then directly generated by the model. Such methods skip the problem of mining negatives which are crucial for a good performance of bi-encoder-based methods. Our method follows the discriminative paradigm.

State-of-the-Art entity linking methods often do not consider out-of-KG entities. This is most apparent when examining the most common EL evaluation dataset. AIDA-CoNLL [5] does contain out-of-KG entities but even in the original paper, they were ignored during evaluation. As a consequence of that, most subsequent methods ignore them as well. However, there exist certain entity linking subtasks where out-of-KG entities are of importance.

NIL-Clustering, introduced at TAC-2011 [2], focuses on linking a whole batch of documents at once. Notably, the entity mentions occurring also contain out-of-KG entities, here called NIL entities. The goal is to not only link the in-KG entities but also the out-of-KG entities. In essence, this leads to a clustering of all documents. Naturally, most methods employ clustering techniques [7–15]. These methods focus on Wikipedia while we focus on Wikidata as a proper KG.

Hoffart et al. [5], introduced the task of emerging entity discovery in 2014. Here, the goal is to link entities, occurring in incoming texts, while also being able to discover emerging entities. These are entities that are out-of-KG and recently created. For example, news articles might contain emerging entities as certain events occurring are entities but might not yet be added to the KG. To solve this problem of discovering emerging entities, auxiliary information is considered. The auxiliary information used is often retrieved from external documents such as crawled webpages [4–6]. In our work, we avoid external documents and solely focus on detecting out-of-KG entities by checking the candidates. Hoffart et al. published the AIDA-EE dataset but it is not freely available.

Since 2022, several new works on the subject matter were published. The EDIN benchmark [31] focuses on the adaptation of an entity linking model to support unknown entities. Here, the training is split into two parts, a regular entity linking training and an adaptation phase where unknown entities are encountered. The EDIN benchmark focuses on Wikipedia and introduces an adaptation phase. In contrast to that, we do not expect any adaptation data to be available.

TempEL [32] is a benchmark focusing on the linking of evolving and emerging entities. However, the assumption here is that long well-formulated descriptions of emerging entities do exist. It is a more specific case of the zero-shot setting. No entities are encountered, for which no description exists. This differs from this work as we do not assume that any additional information is available for the out-of-KG entities.

Agarwal et al. [33] consider the detection of out-of-KG entities but they again assume long well-formulated descriptions of the in-KG entities.

The NILK dataset [34] is a dataset similar to ours but it is not accompanied by an entity linking model. The dataset was excluded from our evaluation due to its mention-focused construction. Specifically, each instance in the dataset consists of a single mention and its corresponding context. The dataset's train/dev/test split was created by partitioning the set of identified mentions. As a result, mentions of same sentences may appear in multiple splits, which poses a challenge for our approach since it depends on contextual information from other mentions within the same sentence. Consequently, the NILK dataset is not suitable for evaluating our method.

The NASTyLinker by Heist et al. [22] introduced a new clustering method and incorporated the scores of a cross-encoder in the clustering process. It was evaluated on the task of NIL-Clustering and and focused on the NILK dataset with Wikipedia descriptions.

The work by Pozzi et al. [35] focuses on Wikipedia and examines the detection and clustering of out-of-KG as well. They modify an existing dataset to include out-of-KG entities as well. We additionally offer a dataset with true out-of-KG entities and provide its KG. Also, we do not assume to have knowledge about out-of-KG entities during training. In our case, they are only encountered during the evaluation. Finally, we put a greater focus on the out-of-KG detection mechanism.

The method by Dong et al. [36] relies on the availability of out-of-KG entities during training time. This differs from our method as we do not assume that this is the case.

The clustering of out-of-KG entities is related to cross-document coreference resolution [9, 37–39]. However, we limit the clustering only to out-of-KG detected entity mentions and include information available in KG to support the clustering.

There exist several other methods [40, 41] which include KG embeddings into the EL process. However, we are the first to examine their impact on the clustering of out-of-KG entities.

5. Future Work

In the future, we want to improve upon the results by using a more sophisticated training regime. For example, hard-negative sampling or arborescence sampling [24] could be employed which could improve the performance. Also, to improve the performance of the out-of-KG detection, it would make sense to introduce more methods from novelty detection [42] or open-set recognition [43] which specifically focus on the detection of instances of classes not encountered during training. Furthermore, the candidate generation can be improved by not relying on TF-IDF but embedding all entity definitions in a latent space and retrieving candidates by a k-nearest neighbor search.

Most importantly, we will look into creating another suitable dataset containing out-of-KG entities. While the Wikievents dataset includes them in a natural way, it is limited to short texts with only a small amount of context information. Alternatively, models which can cope with small amounts of input data need to be developed. Furthermore, Wikievents and AIDA-CoNLL focus on the news domain. Cleaner and less ambiguous texts are more common here. This can be seen by the good performance of the lexical similarity measures when clustering. However, out-of-KG entities also exist in other contexts like historical documents which tend to be noisier and thus challenging [44].

6. Conclusion

We developed the Wikievents entity linking dataset, which contains out-of-KG entities, and demonstrated that it presents a significant challenge for clustering such entities. Moreover, we designed and assessed a sequential method that initially links entities or identifies them as out-of-KG, and subsequently clusters all out-of-KG entities for disambiguation.

Our findings reveal that our sequential method's ability to consider the entire sentence allows it to perform on par with or even surpass methods that cluster all mentions jointly in some cases. We also demonstrated the feasibility of learning out-of-KG entity detection during training and highlighted the importance of incorporating them to an appropriate degree.

Thus, we were able to show, that our approach which relies exclusively on information within a knowledge graph, eliminating dependency on lengthy textual descriptions or external data is an alternative for Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering without a parallel text corpus.

7. Limitations

One limitation of this study is that only one type of graph embedding was considered, and different embeddings like DistMult, ComplEx, etc., might result in different performance outcomes. Additionally, due to hardware limitations, we couldn't perform pre-training of the Entity Linker (EL) on a large corpus like Wikipedia, which is common in many entity linking methods today. Finally, it's important to note that we assumed entity mentions in the text are already detected, but detecting entirely new entity mentions is a challenge and crucial for real-world applications.

Supplemental Material Statement: Source code and datasets are available at https://github.com/cedricmoeller/ELwithOOKGDetectionClustering.

References

[1] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A survey on complex knowledge base question answering: Methods, challenges and solutions. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4483–4491. ijcai.org, 2021.

[2] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1148–1158. The Association for Computer Linguistics, 2011.

[3] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514, 2022.

[4] Zhaohui Wu, Yang Song, and C. Lee Giles. Exploring multiple feature spaces for novel entity discovery. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3073–3079. AAAI Press, 2016.

[5] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 385–396. ACM, 2014.

[6] Lei Zhang, Tianxing Wu, Liang Xu, Meng Wang, Guilin Qi, and Harald Sack. Emerging entity discovery using web sources. In Xiaoyan Zhu, Bing Qin, Xiaodan Zhu, Ming Liu, and Longhua Qian, editors, *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding - 4th China Conference, CCKS 2019, Hangzhou, China, August 24-27, 2019, Revised Selected Papers*, volume 1134 of *Communications in Computer and Information Science*, pages 175–184. Springer, 2019.

[7] Kevin Blissett and Heng Ji. Cross-lingual NIL entity clustering for low-resource languages. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 20–25, Minneapolis, USA, June 2019. Association for Computational Linguistics.

[8] Angela Fahrni, Benjamin Heinzerling, Thierry Göckel, and Michael Strube. Hits' monolingual and cross-lingual entity linking system at TAC 2013. In *Proceedings of the Sixth Text Analysis Conference, TAC 2013, Gaithersburg, Maryland, USA, November 18-19, 2013*. NIST, 2013.

[9] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. Cross-lingual cross-document coreference with entity linking. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*. NIST, 2011.

[10] Huy M. Huynh, Trong T. Nguyen, and Tru Hoang Cao. Using coreference and surrounding contexts for entity linking. In *2013 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, RIVF 2013, Hanoi, Vietnam, November 10-13, 2013*, pages 1–5. IEEE, 2013.

[11] Kara Greenfield, Rajmonda S. Caceres, Michael Coury, Kelly Geyer, Youngjune Gwon, Jason Matterer, Alyssa C. Mensch, Cem Safak Sahin, and Olga Simek. A reverse approach to named entity extraction and linking in microposts. In Aba-Sah Dadzie, Daniel Preotiuc-Pietro, Danica Radovanovic, Amparo Elizabeth Cano Basave, and Katrin Weller, editors, *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016), Montréal, Canada, April 11, 2016*, volume 1691 of *CEUR Workshop Proceedings*, pages 67–69. CEUR-WS.org, 2016.

[12] Taylor Cassidy, Zheng Chen, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jiawei Han, Dan Roth, and Jing Zheng. CUNY-UIUC-SRI TAC-KBP2011 entity linking system description. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*. NIST, 2011.

[13] Suzanne Tamang, Zheng Chen, and Heng Ji. CUNY BLENDER TAC-KBP2012 entity linking system and slot filling validation system. In *Proceedings of the Fifth Text Analysis Conference, TAC 2012, Gaithersburg, Maryland, USA, November 5-6, 2012*. NIST, 2012.

[14] David Graus, Tom Kenter, Marc Bron, Edgar Meij, and Maarten de Rijke. Context-based entity linking - university of amsterdam at TAC 2012. In *Proceedings of the Fifth Text Analysis Conference, TAC 2012, Gaithersburg, Maryland, USA, November 5-6, 2012*. NIST, 2012.

[15] Sourav Dutta and Gerhard Weikum. C3EL: A joint model for cross-document co-reference resolution and entity linking. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 846–856. The Association for Computational Linguistics, 2015.

[16] Amit Singhal. Introducing the Knowledge Graph: Things, not strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/, May 2012. Accessed 2022-03-29.

[17] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1595–1604. Association for Computational Linguistics, 2018.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019.

[20] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795, 2013.

[21] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of the 2nd SysML Conference*, Palo Alto, CA, USA, 2019.

[22] Nicolas Heist and Heiko Paulheim. Nastylinker: Nil-aware scalable transformer -based entity linker. *CoRR*, abs/2303.04426, 2023.

[23] Nora Kassner, Fabio Petroni, Mikhail Plekhanov, Sebastian Riedel, and Nicola Cancedda. EDIN: an end-to-end benchmark and pipeline for unknown entity discovery and indexing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8659–8673. Association for Computational Linguistics, 2022.

[24] Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. Entity linking via explicit mention-mention coreference modeling. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4644–4658. Association for Computational Linguistics, 2022.

[25] Nafise Sadat Moosavi and Michael Strube. Which coreference evaluation metric do you trust? A proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[26] 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 2020, Online, November 16-20, 2020*, pages 6397–6407. Association for Computational Linguistics, 2020.

[27] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3449–3460. Association for Computational Linguistics, 2019.

[28] 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 *NAACL 2022*, 2022.

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

[30] Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. Multilingual autoregressive entity linking. *Trans. Assoc. Comput. Linguistics*, 10:274–290, 2022.

[31] Nora Kassner, Fabio Petroni, Mikhail Plekhanov, Sebastian Riedel, and Nicola Cancedda. EDIN: an end-to-end benchmark and pipeline for unknown entity discovery and indexing. *CoRR*, abs/2205.12570, 2022.

[32] Klim Zaporojets, L-A Kaffee, Thomas Demeester, Chris Develder, and I Augenstein. Tempel: Linking dynamically evolving and newly emerging entities. In *NeurIPS2022, the 36th Conference on Neural Information Processing Systems*, 2022.

[33] Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. Entity linking and discovery via arborescence-based supervised clustering. *CoRR*, abs/2109.01242, 2021.

[34] Anastasiia Iurshina, Jiaxin Pan, Rafika Boutalbi, and Steffen Staab. NILK: entity linking dataset targeting nil-linking cases. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 4069–4073. ACM, 2022.

[35] Riccardo Pozzi, Federico Moiraghi, Fausto Lodi, and Matteo Palmonari. Evaluation of incremental entity extraction with background knowledge and entity linking. *The 11th International Joint Conference on Knowledge Graphs, October 27-29, 2022, Hangzhou, China*, 2022.

[36] Hang Dong, Jiaoyan Chen, Yuan He, Yinan Liu, and Ian Horrocks. Reveal the unknown: Out-of-knowledge-base mention discovery with entity linking. *CoRR*, abs/2302.07189, 2023.

[37] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In Christian Boitet and Pete Whitelock, editors, *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference*, pages 79–85. Morgan Kaufmann Publishers / ACL, 1998.

[38] Sameer Singh, Amarnag Subramanya, Fernando C. N. Pereira, and Andrew McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 793–803. The Association for Computer Linguistics, 2011.

[39] Arie Cattan, Sophie Johnson, Daniel S. Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. Scico: Hierarchical cross-document coreference for scientific concepts. In Danqi Chen, Jonathan Berant, Andrew McCallum, and Sameer Singh, editors, *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021*, 2021.

[40] Özge Sevgili, Alexander Panchenko, and Chris Biemann. Improving neural entity disambiguation with graph embeddings. In Fernando Alva-Manchego, Eunsol Choi, and Daniel Khashabi, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*, pages 315–322. Association for Computational Linguistics, 2019.

[41] Petar Ristoski, Zhizhong Lin, and Qunzhi Zhou. KG-ZESHEL: knowledge graph-enhanced zero-shot entity linking. In Anna Lisa Gentile and Rafael Gonçalves, editors, *K-CAP '21: Knowledge Capture Conference, Virtual Event, USA, December 2-3, 2021*, pages 49–56. ACM, 2021.

[42] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *CoRR*, abs/2110.11334, 2021.

[43] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3614–3631, 2021.

[44] Sina Menzel, Hannes Schnaitter, Josefine Zinck, Vivien Petras, Clemens Neudecker, Kai Labusch, Elena Leitner, and Georg Rehm. Named Entity Linking mit Wikidata und GND – Das Potenzial handkuratierter und strukturierter Datenquellen für die semantische Anreicherung von Volltexten. In Michael Franke-Maier, Anna Kasprzik, Andreas Ledl, and Hans Schürmann, editors, *Qualität in der Inhaltserschließung*, pages 229–258. De Gruyter, September 2021.