





NASTyLinker: NIL-Aware Scalable Transformer-Based Entity Linker

Nicolas Heist^(✉)  and Heiko Paulheim 

Data and Web Science Group, University of Mannheim, Mannheim, Germany
{nico,heiko}@informatik.uni-mannheim.de

Abstract. Entity Linking (EL) is the task of detecting mentions of entities in text and disambiguating them to a reference knowledge base. Most prevalent EL approaches assume that the reference knowledge base is complete. In practice, however, it is necessary to deal with the case of linking to an entity that is not contained in the knowledge base (NIL entity). Recent works have shown that, instead of focusing only on affinities between mentions and entities, considering inter-mention affinities can be used to represent NIL entities by producing clusters of mentions. At the same time, inter-mention affinities can help to substantially improve linking performance for known entities. With NASTyLinker, we introduce an EL approach that is aware of NIL entities and produces corresponding mention clusters while maintaining high linking performance for known entities. The approach clusters mentions and entities based on dense representations from Transformers and resolves conflicts (if more than one entity is assigned to a cluster) by computing transitive mention-entity affinities. We show the effectiveness and scalability of NASTyLinker on NILK, a dataset that is explicitly constructed to evaluate EL with respect to NIL entities. Further, we apply the presented approach to an actual EL task, namely to knowledge graph population by linking entities in Wikipedia listings, and provide an analysis of the outcome.

Keywords: NIL-Aware Entity Linking · Entity Discovery · Knowledge Graph Population · NILK · Wikipedia Listings · CaLiGraph

1 Introduction

1.1 Motivation and Problem

Entity Linking (EL), i.e., the task of detecting mentions of entities in text and disambiguating them to a reference knowledge base (KB), is crucial for many downstream tasks like question answering [9, 40], or KB population and completion [16, 19, 31]. One main challenge of EL is the inherent ambiguity of mentioned entities in the text. Figure 1 shows four homonymous mentions of distinct entities with the name *James Lake* (a lake in Canada, a lake in the US, a musician, and a fictional character). Correctly linking the mentions in Fig. 1a and 1b is especially challenging as both point to lakes that are geographically close.

In a typical EL setting, we assume that the training data contains mentions of all entities to be linked against. This assumption is dropped in Zero-Shot EL [26], where a linking decision is made on the basis of entity information in the reference KB (e.g. textual descriptions, types, relations). In this setting, a seminal approach has been introduced with BLINK [41]. Its core idea is to create dense representations of mentions and entities with a Transformer model [10] in a bi-encoder setting, retrieve mention-entity candidates through Nearest Neighbor Search, and rerank candidates with a cross-encoder.

In a practical setting, we additionally encounter the problem of mentions without a corresponding entity in the reference KB (which we refer to as NIL mentions and NIL entities, respectively). In fact, the mention in Fig. 1a is the only one with a counterpart in the reference KB (i.e., Wikipedia). For the other mentions, a correct prediction based on Wikipedia entities is impossible. Instead, NIL-aware approaches could either (1) create an (intermediate) entity representation for the NIL entity to link, or (2) produce clusters of NIL mentions with all mentions in a cluster referring to the same entity.

While this problem has been largely ignored by EL approaches for quite some time, recent works demonstrate that reasonable predictions for NIL mentions can be made by clustering mentions on the basis of inter-mention affinities [1, 24]. Both compute inter-mention and mention-entity affinities using a bi-encoder architecture on the basis of BLINK [41]. EDIN [24] is an approach of category (1) that uses a dedicated adaptation dataset to create representations for NIL entities in an unsupervised fashion. Hence, the approach can only link to a NIL entity if there is at least one mention of it in the adaptation dataset. For some EL tasks, especially as a prerequisite for KB population, creating an adaptation dataset with good coverage is not trivial because an optimal adaptation dataset has to contain mentions of all NIL entities. Agarwal et al. [1] present an approach of category (2) that creates clusters of mentions and entities in a bottom-up fashion by iteratively merging the two most similar clusters, always under the constraint that a cluster must contain at most one entity.

1.2 Approach and Contributions

With NASTyLinker, we present an EL approach that is NIL-aware in the sense of category (2) and hence avoiding the need for an adaptation dataset. Similar to Agarwal et al. [1], it produces clusters of mentions and entities on the basis of inter-mention and mention-entity affinities from a bi-encoder. NASTyLinker relies on a top-down clustering approach that – in case of a conflict – assigns mentions to the entity with the highest transitive affinity. Contrary to Agarwal et al., who discard cross-encoders completely due to the quadratic growth in complexity when evaluating inter-mention affinities, our experiments show that applying a cross-encoder only for the refinement of mention-entity affinities can result in a considerable increase of linking performance at a reasonable computational cost. Our evaluation on the NILK dataset [21], a dataset especially suited for the evaluation of NIL-aware approaches, shows that NASTyLinker manages to make competitive predictions for NIL entities while even slightly improving

Name	Township(s)	Coordinates	NTS map	Status	CGNDB id
Jackpine Lake	Banting, Chambers	47°8′44″N 79°56′3″W﻿•﻿47°8′44″N 79°56′3″W﻿•﻿47°10′41″N 79°44′26″W	031M04	Official	FBRBM:2
James Lake	Best	47°10′41″N 79°44′26″W﻿•﻿47°10′41″N 79°44′26″W﻿•﻿47°10′41″N 79°44′26″W	031M04	Official	FBRHD:2
Jameson Lake	Banting	47°9′22″N 79°59′37″W﻿•﻿47°9′22″N 79°59′37″W﻿•﻿47°9′22″N 79°59′37″W	031M04	Official	FBRHY:2
Jessie Lake	Strathcona	47°2′28″N 79°48′14″W﻿•﻿47°2′28″N 79°48′14″W﻿•﻿47°2′28″N 79°48′14″W	031M04	Official	FBRJY:2
Jumping Caribou Lake	Las, Olive	46°52′57″N 79°46′32″W﻿•﻿46°52′57″N 79°46′32″W﻿•﻿46°52′57″N 79°46′32″W	031J13	Official	FBSOZ:2
Jumpingcat Lake	Bellast, Joan	47°1′48″N 80°9′59″W﻿•﻿47°1′48″N 80°9′59″W﻿•﻿47°1′48″N 80°9′59″W	041P01	Official	FBSPA:2

(a) A lake in Ontario, CA.
Lakes of Temagami

Members [edit]

- Lionel Williams – vocals, various instruments (2007–present)

Associated musicians [edit]

- Bryan Lee – drums (2007–2010)
- Calin Stephensen – bass (2007–2008)
- James Lake** – drums/synth (2011–2017)
- Ian Gibbs – various instruments (2011–2018)

(c) A musician in the band Vinyl Williams.
Vinyl Williams

- **Elbow Lake**, 46°21′53″N 113°01′31″W﻿•﻿46°21′53″N 113°01′31″W﻿•﻿46°21′53″N 113°01′31″W, el. 7,746 feet (2,361 m)^[14]
- **Evans Lake**, 47°00′15″N 113°04′19″W﻿•﻿47°00′15″N 113°04′19″W﻿•﻿47°00′15″N 113°04′19″W, el. 4,193 feet (1,278 m)^[15]
- **Hagan Pond**, 46°28′46″N 112°52′55″W﻿•﻿46°28′46″N 112°52′55″W﻿•﻿46°28′46″N 112°52′55″W, el. 5,000 feet (1,500 m)^[16]
- **James Lake**, 47°04′32″N 113°12′43″W﻿•﻿47°04′32″N 113°12′43″W﻿•﻿47°04′32″N 113°12′43″W, el. 4,137 feet (1,261 m)^[17]
- **Jones Lake**, 47°02′35″N 113°08′35″W﻿•﻿47°02′35″N 113°08′35″W﻿•﻿47°02′35″N 113°08′35″W, el. 4,088 feet (1,246 m)^[18]
- **Kleinschmidt Lake**, 46°58′33″N 113°02′35″W﻿•﻿46°58′33″N 113°02′35″W﻿•﻿46°58′33″N 113°02′35″W, el. 4,186 feet (1,276 m)^[19]

(b) A lake in Montana, US.
List of lakes of Powell County, Montana

Character	Actor/Actress	Duration
Joe Lacerra	Stephen Liska	1998–2005
Cindy Lake	DeAnna Robbins	1982–83
James Lake	Glenn Corbett	1983
Mary Margaret Lake	Fawne Harriman	1983
Sammy Lake	Danny McCoy Jr.	1978
Hilary Lancaster	Kelly Garrison	1991–93
Dr. Joshua Landers	Heath Kizzier	1996–98

(d) A character in a soap opera.
List of The Young and Restless characters

Fig. 1. Listings in Wikipedia containing the mention *James Lake*. All of the mentions refer to distinct entities. A dedicated Wikipedia page exists only for the entity of the mention in (a).

prediction performance for known entities. The approach is designed in a modular way to make existing EL models NIL-aware by post-processing the computed inter-mention and mention-entity affinities. By applying NASTyLinker to a knowledge graph population task, we demonstrate its ability to reliably link to known entities (up to 87% accuracy) and identify NIL entities (up to 90% accuracy).

To summarize, the contributions of this paper are as follows:

- We introduce the NASTyLinker approach, serving as an extension to existing EL approaches by using a top-down clustering mechanism to consistently link mentions to known entities and produce clusters for NIL mentions (Sect. 4).
- In our experiments, we demonstrate the competitive linking performance and scalability of the presented approach through an evaluation on the NILK dataset (Sect. 5.4).
- We use NASTyLinker for KB population by linking entities in Wikipedia listings. We report on the linking statistics and provide a qualitative analysis of the results (Sect. 5.5).

The produced code is part of the CaLiGraph extraction framework and publicly available on GitHub.¹

2 Related Work

Entity Linking. Entity Linking has been studied extensively in the last two decades [33, 37]. Initially, approaches relied on word and entity frequencies, alias

¹ <https://github.com/nheist/CaLiGraph>.

tables, or neural networks for their linking decisions [7, 13, 28]. The introduction of pre-trained transformer models [10] made it possible to create representations of mentions and entities from text without relying on other intermediate representations. Gillick et al. [14] show how to learn dense representations for mentions and entities, Logeswaran et al. [26] extend this by introducing the zero-shot EL task and demonstrating that reasonable entity embeddings can be derived solely from entity descriptions. Wu et al. [41] introduce BLINK, the prevalent bi-encoder and cross-encoder paradigm for zero-shot EL. Various improvements for zero-shot EL have been proposed based on this paradigm. KG-ZESHEL [35] adds auxiliary entity information from knowledge graph embeddings into the linking process; Partalidou et al. [30] propose alternative pooling functions for the bi-encoder to increase the accuracy of the candidate generation step.

Cross-Document Coreference Resolution. NIL-Aware EL is closely related to Cross-Document Coreference Resolution (CDC), the task of identifying coreferent entity mentions in documents without explicitly linking them to entities in a KB [5]. Dutta and Weikum [11] explicitly tackle CDC in combination with EL by applying clustering to bag-of-words representations of entity mentions. More recently, Logan IV et al. [25] evaluate greedy nearest-neighbour and hierarchical clustering strategies for CDC, however, without explicitly evaluating them with respect to EL.

Entity Discovery and NIL-Aware EL. The majority of EL approaches may identify NIL mentions (for instance, through a binary classifier or a ranking that explicitly includes *NIL*), but does not process them in any way [37, 38]. In 2011, the TAC-KBP challenge [22] introduced a task that includes NIL clustering; in the NEEL challenge [36] that is based on microposts, NIL clustering was part of the task as well. Approaches that tackled these tasks typically applied clustering based on similarity measures over the entity mentions in the text [6, 12, 15, 29, 32]. More recently, Angell et al. [3] train two separate bi-encoders and cross-encoders to compute inter-mention and mention-entity affinities. Subsequently, they apply a bottom-up clustering for refined linking predictions within single biomedical documents. Agarwal et al. [1] extend the approach to cross-document linking through a clustering based on minimum spanning trees over all mentions in the corpus. Clusters are formed by successively adding edges to a graph as long as the constraint that a cluster can contain at most one entity is not violated. They omit the cross-encoder and employ a custom training procedure for the bi-encoder instead. They explicitly evaluate their approach w.r.t. NIL entity discovery by removing a part of the entities in the training set from zero-shot EL benchmark datasets. In our approach, we employ a similar method for computing affinities but employ a top-down clustering approach that aims to better identify clusters of NIL mentions. The EDIN pipeline [24] also applies clustering w.r.t. inter-mention and mention-entity affinities, but only to identify NIL mention clusters on a dedicated adaptation dataset. Subsequently, the entity index is enhanced with pooled representations of these clusters to make a prediction of NIL entities possible. In their clustering phase, they first produce groups of mentions and then identify NIL mention clusters by checking whether less than

70% of the mentions are referring to the same entity. As we aim to apply NIL-aware EL for KB population, relying on an adaptation dataset is not possible. Still, we include the clustering method of the EDIN pipeline in our experiments to compare how well the approaches detect NIL mention clusters.

3 Task Formulation

A document corpus \mathcal{D} contains a set of textual entity mentions \mathcal{M} . Each of the mentions $m \in \mathcal{M}$ refers to an entity e in the set of all entities \mathcal{E} . Given a knowledge base K with known entities \mathcal{E}^k , the task in standard EL is to assign an entity $\hat{e} \in \mathcal{E}^k$ to every mention in \mathcal{M} . In this setting, we assume that $\mathcal{E} = \mathcal{E}^k$, i.e., all entities are contained in K .

In NIL-aware EL, we drop the assumption that every mention links to an entity contained in K . Instead there is a set of NIL entities \mathcal{E}^n with $\mathcal{E}^k \cup \mathcal{E}^n = \mathcal{E}$ and $\mathcal{E}^k \cap \mathcal{E}^n = \emptyset$. For mentions \mathcal{M}^k that refer to entities in K , the task is still to predict an entity $\hat{e} \in \mathcal{E}^k$. For mentions \mathcal{M}^n that refer to entities not contained in K , the task is to predict a cluster identifier $c \in \mathcal{C}$ so that the clustering \mathcal{C} resembles the distribution of mentions in \mathcal{M}^n to entities in \mathcal{E}^n as closely as possible. We assume that we are additionally operating in a zero-shot setting, i.e., the training portion \mathcal{D}_{train} of the document corpus may not contain mentions for all entities in \mathcal{E} .

Note that, similar to related works [1, 26], we assume that the textual entity mentions are already given. Further, we only investigate the relevant steps for KB population, i.e., detection and disambiguation of NIL entities. While we discard the indexing aspect, an EL model which includes the entities in \mathcal{E}^n can still be created in a subsequent step by training a new model on the enhanced KB.

4 NASTyLinker: An Approach for NIL-Aware and Scalable Entity Linking

In this section, we describe our proposed approach for making NIL-aware EL predictions. Figure 2 depicts the three main phases of the NASTyLinker approach. In the *Linking Phase*, we first retrieve inter-mention and mention-entity affinities from an underlying EL model for the subsequent clustering. We define constraints for such a model and describe the one used in our experiments in Sect. 4.1. During the *Clustering Phase*, clusters of mentions and entity candidates are created using greedy nearest-neighbour clustering (Sect. 4.2). Finally, we retrieve entity candidates for every cluster. In the *Conflict Resolution Phase*, clusters are split based on transitive mention-entity affinities to ensure that a cluster contains at most one known entity (Sect. 4.3).

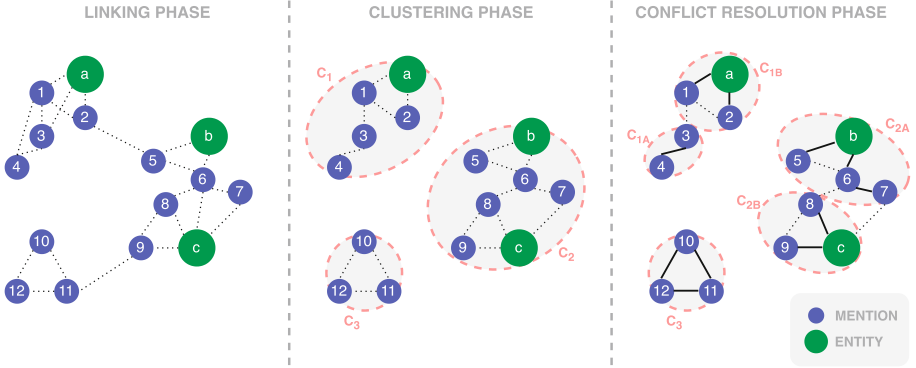


Fig. 2. Main phases of the NASTyLinker approach. Dotted lines show top- k affinity scores, solid lines indicate the highest transitive affinity scores.

4.1 Entity Linking Model

In the *Linking Phase* we compute the k most similar mentions and entities for every mention in \mathcal{M} (dotted lines in Fig. 2). The underlying EL model has to provide a function ϕ with $\phi(m, e) \in [0; 1]$ for the similarity between mention m and entity e as well as $\phi(m, m') \in [0; 1]$ for the similarity between mentions m and m' . In addition to that, it must be possible to retrieve the top k mention and entity candidates for a given mention in an efficient manner.

For our experiments with NASTyLinker, we choose the BLINK architecture [41] as the underlying EL model as it provides the foundation for many state-of-the-art EL models. Furthermore, as the bi-encoder creates embeddings for mentions and entities alike, methods for an approximate nearest neighbour search like FAISS [23] can be used to retrieve linking candidates efficiently. As the application of the cross-encoder is the most time-consuming part of this model, we explore in our experiments the trade-off between linking performance and runtime when reranking only inter-mention affinities, only mention-entity affinities, or both.

Partalidou et al. [30] propose several layouts for structuring the input sequence of mentions and entities for the Transformer model. We achieved the best results with the mention layout

[CLS] [<type>] <mention label> [CTX] <mention context> [SEP]

and the entity layout

[CLS] [<type>] <entity title> [CTX] <entity description> [SEP]

where [CTX] is a special delimiter token and [<type>] is a placeholder for a special token of the mention type (POS-tag) or entity type (top-level type in the KB). For optimization, we stick to Wu et al. [41] and use in-batch (hard) negatives for the bi-encoder, and bi-encoder-generated negatives for the cross-encoder.

4.2 Cluster Initialization

To produce an initial mention clustering, we follow Logan IV et al. [25] and use a greedy nearest-neighbour clustering. Given the mention affinity threshold τ_m , the mentions \mathcal{M} are grouped into clusters \mathcal{C} so that two mentions $m, m' \in \mathcal{M}$ belong to the same cluster if $\phi(m, m') > \tau_m$.

Further, we assign entity candidates to the clusters using a threshold for entity affinity τ_e . For a cluster $C \in \mathcal{C}$ with mentions M_C , we select the known entities with the highest affinity to each cluster mention:

$$E_C^k = \bigcup_{m \in M_C} \{ \underset{e \in \mathcal{E}^k}{\operatorname{argmax}} \phi(m, e) : \phi(m, e) > \tau_e \}. \quad (1)$$

In Fig. 2, the dotted lines represent affinities greater than the thresholds τ_m and τ_e , respectively. Cluster C_1 contains four loosely connected mentions with m_1 and m_2 directly connected to the entity candidate e_a . Either all four mentions refer to e_a as they are transitively connected, or some mentions refer to an entity in \mathcal{E}^n (e.g., a situation like in Fig. 1a and 1b). Cluster C_2 contains several mentions with two known entity candidates e_b and e_c , making a trivial assignment of mentions to entities impossible. Finally, cluster C_3 contains three connected mentions without any assigned entity candidates, most likely representing a NIL entity. Conflicts like the ones occurring in the former two clusters are resolved in the subsequent resolution phase.

4.3 Cluster Conflict Resolution

The objectives of the *Conflict Resolution Phase* are twofold: For every cluster $C \in \mathcal{C}$ we (1) find sub-clusters with $|E_C^k| = 1$ (c.f. C_{1B} , C_{2A} , and C_{2B} in Fig. 2), and (2) identify mentions in M_C that do not refer to any entity in E_C^k . For these, we create one or more sub-clusters representing the NIL entities E_C^n of C (c.f. C_{1A} and C_3 in Fig. 2).

For conflict resolution, we view a cluster $C \in \mathcal{C}$ as a graph G_C with $M_C \cup E_C^k$ as nodes, and affinities above threshold as edges. To ensure objective (1), we assign every mention in a cluster to the candidate entity with the highest transitive, defined as follows:

$$\phi^*(m, e) = \max_{m \sim e \in G_C} \prod_{u, v}^{m \sim e} \phi(u, v) \quad (2)$$

with $m \sim e$ denoting a path from a mention m to an entity e in G_C and (u, v) a single edge. The rationale for this metric is to favour strong contextual similarity between mentions over the mediocre similarity between a mention and an entity. As the entity context is coming from a different data corpus (i.e., information from a KB) than the mention context, it is more likely to happen that the contexts for a mention and its linked entity are dissimilar than the contexts of two mentions linking to the same entity.

Example 1. With affinities $\phi(m_6, e_b) = 0.9$, $\phi(m_6, m_7) = 0.9$, $\phi(m_7, e_c) = 0.8$, and paths $m_7 - m_6 - e_b$, $m_7 - e_c$ from Fig. 2, we find that $\phi^*(m_7, e_b) = 0.81 >$

$\phi^*(m_7, e_c) = 0.8$, resulting in the assignment of m_7 to the cluster of e_b in spite of e_c being the most likely entity for m_7 w.r.t. ϕ .

To ensure objective (2), we introduce a threshold τ_a as a lower limit for the transitive affinity between a mention and an entity. We label mentions as NIL mentions if they do not have a transitive affinity higher than the threshold to any entity in E_c^k :

$$M_c^n = \{m \in M_c \mid \nexists e \in E_c^k : \phi^*(m, e) > \tau_a\} \quad (3)$$

From M_c^n we produce one or more mention clusters similar to the initialization step in Sect. 4.2.

Example 2. With $\tau_a = 0.75$, affinities $\phi(m_1, e_a) = 0.9$, $\phi(m_1, m_3) = 0.8$, $\phi(m_3, m_4) = 0.9$, and path $m_4-m_3-m_1-e_a$ from Fig. 2, we find that $\phi^*(m_3, e_a) = 0.72 < \tau_a$ and $\phi^*(m_4, e_a) = 0.648 < \tau_a$. m_3 and m_4 are labelled as NIL mentions and form - due to their direct connection - the single cluster C_{1B} .

The function ϕ^* can be computed efficiently on a graph using Dijkstra’s algorithm with $-\log\phi$ as a function for edge weights. Edges are only inserted in the graph for $\phi > \tau_a$, avoiding undefined edge weights in the case of $\phi = 0$.

5 Experiments

We first describe the datasets and experimental setup used for the evaluation of NASTyLinker. Then, we compare the performance of our approach with related NIL-aware clustering approaches on the NILK dataset [21] and analyze its potential to scale. Finally, we report on the application of NASTyLinker for KB population by linking entities in Wikipedia listings.

5.1 Datasets

NILK. NILK is a dataset that is explicitly created to evaluate EL both for known and NIL entities. It uses Wikipedia as a text corpus and Wikidata [39] as reference KB. All entities contained in Wikidata up to 2017 are labelled as known entities and entities added to Wikidata between 2017 and 2021 are labelled as NIL entities. Mention and entity counts of NILK are displayed in Table 1. About 1% of mentions in NILK are NIL mentions, and about 6% of entities are NIL entities. NIL entities are probably slightly biased towards more popular entities, as the fact that they are present in Wikidata hints at a certain popularity, which may be higher than the popularity of an average NIL entity. Hence, the average number of mentions per NIL entity is quite high in this dataset: half of the entities are mentioned more than once, and more than 15% are even mentioned more than 5 times. Mention boundaries are already given and the authors define partitions for training, validation, and test, which are split in a zero-shot manner w.r.t. NIL entities. As mention context, the authors provide 500 characters before and after the actual mention occurrence in a Wikipedia page. As entity descriptions, we use Wikipedia abstracts.²

² While there are entities in Wikidata which do not have a Wikipedia page, this case does not occur in NILK by construction.

Table 1. Mention and entity occurrences in the partitions of the datasets. NIL mention counts for \mathcal{D}^L are estimated w.r.t. partial completeness assumption. Furthermore, the number of NIL entities \mathcal{E}^n in the listings dataset is not known. For \mathcal{D}_{pred}^L a single mention count is displayed as we cannot know whether a mention in \mathcal{M} links to an entity in \mathcal{E}^k or \mathcal{E}^n .

Dataset		$ \mathcal{M}^k $	$ \mathcal{M}^n $	$ \mathcal{E}^k $	$ \mathcal{E}^n $
NILK	Training (\mathcal{D}_{train}^N)	85,052,764	1,327,039	3,382,497	282,210
	Validation (\mathcal{D}_{val}^N)	10,525,107	162,948	422,812	35,276
	Test (\mathcal{D}_{test}^N)	10,451,126	162,497	422,815	35,279
LISTING	Training (\mathcal{D}_{train}^L)	11,690,019	6,760,273	3,073,238	?
	Validation (\mathcal{D}_{val}^L)	3,882,641	2,272,941	1,695,156	?
	Test (\mathcal{D}_{test}^L)	3,884,066	2,259,072	1,701,015	?
	Prediction (\mathcal{D}_{pred}^L)	18,658,271		?	?

Wikipedia Listings. The LISTING dataset was extracted in prior work [20] and consists of entity mentions in enumerations and tables of Wikipedia. Instead of all possible mentions, the focus is only on *subject entities*, which we define as *all entities in a listing appearing as instances to a common concept* [19]. So every item in a listing is assumed to have one main entity the item is about. For example, in Fig. 1d, the soap opera characters are considered entity mentions, while the actors are not.

As reference KB we use CaLiGraph [17, 18]. Mention and entity statistics are given in Table 1. We partition the data into train, validation, and test while making sure that listings on a page are all in the same split. Contrary to NILK, the LISTING dataset does not contain explicit labels for NIL entities. Instead, we define NIL entities using the partial completeness assumption (PCA). Given a listing with multiple mentions, we only incorporate them into training or test data if at least one mention is linked to a known entity. Then, by PCA, we assume that all mentions that can be linked are actually linked. All other mentions are assigned a new unique entity identifier. The prediction partition \mathcal{D}_{pred}^L , however, contains all mentions without a linked entity (i.e., they may link to a known or to a NIL entity). We use the text of the listing item as mention context for the dataset, and we use Wikipedia abstracts as entity descriptions.

We have considered further datasets that were used for evaluation of NIL-aware approaches for evaluation (e.g. from challenges like TAC-KBP or Micro-posts [8]), but discarded them due to their small size or not being free to use.

5.2 Metrics

Classification Metrics. We compute precision, recall, and F1-score as well as aggregations of the metrics on the instance level (micro average). As the evaluated approaches are not aware of the true NIL entities, they assign cluster identifiers to (what they assume to be) NIL mentions. To compute the clas-

sification metrics, it is necessary to map the cluster identifiers to actual NIL entities. Kassner et al. [24] allow the assignment of multiple cluster identifiers to the same NIL entity. This assumption would yield overly optimistic results. Instead, we only allow one-to-one mappings between cluster identifiers and NIL entities. Finding an optimal assignment for this scenario is equivalent to solving the linear sum assignment problem [2], for which efficient algorithms exist.

Clustering Metrics. Following related approaches [1, 24], we additionally provide normalized mutual information (NMI) and adjusted rand index (ARI) as clustering metrics for the comparison of the approaches to settings where no gold labels of NIL entities may be available.³ For known entities, however, the classification metrics will most likely be more expressive than the clustering metrics as the latter treat multiple clusters with the same known entity as their label still as separate clusters.

5.3 Evaluated Approaches

EL Model. We compute inter-mention and mention-entity affinities with a bi-encoder similar to BLINK [41]. As the reranking of bi-encoder results with a cross-encoder is costly, we evaluate different scenarios where the cross-encoder is omitted (*No Reranking*), applied to inter-mention affinities only (*Mention Reranking*), applied to mention-entity affinities only (*Entity Reranking*), or applied to both (*Full Reranking*). We use the Sentence-BERT implementation of the bi-encoder and cross-encoder [34] with *all-MiniLM-L12-v2* and *distilbert-base-cased* as respective base models. The base models are fine-tuned for at most one million steps on the training partitions of the datasets. Longer fine-tuning did not yield substantial improvements. We use a batch size of 256 for the bi-encoder and 128 for the cross-encoder. For efficient retrieval of candidates from the bi-encoder, we apply approximate nearest neighbour search with hnswnlib [27].

We use the plain bi-encoder and cross-encoder predictions of the EL model as baselines. Additionally, we evaluate a trivial *Exact Match* approach, where we link a mention to an entity if their textual representations match exactly.⁴ In case of multiple matches, the more popular entity (w.r.t. ingoing and outgoing links in the KB) is selected. Naturally, this approach cannot handle NIL entities.

Clustering Approaches. Apart from the NASTyLinker clustering as described in Sect. 4, we apply the clustering approaches of Kassner et al. [24] and Agarwal et al. [1] for comparison.⁵ The clustering approach of Kassner et al., which we

³ We implement further clustering metrics (B-Cubed+, CEAF, MUC) but do not list them as they are similar to or adaptations of the classification metrics.

⁴ We apply simple preprocessing like lower-casing and removal of special characters.

⁵ We tried to compare with the full approach of Agarwal et al. but they do not provide any code and our efforts to re-implement it did not yield improved results.

call *Majority Clustering*, applies a greedy clustering and assigns a known entity e to a cluster if at least 70% of mentions in the cluster have the highest affinity to e . Similarly to NASTyLinker, they use hyperparameters as thresholds for minimum inter-mention and mention-entity affinities.

The clustering approach of Agarwal et al., which we call *Bottom-Up Clustering*, starts with an empty graph and iteratively adds the edge with the highest affinity, as long as it does not violate the constraint of a cluster having at most one entity. They use a single hyperparameter as a threshold for the minimum affinity of an edge, be it inter-mention or mention-entity.

Hyperparameter Tuning. We select the hyperparameters of the EL model (k , *learning_rate*, *warmup_steps*) and the thresholds of all three clustering approaches w.r.t. micro F1-score on the validation partition of the datasets. For a fair comparison, we also test multiple values for the threshold for entity assignment of Majority Clustering, which in the original paper was fixed at 0.7.

Our experiments are run on a single machine having 96 CPUs, 1 TB of RAM, and an NVIDIA RTX A6000 GPU with 48 GB of RAM.

5.4 Entity Linking Performance

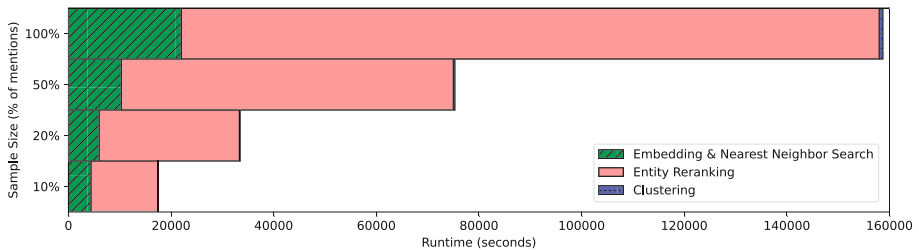
We tune hyperparameters by evaluating on \mathcal{D}_{val}^N . For the EL model, we use a k of 4, a learning rate of $2e-5$, and no warmup steps. For τ_m , a value between 0.8 and 0.9 works best for all approaches. For τ_e , the best values revolve around 0.9 for NASTyLinker and Bottom-Up Clustering, and around 0.8 for Majority Clustering. We use an affinity threshold τ_a of 0.75 for NASTyLinker and find that the 0.7 threshold of Majority Clustering produces the best results.

NILK Results. As shown in Table 2, we evaluate all clustering approaches on \mathcal{D}_{test}^N in different reranking scenarios. We find Exact Match already to be a strong baseline for known entities with an F1 of 79.5%, which the Cross-Encoder outperforms by approximately 10%. Even without reranking, the three clustering approaches are able to achieve an F1-score between 40% and 50% for NIL entities. Overall, Majority Clustering is best suited to identify NIL entities. It is the only one to substantially benefit from reranking, increasing the F1-score by 10% when applying entity reranking. Especially for linking known entities, applying only entity reranking is the most favourable scenario, leading even to slight improvements over the baseline approaches that focus only on known entities.

As the reranking of mentions tends to lead to a decrease in results while considerably increasing runtime, we omit mention reranking (and hence, full reranking) in experiments with Wikipedia listings. In the remaining scenarios, NASTyLinker finds the best balance between the linking of known entities and the identification of NIL entities w.r.t. F1-score and NMI.

Table 2. Results for the test partition \mathcal{D}_{test}^N of the NILK dataset.

Approach		Known			NIL			Micro		
		F1	NMI	ARI	F1	NMI	ARI	F1	NMI	ARI
No Clustering	Exact Match	79.5	—	—	0.0	—	—	78.1	—	—
	Bi-Encoder	80.8	—	—	0.0	—	—	79.1	—	—
	Cross-Encoder	89.0	—	—	0.0	—	—	87.1	—	—
Clustering & No Reranking	Bottom-Up	64.6	99.0	97.5	41.6	94.8	81.8	64.1	96.8	93.5
	Majority	59.4	99.3	98.0	49.8	92.7	82.7	59.2	96.6	94.6
	NASTyLinker	76.8	98.6	95.3	40.8	95.2	76.8	76.0	97.3	90.3
Clustering & Mention Reranking	Bottom-Up	65.7	97.1	98.9	41.5	94.6	10.0	65.1	96.0	66.0
	Majority	66.6	92.4	74.8	44.0	94.4	73.2	66.1	92.4	70.4
	NASTyLinker	74.2	99.0	96.6	39.2	85.6	16.5	73.5	95.5	81.6
Clustering & Entity Reranking	Bottom-Up	89.0	99.3	96.2	41.6	94.1	58.0	87.9	98.2	92.6
	Majority	74.2	99.1	99.3	54.1	89.3	92.5	73.7	96.6	94.6
	NASTyLinker	90.4	99.3	95.5	43.7	94.6	85.3	89.4	98.5	84.1
Clustering & Full Reranking	Bottom-Up	84.2	99.6	98.9	41.8	84.6	3.2	83.3	96.2	65.5
	Majority	80.3	95.1	95.9	51.7	90.0	39.2	79.6	93.9	70.4
	NASTyLinker	87.9	99.5	99.2	42.5	87.6	33.6	86.9	97.4	71.7


Fig. 3. Runtime of NASTyLinker components for predictions on samples of \mathcal{D}_{test}^N .

Runtime and Scalability. The fine-tuning of the bi-encoder and cross-encoder models took 2 h and 12 h, respectively. For prediction with a k of 4 on \mathcal{D}_{test}^N , the bi-encoder needed 6 h. Reranking entity affinities with the cross-encoder took 38 h. Clustering the results with any of the three approaches took an additional 8 to 12 min.

In Fig. 3 we give an overview of the runtime of NASTyLinker components, compared over various sample sizes of \mathcal{D}_{test}^N . Overall, we can see that the total runtime scales linearly. With a smaller sample size, the computation of embeddings and nearest neighbour search with the bi-encoder is responsible for a larger fraction of the total runtime. We find that this is due to the relatively large overhead of creating the index for the approximate nearest neighbour search. With increasing sample size, this factor is less important for the overall runtime. In general, entity reranking is responsible for most of the total runtime.

The runtime of the clustering itself is responsible for approximately 1% of total runtime and we do not expect it to increase substantially, as Dijkstra’s algorithm has log-linear complexity and the size of mention clusters can be controlled by the threshold τ_m . Hence, the runtime of NASTyLinker is expected to grow proportionally to the runtime of BLINK [41] for increasing sizes of datasets. If runtime is an important factor, one might consider skipping entity reranking as NASTyLinker still produces reasonable results when relying on bi-encoder affinities only.

5.5 Linking Entities in Wikipedia Listings

As the average mention context length in the LISTING dataset is lower than the one in NILK, fine-tuning the EL models took only a total of 8 h. We find that most of the hyperparameters chosen for NILK are a reasonable choice for this dataset as well. For entity reranking, however, the approaches produce better results when the thresholds τ_m and τ_a are slightly increased to 0.9 and 0.85.

Results on Test Partition. Linking results for \mathcal{D}_{test}^L are provided in Table 3. As we rely on PCA for the labelling of NIL mentions, we only know whether a mention is a NIL mention without knowing which NIL mentions refer to the same entity. Hence, we can only compute results for known entities and for overall predictions. For the latter, we simply assume that any prediction made for a NIL mention is incorrect. With this assumption, we are obviously not able to produce realistic performance estimates, but we are able to see the impact of being NIL-aware (and hence, make no prediction for NIL mentions) on the overall linking performance.

Due to their majority mechanism, Majority Clustering identifies known entities with very high precision, but at the cost of a reduced recall. The scores of Bottom-Up Clustering and NASTyLinker are comparable when considering known entities, but diverge w.r.t. the micro average. In the entity reranking scenario, NASTyLinker achieves the overall best micro F1-score with 86.7%. This, however, has to be taken with a grain of salt as we do not know how many of the heuristically labelled NIL mentions are actually referring to NIL entities and how many refer to known entities.

Knowledge Graph Population Statistics. The partition \mathcal{D}_{pred}^L of the LISTING dataset contains only mentions for which we don’t know whether they link to a known or to a NIL entity. To make predictions for these mentions, we run the NASTyLinker approach on the whole LISTING corpus, i.e. on a total of 38 million mentions, as we need representations of all known entities for the clustering step. These mentions were extracted from 2.9 million listings on 1.4 million Wikipedia pages. As reference KB, we use the knowledge graph CaLi-Graph which is based on Wikipedia and hence contains entities for all 5.8 million Wikipedia articles.

Table 3. Results for the test partition \mathcal{D}_{test}^L of the LISTING dataset. No results for *NIL* are given because the real NIL entities \mathcal{E}^n are not available for this dataset. For the micro average, we label every prediction made for a mention linked to an entity in \mathcal{E}^n as incorrect.

Approach		Known			Micro		
		P	R	F1	P	R	F1
No Clustering	Exact Match	91.4	73.5	81.5	81.1	73.5	77.1
	Bi-Encoder	88.6	88.6	88.6	62.6	88.6	73.4
	Cross-Encoder	93.7	93.8	93.8	66.2	93.8	77.6
Clustering & No Reranking	Bottom-Up	89.7	84.9	87.2	63.9	84.9	72.9
	Majority	95.2	67.9	79.2	78.1	67.9	72.6
	NASTyLinker	90.6	78.5	84.1	70.7	78.5	74.4
Clustering & Entity Reranking	Bottom-Up	94.2	90.8	92.5	75.3	90.8	82.3
	Majority	98.8	76.2	86.0	93.4	76.2	83.9
	NASTyLinker	97.0	87.0	91.8	88.5	87.0	87.7

The total runtime was 62h, with 14h for the bi-encoder, 47h for the cross-encoder, and 45 min for the clustering. We find 13.4 million mentions (i.e., 70%) to be NIL mentions which refer to 7.6 million NIL entities. The remaining 5.2 million mentions refer to 1.4 million entities that exist in CaLiGraph already. By integrating the discovered NIL entities into CaLiGraph, we would increase its entity count by 130%. Further, the discovered mentions for known entities can be used to enrich the representations of the entities in the knowledge graph through various knowledge graph completion methods [19].

Qualitative Analysis. To evaluate the actual linking performance on the set of unlabeled mentions \mathcal{D}_{pred}^L , we conducted a manual inspection of the results. We randomly picked 100 mentions and 100 clusters⁶ and identified, if incorrect, the type of error.⁷ The results of this evaluation are given in Table 4. Overall, we find the outcome to agree with the results of NASTyLinker on \mathcal{D}_{test}^L . Hence, the approach produces highly accurate results, which we observed even for difficult cases. For example, the approach correctly created NIL entity clusters for the mention *North Course* referring to a racing horse (in pages *Appleton Stakes* and *Oceanport Stakes*), a golf course in Ontario, CA (in page *Tournament Players Club*), and a golf course in Florida, US (in page *Pete Dye*).

While the linking performance is quite consistent for mentions, the correctness of clusters for known entities is significantly lower than for NIL entities.⁸

⁶ The sampling of clusters was stratified w.r.t. cluster size.

⁷ We evaluated the linking and clustering decision w.r.t. the top-4 mention and entity candidates produced by the bi-encoder. Although recall@4 for the bi-encoder is 97%, some relevant candidates might have been missed.

⁸ For the evaluation to be significant, we treat all clusters referring to the same known entity as a single cluster.

Table 4. Results of the manual evaluation of 100 clusters and 100 mentions. Columns group the results by actual entity type (known, NIL, overall), rows group by prediction outcome. Accuracy values may deviate by $\pm 9.6\%$ for mentions and by $\pm 7.0\%$ for clusters (95% confidence).

Prediction	Mentions			Clusters		
	\mathcal{E}^k	\mathcal{E}^n	\mathcal{E}	\mathcal{E}^k	\mathcal{E}^n	\mathcal{E}
Correct	20	64	84	8	71	79
Incorrectly linked to NIL entity	3	—	3	1	—	1
Incorrectly linked to known entity	—	7	7	—	3	3
Not all mentions of entity in cluster	—	—	—	8	0	8
Mentions from multiple entities in cluster	—	—	—	1	4	5
Ignored (mention extracted incorrectly)	—	—	6	—	—	4
Total Count	23	71	94	18	78	96
Accuracy (%)	87.0	90.1	89.4	44.4	91.0	82.3

This drop in performance is not due to NASTyLinker being incapable of linking to known entities correctly (as the accuracy of 87% on mention-level shows). Instead, it can rather be attributed to the fact that clusters of known entities contain 3.8 mentions on average, while clusters of NIL entities contain 1.7 mentions on average. Hence, the likelihood of missing at least one mention is a lot higher, which is also the main error for known clusters.

Compared to the results on NILK, the linking accuracy for NIL mentions is much higher. We explain this with the different kinds of NIL entities contained in the two datasets. While an average NIL entity is mentioned 4.6 times in NILK, our results indicate that this number is approximately 1.7 for the LISTING dataset. The latter dataset may hence contain a lot of easy-to-link mentions by assigning them their own cluster.

6 Conclusion and Outlook

With NASTyLinker, we introduce a NIL-aware EL approach that is capable of making high-quality predictions for known and for NIL entities. In the practical setting of EL in Wikipedia listings, we show that our approach can be used to populate a knowledge graph with a large number of additional entities as well as to enrich representations of existing entities.

Although the results look promising at a first glance, there is still a lot to improve as even small errors can multiply in downstream applications. For future work, we plan to concentrate on establishing a full end-to-end pipeline that includes the detection of mention, as recent works demonstrate how this can substantially reduce runtime without a decrease in performance [4, 24]. This will also open the path to a training procedure that considers NIL entities already during the creation of embeddings. Additionally, we will explore how the dependencies between items in listings can be exploited to further improve predictions.

References

1. Agarwal, D., Angell, R., Monath, N., McCallum, A.: Entity linking and discovery via arborescence-based supervised clustering. arXiv preprint [arXiv:2109.01242](https://arxiv.org/abs/2109.01242) (2021)
2. Alfaro, C.A., Perez, S.L., Valencia, C.E., Vargas, M.C.: The assignment problem revisited. *Optim. Lett.* **16**(5), 1531–1548 (2022). <https://doi.org/10.1007/s11590-021-01791-4>
3. Angell, R., Monath, N., Mohan, S., Yadav, N., McCallum, A.: Clustering-based inference for biomedical entity linking. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2598–2608 (2021)
4. Ayoola, T., Tyagi, S., Fisher, J., Christodoulopoulos, C., Pierleoni, A.: ReFinED: an efficient zero-shot-capable approach to end-to-end entity linking. arXiv preprint [arXiv:2207.04108](https://arxiv.org/abs/2207.04108) (2022)
5. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics* (1998)
6. Blissett, K., Ji, H.: Cross-lingual NIL entity clustering for low-resource languages. In: *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pp. 20–25 (2019)
7. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 708–716 (2007)
8. Dadzie, A., Preotiuc-Pietro, D., Radovanovic, D., Basave, A.E.C., Weller, K. (eds.): *Proceedings of the 6th Workshop on ‘Making Sense of Microposts’ Co-Located with the 25th International World Wide Web Conference (WWW 2016)*, Montréal, Canada, 11 April 2016, CEUR Workshop Proceedings, vol. 1691. CEUR-WS.org (2016). <http://ceur-ws.org/Vol-1691>
9. Das, R., et al.: Multi-step entity-centric information retrieval for multi-hop question answering. In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 113–118 (2019)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
11. Dutta, S., Weikum, G.: C3EL: a joint model for cross-document co-reference resolution and entity linking. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 846–856 (2015)
12. Fahrni, A., Heinzerling, B., Göckel, T., Strube, M.: HITS’Monolingual and cross-lingual entity linking system at TAC 2013. In: *TAC. Citeseer* (2013)
13. Ganea, O.E., Hofmann, T.: Deep joint entity disambiguation with local neural attention. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2619–2629 (2017)
14. Gillick, D., et al.: Learning dense representations for entity retrieval. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 528–537 (2019)
15. Greenfield, K., et al.: A reverse approach to named entity extraction and linking in microposts. In: *# Microposts*, pp. 67–69 (2016)

16. Heist, N., Hertling, S., Ringler, D., Paulheim, H.: Knowledge graphs on the web-an overview. In: Knowledge Graphs for eXplainable Artificial Intelligence, pp. 3–22 (2020)
17. Heist, N., Paulheim, H.: Uncovering the semantics of Wikipedia categories. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11778, pp. 219–236. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_13
18. Heist, N., Paulheim, H.: Entity extraction from Wikipedia list pages. In: Harth, A., et al. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 327–342. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_19
19. Heist, N., Paulheim, H.: Information extraction from co-occurring similar entities. In: Proceedings of the Web Conference 2021, pp. 3999–4009 (2021)
20. Heist, N., Paulheim, H.: Transformer-based subject entity detection in Wikipedia listings. arXiv preprint [arXiv:2210.01482](https://arxiv.org/abs/2210.01482) (2022)
21. Iurshina, A., Pan, J., Boutalbi, R., Staab, S.: NILK: entity linking dataset targeting NIL-linking cases. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 4069–4073 (2022)
22. Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J.: Overview of the TAC 2010 knowledge base population track. In: Third Text Analysis Conference (TAC 2010), vol. 3, p. 3 (2010)
23. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2019)
24. Kassner, N., Petroni, F., Plekhanov, M., Riedel, S., Cancedda, N.: EDIN: an end-to-end benchmark and pipeline for unknown entity discovery and indexing. arXiv preprint [arXiv:2205.12570](https://arxiv.org/abs/2205.12570) (2022)
25. Logan IV, R.L., McCallum, A., Singh, S., Bikel, D.: Benchmarking scalable methods for streaming cross document entity coreference. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4717–4731 (2021)
26. Logeswaran, L., Chang, M.W., Lee, K., Toutanova, K., Devlin, J., Lee, H.: Zero-shot entity linking by reading entity descriptions. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3449–3460 (2019)
27. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 824–836 (2018)
28. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 509–518 (2008)
29. Monahan, S., Lehmann, J., Nyberg, T., Plymale, J., Jung, A.: Cross-lingual cross-document coreference with entity linking. In: TAC (2011)
30. Partalidou, E., Christou, D., Tsoumakas, G.: Improving zero-shot entity retrieval through effective dense representations. In: Proceedings of the 12th Hellenic Conference on Artificial Intelligence, pp. 1–5 (2022)
31. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. Semant. Web **8**(3), 489–508 (2017)
32. Radford, W., Hachey, B., Honnibal, M., Nothman, J., Curran, J.R.: Naive but effective NIL clustering baselines-CMCRC at TAC 2011. In: Proceedings of Text Analysis Conference (TAC 2011). Citeseer (2011)

33. Rao, D., McNamee, P., Dredze, M.: Entity linking: finding extracted entities in a knowledge base. In: Poibeau, T., Saggion, H., Piskorski, J., Yangarber, R. (eds.) *Multi-Source, Multilingual Information Extraction and Summarization. NLP*, pp. 93–115. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-28569-1_5
34. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992 (2019)
35. Ristoski, P., Lin, Z., Zhou, Q.: KG-ZESHEL: knowledge graph-enhanced zero-shot entity linking. In: *Proceedings of the 11th on Knowledge Capture Conference*, pp. 49–56 (2021)
36. Rizzo, G., Pereira, B., Varga, A., Van Erp, M., Cano Basave, A.E.: Lessons learnt from the Named Entity rEcognition and linking (NEEL) challenge series. *Semant. Web* **8**(5), 667–700 (2017)
37. Sevgili, Ö., Shelmanov, A., Arkhipov, M., Panchenko, A., Biemann, C.: Neural entity linking: a survey of models based on deep learning. *Semant. Web* **13**(3), 527–570 (2022)
38. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* **27**(2), 443–460 (2014)
39. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. *Commun. ACM* **57**(10), 78–85 (2014)
40. Wang, Z., Ng, P., Nallapati, R., Xiang, B.: Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 347–357 (2021)
41. Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L.: Scalable zero-shot entity linking with dense entity retrieval. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6397–6407 (2020)