Neural Entity Linking: A Survey of Models Based on Deep Learning

Özge Sevgili [a,*], Artem Shelmanov [b,c,**], Mikhail Arkhipov [c], Alexander Panchenko [b], Chris Biemann [a]

[a] *Language Technology Group, Department of Informatics, University of Hamburg, Germany*
E-mails: sevgili@informatik.uni-hamburg.de, biemann@informatik.uni-hamburg.de
[b] *Center for Data Intensive Science and Engineering, Skolkovo Institute of Science and Technology, Russia*
E-mails: a.shelmanov@skoltech.ru, a.panchenko@skoltech.ru
[c] *Research Computing Center, Lomonosov Moscow State University, Russia*
[d] *Neural Networks and Deep Learning Laboratory, Moscow Institute of Physics and Technology, Russia*
E-mail: arkhipov@yahoo.com

Editors: First Editor, University or Company name, Country; Second Editor, University or Company name, Country
Solicited reviews: First Solicited Reviewer, University or Company name, Country; Second Solicited Reviewer, University or Company name, Country
Open reviews: First Open Reviewer, University or Company name, Country; Second Open Reviewer, University or Company name, Country

Abstract. In this survey, we provide a comprehensive description of recent neural entity linking (EL) systems developed since 2015 as a result of the "deep learning revolution" in NLP. Our goal is to systemize design features of neural entity linking systems and compare their performances to the best classic methods on the common benchmarks. We distill generic architectural components of a neural EL system, like candidate generation and entity ranking summarizing the prominent methods for each of them, such as approaches to mention encoding based on the self-attention architecture. The vast variety of modifications of this general neural entity linking architecture are grouped by several common themes: joint entity recognition and linking, models for global linking, domain-independent techniques including zero-shot and distant supervision methods, and cross-lingual approaches. Since many neural models take advantage of pre-trained entity embeddings to improve their generalization capabilities, we provide an overview of popular entity embedding techniques. Finally, we briefly discuss applications of entity linking, focusing on the recently emerged use-case of enhancing deep pre-trained masked language models such as BERT.

Keywords: Entity Linking, Deep Learning, Neural Networks, Natural Language Processing, Knowledge Bases

1. Introduction

1.1. Entity Linking

Knowledge Bases (KBs), such as Freebase [7], DBpedia [2], and Wikidata [110], contain rich and precise information about entities of all kinds, such as persons, locations, organizations, movies, scientific theories to name a few. Each entity has a set of carefully defined relations and attributes, e.g. "was born in" or "play for". This wealth of structured information gives rise and facilitates the development of semantic processing algorithms as they can directly operate on and benefit from such entity representations. For instance, imagine a search engine that is able to retrieve mentions in the news during the last month of all retired NBA players with a net income of more than 1 billion US dollars. The list of players together with their income and retirement information may be available in a knowledge base. Equipped with this information, it appears to be straightforward to look up mentions of

*Equal contribution. Corresponding author. E-mail: sevgili@informatik.uni-hamburg.de.
**Equal contribution. Corresponding author. E-mail: a.shelmanov@skoltech.ru.

such retired basketball players in the newswire. However, the main obstacle for such a direct counting algorithm is the lexical ambiguity of entities. In the context of this application, one would want to only retrieve all mentions of "Michael Jordan (basketball player)"[1] and exclude mentions of other persons with the same name such as "Michael Jordan (mathematician)"[2].

This is why Entity Linking (EL) – the process of matching a mention, e.g. "Michael Jordan", in a textual context to a KB record (e.g. "basketball player" or "mathematician") fitting the context – is the key technology enabling various semantic applications. Thus, EL is the task of identifying an entity mention in text and establishing a link to an entry in a knowledge base (therefore connecting an unstructured data to a structured data).

Entity linking is an essential component of many information extraction and Natural Language Understanding (NLU) pipelines since it resolves the lexical ambiguity of entity mentions and determines their meanings. A link between a textual mention and an entity in a knowledge base also allows to take advantage of the information encompassed in a semantic graph, which is shown to be useful in such NLU tasks as information extraction [73], biomedical text processing [103], or semantic parsing and question answering [6, 119]. This wide range of direct applications is the reason why entity linking is enjoying a great interest from both academy and industry for more than two decades.

1.2. Goal and Scope of this Survey

Recently, a new generation of approaches for entity linking based on the neural models and deep learning emerged pushing the state-of-the-art to the new level. The goal of this survey is to provide an overview of this latest wave of models, emerging from 2015 until now.

Models based on neural networks have managed to excel in EL as in many other natural language processing tasks due to their ability to learn useful deep distributed representations of linguistic data [18, 120]. The state-of-the-art neural entity linking models have shown significant improvements over "classical"[3] machine learning approaches [15, 53, 89] that are based on shallow architectures, e.g. Logistic Regression, and/or depend mostly on hand-engineered features. Such models often cannot capture all relevant statistical dependencies and interactions [32]. In contrast, deep neural networks are able to learn sophisticated representations within their deep layered architectures reducing the burden of manual feature engineering. This capability enabled improvements on various tasks, including EL as will be discussed in detail in Section 4.

In this survey, we systemize recently proposed neural models, distilling one generic architecture commonly used by the popular neural EL models (illustrated in Figures 2 and 5). We categorize and summarize the models used in each component of this architecture, e.g. candidate generation or ranking. The prominent variations of this generic architecture, e.g. end-to-end EL or global models, are also categorized and discussed. To better structure the sheer amount of available models, various types of methods are illustrated in the form of taxonomies (Figures 3 and 6) while notable features of each model are carefully assembled in tabular form (Tables 2 and 3).

An important component of neural entity linking systems is entity vector representations and entity encoding methods. It has been shown that encoding in low-dimensional vectors the KB structure (entity relationships), entity definitions, as well as textual information in large annotated corpora, helps to improve the generalization capabilities of EL models significantly. We summarize novel methods for entity encoding, as well as context/mention encoding techniques.

Many natural language processing systems take advantage of deep pre-trained language models like ELMo [84], BERT [22], and their modifications. EL made its path into these models as a way of introducing information stored in KBs, which helps to adopt word representations to some text processing tasks. We discuss this novel application of EL and how it can be further developed.

1.3. Previous Surveys

One of the first surveys on EL is provided by Ling et al. [60], in 2015. They aim at providing (1) a standard problem definition to reduce a confusion that appears due to existence of variant similar tasks related to EL (e.g., Wikification [68] and named entity linking [44]), and (2) a clear comparison of models and their various aspects. In the same year, Shen et al. [96] published a survey covering the main approaches to en-

[1] https://en.wikipedia.org/wiki/Michael_Jordan
[2] https://en.wikipedia.org/wiki/Michael_I._Jordan
[3] https://towardsdatascience.com/deep-learning-vs-classical-machine-learning-9a42c6d48aa

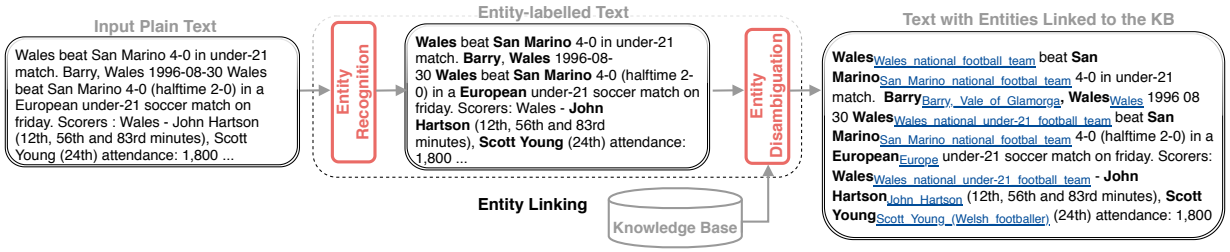

Fig. 1. **The entity linking task**. EL model takes a raw textual input and enriches it with entity mention links in a KB. Commonly the task is split into entity recognition and entity disambiguation sub-tasks.

tity linking, its applications, evaluation methods, and future directions.

There are also other surveys, which address a wider scope. The work of Martínez-Rodríguez et al. [64], published in 2020, involves information extraction models and semantic web technologies. Namely, they consider named entity recognition, entity linking, terminology extraction, keyphrase extraction, topic modeling, topic labeling, and relation extraction tasks for information extraction side. In a similar vein, Al-Moslmi et al. [1], released in 2020, overview the research in named entity recognition and named entity disambiguation/linking published between 2014-2019.

Another recent survey paper by Oliveira et al. [78], published in 2020, analyses and summarizes EL approaches that exhibit some holism. This viewpoint limits the survey to the works that exploit various peculiarities of the EL task: additional metadata stored in specific input like microblogs, specific features that can be extracted from this input like geographic coordinates in tweets, timestamps, interests of users posted these tweets, and specific disambiguation methods that take advantage of these additional features.

The previous surveys (a) do not cover many recent publications [60, 96], (b) broadly cover numerous topics [1, 64], or (c) are focused on the specific types of methods [78]. There is not yet, to our knowledge, a detailed survey specifically devoted to recent neural entity linking models. The previous surveys also do not address the topics of entity and context/mention encoding, applications of EL to deep pre-trained language models, and cross-lingual EL. We also the first to summarize the domain-independent approaches to EL, several of which are based on zero-shot techniques.

1.4. Contributions

More specifically, this paper makes the following contributions:

- a survey of state-of-the-art neural entity linking models;
- feature tables for neural EL methods;
- a survey of entity and context/mention embedding techniques;
- a discussion of recent domain-independent (zero-shot) and cross-lingual EL approaches;
- a survey of EL applications to modeling word representations.

The structure of this survey is the following. We start with defining the task of EL in Section 2. In Section 3.1, the common architecture of neural entity linking systems is presented. Modifications and variations of this basic pipeline are discussed in Section 3.2. In Section 4, we summarize the evaluation results for EL and entity representation models. Section 5 is dedicated to the application of EL by highlighting recently emerged applications for improving neural language models. Finally, Section 6 summarizes the survey and suggests a prominent direction of future work in neural EL.

2. Task Description

2.1. Informal Definition

Consider the example presented in Figure 1 with an entity mention *Scott Young*. Literally, this common name can at least refer to an *American football player*, *Welsh football player*, or a *writer*. The EL task is to correctly determine the mention in the text, resolve its ambiguity, and ultimately provide a link to a corresponding entity entry in a KB. To achieve this goal, commonly the task is decomposed into two stages, as illustrated in Figure 1: Entity Recognition (ER) and Entity Disambiguation (ED).

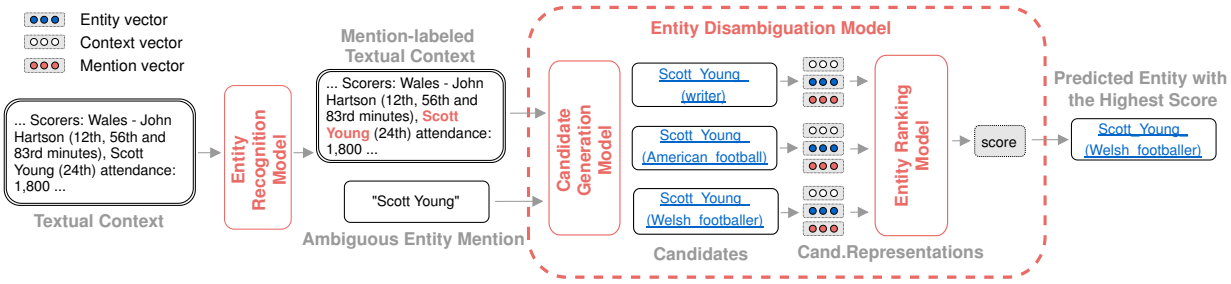

Fig. 2. **General architecture for neural entity linking.** EL contains two main steps: *Entity Recognition*, mentions in a plain text are distinguished, and *Entity Disambiguation*, a corresponding entity is predicted for the given mention. Entity Disambiguation is further divided into two steps: *Candidate Generation*, possible entities are produced for the mention, and *Entity Ranking*, a score between context/mention and a candidate is computed through the representations.

2.2. Formal Definition

The goal of entity recognition is just to identify the position of an entity mention, while entity disambiguation is performing linking of these mentions to entries of a KB. Formally, we use an ER function takes as input a textual context (e.g. a document) C and outputs a sequence of n mentions in this context:[4]

$$\mathsf{ER} : C \rightarrow M^n. \tag{1}$$

Formally the entity disambiguation task can be represented as a function ED, in which given sequence of n mentions in a document, $(m_1, \ldots m_n), m_i \in M$, and their contexts $(c1, \ldots, c_n), c_i \in C$, a function predicts an entity assignment $(e_1, \ldots, e_n), e_i \in E$:

$$\mathsf{ED} : (M, C)^n \rightarrow E^n. \tag{2}$$

To learn a mapping from entity mentions in a context to entity entries in a KB, EL models use supervision signals like manually annotated mention-entity pairs. The size of KBs vary; they can contain hundreds of thousands or millions of entities. Due to their large size, training data for EL would be extremely unbalanced; training sets can lack even a single example for a particular entity or mention, e.g. AIDA training set [44]. To deal with this problem, EL models should have wide generalization capabilities. Despite their large size, KBs are incomplete. Therefore, some mentions in the text cannot be correctly mapped to any KB entry. Determining such unlinkable mentions is one of the EL challenges. Furthermore, it is customary to distinguish "local" and "global" EL tasks. Local EL performs disambiguation of each mention in text independently using only context near target mentions, while global EL deals with interdependent disambiguation of all the mentions and can engage features extracted from the whole document. We further discuss global models in Section 3.2.1 providing a formal definition in Equation 16.

[4] We adopt and extend notation presented by Ganea et al. [33].

2.3. Terminological Aspects

More or less the same technologies and models sometimes called differently in the literature. Namely, Wikification [14] and Entity Disambiguation (ED) are considered as subtypes of EL [74]. To be comprehensive in this survey, we assume that entity linking task encompasses both entity recognition (ER) and entity disambiguation (ED). However, only few studies suggest models that perform ER and ED jointly, while the majority of papers on EL focus exclusively on ED and assume that mention boundaries are given by an external entity recogniser [91] (which may lead to some terminological confusions). Numerous techniques that perform ER only without disambiguation are considered in many previous surveys [36, 72, 95, 115] and are out of the scope of this work.

3. Neural Entity Linking

We start the discussion of neural entity linking approaches from the most general structure of pipelines and continue with various specific modifications like joint entity recognition and linking, using global context, domain-independent approaches including zero-shot methods, and cross-lingual models.

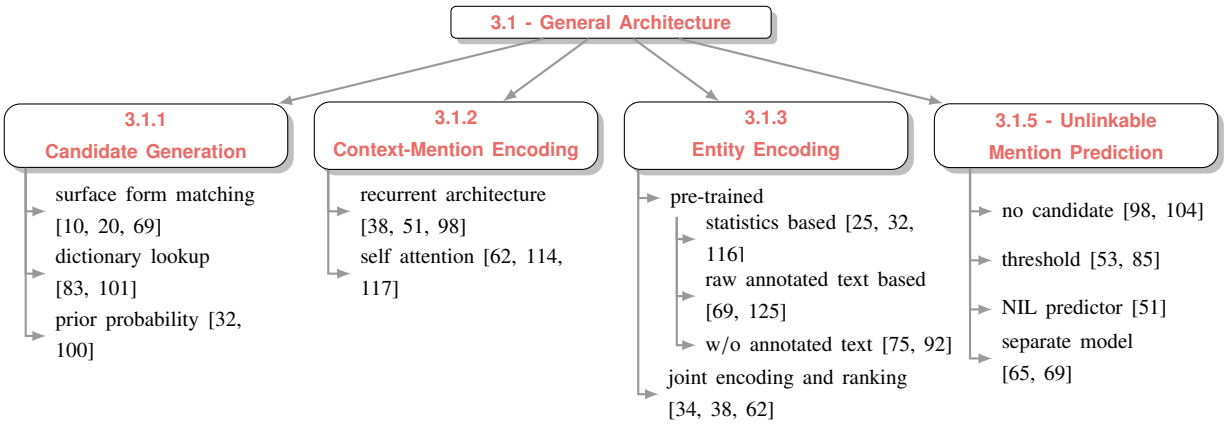

Fig. 3. **Reference map of general architecture for neural EL.** The categorization of each step in the general neural EL architecture with alternative design choices and sample references illustrating each of the choices.

3.1. General Architecture

Some of the attempts to EL based on neural networks treat it as a multi-class classification task, in which entities correspond to classes. However, the straightforward approach results in a large number of classes, which leads to suboptimal performance without task sharing [50]. The streamlined approach to EL is to treat it as a ranking problem. We present the generalized EL architecture in Figure 2, which is applicable to main neural approaches. Here, the entity recognition model identifies the mention boundaries in text. The next step is to produce a short list of possible entities (candidates) for the mention. Then, the mention encoder produces a semantic vector representation of a mention in a context. The entity encoder produces a set of vector representations of candidates. Finally, the entity ranking model compares mention and entity representations and estimates how well candidate entities match the context. An optional step is to determine unlinkable mentions, for which KBs do not contain a corresponding entity. The categorization of each step in the general neural EL architecture is summarized in Figure 3.

3.1.1. Candidate Generation

An essential part of EL is candidate generation. The goal of this step is given an ambiguous entity mention, such as "Scott Young", to provide a list of its possible "senses" as specified by entities in a KB. EL is analogous to the Word Sense Disambiguation (WSD) task [70, 74] as it resolves lexical ambiguity. Yet in WSD, each sense of a word can be clearly defined by WordNet [27], while in EL, KBs do not provide such an exact mapping between mentions and entities. Therefore, a mention potentially can be linked to any entity in a KB, resulting in large search space, e.g. the "Big Blue" referring to IBM. To address this issue, preliminary filtering of an entity list, called candidate generation, is performed.

Formally, given a mention m_i, a candidate generator provides a list of probable entities, $e_1, e_2, ..., e_k$, for all n entity mentions in a document.

$$\mathsf{CG} : M^n \rightarrow (e_1, e_2, ..., e_k)^n \tag{3}$$

There are three prominent candidate generation methods: a surface form matching, a dictionary lookup, and a prior probability computation. In the first approach, a candidate list is composed of entities, which match various surface forms of mentions in the text [10, 20, 21, 40, 55, 69, 83, 125]. There are many heuristics for generation of mention forms and matching criteria like the Levenshtein distance, n-grams, and normalization. For the example mention of "Big Blue", this approach could not work well, as the referent entity "IBM" does not contain a mention string. Examples of candidate entity sets are presented in Table 1, where we searched a name matching of the mention "Big Blue" in the titles of the all Wikipedia articles present in DBpedia[5].

In the second approach, a dictionary of additional aliases is constructed using KB metadata like disambiguation/redirect pages of Wikipedia [26]. This helps to improve the candidate generation recall. Pershina et al. [83] provide a resource of this type used in

[5]http://downloads.dbpedia.org/2016-10/core-i18n/en/labels_en.ttl.bz2

Table 1

Candidate generation examples. Ten sample candidate entities for the example mention "Big Blue" for each method. The highlighted are "correct" candidates assuming that given mention refers to the IBM corporation and not a river, e.g. Big_Blue_River_(Kansas).

Method	10 sample candidate entities for the example mention "Big Blue"
surface form matching based on DBpedia	Santa_Monica_Big_Blue_Bus, Bear_in_the_big_blue_house, The_Big_Blue_Bug, The_Big_Blue_Marble, IBM_Big_Blue_(rugby_union), The_Blue_Mouse_and_the_Big_Faced_Cat, The_Big_Blue_(A-League), The_Big_Blue_Megamix, Millikin_Big_Blue_football, IBM_Big_Blue_(disambiguation)
dictionary lookup based on YAGO-means	Big_Blue_River_(Indiana), Big_Blue_River_(Kansas), Big_Blue_(crane), Big_Red_(drink), **IBM**, IBM_Big_Blue, Millville_Football_&_Athletic_Club, Our_Lady_of_Mount_Carmel_High_School_(Baltimore,_Maryland), The_Big_Blue, Tift_County_High_School
prior probability based on CrossWikis	**IBM**, Big_Blue_River_(Kansas), The_Big_Blue, Utah_State_University, New_York_Giants, Big_Blue_River_(Indiana), Big_Blue_(crane), Big_Blue_(disambiguation), Deep_Blue_(chess_computer), Superman

many EL models [11, 65, 75, 87, 116]. Another well-known alternative is the YAGO ontology [101] – automatically constructed from Wikipedia and WordNet. Among many other relations, it provides *'means'* relations between mentions and entities, and this mapping is utilized as a candidate generator [12, 32, 44, 51, 85, 116, 117]. In this technique, the external dictionaries would help to disambiguate "Big Blue" as "IBM". In Table 1, sample candidate entity sets of the YAGO-means dataset[6] are shown.

The candidates are also generated based on pre-calculated prior probabilities of correspondence between certain mentions and entities, $p(e|m)$. Most of the studies rely on priors computed on the basis of Wikipedia anchor links [12, 32, 51, 56, 85, 98, 104, 117, 118, 125]. Another widely used option is CrossWikis [100], which is an extensive dictionary computed from the frequency of mention-entity links of web crawl data [12, 32, 38, 51, 85, 117]. Using the CrossWikis dictionary, the example mention string "Big Blue" can be labeled as its referent entity "IBM" with pre-computed priors, as shown in Table 1.

Recent zero-shot models [34, 62, 114] perform candidate generation without external knowledge. Section 3.2.3 describes them in detail.

3.1.2. Context-mention Encoding

To correctly disambiguate an entity mention, it is crucial to thoroughly capture the information from its context. The streamline approach is to construct a dense contextualized vector representation of a mention $\boldsymbol{y_m}$ using an encoder network.

$$\mathsf{mENC} : (C, M)^n \rightarrow (\boldsymbol{y_{m_1}}, \boldsymbol{y_{m_2}}, ..., \boldsymbol{y_{m_n}}) \tag{4}$$

[6] http://resources.mpi-inf.mpg.de/yago-naga/aida/download/aida_means.tsv.bz2

Several early techniques in neural EL utilize a convolutional encoder [29, 77, 99, 102], as well as attention between candidate entity embeddings and embeddings of words surrounding a mention [32, 56]. However, in recent models, two approaches prevail: recurrent networks and self-attention [108].

A recurrent architecture with LSTM cells [43] that has been a backbone model for many NLP applications, is adopted to EL in [26, 38, 51, 55, 65, 98] inter alia. Gupta et al. [38] concatenate outputs of two LSTM networks that independently encode left and right contexts of a mention (including the mention itself). In the same vein, Sil et al. [98] encode left and right local contexts via LSTMs but also pool the results across all mentions in a coreference chain and postprocess left and right representations with a tensor network. A modification of LSTM – GRU [17] is used by Eshel et al. [24] in conjunction with an attention mechanism [3] to encode left and right context of a mention. Kolitsas et al. [51] represent an entity mention as a combination of LSTM hidden states included in the mention span. Le and Titov [55] simply run a bidirectional LSTM network on words complemented with embeddings of word positions relative to a target mention. Shahbazi et al. [94] adopt pre-trained ELMo [84] for mention encoding by averaging mention word vectors.

Encoding methods based on self-attention have recently become ubiquitous. The EL models presented in [62, 85, 114, 117] rely on the outputs from pre-trained BERT layers [22] for context and mention encoding. In Peters et al. [85], a mention representation is modeled by pooling over word pieces in a mention span. The authors also put an additional self-attention block over all mention representations that encode interactions between several entities in a sentence. Another approach to modeling mentions is to insert special tags around them and perform a reduction of the whole encoded sequence. Wu et al. [114] reduce a sequence by

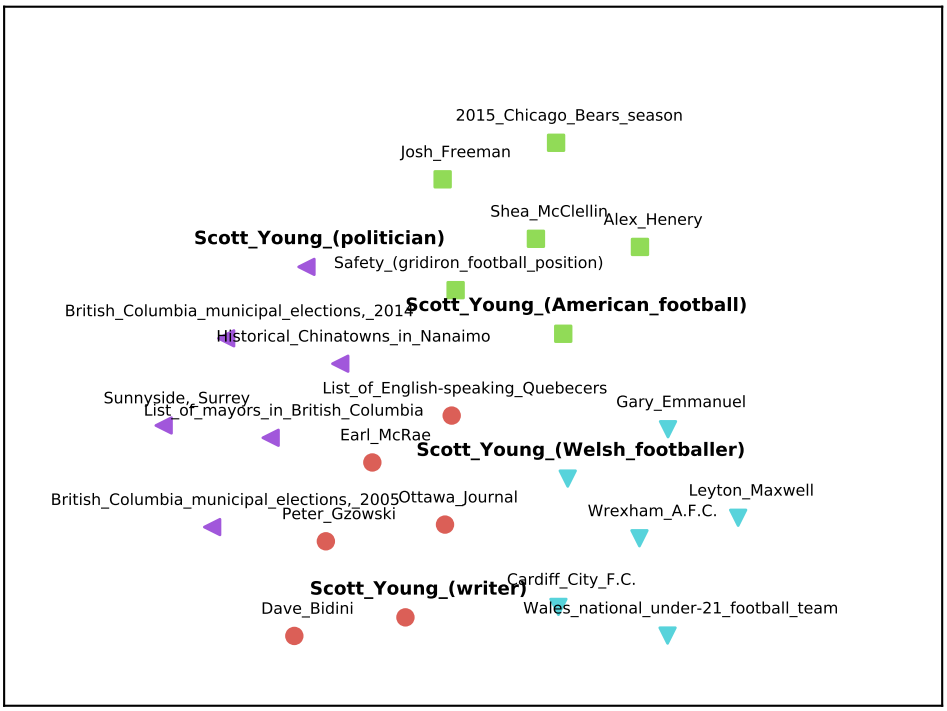

Fig. 4. **Visualization of entity embeddings.** Entity embedding space for entities related to the ambiguous entity mention "Scott Young". Four candidate entities from Wikipedia/DBpedia are illustrated. For each entity, their most similar 5 entities are shown in the same colors. Entity embeddings are visualized with t-SNE using pre-trained embeddings by [92].

keeping the representation of the special pooling symbol '[CLS]' inserted at the beginning of a sequence. Logeswaran et al. [62] mark positions of a mention span by summing embeddings of words within the span with a special vector and use the same reduction strategy as Wu et al. [114]. Yamada et al. [117] concatenate text with all mentions in it and jointly encode this sequence via a self-attention model based on pre-trained BERT.

3.1.3. Entity Encoding

Good representations $\boldsymbol{y}_e$ of entity candidates that capture various semantic information are essential for making EL systems robust.

$$\text{eENC} : E^k \rightarrow (\boldsymbol{y}_{e_1}, \boldsymbol{y}_{e_2}, ..., \boldsymbol{y}_{e_k}) \tag{5}$$

Usually, entities are encoded into low-dimensional vectors in such a way that spatial proximity between them in a vector space correlates with their semantic relatedness. For instance, in Figure 4, *Scott Young* in the Scott_Young_(American_football) sense is close to football players, e.g. Alex_Henery, whereas Scott_Young_(politician) sense is in the proximity of other politicians.

Early EL methods, such as Milne and Witten [68] or He et al. [40] depend on hand-engineered features, e.g. one-hot vectors, to represent entities. These representations exhibit sparsity issues and are superseded by dense representations that follow the line of work on word embeddings like word2vec [67]. Huang et al. [45] train a model that generates from sparse entity features dense entity representations based on the entity relatedness in a KB. Several works expand entity relatedness objective with functions that align words and entities in the unified vector space using several features like entity-word co-occurrence statistics in

Table 2

Features of entity embeddings. Entity embedding models in terms of their data requirements and architectural features: the first six columns denote data related features; the remaining one refers to the architectural feature. (The footnotes in the table are explained in the text.)

	Annotated Text	Entity-Entity Links	Entity-Mention Links	Entity Descriptions	Entity Titles	Entity Types	Joint Space of Entities and Words
Huang et al. (2015) [45]		✘	✘	✘		✘	
Sun et al. (2015) [102]	✘				✘	✘	✘[1,6]
Fang et al. (2016) [25]	✘	✘	✘	✘			✘
Yamada et al. (2016) [116]	✘	✘					✘
Zwicklbauer et al. (2016) [125]	✘[2]			✘			
Tsai and Roth (2016) [104]	✘				✘		✘
Ganea and Hofmann (2017) [32]	✘						✘
Cao et al. (2017) [11]	✘	✘	✘				✘
Moreno et al. (2017) [69]	✘						✘
Gupta et al. (2017) [38]	✘			✘		✘	✘[4,6]
Sil et al. (2018) [98]				✘			✘
Upadhyay et al. (2018) [106]	✘		✘			✘	✘
Newman-Griffis et al. (2018) [75]					✘	✘	✘
Radhakrishnan et al. (2018) [87]	✘						✘
Rijhwani et al. (2019) [90]	✘	✘			✘		✘
Logeswaran et al. (2019) [62]				✘			✘[3,6]
Gillick et al. (2019) [34]	✘			✘	✘	✘	✘[6]
Le and Titov (2019) [55]						✘	✘[6]
Sevgili et al. (2019) [92]		✘		✘			
Shahbazi et al. (2019) [94]	✘						✘
Shi et al. (2020) [97]	✘	✘				✘	✘
Zhou et al. (2020) [124]	✘	✘	✘		✘		✘
Wu et al. (2019) [114]				✘	✘		✘[5,6]
Yamada et al. (2020) [117]	✘						✘[6]

Wikipedia [11, 25, 97, 116]. Some works rely solely on entity-word co-occurrence statistics in extensive textual resources [32, 87]. For example, Ganea and Hofmann [32] collect statistics through entity description pages and the mentions surrounding word in an annotated Wikipedia text. They train the embeddings so the vectors of positive words are closer (in terms of dot product) to the co-occurring embedding of entities compared to vectors of random words.

There are some other models, which directly replace the anchor text with an entity descriptor and train the word representation model like word2vec [69, 104, 125].

There are few recent studies, which perform entity encoding without entity annotated text data. For example, Newman-Griffis et al. [75] expand the word2vec architecture with a distant supervision setup based on the terminology of Wikipedia's page titles and redirects. Sevgili et al. [92] build a graph from entity-entity hyperlinks and execute a graph embedding algorithm, namely DeepWalk [82].

There are models that propose joint architectures, in which parameters for mention/context encoding or parameters for entity ranking are trained jointly with parameters for entity embedding. Sun et al. [102] initialize entity embeddings using word2vec through description page words, surface forms words, and entity category words, and fine-tune these representations during training of the ranking model. In the same vein, Francis-Landau et al. [29] and Nguyen et al. [77] use entity titles and description pages for embedding initialization. Gupta et al. [38] train several encoders for multiple types of information: entity types, entity description page, local and global context jointly with entity vector representations. Gillick et al. [34] encode entities based on entity page titles, the short entity description on the corresponding Wikipedia page, and category information of an entity. Logeswaran et al. [62] and Wu et al. [114] depend on BERT to create representations through the description pages. Le and Titov [55] propose a scalable approach for computing entity embeddings without relying on any pre-trained words using only types associated with entities specified in a KB. Yamada et al. [117] modify the BERT architecture so that it can predict an entity or a word corresponding to the masked term. Recently, Shahbazi

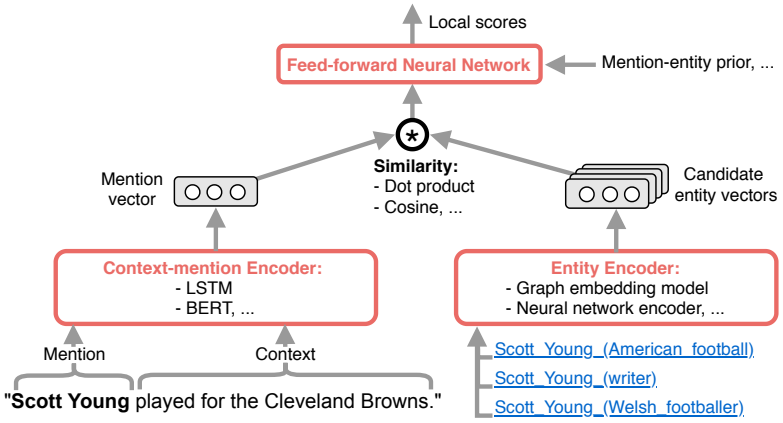

Fig. 5. **Entity ranking**. A generalized candidate entity ranking neural architecture: pairs composed of an entity candidate and a context are classified according to their goodness of fit.

et al. [94] introduce E-ELMo that extends the contextual language embedding model ELMo [84] with an additional objective. The model is trained in a multi-task fashion: to predict next/previous word, as in a standard bidirectional language model [84], and to predict the target entity when encountering its mentions. As a result, entity representations are obtained, as well as a model for mention encoding.

Finally, the entity representation models also greatly vary in terms of data sources, which can be structured (e.g. hyperlinks) or textual (i.e. a description of an entity, anchor texts, or annotated texts). The detailed model-wise comparison can be found in Table 2. The columns refer to the data/architectural features of models. The **annotated text** column corresponds to the models that use text explicitly annotated with entity mentions, e.g. coming from Wikipedia hyperlinks inside an article. The **entity-entity links** column specifies whether the model uses hyperlinks to learn entity representations, e.g. ontological relations in a KB (Jordan, played-for, Chicago Bulls), etc. The **entity-mention links** column corresponds to another relational data of entity provided by KBs, e.g. anchor links ("Big Blue" mention can refer to entity IBM). The **entity descriptions** column refers to description pages of entities, which are accepted as another text resource provided by KBs. The **entity titles** column is for the models, which include entity title information to represent entities. The **entity types or redirect** column refers to either redirect pages, e.g. Michael_Jordan_(basketball) is a redirect page for the Michael_Jordan entity, or entity types like person, organization, etc. The **joint space entity and word** column specifies whether the models aim at representing words and entities in the common vector space.

The following notations are used to present the external features of the model shown as a footnote in Table 2:

1. Encode mention and context in a single vector space.
2. Only entities are remained in the anchor text.
3. Mention and entity are in a single vector space, known as cross-encoder.
4. They use bi-encoders that consist of two independent encoders for entity and context [114].
5. They use bi-encoders and cross-encoders for processing mention, entity, and context.
6. They are trained jointly with a ranking component, so their score function is based on ranking.

3.1.4. Entity Ranking

The goal of this stage is given a list of entity candidates $(e_1, e_2, ..., e_k)$ from a KB and a context C with a mention M to rank these entities assigning a score to each of them, as in Equation 6, where n is a number of entity mentions in a document, k is a number of candidate entities. Figure 5 depicts the typical architecture of the ranking component.

$$\text{RNK} : ((e_1, e_2, ..., e_k), C, M)^n \rightarrow \mathbb{R}^{n \times k} \tag{6}$$

The mention representation $\boldsymbol{y_m}$ generated in the mention encoding step is compared with candidate entity representations $\boldsymbol{y_{e_i}}(i = 1, 2, \ldots, k)$ according to the similarity measure $s(m, e_i)$. Entity representations

can be pre-trained (see Section 3.1.3) or generated by another encoder as in some zero-shot approaches (see Section 3.2.3). The BERT-based model of Yamada et al. [117] simultaneously learns how to encode mentions and entity embeddings in the unified architecture.

Most of the state-of-the-art studies compute similarity $s(m,e)$ between representations of a mention m and an entity e using a dot product as in [32, 38, 51, 85, 114]:

$$s\left(m,e_i\right) = \boldsymbol{y}_m \cdot \boldsymbol{y}_{e_i}; \tag{7}$$

or cosine similarity as in [29, 34, 102]:

$$s\left(m,e_i\right) = \cos(\boldsymbol{y}_m, \boldsymbol{y}_{e_i}) = \frac{\boldsymbol{y}_m \cdot \boldsymbol{y}_{e_i}}{\|\boldsymbol{y}_m\| \cdot \|\boldsymbol{y}_{e_i}\|}. \tag{8}$$

The final disambiguation decision is inferred via a probability distribution $P(e_i|m)$, which is usually approximated by a softmax function over the candidates. The calculated similarity score or probability can be combined with mention-entity priors obtained during the candidate generation phase [29, 32, 51] or other features $f(e_i,m)$ such as various similarities, a string matching indicator, and entity types [29, 93, 94, 98, 118]. One of the common techniques for that is to use an additional one or two-layer feedforward network $\phi(\cdot,\cdot)$ [29, 32, 94]. The obtained local similarity score $\Phi(e_i,m)$ or the probability distribution can be further utilized for global scoring (see Section 3.2.2).

$$P(e_i|m) = \frac{\exp(s(m,e_i))}{\sum_{i=1}^{k}\exp(s(m,e_i))} \tag{9}$$

$$\Phi(e_i,m) = \phi(P(e_i|m), f(e_i,m)) \tag{10}$$

There are several approaches to frame a training objective in the literature on EL. Consider we have k candidates for the target mention m, one of which is a true entity e_*. In some works, the models are trained with the standard negative log likelihood objective like in classification tasks [62, 114]. However, instead of classes, negative candidates are used:

$$\mathcal{L}\left(m\right) = -s\left(m,e_*\right) + \log\sum_{i=1}^{k}\exp\left(s\left(m,e_i\right)\right) \tag{11}$$

Instead of the the negative log likelihood, some works use variants of a ranking loss. The idea behind such an approach is to enforce a positive margin $\gamma > 0$ between similarity scores of mentions to positive and negative candidates [32, 51, 85]:

$$\mathcal{L}(m) = \sum_i \ell(e_i,m), \text{ where} \tag{12}$$

$$\ell(e_i,m) = \left[\gamma - \Phi\left(e_*,m\right) + \Phi(e_i,m)\right]_+ \tag{13}$$

or

$$\ell(e_i,m) = \begin{cases} \left[\gamma - \Phi(e_i,m)\right]_+, & \text{if } e_i \text{ equal } e_* \\ \left[\Phi(e_i,m)\right]_+, & \text{otherwise} \end{cases} \tag{14}$$

3.1.5. Unlinkable Mention Prediction

The referent entities of some mentions can be absent in the KBs, e.g. there is no Wikipedia entry about *Scott Young* as a cricket player of the Stenhousemuir cricket club.[7] Therefore, an EL system should be able to predict the absence of a reference if a mention appears in specific contexts, which is known as NIL prediction task.

$$\mathsf{NIL} : (C,M)^n \rightarrow \{0,1\}^n \tag{15}$$

The NIL prediction task is essentially a classification with a reject option [30, 41, 42]. There are four common ways to perform NIL prediction. Sometimes a candidate generator does not yield any corresponding entities for a mention; such mentions are trivially considered unlikable [98, 104]. One can set a threshold for the best linking probability (or a score), below which a mention is considered unlinkable [53, 85]. Some models introduce an additional special 'NIL' entity in the ranking phase, so models can predict it as the best match for the mention [51]. It is also possible to train an additional binary classifier that accepts mention-entity pairs after the ranking phase, as well as several additional features (best linking score, whether mentions are also detected by a dedicated NER system, etc.), and makes the final decision about whether a mention is linkable or not [65, 69].

3.2. Modifications of the General Architecture

This section presents the most notable modifications and improvements of the general architecture of neural

[7] https://www.stenhousemuircricketclub.com/teams/171906/player/scott-young-1828009

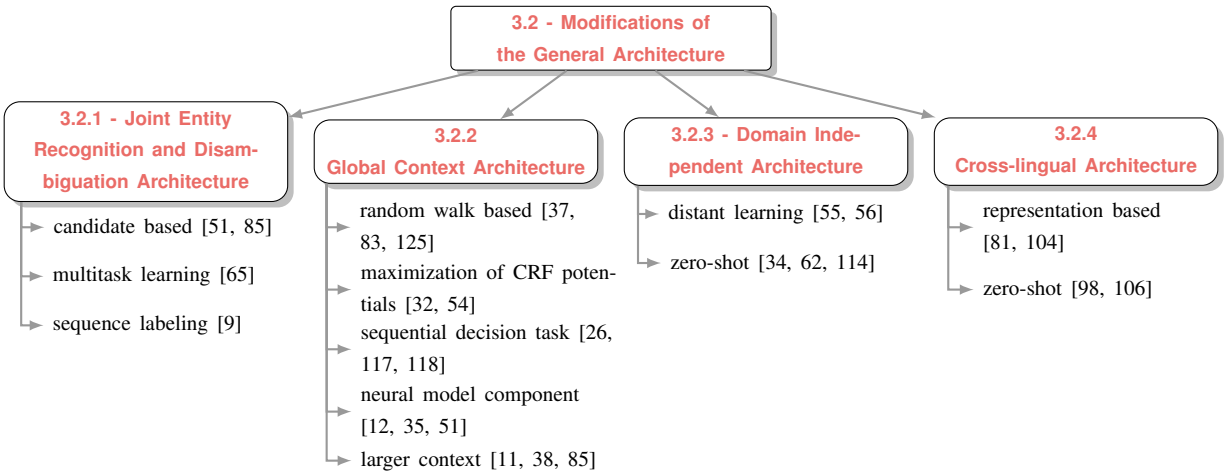

Fig. 6. **Reference map of the modifications of the general architecture for neural EL.** The categorization of each modification with various design choices and sample references illustrating each choice. Sections 3.2.3 and 3.2.4 are categorized based on their EL solutions, here.

entity linking models presented in Section 3.1 and Figures 2 and 5. The categorization of each modification is summarized in Figure 6.

3.2.1. Joint Entity Recognition and Disambiguation

While it is common to separate the entity recognition (cf. Equation 1) and entity disambiguation stages (cf. Equation 2) as illustrated in Figure 1, a few systems provide a *joint* solution for entity linking where entity recognition and disambiguation are done at the same time by the same model. Formally, the task becomes to detect a mention $m_i \in M$ and predict an entity $e_i \in E$ for given context $c_i \in C$, for all n entity mentions in the context:

$$\mathsf{EL} : C \rightarrow (M, E)^n. \tag{16}$$

Undoubtedly, solving these two problems simultaneously makes the task more challenging. However, the interaction between these steps can be beneficial for improving the quality of the overall pipeline due to their natural mutual dependency. While first competitive models that provide joint solutions were probabilistic graphical models [63, 76], we focus on purely neural approaches proposed recently [9, 51, 65, 85, 99].

The main difference of joint models is the necessity to produce also mention candidates. For this purpose, Kolitsas et al. [51] and Peters et al. [85] enumerate all spans in a sentence with a certain maximum width, filter them by several heuristics (remove mentions with stop words, punctuation, ellipses, quotes, and currencies), and try to match them to a pre-built index of entities used for the candidate generation. If a mention candidate has at least one corresponding entity candidate, it is further treated by a ranking neural network that can also discard it by considering it unlinkable to any entity in a KB (see Section 3.1.4). Therefore, the decision during the entity disambiguation phase affects entity recognition. In the same vein, Sorokin and Gurevych [99] treat each token n-gram up to a certain length as a possible mention candidate. Sorokin and Gurevych [99] use an additional binary classifier for filtering candidate spans, which is trained jointly with an entity linker. Banerjee et al. [5] also enumerates all possible n-grams and expands each of them with candidate entities, which results in a long sequence of points corresponding to a candidate entity for a particular mention n-gram. This sequence is further processed by a single-layer BiLSTM pointer network [109] that generates index numbers of potential entities in the input sequence.

Martins et al. [65] describe the approach with a tighter integration between recognition and linking phases via multi-task learning. The authors propose a stack-based bidirectional LSTM network with a shift-reduce mechanism and attention for entity recognition that propagates its internal states to the linker network for candidate entity ranking. The linker is supplemented with a NIL predictor network. The networks are trained jointly by optimizing the sum of losses from all three components.

Broscheit [9] goes further by suggesting a completely end-to-end method that deals with entity recognition and linking jointly without explicitly executing

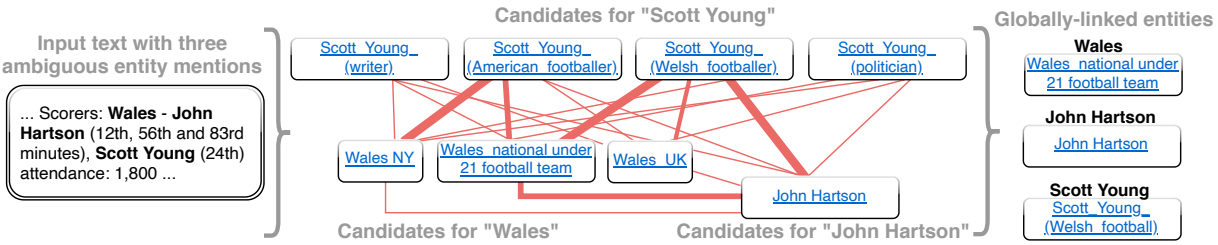

Fig. 7. **Global entity disambiguation**. The global entity linking resolves all mentions simultaneously based on entity coherence. Bolder lines indicate higher degrees of entity-entity similarity.

a candidate generation step. They formulate the task as a sequence labeling problem, where each token in the text is assigned an entity link or a NIL class. They leverage a sequence tagger based on pre-trained BERT for this purpose. This simplistic approach does not supersede [51] but outperforms the baseline, in which candidate generation, entity recognition, and linking are performed independently.

3.2.2. Global Context Architectures

Two kinds of contextual information are available in entity disambiguation: local and global. In local approaches to ED, each mention is disambiguated independently based on the surrounding words, as in the following function:

$$\text{LED} : (M, C) \rightarrow E \tag{17}$$

Global approaches to ED take into account semantic consistency across multiple entities in a context. In this case, all q entity mentions in a group are disambiguated interdependently: a disambiguation decision for one entity is affected by decisions made for other entities in the context as illustrated in Figure 7 and Equation 18. Here, the context refers to a larger number of surrounding words or the whole document.

$$\text{GED} : ((m_1, m_2, ..., m_q), C) \rightarrow E^q \tag{18}$$

Although the extra information of the global context improves the disambiguation accuracy, the number of possible entity assignments is combinatorial, which results in a high time complexity of disambiguation [32, 118]. Another difficulty is an attempt to assign an entity its coherence score, since this score is not possible to compute in advance due to the simultaneous disambiguation [116].

The typical approach to global disambiguation is to generate a graph containing candidate entities of mentions in a context and perform random walk algorithms, e.g. PageRank [80], over it to select highly consistent entities [37, 83, 125]. In this category, Globerson et al. [35] introduce a model with an attention mechanism that takes into account only the subgraph of the target mention, instead of all the mention candidates in a document.

Some works approach global ED by maximizing the Conditional Random Field (CRF) potentials, where the first component Φ represents a local entity-mention score, and the other component Ψ measures coherence among selected candidates [32, 33, 54, 56]:

$$g(e, m, c) = \sum_{i=1}^{n} \Phi(e_i, m_i, c_i) + \sum_{i<j} \Psi(e_i, e_j). \tag{19}$$

However, model training and its exact inference are NP-hard. Ganea and Hofmann [32] adapt loopy belief propagation [33, 35] with message passing iterations using pairwise entity scores to reduce the complexity. Le and Titov [54] expand it by modelling coreference relations of mentions as latent variables (the mentions are coreferent if they refer to the same entity). Shahbazi et al. [93] develop a greedy beam search strategy, which starts from a locally optimal initial solution and that is improved by searching possible corrections with the focus on the least confident mentions.

However, the models like aforementioned one take into account coherence among candidate entities of all mentions, which can result in high computational complexity and also be malicious due to erroneous coherence among wrong entities [26]. For example, if two mentions have coherent erroneous candidates, this noisy information can mislead the final global scoring. To resolve this issue, some studies define the problem as a sequential decision task, where the disambiguation of new entities is based on the already disambiguated ones. Fang et al. [26] train a policy network for sequential selection of entities using reinforcement learning. The disambiguation of mentions is ordered according

to the local score, so the mentions with high confident entities are resolved earlier. The policy network takes advantage of output from the LSTM global encoder that maintains the information about earlier disambiguation decisions. In the same vein, Yang et al. [118] use reinforcement learning to determine ordering for mention disambiguation. They also use an attention model to leverage knowledge from previously linked entities. The model dynamically selects the most relevant entities for the target mention and calculates the coherence scores. Yamada et al. [117] iteratively predict entities for yet unresolved mentions with a BERT model, while attending on the previous most confident entity choices. Yamada et al. [116] and Radhakrishnan et al. [87] measure the similarity first based on unambiguous mentions and then predict entities for complex cases.

Many studies rely on the idea of attaching an entity coherence component to the local scoring model and train their parameters jointly. In this case, local models can directly benefit from the pairwise coherence score without a necessity of handling the optimization of the global objective. The coherence component of Kolitsas et al. [51] is an additional feed-forward neural network that uses the similarity score between the target entity and an average embedding of the candidates with a high local score. Fang et al. [25] use the similarity score between the target entity and its surrounding entity candidates in a specified window as feature for the disambiguation model. In the same vein, Yamada et al. [116] and Radhakrishnan et al. [87] treat the global coherence as a feature for the final disambiguation model. Instead of computing entity coherence scores, Tsai and Roth [104] directly use embeddings of previously linked entities as features for the disambiguation model. Distinctively, Cao et al. [12] integrate a graph convolutional network into a disambiguation model that takes advantage of the knowledge provided by the a subgraph of candidate entities in a documents. Nguyen et al. [77] use a RNN to store information about previously seen mentions and corresponding entities. They leverage the hidden states of the RNN to reach this information as a feature for computation of the global score.

Another approach that can be considered as global model is to make use of a larger context to capture the coherence implicitly instead of explicitly designing an entity coherence component [11, 29, 38, 53, 69, 85, 98].

3.2.3. Domain-Independent Architectures

Domain independence is one of the most desired properties of EL systems. Annotated resources are very limited and exist only for a few domains. Obtaining labeled data in a new domain requires much labor. Earlier, this problem is tackled by few domain-independent approaches based on unsupervised [11, 54, 75, 112] and semi-supervised models [53]. Recent studies provide solutions based on distant learning and zero-shot methods.

Le and Titov [55, 56] propose distant learning techniques that use only unlabeled documents. They rely on the weak supervision coming from a surface matching heuristic, and the EL task is framed as binary multi-instance learning. The algorithm learns to distinguish between positive entities set and a set of random negatives. The positive set is obtained by retrieving entities with a high word overlap with the mention and that have relations in a KB to candidates of other mentions in the sentence. While showing promising performance, which in some cases rivals fully supervised systems, these approaches require either a KB describing relations of entities [55] or mention-entity priors computed from entity hyperlink statistics extracted from Wikipedia [56].

Recently proposed zero-shot techniques [34, 62, 114] pushed EL labeled data requirements to the minimum. In zero-shot setting, the only entity information available is its description. As well as in other settings, texts with mention-entity pairs are also available. The key idea of zero-shot is to train the system on domain with rich labeled data resources and apply it to a new domain with minimal data like descriptions of the domain-specific entities. One of the first studies that proposes such a technique is Gupta et al. [38] (not purely zero-shot because they use entity typings). Existing zero-shot systems do not require such information resources as surface form dictionaries, prior entity-mention probability, KB entity relations, and entity typing, which makes them particularly suited for building domain-independent solutions. However, the limitation of information sources raises new challenges.

Since only textual descriptions of entities are available for the target domain, one cannot rely on pre-build dictionaries for candidate generation. All zero-shot works rely on the same strategy to tackle candidate generation: pre-compute representations of the descriptions of the entities (sometimes referred as caching), compute representations of the mention and calculate its similarity with all description represen-

tations. Pre-computed representations of descriptions save a lot of time at the inference stage. Particularly, Logeswaran et al. [62] use the BM25 information retrieval formula [49], which is a similarity function of count-based representations. A natural extension of count-based approaches is embeddings. Gillick et al. [34] used average unigram and bigram embeddings followed by dense layers to obtain representations of mentions and descriptions. Cosine similarity is used for comparison of representations. Because of computational simplicity of the approach, it can be used in a single stage fashion where candidate generation and ranking are identical. However, for further speedup, approximate search can be used for candidate set retrieval. This set can be used for exact similarity computation, making the procedure two-stage as usual. Going neural, Wu et al. [114] introduce a BERT-based bi-encoder for candidate generation. Two separate encoders are used to get mention and description vectors. The ranking is performed by comparing dot-products of representations.

Zero-shot approaches use descriptions for entity ranking as well. Surprisingly, a very simple embedding-based approach [34] described above shows very competitive scores on TACKBP-2010, outperforming some complex neural architectures. The recent studies of Logeswaran et al. [62] and Wu et al. [114] utilize a BERT-based cross-encoder to perform joint encoding of mentions and entities. The cross-encoder takes concatenation of context with a mention and an entity description to produce a scalar score for each candidate. In both studies, cross-encoders achieve superior results compared to bi-encoders and count-based approaches.

Evaluation of zero-shot systems requires data from different domains in the format discussed at above. Logeswaran et al. [62] proposed *Zero-shot EL*[8] dataset, constructed from different Wikias[9]. In the proposed setting training is performed on one set of Wikias while evaluating on others. Gillick et al. [34] constructed Wikinews dataset. This dataset is used for evaluation after training on Wikipedia data.

Clearly, heavy neural architectures pre-trained on common open corpora push the limits of zero-shot further. As highlighted by Logeswaran et al. [62] further unsupervised pre-training on source as well as on the target data are beneficial. Development of better approaches to utilization of unlabeled data might be a fruitful direction of research. Furthermore, closing the entity ranking gap between fast representation based bi-encoder and computationally intensive cross-encoder is an open question.

[8]https://github.com/lajanugen/zeshel
[9]https://www.wikia.com

3.2.4. Cross-lingual Architectures

Abundance of labeled data for EL in English language contrasts with amount of data available in other languages. At the same time, such a unique source of supervision as Wikipedia is available for a variety of languages. However, there is still a big gap between resource-rich Wikipedia languages and low-resource ones.

The cross-lingual EL methods [48] aim at overcoming the lack of annotation for some languages by leveraging supervision coming from their high-resource counterparts. The inter-language links in Wikipedia is one of the most widely used sources of cross-lingual supervision. These links map pages to equivalent pages in another language.

Challenges in cross-lingual EL start at candidate generation and entity recognition steps, since the low-resource language can lack mappings between mention strings and entities. In addition to the standard methods with mention-entity priors [98, 104, 106], candidate generation can be approached by mining a translation dictionary [81], training a translation and alignment model [105], or applying a neural character-level string matching model [90]. The latter relies on training on a high-resource pivot language, similar to the target low-resource one. The neural string matching approach can be further improved with simpler average n-gram encoding and extending entity-entity pairs with mention-entity examples [124]. For entity recognition, the transfer of BiLSTM-CRF with a character encoding network from a similar high-resource pivot language can be applied [19].

There are several approaches to candidate ranking that take advantage of cross-lingual data for dealing with the lack of annotated examples. Pan et al. [81] uses the Abstract Meaning Representation (AMR) Banarescu et al. [4] statistics in English Wikipedia and mention context for ranking. To train an AMR tagger, pseudo-labeling [57] is used. Tsai and Roth [104] train monolingual embeddings for words and entities jointly by replacing every entity mention with corresponding entity tokens. Using the inter-language links, they learn the projection functions from multiple languages into the English embedding space. For ranking, context embeddings are averaged, projected into the English space, and compared with entity embeddings. The au-

thors demonstrate that this approach helps to build better entity representations and boosts the EL accuracy in cross-lingual setting by more than 1% for Spanish and Chinese. Sil et al. [98] propose a method for zero-shot transfer from a high-resource language. The authors extend the previous approach with the least squares objective for embedding projection learning, the CNN context encoder, and a trainable re-weighting of each dimension of context and entity representations. The proposed approach demonstrates improved performance compared to previous non-zero-shot approaches. Upadhyay et al. [106] argued that the success of zero-shot cross-lingual approaches [98, 104] might be highly related to mention-entity prior probabilities used as features. Their approach extends [98] with global context information and incorporation of typing information into context and entity representations (the system learns to predict typing during the training). The authors report a significant drop in performance for zero-shot cross-lingual EL with an excluded mention-entity prior, while showing state-of-the-art results with prior. They also show that training on the high-resource language might be very beneficial for the low-resource settings.

Existing techniques of cross-lingual entity linking heavily rely on pre-trained multilingual embeddings for entity ranking. While being effective in settings with at least prior probabilities available, the performance in realistic zero-shot scenarios drops drastically. Along with recent success of zero-shot multilingual transfer of large pre-trained language models, this might be a motivation to utilize powerful multilingual self-supervised models.

3.3. Summary

We summarize design features for neural EL models in Table 3. The mention encoders have made a shift to self-attention architectures and start using deep pre-trained models like BERT. The majority of studies still rely on external knowledge for the candidate generation step. There is a surge of models that tackle the domain adaptation problem in a zero-shot fashion. However, the task of zero-shot joint entity recognition and linking has not been addressed yet. It is shown in several works that the cross-encoder architecture is superior compared to models with separate mention and entity encoders. Many approaches rely on pre-trained entity representations, only few take advantage of a trainable entity encoder inside an EL model. The global context is widely used, but there are few recent studies that focus only on local EL.

Each column in Table 3 corresponds to a model feature. The **encoder arch.** column presents the architecture of the encoder of the neural entity linking model. It contains the following options:

- n/a – a model does not have a neural encoder for mentions / contexts. It can be a simplistic embedding averaging method or a feature-engineering approach.
- CNN – an encoder based on convolutional layers (usually with pooling).
- Tensor net. – an encoder that uses a tensor network.
- Atten. – means that an encoder uses an attention mechanism.
- GRU – an encoder based on a recurrent neural network and gated recurrent units [17].
- LSTM – an encoder based on a recurrent neural network and long short-term memory cells [43] (might be also bidirectional).
- FFNN – an encoder based on a simple feedforward neural network.
- ELMo – an encoder based on a pre-trained ELMo model [84].
- BERT – an encoder based on a pre-trained BERT model [22].

The **global** column shows whether a system uses a global solution (see Section 3.2.2). The **recognition** column refers to joint entity recognition and disambiguation models, where recognition and disambiguation of entities are performed collectively (Section 3.2.1). The **NIL prediction** column points out models that also label unlinkable mentions. The **entity embedding** column presents how entity representations are trained, where

- *joint architecture* means that the entity representations and the parameters of the disambiguation model are learned in the unified architecture;
- *pre-trained* denotes that the model uses entity representations pre-trained independently from the parameters for disambiguation and the context encoder parameters (Section 3.1.3).

In the **candidate generation** column, the candidate generation methods are noted (Section 3.1.1). It contains the following options:

Table 3

Features of neural EL models. Neural entity linking models compared according to their architectural features. (The footnotes in the table are explained in the text.)

	Encoder Type	Global	Recog-nition	NIL Prediction	Entity Embeddings	Candidate Generation	Zero-shot	Annotated Text Data	Cross-lingual
Sun et al. (2015) [102]	CNN+ Tensor net.				joint architecture	surface match dictionary		✘	
Francis-Landau et al. (2016) [29]	CNN	✘[4]			joint architecture	surface match prior		✘	
Fang et al. (2016) [25]	n/a	✘			pre-trained[2]	prior[1]		✘	
Yamada et al. (2016) [116]	n/a	✘			pre-trained[2]	prior or dictionary		✘	
Zwicklbauer et al. (2016) [125]	n/a	✘		✘	pre-trained[2]	surface match prior nearest neighbors		✘	
Tsai and Roth (2016) [104]	n/a	✘		✘	pre-trained[2]	prior		✘	✘
Nguyen et al. (2016) [77]	CNN	✘		✘	joint architecture	surface match prior		✘	
Cao et al. (2017) [11]	n/a	✘			pre-trained[2]	dictionary		in entity embedding	
Eshel et al. (2017) [24]	GRU+ Atten.				joint architecture	dictionary		✘	
Ganea and Hofmann (2017) [32]	Atten.	✘			pre-trained[2]	prior+ nearest neighbors		✘	
Moreno et al. (2017) [69]	n/a	✘[4]		✘	pre-trained[2]	surface match		✘	
Gupta et al. (2017) [38]	LSTM	✘[4]			joint architecture	prior	✘		
Sorokin and Gurevych (2018) [99]	CNN	✘	✘		pre-trained[2]	surface match		✘	
Shahbazi et al. (2018) [93]	Atten.	✘			pre-trained	prior		✘	
Le and Titov (2018) [54]	Atten.	✘			pre-trained	prior		✘	
Newman-Griffis et al. (2018) [75]	n/a				pre-trained[2]	dictionary			
Radhakrishnan et al. (2018) [87]	n/a	✘			pre-trained[2]	dictionary		✘	
Kolitsas et al. (2018) [51]	LSTM	✘	✘		pre-trained	prior		✘	
Sil et al. (2018) [98]	LSTM+ Tensor net.	✘[4]		✘	joint architecture	prior	✘[5]	✘	✘
Upadhyay et al. (2018) [106]	CNN				joint architecture	prior		✘	✘
Cao et al. (2018) [12]	FFNN	✘[4]			pre-trained[2]	prior		✘	
Raiman and Raiman (2018) [88]	n/a	✘			n/a	prior type classifier		✘	✘
Mueller and Durrett (2018) [71]	GRU+ Atten.+ CNN				joint architecture	dictionary		✘	
Shahbazi et al. (2019) [94]	ELMo				pre-trained[2]	prior or dictionary		✘	
Logeswaran et al. (2019) [62]	BERT				joint architecture	BM25	✘		
Gillick et al. (2019) [34]	FFNN				joint architecture	nearest neighbors	✘	in entity embedding	
Peters et al. (2019) [85][3]	BERT	✘[4]	✘	✘	pre-trained	prior		in entity embedding	
Le and Titov (2019) [55]	LSTM				joint architecture	surface match			
Le and Titov (2019) [56]	Atten.	✘			pre-trained	prior		in entity embedding	
Fang et al. (2019) [26]	LSTM	✘			pre-trained	dictionary		✘	
Martins et al. (2019) [65]	LSTM		✘	✘	pre-trained	dictionary		✘	
Yang et al. (2019) [118]	Atten. or CNN	✘			pre-trained	prior		✘	
Broscheit (2019) [9]	BERT		✘		n/a	n/a		✘	
Onoe and Durrett (2020) [79]	ELMo+ Atten.+ CNN				n/a	prior or dictionary		✘	
Wu et al. (2019) [114]	BERT				joint architecture	nearest neighbors	✘		
Yamada et al. (2020) [117]	BERT	✘			joint architecture	prior		✘	

Table 4
Evaluation datasets. Descriptive statistics of the evaluation datasets used in this survey to compare the models.

Corpus	Text Type	# of Docs	# of Mentions
AIDA-B [44]	News	231	4485
MSNBC [20]	News	20	656
AQUAINT [68]	News	50	727
ACE2004 [89]	News	36	257
CWEB [31, 37]	ClueWeb & Wikipedia	320	11154
WW [31, 37]	ClueWeb & Wikipedia	320	6821
TAC KBP 2010 [47]	News & Web	1013	1020[1]
TAC KBP 2015 Chinese [48]	News & Forums	166	11066
TAC KBP 2015 Spanish [48]	News & Forums	167	5822

[1] # of mention/entity pairs

– n/a – the solution presented by Broscheit [9] does not have an explicit candidate generation step;
– surface match – surface match heuristics;
– dictionary – a dictionary with supplementary aliases for entities;
– prior – filtering candidates with pre-calculated mention-entity prior probabilities;
– type classifier – Raiman and Raiman [88] filter candidates using a classifier for an automatically learned type system;
– BM25 – Logeswaran et al. [62] a variant of TF-IDF to measure similarity between a mention and a candidate entity based on description pages;
– nearest neighbors – the similarity between mention and entity representations is calculated, and entities that are nearest neighbors of mentions are retrieved as candidates. Wu et al. [114] train a supplementary model for this purpose.

The **zero-shot** column displays whether an EL system provides a zero-shot approach (see Section 3.2.3). The **annotated text data** column shows whether a model uses an annotation or not. *'In entity embedding'* denotes the models that do not use the annotated data for training, but the annotated data is used for training entity representations (see Section 3.2.3). The **cross-lingual** column refers to models, which provide cross-lingual EL solutions (Section 3.2.4).

Besides, the following superscript notations are used to denote specific features of methods shown as a note in the Table 3:

1. In classification, the prior is checked by a threshold. This can be considered as a candidate selection step.
2. In these works, the authors pre-train their own entity embedding models.
3. The authors provide EL as a subsystem of language modeling.
4. These solutions do not rely on global coherence but are marked as "global", because they use document-wide context or multiple mentions at once for resolving entity ambiguity.
5. Zero-shot in the sense of model adaptation to a new language using English annotated data, while the other zero-shot works solve the problem of model adaptation to a new domain without switching the language.

4. Evaluation

In this section, we present evaluation of the models on the entity linking and entity relatedness tasks over the commonly used datasets.

4.1. Entity Linking

4.1.1. Experimental Setup

The most widely-used datasets for evaluation of EL systems are: AIDA [44], TAC KBP 2010 [47], MSNBC [20], AQUAINT [68], ACE2004 [89], CWEB [31, 37], and WW [31, 37]. Among them, CWEB and WW are large datasets that are annotated automatically, while AIDA is also a large dataset, annotated manually [32]. The cross-lingual EL results are reported for the TAC KBP 2015 [48] Spanish (es) and Chinese (zh) datasets.

Some of the systems are evaluated using GERBIL [107], a benchmarking platform for entity recogni-

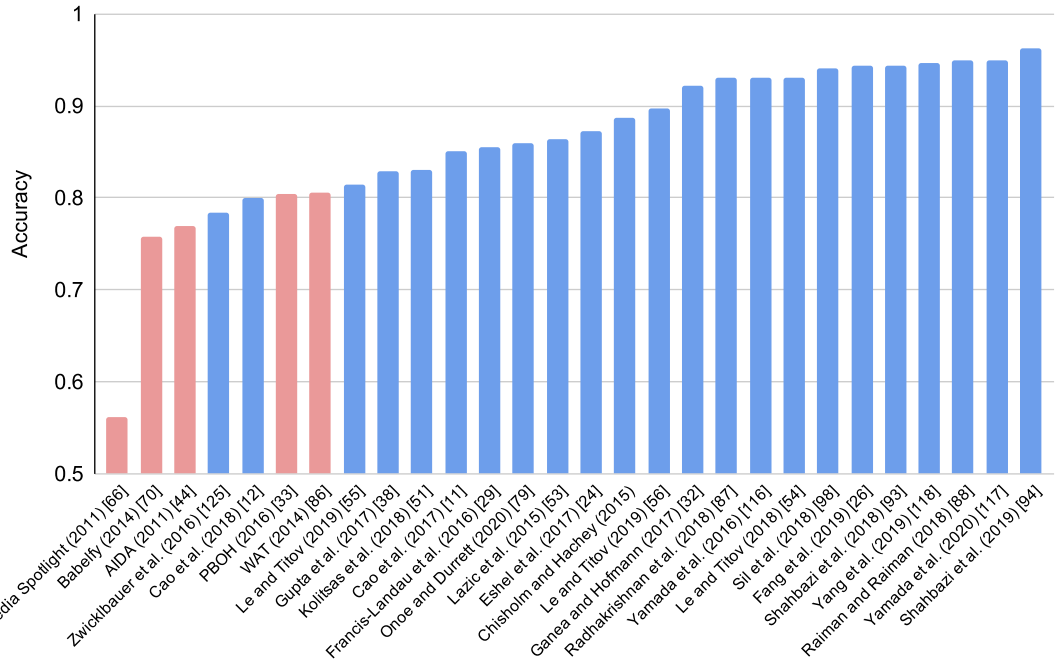

Fig. 8. **Entity disambiguation progress**. Performance of the best classic entity linking models (red) with the more recent neural models (blue) on the AIDA dataset shows a substantial improvement (over 15 points of accuracy).

tion, disambiguation and typing systems with different types of available experiments, e.g. entity disambiguation (D2KB), combination of entity recognition and disambiguation (A2KB). In Table 4, we present the topic or source feature and statistics of datasets as reported in [32], [34], [48] and overview file[10].

In this section, we present accuracy and micro F1 scores achieved by the EL systems. The results are reported based on two different evaluation settings. The inputs for the systems that perform ER and ED jointly are plain text, whereas the disambiguation systems have an access to the mention boundaries. We stated their results in separate tables since the scores for the joint models take into account also mistakes in entity recognition.

We also include some baseline systems to underline improvements of the neural based models over them. For each work, we present the best scores reported by the authors and, for baseline systems, the results are presented as reported in Kolitsas et al. [51] and Ganea and Hofmann [32].

[10]https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/tackbp2015_overview.pdf

4.1.2. Discussion of Results

We start our presentation of results of from the disambiguation only models (for which entity boundaries are already provided). Figure 8 shows how performance of the entity disambiguation models improved during the course of the last decade and how the best classic models correspond to the recent neural state-of-the-art models for entity linking. As one may observe the models based on deep learning substantially improve the performance pushing the state of the art by more than 15 points. AIDA is the most widely used dataset (but also one of the largest), but we also report results on other datasets in Table 5.

Among local models for disambiguation, the best results are reported by Shahbazi et al. [93] and Wu et al. [114]. It is worth noting that the latter model can be used in a zero-shot setting. Shahbazi et al. [94] has the best score on AIDA-B among other models. However, this is due to the use of less-ambiguous resource of Pershina et al. [83] for candidate generation, while many of other works use the YAGO-based resource provided by Ganea and Hofmann [32], which typically yields lower results.

The common trend is that the global models (those trying to disambiguate several entity occurrences at

Table 5

Entity disambiguation evaluation. Performance of neural entity disambiguation as compared to the selected classic models on common evaluation datasets.

	AIDA-B	**KBP'10**	**MSNBC**	**AQUAINT**	**ACE-2004**	**CWEB**	**WW**	**KBP'15 (es)**	**KBP'15 (zh)**
	Accuracy	Accuracy	Micro F1	Micro F1	Micro F1	Micro F1	Micro F1	Accuracy	Accuracy
Non-Neural Baseline Models									
DBpedia Spotlight (2011) [66]	0.561	-	0.421	0.518	0.539	-	-	-	-
AIDA (2011) [44]	0.770	-	0.746	0.571	0.798	-	-	-	-
Ratinov et al. (2011) [89]	-	-	0.750	0.830	0.820	0.562	0.672	-	-
WAT (2014) [86]	0.805	-	0.788	0.754	0.796	-	-	-	-
Babelfy (2014) [70]	0.758	-	0.762	0.704	0.619	-	-	-	-
Lazic et al. (2015) [53]	0.864	-	-	-	-	-	-	-	-
Chisholm and Hachey (2015) [15]	0.887	-	-	-	-	-	-	-	-
PBOH (2016) [33]	0.804	-	0.861	0.841	0.832	-	-	-	-
Neural Models									
Sun et al. (2015) [102]	-	0.839	-	-	-	-	-	-	-
Tsai and Roth (2016) [104]	-	-	-	-	-	-	-	0.824	0.851
Fang et al. (2016) [25]	-	0.889	0.755	0.852	0.808	-	-	-	-
Yamada et al. (2016) [116]	0.931	0.855	-	-	-	-	-	-	-
Zwicklbauer et al. (2016) [125]	0.784	-	0.911	0.842	0.907	-	-	-	-
Francis-Landau et al. (2016) [29]	0.855	-	-	-	0.899	-	-	-	-
Eshel et al. (2017) [24]	0.873	-	-	-	-	-	-	-	-
Ganea and Hofmann (2017) [32]	0.922	-	0.937	0.885	0.885	0.779	0.775	-	-
Gupta et al. (2017) [38]	0.829	-	-	-	0.907	-	-	-	-
Cao et al. (2017) [11]	0.85	-	-	-	-	-	-	-	-
Sil et al. (2018) [98]	0.940	0.874	-	-	-	-	-	0.823	0.844
Shahbazi et al. (2018) [93]	0.944	0.879	-	-	-	-	-	-	-
Kolitsas et al. (2018) [51]	0.831	-	0.864	0.832	0.855	-	-	-	-
Le and Titov (2018) [54]	0.931	-	0.939	0.884	0.899	0.775	0.780	-	-
Radhakrishnan et al. (2018) [87]	0.930	0.896	-	-	-	-	-	-	-
Cao et al. (2018) [12]	0.800	0.910	-	0.870	0.880	-	0.860	-	-
Raiman and Raiman (2018) [88]	0.949	0.909	-	-	-	-	-	-	-
Upadhyay et al. (2018) [106]	-	-	-	-	-	-	-	0.844	0.860
Gillick et al. (2019) [34]	-	0.870	-	-	-	-	-	-	-
Le and Titov (2019) [55]	0.815	-	-	-	-	-	-	-	-
Le and Titov (2019) [56]	0.897	-	0.922	0.907	0.881	0.782	0.817	-	-
Fang et al. (2019) [26]	0.943	-	0.928	0.875	0.912	0.785	0.828	-	-
Yang et al. (2019) [118]	0.946	-	0.946	0.883	0.901	0.756	0.788	-	-
Shahbazi et al. (2019) [94]	0.962	0.883	-	-	-	-	-	-	-
Onoe and Durrett (2020) [79]	0.859	-	-	-	-	-	-	-	-
Wu et al. (2019) [114]	-	0.940	-	-	-	-	-	-	-
Yamada et al. (2020) [117]	0.950	-	0.963	0.935	0.919	0.789	0.892	-	-

once) outperform the local ones (relying on a single context). The global model of Yamada et al. [117] produce results that are consistently better compared to other solutions including the results of Shahbazi et al. [94] reported for the YAGO-based resource. The performance improvements are explained by the authors by the novel masked entity prediction objective that helps to fine-tune pre-trained BERT for producing contextualized entity embeddings and the multi-step global disambiguation algorithm.

Table 6 presents results of the joint ER and ED models. Only a fraction of the models presented in above is capable of performing both entity recognition and disambiguation thus a much shorter list of results. Among the joint recognition and disambiguation solutions, the leadership is owned by Kolitsas et al. [51]. This system and others that solve also the ER task fall behind the disambiguation-only systems since they rely on noisy mention boundaries produced by themselves. As one may observe, in the joint setting the neural models also

Table 6
Evaluation of joint NER-EL models. Results for joint entity recognition and entity disambiguation evaluation on AIDA-B and MSNBC datasets.

	AIDA-B	MSNBC
	Micro F1	Micro F1
Non-Neural Baseline Models		
DBpedia Spotlight (2011) [66]	0.578	0.406
AIDA (2011) [44]	0.728	0.651
WAT (2014) [86]	0.730	0.645
Babelfy (2014) [70]	0.485	0.397
Neural Models		
Kolitsas et al. (2018) [51]	**0.824**	**0.724**
Martins et al. (2019) [65]	0.819	-
Peters et al. (2019) [85]	0.744	-

substantially (up to 10 points) outperform the classic models, showing the state-of-the-art results.

4.2. Entity Relatedness

4.2.1. Experimental Setup

The evaluation data is provided by Ceccarelli et al. [13] using the dataset of Hoffart et al. [44]. It is in the form of queries, where the first entity is accepted as correctly linked and the second entity is the candidate.

Entity representation performance is evaluated through an entity relatedness task. Namely, the task is to rank entities for the target one, which is performed using cosine similarity of entity representations except for two studies: Milne and Witten [68] introduce a Wikipedia hyperlink-based measure, known as WLM, and recently El Vaigh et al. [23] provide a weighted semantic relatedness measure.

The evaluation of ranking quality is performed with a normalized discounted cumulative gain (nDCG) [46] and a mean average precision (MAP) [121]. nDCG is commonly used in information retrieval and provides a fair evaluation by measuring the position impressiveness. Similarly, MAP measures how accurately the model performs for the target entity.

4.2.2. Discussion of Results

In Table 7, the entity relatedness scores are reported. The highest score is reported by Huang et al. [45] since they specifically train the embeddings based on a pairwise entity score function in a supervised way. Instead, in other models, entity embeddings are mostly trained in a joint space with word embeddings based on the relatedness between mentions and words. Therefore, the results of Huang et al. [45] are distinctively higher. Ganea and Hofmann [32] and Cao et al. [11] achieve good scores, and recently, Shi et al. [97] also present an excellent performance using a large amount of data sources based on textual and KB information.

5. Applications of Entity Linking

In this section, we first give a brief overview of the "classical" established applications of the entity linking technology and then discuss the recently emerged use-case specific to the neural entity linking based on injection of these models as a part of a larger neural network, e.g. a neural language model.

5.1. Established Applications

Any entity linking model can be used as a component to solve downstream tasks, e.g. semantic parsing and question answering [6, 119], information extraction [73], or biomedical text processing [58], as mentioned in the introductory part of this survey.

EL is a typical building block [103] for biomedical and clinical text process systems like electronic health record mining tools [52], literature search engines [59], and question answering systems [58]. For example, COVIDASK [58], recently presented real-time question answering system that helps researchers to retrieve information related to coronavirus, uses BioSyn [103] EL model for linking in COVID-19 articles objects like drugs, symptoms, disease mentions to biomedical ontologies like Medical Subject Headings (MeSH)[11].

Entity linking in information extraction systems is also widely used for relation extraction (extraction of relations between mentions or entities such as "child-of", "politician-from", "born-in", etc.) For example, PATTY [73] uses entity linking to build a resource of relational patterns by imposing a semantically typed structure. It extracts patterns from an input text corpus and links entity mentions to KB entries Detecting and disambiguating entities during relation extraction enable also some other applications like KB population, fact verification, machine reading [64, 96].

Question answering systems typically leverage semantic parsing to convert a question into a formal meaning representation (e.g., logical form) that can

[11] https://www.nlm.nih.gov/mesh/meshhome.html

Table 7
Entity relatedness evaluation. Reported results for entity relatedness evaluation on the dataset of Ceccarelli et al. [13] .

	nDCG@1	**nDCG@5**	**nDCG@10**	**MAP**
Milne and Witten (2008) [68]	0.540	0.520	0.550	0.480
Huang et al. (2015) [45]	0.810	0.730	0.740	0.680
Yamada et al. (2016) [116]	0.590	0.560	0.590	0.520
Ganea and Hofmann (2017) [32]	0.632	0.609	0.641	0.578
Cao et al. (2017) [11]	0.613	0.613	0.654	0.582
El Vaigh et al. (2019) [23]	0.690	0.640	0.580	-
Shi et al. (2020) [97]	0.680	0.814	0.820	-

used for directly querying a KB [119]. EL helps to restrict the search space of a query. For example, Yih et al. [119] apply EL for pruning the search space resulting in simplification of the semantic matching. For the query: "Who first voiced Meg on Family Guy?", after linking "Meg" and "Family Guy" to entities in a KB, the task becomes to resolve the predicates to the "Family Guy (the TV show)" entry rather than all entries in the KB.

5.2. Novel Applications: Neural Entity Linking for Training of Neural Language Models

In addition to aforementioned applications, neural EL models have unlocked the new category of applications that have not been available for the classical machine learning methods. Namely neural models allow the integration of an entire entity linking system inside a larger neural network such as BERT. As they both just some neural networks, such kind of integration becomes possible. After integrating an entity linker in another model architecture, we also expand the training objective with an additional EL-related task and train parameters of all neural components jointly:

$$\mathcal{L}_{\text{JOINT}} = \mathcal{L}_{\text{BERT}} + \mathcal{L}_{\text{EL-related}} . \tag{20}$$

Neural entity linkers can be integrated in any other networks. The main novel trend is the use of EL information for representation learning. Namely, several studies have shown that contextual word representations could benefit from information stored in KBs by incorporating EL into deep models for transfer learning.

KnowBERT [85] injects one or several entity linkers between top layers of the BERT architecture and optimizes the whole network for multiple tasks: the masked language model (MLM) task and next sentence prediction (NSP) from the original BERT model, as well as EL:

$$\mathcal{L}_{\text{BERT}} = \mathcal{L}_{\text{NSP}} + \mathcal{L}_{\text{MLM}}. \tag{21}$$

$$\mathcal{L}_{\text{KnowBert}} = \mathcal{L}_{\text{NSP}} + \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{EL}} . \tag{22}$$

The authors adopt the general end-to-end EL architecture of [51] but use only the local context for disambiguation and use an encoder based on self-attention over the representations generated by underlying BERT layers. If the EL subsystem detects an entity mention in a given sentence, corresponding pre-built entity representations of candidates are utilized for calculating the updated contextual word representations generated on the current BERT layer. These representations are used as input in a subsequent layer and can also be modified by a subsequent EL subsystem. Experiments with two EL subsystems based on Wikidata and WordNet show that presented modifications in KnowBERT help it to slightly surpass other deep pre-trained language models in tasks of relationship extraction, WSD, and entity typing.

ERNIE [123] expands the BERT [22] architecture with a knowledgeable encoder (K-Encoder), which fuses contextualized word representations obtained from the underlying self-attention network with entity representations from a pre-trained TransE model [8]. EL in this study is performed by an external tool TAGME [28]. For model pre-training, in addition to the MLM task, the authors introduce the task of restoring randomly masked entities in a given sequence keeping the rest of the entities and tokens. They refer to this procedure as a denoising entity auto-encoder (dEA):

$$\mathcal{L}_{\text{ERNIE}} = \mathcal{L}_{\text{NSP}} + \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{dEA}} . \tag{23}$$

Using English Wikipedia and Wikidata as training data, the authors show that introduced modifications

provide performance gains in entity typing, relation classification, and several GLUE tasks [111].

Wang et al. [113] train a disambiguation network using the composition of two losses: regular MLM and a Knowledge Embedding (KE) loss based on the TransE [8] objective for encoding graph structures:

$$\mathcal{L}_{\text{KEPLER}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{KE}}. \quad (24)$$

In KE loss, representations of entities are obtained from their textual descriptions encoded with a self-attention network [61], and representations of relations are trainable vectors. The network is trained on a new dataset of entity-relation-entity triplets with descriptions gathered from Wikipedia and Wikidata. Although the system exhibits a significant drop in performance on general NLP benchmarks such as GLUE [111], it shows increased performance on a wide range of KB-related tasks such as TACRED [122], FewRel [39], and OpenEntity [16].

6. Conclusion

In this survey, we have analyzed recently proposed neural entity linking models, which generally perform the task with higher accuracy than classical methods scores. We provide a generic neural entity linking architecture, which are applicable for most of the neural EL systems, including components e.g. candidate generation, entity ranking, mention and entity encoding. The various modifications of general architecture are grouped into four common directions: (1) joint entity recognition and linking models, (2) global entity linking models, (3) domain-independent approaches including zero-shot and distant supervision methods, and (4) cross-lingual techniques. The taxonomy figures and feature tables are provided to explain categorization and to show which prominent features are used in each method.

The majority of studies still rely on external knowledge for the candidate generation step. The mention encoders have made a shift from convolutional and recurrent models to self-attention architectures and start using pre-trained contextual language models like BERT. There is a surge of models that tackle the problem of adapting a model trained on one domain to another domain in a zero-shot fashion. These approaches do not need any annotated data in the target domain, but only descriptions of entities from this domain to make such adaptation. It is shown in several works that the cross-encoder architecture is superior as compared to models with separate mention and entity encoders. Many approaches rely on pre-trained entity representations, only few take advantage of a trainable entity encoder inside an EL model. The global context is widely used, but there are few recent studies that focus only on local EL.

Among the joint recognition and disambiguation solutions, the leadership is still owned by Kolitsas et al. [51]. Among published local models for disambiguation, the best result is reported by Wu et al. [114]. It is worth noting that this model can be used in a zero-shot setting. The global models outperform the local ones. The work of Yamada et al. [117] reports results that are consistently better in comparison to other solutions. The performance improvements are attributed to the masked entity prediction mechanism for entity embedding and to the usage of the pre-trained model based on BERT with a multi-step global scoring function.

7. Future Directions

We identify four promising directions of future work in entity linking listed below:

1. **End-to-end models featuring the candidate generation step**: Candidate generation step requires to collect information from a large amount of data, as described in the Section 3.1.1. Although the models could create a domain-independent architecture, they are still based on data for candidate generator. Therefore, the possible direction would be to handle the candidate generation step without the requisite of external data or directly eliminate this step. There are some studies, which use either the representations [34, 114] or BM25 scores computed from entity descriptions [62] to find out candidates. However, these models do not provide completely end-to-end solutions. Thus, future approaches could tackle the challenge of a complete end-to-end solution without a candidate generator.
2. **Further development of zero-shot approaches to address emerging entities**: We also expect that zero-shot EL will rapidly evolve engaging other features like global coherence across all entities in a document, NIL prediction, joining ER and EL steps together, or providing completely end-to-end solutions. The latter would be an es-

pecially challenging task but also a fascinating research direction.

3. **More use-cases of EL-enriched language models**: Some studies [85, 113, 123] have shown improvements over contextual language models by including knowledge stored in KBs. They incorporate entity linking into these deep models to use information in KBs. In the future work, more use-cases are expected to enhance language models by using entity linking. The enriched representations would be used in downstream tasks, enabling improvements over there.
4. **Integration of EL loss in more neural models**: It may be interesting to integrate EL loss in other neural models distinct from the language models, but in the similar fashion as the models described in Section 5.2. Due to the fact that an end-to-end EL model is also just a neural network, such integration with other networks is technically straightforwards and may be useful to inject information about entities contained in an EL model into other possibly specialized architectures.

Acknowledgements

The work was partially supported by a Deutscher Akademischer Austauschdienst (DAAD) doctoral stipend and the DFG-funded JOIN-T project BI 1544/4. The work of Artem Shelmanov in the current study (preparation of sections related to application of entity linking to neural language models, entity ranking, context-mention encoding, and overall harmonization of the text and results) is supported by the Russian Science Foundation (project 20-11-20166). Finally, this work was partially supported by the joint MTS-Skoltech laboratory.

References

[1] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881, 2020. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8999622.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3540762973.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, San-Diego, California, USA, 2015. URL http://arxiv.org/abs/1409.0473.

[4] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, Sofia, Bulgaria, 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-2322/.

[5] Debayan Banerjee, Debanjan Chaudhuri, Mohnish Dubey, and Jens Lehmann. Pnel: Pointer network based end-to-end entity linking over knowledge graphs. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 21–38, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62419-4.

[6] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, 2014. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P14-1133.

[7] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581026. URL https://doi.org/10.1145/1376616.1376746.

[8] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, Stateline, Nevada, USA, 2013. URL https://papers.nips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[9] Samuel Broscheit. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K19-1063.

[10] Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E06-1002.

[11] Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1623–1633, Vancouver, Canada, 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P17-1149.

[12] Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. Neural collective entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686, Santa Fe, New Mexico, USA, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1057.

[13] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Learning relatedness measures for entity linking. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 139–148, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. URL http://doi.acm.org/10.1145/2505515.2505711.

[14] Xiao Cheng and Dan Roth. Relational inference for Wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Seattle, Washington, USA, 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1184.

[15] Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015. URL https://www.aclweb.org/anthology/Q15-1011.

[16] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1009.

[17] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, Montréal, Canada, 2014. URL https://arxiv.org/abs/1412.3555.

[18] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[19] Ryan Cotterell and Kevin Duh. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 91–96, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-2016.

[20] Silviu Cucerzan. 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)*, pages 708–716, Prague, Czech Republic, 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D07-1074/.

[21] Silviu Cucerzan. TAC entity linking by performing full-document entity extraction and disambiguation. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011*. NIST, 2011. URL http://www.nist.gov/tac/publications/2011/participant.papers/MS_MLI.proceedings.pdf.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1423.

[23] Cheikh Brahim El Vaigh, François Goasdoué, Guillaume Gravier, and Pascale Sébillot. Using knowledge base semantics in context-aware entity linking. In *Proceedings of the ACM Symposium on Document Engineering 2019*, DocEng '19, New York, NY, USA, 2019. ACM. ISBN 9781450368872. . URL https://doi.org/10.1145/3342558.3345393.

[24] Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. Named entity disambiguation for noisy text. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 58–68, Vancouver, Canada, 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K17-1008.

[25] Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 260–269, Berlin, Germany, 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K16-1026.

[26] Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. Joint entity linking with deep reinforcement learning. In *The World Wide Web Conference*, WWW '19, pages 438–447, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6674-8. URL http://doi.acm.org/10.1145/3308558.3313517.

[27] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[28] Paolo Ferragina and Ugo Scaiella. TAGME: On-the-Fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1625–1628, New York, NY, USA, 2010. ACM. ISBN 9781450300995. URL https://doi.org/10.1145/1871437.1871689.

[29] Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261, San Diego, California, USA, 2016. URL https://www.aclweb.org/anthology/N16-1150.

[30] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Reject option with multiple thresholds. *Pattern recognition*, 33(12):2099–2101, 2000.

[31] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. FACC1: Freebase annotation of ClueWeb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0), 2013. Note: http://lemurproject.org/clueweb09/.

[32] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1277.

[33] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 927–938, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4143-1. URL https://doi.org/10.1145/2872427.2882988.

[34] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K19-1049.

[35] Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, Berlin, Germany, 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P16-1059.

[36] Archana Goyal, Vishal Gupta, and Manish Kumar. Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, 29:21–43, 2018. URL https://doi.org/10.1016/j.cosrev.2018.06.001.

[37] Zhaochen Guo and Denilson Barbosa. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4): 459 – 479, 2018. URL https://content.iospress.com/articles/semantic-web/sw273.

[38] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark, 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1284.

[39] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1514.

[40] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34, Sofia, Bulgaria, 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2006.

[41] Martin E Hellman. The nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, 6(3):179–185, 1970.

[42] Radu Herbei and Marten H Wegkamp. Classification with reject option. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, pages 709–721, 2006.

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

[44] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792. Association for Computational Linguistics, 2011. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145521.

[45] Hongzhao Huang, Larry Heck, and Heng Ji. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*, 2015. URL https://arxiv.org/abs/1504.07678.

[46] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20 (4):422–446, October 2002. ISSN 1046-8188. URL https://doi.org/10.1145/582415.582418.

[47] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the TAC 2010 knowledge base population track. In *Third Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA, 2010. URL https://blender.cs.illinois.edu/paper/kbp2010overview.pdf.

[48] Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. Overview of TAC-KBP2015 tri-lingual entity discovery and linking. In *Proceedings of the 2015 Text Analysis Conference, TAC 2015*, pages 16–17, Gaithersburg, Maryland, USA, 2015. NIST. URL https://tac.nist.gov/publications/2015/additional.papers/TAC2015.KBP_Trilingual_Entity_Discovery_and_Linking_overview.proceedings.pdf.

[49] Karen Spärck Jones, Shelia Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments part 2. *Information Processing & Management*, 36(6):809–840, 2000. ISSN 0306-4573. . URL https://doi.org/10.1016/S0306-4573(00)00016-9.

[50] Rijula Kar, Susmija Reddy, Sourangshu Bhattacharya, Anirban Dasgupta, and Soumen Chakrabarti. Task-specific representation learning for web-scale entity disambiguation. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5812–5819, New Orleans, Louisiana, USA, 2018. AAAI Press. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewFile/17281/16144.

[51] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K18-1050.

[52] Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A Folarin, Angus Roberts, Rebecca Bendayan, Mark P Richardson, Robert Stewart, Anoop D Shah, Wai Keong Wong, Zina Ibrahim, James T Teo, and Richard JB Dobson. Multi-domain clinical natural language processing with medcat: the medical concept annotation toolkit, 2020.

[53] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515, 2015. URL https://www.aclweb.org/anthology/Q15-1036.

[54] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1148.

[55] Phong Le and Ivan Titov. Distant learning for entity linking with automatic noise detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4081–4090, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1400.

[56] Phong Le and Ivan Titov. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1187.

[57] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, Atlanta, USA, 2013. JMLR. URL http://deeplearning.net/wp-content/uploads/2013/03/pseudo_label_final.pdf.

[58] Jinhyuk Lee, Sean S. Yi, Minbyul Jeong, Mujeen Sung, Won-Jin Yoon, Yonghwa Choi, Miyoung Ko, and Jaewoo Kang. Answering questions on COVID-19 in real-time. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online, December 2020. Association for Computational Linguistics. . URL https://www.aclweb.org/anthology/2020.nlpcovid19-2.1.

[59] Sunwon Lee, Donghyeon Kim, Kyubum Lee, Jaehoon Choi, Seongsoon Kim, Minji Jeon, Sangrak Lim, Donghee Choi, Sunkyu Kim, Aik-Choon Tan, and Jaewoo Kang. Best: Next-generation biomedical entity search tool for knowledge discovery from biomedical literature. *PLOS ONE*, 11(10):1–16, 10 2016. . URL https://doi.org/10.1371/journal.pone.0164680.

[60] Xiao Ling, Sameer Singh, and Daniel S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015. URL https://www.aclweb.org/anthology/Q15-1023/.

[61] 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. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

[62] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1335.

[63] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal, 2015. URL https://www.aclweb.org/anthology/D15-1104/.

[64] José L. Martínez-Rodríguez, A. Hogan, and I. López-Arévalo. Information extraction meets the semantic web: A survey. *Semantic Web*, 11(2):255–335, 2020.

[65] Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-2026.

[66] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0621-8. URL http://doi.acm.org/10.1145/2063518.2063519.

[67] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc. URL https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

[68] David Milne and Ian H. Witten. Learning to link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. URL http://doi.acm.org/10.1145/1458082.1458150.

[69] Jose G. Moreno, Romaric Besançon, Romain Beaumont, Eva D'hondt, Anne-Laure Ligozat, Sophie Rosset, Xavier Tannier, and Brigitte Grau. Combining word and entity embeddings for entity linking. In *Extended Semantic Web Conference (1)*, volume 10249 of *Lecture Notes in Computer Science*, pages 337–352, 2017. URL https://perso.limsi.fr/bg/fichiers/2017/combining-word-entity-eswc2017.pdf.

[70] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014. URL https://www.aclweb.org/anthology/Q14-1019/.

[71] David Mueller and Greg Durrett. Effective use of context in noisy entity linking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1029, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1126.

[72] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007. URL https://nlp.cs.nyu.edu/sekine/papers/li07.pdf.

[73] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, page 1135–1145, USA, 2012. Association for Computational Linguistics.

[74] Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, 2009. ISSN 0360-0300. URL http://doi.acm.org/10.1145/1459352.1459355.

[75] Denis Newman-Griffis, Albert M. Lai, and Eric Fosler-Lussier. Jointly embedding entities and text with distant supervision. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 195–206, Melbourne, Australia, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-3026.

[76] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-NERD: joint named entity recognition and disambiguation with rich linguistic features. *Transactions of the Association for Computational Linguistics*, 4:215–229, 2016. URL https://www.aclweb.org/anthology/Q16-1016/.

[77] Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. Joint learning of local and global features for entity linking via neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2310–2320, Osaka, Japan, 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1218.

[78] Italo L. Oliveira, Renato Fileto, René Speck, Luís P.F. Garcia, Diego Moussallem, and Jens Lehmann. Towards holistic entity linking: Survey and directions. *Information Systems*, 95:101624, 2021. ISSN 0306-4379. URL http://www.sciencedirect.com/science/article/pii/S0306437920300958.

[79] Yasumasa Onoe and Greg Durrett. Fine-grained entity typing for domain independent entity linking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8576–8583, New York, NY, USA, 2020. AAAI Press. URL https://arxiv.org/abs/1909.05780.

[80] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

[81] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada, 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P17-1178/.

[82] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9. . URL http://doi.acm.org/10.1145/2623330.2623732.

[83] Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243, Denver, Colorado, USA, 2015. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N15-1026.

[84] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2227–2237, New Orleans, Louisiana, USA, 2018. AAAI Press. URL https://arxiv.org/abs/1802.05365.

[85] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. 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)*, pages 43–54, Hong Kong, China, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1005.

[86] Francesco Piccinno and Paolo Ferragina. From tagme to wat: A new entity annotator. In *Proceedings of the First International Workshop on Entity Recognition \& Disambiguation*, ERD '14, pages 55 – 62, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330237. URL https://doi.org/10.1145/2633211.2634350.

[87] Priya Radhakrishnan, Partha Talukdar, and Vasudeva Varma. ELDEN: Improved entity linking using densified knowledge graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1844–1853, New Orleans, Louisiana, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N18-1167.

[88] Jonathan Raiman and Olivier Raiman. Deeptype: Multilingual entity linking by neural type system evolution. In *AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA., 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17148.

[89] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Portland, Oregon, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002642.

[90] Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime Carbonell. Zero-shot neural transfer for cross-lingual entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6924–6931, Honolulu, Hawaii, USA, 2019. URL https://ojs.aaai.org/index.php/AAAI/article/download/4670/4548.

[91] Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. Benchmarking the extraction and disambiguation of named entities on the Semantic Web. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4593–4600, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/176_Paper.pdf.

[92] Özge Sevgili, Alexander Panchenko, and Chris Biemann. Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 315–322, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-2044/.

[93] Hamed Shahbazi, Xiaoli Fern, Reza Ghaeini, Chao Ma, Rasha Mohammad Obeidat, and Prasad Tadepalli. Joint neural entity disambiguation with output space search. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2170–2180, 2018. URL https://www.aclweb.org/anthology/C18-1184/.

[94] Hamed Shahbazi, Xiaoli Z Fern, Reza Ghaeini, Rasha Obeidat, and Prasad Tadepalli. Entity-aware elmo: Learning contextual entity representation for entity disambiguation. *arXiv preprint arXiv:1908.05762*, 2019. URL https://arxiv.org/abs/1908.05762.

[95] Rahul Sharnagat. Named entity recognition: A literature survey. *Center For Indian Language Technology*, 2014. URL http://www.cfilt.iitb.ac.in/resources/surveys/rahul-ner-survey.pdf.

[96] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *Transactions on Knowledge & Data Engineering*, 27(2):443–460, 2015. URL http://www.computer.org/csdl/trans/tk/2015/02/06823700-abs.html.

[97] Wei Shi, Siyuan Zhang, Zhiwei Zhang, Hong Cheng, and Jeffrey Xu Yu. Joint embedding in named entity linking on sentence level. *arXiv preprint arXiv:2002.04936*, 2020. URL https://arxiv.org/abs/2002.04936.

[98] Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. Neural cross-lingual entity linking. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA., 2018. AAAI Press. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16501/16101.

[99] Daniil Sorokin and Iryna Gurevych. Mixing context granularities for improved entity linking on question answering data across entity categories. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 65–75, 2018. URL https://www.aclweb.org/anthology/S18-2007/.

[100] Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3168–3175, Istanbul, Turkey, 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/266_Paper.pdf.

[101] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA, 2007. ACM. ISBN 9781595936547. . URL https://doi.org/10.1145/1242572.1242667.

[102] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1333–1339. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL http://dl.acm.org/citation.cfm?id=2832415.2832435.

[103] Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. Biomedical entity representations with synonym marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3641–3650, Online, July 2020. Association for Computational Linguistics. . URL https://www.aclweb.org/anthology/2020.acl-main.335.

[104] Chen-Tse Tsai and Dan Roth. Cross-lingual Wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598, San Diego, California, USA, 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N16-1072.

[105] Chen-Tse Tsai and Dan Roth. Learning better name translation for cross-lingual Wikification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, 2018. AAAI Press. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17318/16109.

[106] Shyam Upadhyay, Nitish Gupta, and Dan Roth. Joint multilingual supervision for cross-lingual entity linking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2495, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1270.

[107] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1133–1143, Florence, Italy, 2015. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-3469-3. URL https://doi.org/10.1145/2736277.2741626.

[108] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. URL https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[109] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5866-pointer-networks.pdf.

[110] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014. ISSN 0001-0782. URL https://doi.org/10.1145/2629489.

[111] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-5446.

[112] Han Wang, Jin Guang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji. Language and domain independent entity linking with quantified collective validation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 695–704, Lisbon, Portugal, 2015. Asso-

ciation for Computational Linguistics. URL https://www.aclweb.org/anthology/D15-1081.

[113] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *arXiv preprint arXiv:1911.06136*, 2019. URL https://arxiv.org/abs/1911.06136.

[114] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*, 2019. URL https://arxiv.org/abs/1911.03814.

[115] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, NM, USA, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1182.

[116] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/K16-1025.

[117] Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. Global entity disambiguation with pretrained contextualized embeddings of words and entities. *arXiv preprint arXiv:1909.00426v2*, 2020. URL https://arxiv.org/abs/1909.00426v2.

[118] Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. Learning dynamic context augmentation for global entity linking. 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)*, pages 271–281, Hong Kong, China, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1026.

[119] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, 2015. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P15-1128/.

[120] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018. URL https://ieeexplore.ieee.org/document/8416973.

[121] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 271–278, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. URL https://doi.org/10.1145/1277741.1277790.

[122] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, Copenhagen, Denmark, 2017. ACL. URL https://nlp.stanford.edu/pubs/zhang2017tacred.pdf.

[123] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1139.

[124] Shuyan Zhou, Shruti Rijhwani, John Wieting, Jaime Carbonell, and Graham Neubig. Improving candidate generation for low-resource cross-lingual entity linking. *Transactions of the Association for Computational Linguistics*, 8:109–124, 2020. URL https://www.mitpressjournals.org/doi/full/10.1162/tacl_a_00303.

[125] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 425–434, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. URL http://doi.acm.org/10.1145/2911451.2911535.