A Sequence Learning Method for Domain-Specific Entity Linking

Emrah Inan
Department of Computer Engineering
Ege University
emrah.inan@ege.edu.tr

Oguz Dikenelli
Department of Computer Engineering
Ege University
oguz.dikenelli@ege.edu.tr

Abstract

Recent collective Entity Linking studies usually promote global coherence of all the mapped entities in the same document by using semantic embeddings and graph-based approaches. Although graph-based approaches are shown to achieve remarkable results, they are computationally expensive for general datasets. Also, semantic embeddings only indicate relatedness between entity pairs without considering sequences. In this paper, we address these problems by introducing a two-fold neural model. First, we match easy mention-entity pairs and using the domain information of this pair to filter candidate entities of closer mentions. Second, we resolve more ambiguous pairs using bidirectional Long Short-Term Memory and CRF models for the entity disambiguation. Our proposed system outperforms state-of-the-art systems on the generated domain-specific evaluation dataset.

1 Introduction

Entity Linking is the task of matching ambiguous mentions in a text to the corresponding entities in the given knowledge base. The output of the entity linking is a crucial step for many tasks, including relation extraction (Weston et al., 2013), link prediction (Nickel et al., 2015) and knowledge graph completion (Minervini et al., 2016). The main challenge is to disambiguate candidate entities for the given mentions. For instance, it requires to resolve the mention *Wicker Park* in the following text *"Wicker Park is a 2004 American psychological drama mystery film directed by Paul McGuigan and starring Josh Hartnett..."* to the referent entity *Wicker_Park_(film)*[1] in DBpedia. But the mention *Wicker Park* has three different candidate entities as indicated in the Wikipedia disambiguation page of this mention.

The key step for entity disambiguation is the similarity computation between mention-entity and entity-entity pairs. Early studies focused on modeling the similarity between local context that computes the similarity between mention context and relevant candidate entities (Bunescu and Paşca, 2006; Mihalcea and Csomai, 2007). Recent state-of-the-art methods consider global coherence that is the relatedness between all candidate entities in the same document (Milne and Witten, 2008; Kulkarni et al., 2009; Ratinov et al., 2011). These methods depend on well-defined link structures as seen in Wikipedia to compute global coherence. After the emergence of word embeddings (Mikolov et al., 2013), it facilitates to produce more generalized coherence computations without using hand-crafted features. Hence, the dependency of well-defined knowledge bases has decreased and knowledge base agnostic approaches become revealed (Zwicklbauer et al., 2016). Most recent deep learning approaches have been presented as a way to support better generalization for the similarity measurement of context, mention and entity (Sun et al., 2015). Also, mentions and entities are combined into the same continuous vector space for the entity disambiguation (Yamada et al., 2016). From a different perspective, the entity disambiguation should be transformed into as a sequence learning task to capture more generalized semantics between candidate entities and also mentions.

In this paper, we generate RDF embeddings (Ristoski and Paulheim, 2016) as the input of a sequence learning model using bidirectional

[1] http://dbpedia.org/resource/Wicker_Park_(film)

Proceedings of the Seventh Named Entities Workshop, pages 14–21
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

Long Short-Term Memory (LSTM) (Graves and Schmidhuber, 2005). Then, we perform Conditional Random Field (CRF) to match the best mention-entity pairs. LSTM networks are not suitable for large entity vocabularies since English DBpedia contains more than 5M entities. To reduce the size of these vocabularies, our study employs the two-fold method. First, we match easy mention-entity pairs in which each mention contains only one candidate entity. Similar to AIDA-light study (Hoffart et al., 2013) we identify the domain of the given text and the size of candidate entities are reduced to reasonable dimensions for the detected domain. The contributions of our study can be summarized as below:

- Our study proposes a novel algorithm that first disambiguates easy mention-entity pairs for a specific domain. Thereafter, it applies CRF model to link more ambiguous entities.
- Our study provides a sequence learning model like a translation task in which a sequence of mentions will be translated into a sequence of referent entities in the domain-specific knowledge base.

Our method employs one of prominent Named Entity Recognition approaches (Lample et al., 2016) to perform a domain-specific Entity Linking. We aim to model the topical coherence of the mention-entity pairs in terms of a sequence labeling task. We conduct the experimental setup using the well-known evaluation framework called GERBIL (Usbeck et al., 2015) to compare our study with the state-of-the-art Entity Linking systems. The rest of this paper is organized as follows: In Section 2, it gives an overview of related work. In Section 3, the sequence learning method is proposed for a specific domain. Section 4 presents the experiments are for the selected approaches on the prepared evaluation dataset. We conclude our study and highlight the research questions in Section 5.

2 Related Work

Common trends in Entity Linking employs the global coherence to identify entities. Traditional studies mainly depend on Wikipedia link structure to disambiguate entities (Milne and Witten, 2008; Cucerzan, 2007). Also, TAGME (Ferragina and Scaiella, 2010) exploits Wikipedia anchor link texts for the mention detection and aims on-the-fly annotation of short texts using agreement approach based on Wikipedia link structure. Moreover, these approaches focus on global coherence approaches that emphasize the consistency of all mention-entity pairs in the given text. AIDA-light (Nguyen et al., 2014) considers global coherence to disambiguate the entities and exploits YAGO2 (Hoffart et al., 2013) and Wikipedia domain hierarchy. DBpedia Spotlight (Mendes et al., 2011), Babelfy (Moro et al., 2014) and WAT (Piccinno and Ferragina, 2014) have achieved remarkable results while using open domain knowledge bases. However, these type of systems tends to work inherently worse in domain intensive environment. These studies generally exploit hand-crafted features to represent mentions and entities. Methods based on word embeddings (Mikolov et al., 2013) are recently popular including continuous word vectors representations from large unstructured texts. Doser (Zwicklbauer et al., 2016) leverages word embeddings as the input of Personalized-PageRank algorithm to disambiguate candidate entities.

Most recently, neural models have been presented as a way to promote better generalization without hand-crafted features. Sun et al. (2015) presents a neural network approach using mention, entity and context embeddings in a unified way. They leverage a Convolutional Neural Network model for context representation and consider positions of context words around mentions. They identify entity disambiguation as a ranking task that computes the similarity between mention-context inputs and candidate entities. Yamada et al. (2016) present a joint learning method combining word and entity embeddings into the same continuous vector space to disambiguate entities. Similar to these two studies, Gupta et al. (2017) extends the joint encoding of context, mention, and entities with a fine-grained type information defined for candidate entities. Similarly, we use this entity type information as a domain indicator to filter the candidate entities.

NeuPL (Phan et al., 2017) employs LSTM and attention mechanism to disambiguate entities. Also, it provides a fast Pair-Linking algorithm which matches mention-entity pairs starting from the easiest pair. NeuPL considers positional information and word orderings. Therefore, two LSTM networks are used to model the context of left and right sides of each mention. Our study is similar to NeuPL in terms of resolving closer mention-entity pairs rather than all pairs. Our disambiguation method of closer mention-entity pairs is different from the pair linking method of NeuPL. Our method leverages CRFs to disambiguate closer neighbors as a named entity recognition task for a specific domain.

3 Method

Our study implies Entity Linking as a sequence learning task. For a given sequence mapping between mentions and entities, it consists a set of mentions $M = \{m_1, m_2, ..., m_N\}$ and a set of referent entities $E = \{e_1, e_2, ..., e_N\}$ in the Entity Linking task. In our work, the input size of mentions are equal to the output size of entities for each sequence mapping and N indicates the size of sequence elements rather than defining the size of the entire dictionary. The mention dictionary may contain variations of a proper noun. For instance, founder of the Republic of Turkey is ”Mustafa Kemal Ataturk” and M may include ”Ataturk”, ”Mustafa Kemal”. Therefore, the size of the entity dictionary is much less than the mention dictionary. Also, sequence mappings contain duplicates and the entity dictionary includes a limited number of unique entities for a specific domain.

In this study, we aim to map each mention to a corresponding entity ($M_i \rightarrow E_i$) in the given knowledge base for specific domains. Similar to recent studies (Zwicklbauer et al., 2016; Usbeck et al., 2014) we assume that documents with already detected mentions are the inputs of our method. Also, we have another assumption in which every mention contains one or more referent entities in the given knowledge base.

Figure 1 illustrates the general architecture of our method including 3 layers

1. **RDF2Vec Layer:** RDF2Vec (Ristoski and Paulheim, 2016) layer transforms each mention into a numerical d-dimensional vectors. We focus on entities and mentions, relations are not taken into considerations in this study.

2. **Bi-LSTM Layer:** Bidirectional LSTM layer will output hidden vector h_t per time step t and h_t is computed for the given sequence $S = \{r_1, r_2, ..., r_N\}$ where $\overrightarrow{r_t}$ and $\overleftarrow{r_t}$ are forward and backward pass vectors of RDF elements.

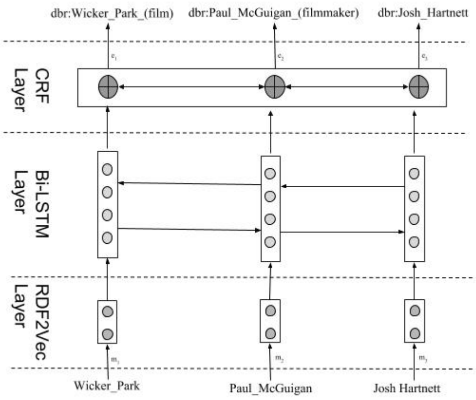

Figure 1: General structure of our method.

3. **CRF Layer:** This layer is composed of full-connected CRFs for ambiguous candidate entities and maps the best mention-entity pairs as a joint disambiguation task for specific domains.

We obtain training data by extracting mentions from Wikipedia and corresponding entities from DBpedia for the movie domain. In the previous sample text about the mention *Wicker Park*, three mentions are detected and these are inputs of RDF2Vec layer. On the other hand, three referent entities of these mentions are given as the output of CRF layer.

To increase ambiguity of training data, we extract Wikipedia disambiguation pages for this type of mentions. For instance, the mention *Wicker Park* has three candidate entities such as *Wicker_Park_(film)*[2], *Wicker_Park_(Chicago_park)*[3] and *Wicker_Park_(soundtrack)*[4]. Also, another mention *Paul_McGuigan* includes two different candidate entities. In the sample text, the last mention *Josh_Hartnett* has only one candidate entity and it can be recognized as an easy tag and is an indicator that this text might be related to the movie domain.

3.1 Candidate Entity Generation

To generate candidate entities we select DBpedia as the base knowledge base. We gather texts with

[2] http://dbpedia.org/resource/Wicker_Park_(film)

[3] http://dbpedia.org/resource/Wicker_Park_(Chicago_park)

[4] http://dbpedia.org/resource/Wicker_Park_(soundtrack)

already detected mentions for these entities from Wikipedia pages. For each mention, we query candidate entities and their domain information from DBpedia. To do it, we check ”dct:subject” and ”rdf:type” properties of entities. Then, mentions and candidate entities with their domain information are recorded in a key-value store.

Wikipedia articles are separated into paragraphs and each paragraph is retrieved in the key-value store whether the paragraph includes any annotated entity. If there is one or more entity in the given paragraph and this paragraph does not exist in the annotated text list, the given paragraph is loaded into a document store. At the same time, Wikipedia disambiguation pages are searched for each mention found in the paragraph. If there exists any disambiguation page for any mention, annotated texts with ambiguous entities are also stored.

3.2 RDF2Vec

RDF2Vec model (Ristoski and Paulheim, 2016) transforms word representations of Word2Vec model into representations of RDF elements such as classes, relations and instances. Instead of using words, dense RDF vectors are generated by entities and relations from the given RDF model. The overall structure of RDF2Vec model is listed as follows:

1. Entity-relation sequences are constructed from one of the strategies (Weistfeiler-Lehman Subtree RDF Graph Kernels, graph walks, etc.)
2. Neural language model is built by either Skip-gram or CBOW algorithm from entity-relation sequences
3. Entity relatedness is computed with Softmax function from the neural language model

Before the neural language model is trained, RDF model is transformed into the form of RDF embeddings. Consequently, each embedding can be represented as a numerical vector in Latent Feature Space. In this study, we do not use graph walks to transform RDF model into entity-relation sequences. Instead, we generate sequences of mentions and their entities considering their positions in the Wikipedia pages. Then, we obtain two different sequence documents for mention and entity sequences like a neural translation model as denoted in Figure 1. As sketched in this figure, mentions of the sample text of *Wicker Park* and corresponding entity sequences without relations are the input of RDFVec model.

3.3 Bi-LSTM Layer

Recurrent Neural Networks (RNNs) employ sequential information to make predictions. It differs from classical neural networks by considering dependency between sequences of inputs and outputs. RNNs operate recurrent tasks for every element of the sequence and memorizes information what has been computed so far. Bengio et al. (Bengio et al., 1994) emphasizes that RNNs can operate on long sequences in theory but in practice, they fail because they remember their most recent inputs in the sequence. Long Short-term Memory Networks(LSTMs) have been proposed to overcome this problem by producing a memory-cell to operate on long sequences (Hochreiter and Schmidhuber, 1997).

To disambiguate entities, we first operate an LSTM over input and output sequences. It will output the hidden state h_t at timestep *t*. Then the entity disambiguation rule for e_i^*

$$e_i^* = argmax_j(log\sigma(Ohi + b))_j \tag{1}$$

where $log\sigma$ is the log softmax function of the hidden state, and e_i^* is the annotated entity which has the highest score in this vector. The output space of O is $|E|x|E|$ dimensions in which E is the length of candidate entities.

For the given input sequences of mentions, this LSTM model computes a representation $\overrightarrow{h_t}$ from beginning to end in this sequence at every mention t. But it may ignore the critical information from the reverse order. To achieve it, a second LSTM operates over the same sequence from end to beginning. Then, this forward and backward LSTM pair denotes as a bidirectional LSTM (Graves and Schmidhuber, 2005). Bi-LSTM represents mentions with its left and right context and it is useful to gather more comprehensive information from the sequences.

3.4 Entity Disambiguation with CRFs

We model the entity disambiguation jointly using a Conditional Random Field (CRF) (Lafferty et al., 2001). In this situation, we use CRF as a sequence model where Bi-LSTM provides features.

Hence, CRF computes a conditional probability as a log-linear formulation

$$p(e|m) = \frac{exp(lp(m,e))}{\sum_{e'} exp(lp(m,e'))} \quad (2)$$

where m is the input sequence of mentions, e is the output sequence of entities. Then, lp indicates the log potential score of mention and entity sequences. To generate a tractable function, the potentials should be only included at local features. Then we define Emission and Transition as two types of potential scores in the Bi-LSTM CRF. Then, the score is determined for these log potentials such that

$$lp(e,m) = \sum_i log\theta_E(e_i \rightarrow m_i) + log\theta_T(e_{i-1} \rightarrow e_i) \quad (3)$$

where ($log\theta_E$) is the emission potential score for the mention at index i comes from the hidden state of the Bi-LSTM at timestep i. The transition potential scores ($log\theta_T$) are stored in a $|E|x|E|$ matrix P, where E is the entity dictionary and consists of unique entities from short texts.

We use PyTorch[5] to compute LSTM, Bi-LSTM and CRF models. PyTorch is a dynamic neural network tool in which we can define a computation graph for each instance and can be executed on-the-fly.

4 Experimental Setup

4.1 Dataset Generation

Manually annotated texts tend to be biased because people usually select familiar terms for the entity annotation. Also, this annotation process is sometimes noisy for unpopular terms. Therefore, Wikipedia should be chosen because it is curated by crowdsourcing and involves structured annotation process. MSNBC (Cucerzan, 2007), IITB (Kulkarni et al., 2009) and Wikilinks (Singh et al., 2012) proposes experimental datasets for general entity annotation tasks. Wikilinks provides a large-scale labeled corpus automatically constructed via links to Wikipedia. Wikilinks presents an automated method to identify a collection of massive amounts of entity mentions and is based on crawling anchor links in Wikipedia pages and exploiting anchor text as mentions. However, Wikipedia can also be employed for the level of ambiguity adjustments in order to use disambiguation pages and this is not directly indicated in Wikilinks.

Ambiguity is the ratio between ambiguous and unique entities and provides more realistic environment to entity annotators (Li et al., 2012). To adjust ambiguity and generate annotated texts for specific domains, we use a recent study (Inan and Dikenelli, 2017) which extracts the latest Wikipedia dump in English[6] for specific domains. To do it, they use Wikipedia category pages and DBpedia ”dct:subject”[7] property. Also, they provide an ambiguous environment in which, Wikipedia disambiguation pages are used for the selected domains. As an example, the mention *Wicker Park* has a Wikipedia disambiguation page[8] and it can be used to increase ambiguity in the movie domain.

The movie evaluation dataset involves 123 annotated texts in English. For each text, the average number of entities is 4.99 and there are 614 entities in total. Entities such as movies, directors, and starring are extracted from infoboxes of Wikipedia articles and mapped with referent entities by DBpedia. Disambiguation pages of these entities are extracted in other domains such as music and location to increase the ratio of ambiguity in the evaluation dataset for the movie domain. The ambiguity ratio of the evaluation dataset is 48.79% computed as the division of all ambiguous entities to the total number of unique entities extracted for the movie domain. Therefore, a more realistic ambiguous dataset can be generated to evaluate Entity Linking systems.

4.2 Results

We evaluate our method with several Entity Linking approaches from GERBIL benchmarking framework (Usbeck et al., 2015). We select Disambiguate to Knowledge Base (D2KB) task which focuses on the disambiguation of detected mentions to the related entities in the knowledge base. In this task, a given mention is guaranteed to map to the corresponding entity.

AGDISTIS (Usbeck et al., 2014) chooses candidate entities for the detected mentions from surface forms and generates a disambiguation graph for these candidates. The generated disambiguation graph is used in graph-based HITS algorithm

[5] http://pytorch.org/tutorials/index.html

[6] https://dumps.wikimedia.org/enwiki/20170420/

[7] http://purl.org/dc/terms/subject

[8] https://en.wikipedia.org/wiki/Wicker_Park

EL System	Micro-F1	Micro-P	Micro-R	Macro-F1	Macro-P	Macro-R
AGDISTIS	0.2063	0.2097	0.2031	0.3093	0.3098	0.309
AIDA	0.1485	0.1559	0.1417	0.1975	0.2035	0.1932
Babelfy	0.2101	0.2273	0.1953	0.284	0.2887	0.2812
Dbpedia Spotlight	0.1515	0.1515	0.1515	0.2044	0.2044	0.2044
Kea	0.1478	0.149	0.1466	0.1942	0.1976	0.1922
PBOH	0.2193	0.25	0.1953	0.282	0.282	0.282
WAT	0.2174	0.2451	0.1953	0.3124	0.3439	0.2967
LSTM	0.336	0.342	0.33	0.436	0.45	0.422
Bi-LSTM+CRF	0.446	0.488	0.41	0.546	0.564	0.53

Table 1: Evaluation scores of Entity Linking (EL) systems in GERBIL.

to match the best mention-entity pairs in the disambiguation step.

AIDA (Hoffart et al., 2011b) relies on a computation of global coherence between candidate entities and dense subgraph algorithms executing on the YAGO (Hoffart et al., 2011a) knowledge base.

Babelfy uses a graph-based disambiguation algorithm and finds the densest subgraph surrounded by candidate entities for the given mention. Then, Babelfy leverages the densest subgraph to match the best mention and entity pair.

DBpedia Spotlight (Mendes et al., 2011) uses a Vector Space Model (VSM) including DBpedia entity occurrences where a multidimensional word space has a representation per entity. Disambiguation task of DBpedia spotlight transforms Inverse Term Frequency (ITF) into an Inverse Candidate Frequency (ICF) which depends on candidate entities rather than terms and is an inverse proportion of candidate entities associated with words in VSM.

KEA (Waitelonis and Sack, 2016) proposes a combination of dictionary and knowledge based approaches. They analyze word co-occurrences of Wikipedia pages and merge these co-occurences with a graph analysis on the Wikipedia link structure and DBpedia.

PBOH (Ganea et al., 2016) is a collective entity linking system that is based on lightweight Wikipedia statistics. PBOH computes co-occurrence of words and entities for a probabilistic graphical model.

WAT (Piccinno and Ferragina, 2014) system is a complex version of TagMe (Ferragina and Scaiella, 2010). WAT depends on graph-based algorithm and selection of the best mention-entity pair from a vote-based algorithm.

Cornolti et al. (2013) expands general F1 measures to the Macro- and Micro- measures. While Macro- measures are the average of the corresponding measure over each document in all annotated documents, the Micro- measures consider all tags together thus giving more importance to documents having more tags. Table 1 illustrates the overall scores for Entity Linking task is measured in the generated evaluation set with respect to precision, recall, and F1-score. All scores are low because of high ambiguity of the generated evaluation dataset. F1 scores show that our study outperforms state-of-the-art studies using Bi-LSTM+CRF model on the generated evaluation dataset in the movie domain.

5 Conclusion

This study mainly presents a sequence learning method for domain-specific entity linking using sequence learning as a neural machine translation task. We filter candidate entities leveraging domain information and eliminating easy matches of mention-entity pairs. We employ a domain-specific dataset to compare our work with existing studies in GERBIL. Our method outperforms the state-of-the-art methods in the domain-specific configuration.

In the future, we apply other decoder models using the attention mechanism to the current model as a different joint disambiguation method of candidate entities. Also, we will examine many domain-specific datasets on this method.

References

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 249–260. ACM.

Silviu Cucerzan. 2007. 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)*.

Paolo Ferragina and Ugo Scaiella. 2010. 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, pages 1625–1628, New York, NY, USA. ACM.

Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. 2016. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. 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.

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

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard De Melo, and Gerhard Weikum. 2011a. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, pages 229–232. ACM.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61.

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

Emrah Inan and Oguz Dikenelli. 2017. Wedgem: A domain-specific evaluation dataset generator for multilingual entity linking systems. In *International Conference on Web Information Systems Engineering*, pages 221–228. Springer.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Xuansong Li, Stephanie M Strassel, Heng Ji, Kira Griffitt, and Joe Ellis. 2012. Linguistic resources for entity linking evaluation: from monolingual to cross-lingual. *Annotation*, 1:1.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. 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. ACM.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

David Milne and Ian H. Witten. 2008. 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. ACM.

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. 2016. Leveraging the schema in latent factor models for knowledge graph completion. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016*, pages 327–332.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the*

Association for Computational Linguistics (TACL), 2:231–244.

Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. Aida-light: High-throughput named-entity disambiguation. In *LDOW*, volume 1184 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *CoRR*, abs/1503.00759.

Minh C Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2017. Neupl: Attention-based semantic matching and pair-linking for entity disambiguation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1667–1676. ACM.

Francesco Piccinno and Paolo Ferragina. 2014. From tagme to WAT: a new entity annotator. In *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation, July 11, 2014, Gold Coast, Queensland, Australia*, pages 55–62.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. 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*, pages 1375–1384. Association for Computational Linguistics.

Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: RDF graph embeddings for data mining. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, pages 498–514.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2012-015*.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*, pages 1333–1339.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, SandroAthaide Coelho, Sören Auer, and Andreas Both. 2014. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 457–471. Springer International Publishing.

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. 2015. GERBIL – general entity annotation benchmark framework. In *24th WWW conference*.

Jörg Waitelonis and Harald Sack. 2016. Named entity linking in# tweets with kea. In *# Microposts*, pages 61–63.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1366–1371.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. *DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings*. Springer International Publishing, Cham.