OPTIC: A Deep Neural Network Approach for Entity Linking using Word and Knowledge Embeddings

Italo Lopes Oliveira[1][a], Diego Moussallem[2][b], Luís Paulo Faina Garcia[3][c] and Renato Fileto[1][d]

[1]*Department of Informatics and Statistics, Federal University of Santa Catarina, Florianópolis, Santa Catarina, Brazil*

[2]*Data Science Group, University of Paderborn, North Rhine-Westphalia, Germany*

[3]*Computer Science Department, University of Brasilia, Brasília, Distrito Federal, Brazil*

Keywords: Entity Linking, Knowledge Embedding, Word Embedding, Deep Neural Network.

Abstract: Entity Linking (EL) for microblog posts is still a challenge because of their usually informal language and limited textual context. Most current EL approaches for microblog posts expand each post context by considering related posts, user interest information, spatial data, and temporal data. Thus, these approaches can be too invasive, compromising user privacy. It hinders data sharing and experimental reproducibility. Moreover, most of these approaches employ graph-based methods instead of state-of-the-art embedding-based ones. This paper proposes a knowledge-intensive EL approach for microblog posts called OPTIC. It relies on a jointly trained word and knowledge embeddings to represent contexts given by the semantics of words and entity candidates for mentions found in the posts. These embedded semantic contexts feed a deep neural network that exploits semantic coherence along with the popularity of the entity candidates for doing their disambiguation. Experiments using the benchmark system GERBIL shows that OPTIC outperforms most of the approaches on the NEEL challenge 2016 dataset.

1 INTRODUCTION

A massive amount of short text documents such as microblog posts (e.g., tweets) is produced and made available on the Web daily. However, applications have difficulties in automatically making sense of their contents for correctly using them (Laender et al., 2002). One way to circumvent this problem is by using Entity Linking (EL).

The EL task links each named entity mention (e.g., place, person, institution) found in a text to an entity that precisely describes the mention (Shen et al., 2015; Trani et al., 2018) in a Knowledge Graph (KG), such as DBpedia[1] (Auer et al., 2007; Lehmann et al., 2009), Yago[2] (Fabian et al., 2007) or Freebase[3] (Bollacker et al., 2008). The disambiguated named entity mentions can be used to identify things that the users talk about. It can help to recommend new products for a user or to determine if a user is a good potential client for a particular company, for example.

Several EL approaches have been successfully applied to long formal texts, with F1 scores above 90% for some datasets (Liu et al., 2019; Parravicini et al., 2019). However, microblog posts still present a challenge for EL (Guo et al., 2013; Shen et al., 2013; Fang and Chang, 2014; Hua et al., 2015; Han et al., 2019; Plu et al., 2019). This happens because those posts are usually informal and, therefore, prone to problems like typos, grammatical errors, slangs, and acronyms, among other kinds of noise. Besides, microblog posts have a limited textual context. For example, Twitter only allows posts having up to 280 characters.

Although limited, the textual context present in microblog posts is still essential to correctly disambiguate named entity mentions, as highlighted by Han et al. (Han et al., 2019). Some approaches expand the post context by considering related posts (Guo et al., 2013; Shen et al., 2013) and extra information, like social interactions between users (Hua et al., 2015) and spatial and temporal data (Fang and Chang, 2014). However, we believe that overworking this kind of extra context can be too invasive, compromis-

[a] https://orcid.org/0000-0002-2357-5814
[b] https://orcid.org/0000-0003-3757-2013
[c] https://orcid.org/0000-0003-0679-9143
[d] https://orcid.org/0000-0002-7941-6281

[1]https://wiki.dbpedia.org
[2]http://www.yago-knowledge.org/
[3]https://developers.google.com/freebase/

Oliveira, I., Moussallem, D., Garcia, L. and Fileto, R.
OPTIC: A Deep Neural Network Approach for Entity Linking using Word and Knowledge Embeddings.
DOI: 10.5220/0009351203150326
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 315-326
ISBN: 978-989-758-423-7

ing the privacy of the users. EL approaches should avoid so much intrusion and, as much as possible, focus on the context present on the text of each post being semantically enriched.

Recently, the use of embeddings to represent words and Knowledge Graph (KG) entity candidates for mentions spotted in formal texts has been gaining traction in EL approaches based on Deep Neural Network (DNN) (Fang et al., 2016; Yamada et al., 2016; Moreno et al., 2017; Ganea and Hofmann, 2017; Chen et al., 2018; Kolitsas et al., 2018). Word embedding and knowledge embedding techniques aim to represent words and entities, respectively, in some n-dimensional continuous vector space. Word embeddings (Li and Yang, 2018) trained with large volumes of text capture relations between words. Knowledge embeddings (Wang et al., 2017), on the other hand, capture relationships between unambiguous entities, which can be represented as triples in some KG. One reason why DNNs have been used with embeddings is that DNN may capture linear and non-linear relations between embeddings. However, microblog posts are not the focus of approaches that employ embeddings and DNN (Shen et al., 2013; Han et al., 2019), and only (Fang et al., 2016) has exploited graph-based knowledge embeddings in EL yet.

This work proposes OPTIC, a knOwledge graPh-augmented enTity lInking approaCh. OPTIC is based on a DNN model that exploits the embeddings of words and knowledge in a shared space to tackle the EL task for microblog posts. Firstly, we jointly train word embeddings and knowledge embeddings in fastText (Bojanowski et al., 2017; Joulin et al., 2017b). Then, OPTIC employs these embeddings to represent the text documents and their entity candidates for each recognized mention. Differently from other approaches, we replace the named entity mentions by their respective entity candidates. Our DNN model uses the embeddings to determine if an entity candidate (represented by a knowledge embedding) matches the textual context (represented by word embeddings) that surround it. Experiments with microblog posts, more specifically tweets, show the viability and the benefits of our approach. At the best of our knowledge, we are the first to use in an EL approach word and knowledge embedding trained jointly by fastText. Finally, we evaluate OPTIC using the EL benchmark system GERBIL (Usbeck et al., 2015) with public datasets.

The main contributions of this work are: (*i*) an EL process that jointly trains word embeddings and knowledge embeddings for the EL task using fastText and selects entity candidates for each named entity mention by using an index of surface forms built-in ElasticSearch; (*ii*) a neural network model to disambiguate named entity mentions by exploiting semantic coherence of embeddings along with entity popularity and; (*iii*) the evaluation of the proposal using public datasets on the EL benchmark system GERBIL. The version of OPTIC used in this paper is publicly available[4].

The remaining of this paper is organized as follows. Section 2 reviews literature about the use of embeddings in EL approaches. Section 3 details our EL approach as a process that selects candidate entities for mentions using an index of surface forms and disambiguates them using a DNN model fed with jointly trained embeddings for words and knowledge. Section 4 reports experiments to evaluate our approach and discusses their results. Lastly, Section 5 presents the conclusions and possible future works.

2 RELATED WORKS

In this paper, we use the following formal definition for the EL task, extracted from Shen, Wang and Han (Shen et al., 2015). Given a set of entities E and a set of named entity mentions M within a text document T, the EL task aims to map each mention $m \in M$ to its corresponding entity $e \in E$. If the entity e for a mention m does not exist in E (i.e., $e \notin E$), m is labeled as "NIL", whose meaning is non-linked.

Existing EL approaches for microblog posts, to the best of our knowledge, do not employ word, knowledge, and entity embeddings. Thus, in the following Section 2.1, we discuss these embeddings and approaches that employ them successfully for EL in long formal texts. Then, in Section 2.2 we review EL approaches particularly intended for microblogs.

2.1 Embeddings and EL Approaches

As discussed in Section 1, many approaches employ embeddings successfully for doing EL in long formal texts (Fang et al., 2016; Yamada et al., 2016; Moreno et al., 2017; Ganea and Hofmann, 2017; Chen et al., 2018; Kolitsas et al., 2018). Nevertheless, except for (Fang et al., 2016), these works use entity embedding instead of knowledge embedding. Similarly to knowledge embedding, entity embedding aims to represent entities as vectors in an n-dimensional continuous space. However, entity embeddings are derived from textual contents (Moreno et al., 2017; Ganea and Hofmann, 2017; Kolitsas et al., 2018), in a similar way as word embeddings, or hyperlinks (Yamada

[4]https://github.com/ItaloLopes/optic

et al., 2016; Chen et al., 2018) of semi-structured and unstructured data sources, like Wikipedia pages.

Entity embedding has a few drawbacks compared with knowledge embedding. Firstly, documents like Wikipedia pages are published in HTML format, whose contents can be interpreted and handled in different ways. It hampers the replication of entity embedding techniques. On the other hand, most knowledge embedding techniques (e.g., Trans(E, H, R) (Bordes et al., 2013; Wang et al., 2014b; Lin et al., 2015), HoLE (Nickel et al., 2016), fastText knowledge embedding (Joulin et al., 2017b)) take as input triples of the form $\langle subject, predicate, object \rangle$ from KGs (e.g., DBpedia, Yago, Freebase) that follow the Linked Open Data (LOD) guidelines. Consequently, they use the RDF standard, allowing triples interchange with little effort while keeping their precise semantics.

Secondly, when dealing with different types of data (e.g., hyperlinks instead of textual contents), it is necessary to adapt entity embedding techniques. If someone wants to combine textual contents and hyperlinks to produce embeddings, it is necessary to propose a new embedding technique or adapt an existing one. Although some knowledge embedding techniques suffer from a similar problem (e.g., Trans(E, H, R), HoLE), a few techniques, like fastText and techniques proposed by (Wang et al., 2014a; Xie et al., 2016), already surpass this limitation by allowing the combination of triples with text about entities.

Finally, most of the entity embedding techniques work with any text (considering the ones based on texts) or any graph structure (considering the ones based on hyperlinks). Knowledge embedding techniques, on the other hand, are tailored for KGs, considering features like distinct relations, and may impose restrictions such as the number of distinct relations (e.g., subclass, type) being far smaller than the number of entities. Therefore, knowledge embedding may represent the entities and relations of a KG in a more meaningful way than entity embedding.

Differently from most EL approaches that employ embeddings, (Fang et al., 2016) uses knowledge embedding jointly with word embedding, instead of entity embedding. The knowledge embedding technique used in that paper is similar to the TransE knowledge embedding technique (Bordes et al., 2013). To guarantee that knowledge embeddings and word embeddings are compatible, (Fang et al., 2016) employs methods for jointly embedding entities and words into the same continuous vector space (Wang et al., 2014a) and for aligning text embeddings with knowledge embeddings (Zhong et al., 2015). However, meaningful relations between words and entities may be lost by separately training word embeddings and knowledge embeddings. Thus, in this work, we use the fastText technique to train word and knowledge embedding jointly in the same vector space. We chose fastText as it was state of the art for doing that at the time we prepared this paper (Joulin et al., 2017b).

The FastText word embedding model efficiently achieves state-of-the-art results for text classification (Joulin et al., 2016; Joulin et al., 2017a; Bojanowski et al., 2017). It reaches this competitiveness by training a linear model with a low-rank constraint. It represents sentences in a Bag of Words (BoW) model, besides considering n-gram features. According to (Joulin et al., 2017b), fastText "can be applied to any problem where the input is a set of discrete tokens".

The fastText model for knowledge embedding also achieves state-of-the-art results, especially for tasks like KG completion and question answering. As fastText models the sentences of a text and facts of a KG as a BoW, it is possible to train a linear model for both word and knowledge embedding. This approach has the advantage of producing aligned embeddings, besides providing more context for both types of embeddings. At the best of our knowledge, such an approach has not been considered for EL yet.

2.2 EL Approaches for Microblogs

Current EL approaches for microblogs that we found in the literature (Guo et al., 2013; Shen et al., 2013; Fang and Chang, 2014; Hua et al., 2015; Han et al., 2019; Plu et al., 2019) do not use embeddings. One possible reason for this is that microblog posts are short and, consequently, have a little context. It hampers the effectiveness of embedding-based EL techniques, which are heavily based on the textual context.

EL approaches for microblog posts tackle the disambiguation of mentions in different ways, like (*i*) collecting extra posts to increase the context size (Guo et al., 2013; Shen et al., 2013); (*ii*) modeling user interest information based on social interactions between users (Hua et al., 2015); (*iii*) using spatial and temporal data associated with microblog posts (Fang and Chang, 2014) and; (*iv*) exploiting the relationships between entities in a KG to determine scores for disambiguation (Shen et al., 2013; Han et al., 2019). However, these approaches have some drawbacks. The approaches in the groups (*i*), (*ii*), and (*iii*) can be considered too invasive, as they handle lots of data about the users and can compromise privacy. Moreover, privacy issues hinder dataset sharing and, consequently, experimental reproducibility. Regarding group (*iv*), the approaches that have been successfully applied to long formal texts (Han et al., 2011; Huang et al., 2014; Guo and Barbosa, 2014; Kalloubi et al., 2016; Li et al., 2016; Ganea et al., 2016; Chong et al., 2017; Wei et al., 2019; Parravicini et al., 2019; Liu et al., 2019) are tailored for documents with a high number of entity mentions, which is usually not the case for microblog posts.

In Shen et al. (Shen et al., 2013) and Han et al. (Han et al., 2019), the graph-structure of a KG is used to extract scores like prior probability and topical coherence. Although these scores have been useful in several EL approaches, utilizing only them neglects the context present in the KGs. Han et al. (Han et al., 2019) circumvent this limitation by comparing the embedded contexts of the microblog posts and each entity mention with the embedded first paragraph of the Wikipedia page of the respective entity candidates. However, their paper does not detail the embedding used (e.g., word embedding, entity embedding, knowledge embedding).

Finally, among all works that we analyzed, only (Plu et al., 2019) proposes an EL approach suitable for both formal long text and microblog posts. It disambiguates entity candidates by using a combination of the previously obtained PageRank of each entity candidate, the Wikipedia page title referring to each mention candidate, the Levenshtein distance between mentions and, the maximum Levenshtein distance between the mention and each element in the respective Wikipedia disambiguation page. The performance of the (Plu et al., 2019) approach is evaluated for microblog posts using only the NEEL challenge public dataset (Rizzo et al., 2015; Rizzo et al., 2016) and the GERBIL benchmark system (Usbeck et al., 2015).

Differently from the existing EL approaches, OPTIC uses jointly trained knowledge embeddings and word embeddings to tackle EL in microblog posts. Moreover, to the best of our knowledge, we are the first to propose the use of knowledge and word embedding trained jointly by fastText for doing EL using a neural network. Finally, our neural network is trained only with tweets available in the NEEL 2016 challenge dataset, which lessens privacy issues.

3 PROPOSED APPROACH

OPTIC employs jointly trained knowledge embeddings and word embeddings as microblog post semantic features that are fed to a DNN model that disambiguates entity candidates for each mention spotted in the posts. Figure 1 provides an overview of the OPTIC architecture and EL process. Like most approaches proposed in the literature, OPTIC does EL in two stages: (*i*) selection of entity candidates for each mention and; (*ii*) disambiguation of entity candidates. Prior to these stages, it is necessary to build an index of surface forms to support efficient entity candidate selection for each mention recognized in the text. Word embeddings and knowledge embeddings are also jointly generated prior to named entity recognition and disambiguation. All these tasks are explained in more detail in the following subsections.

3.1 Indexing Surface Forms for Entity Candidates Selection

The selection of entity candidates is the stage of the EL task that chooses a set of candidate entities C_i for each mention $m_i \in M$ found in the text. It is essential properly select entity candidates for two main reasons: (*i*) if the search scope is too narrow or imprecise, the correct entity that describes m_i may not be in C_i; and (*ii*) if the search scope is too broad, it may generate noise that increases the running time and hinders the results of the disambiguation stage, depending on the adopted disambiguation strategy (e.g., collective graph-based disambiguation).

Several works (Moussallem et al., 2017; Parravicini et al., 2019; Wei et al., 2019; Plu et al., 2019) use index-based string search systems to find entity candidates for each mention. We also employ this strategy in OPTIC. More specifically, we implement the entity candidate selection strategy proposed in (Moussallem et al., 2017) on top of ElasticSearch[5].

The strategy of Moussallem et al. (Moussallem et al., 2017) is based on five indexes, respectively, for surface forms, person names, rare references, acronyms, and context. Surface forms are the possible names that can be used to refer to an entity. In this work, we obtained the surface names from the KG triples by taking the values of the property *rdfs:label* of each entity. Person names consider all the possible permutations of the words constituting each surface form, in order to represent possible variations of person names in textual mentions. Rare references refer to surface names that appear in the entity textual description but are not available in KG triples. We take them by applying a POS tagger to the first line of the entity description text. We employ the Stanford POS Tagger (Toutanova and Manning, 2000) for doing this, in the same way as (Moussallem et al., 2017). Acronyms refer to the possible meanings of each entity acronym (e.g., BR to Brazil). Lastly, context is the Concise Bounded Description[6] (CBD) of each entity.

[5]https://www.elastic.co/products/elasticsearch
[6]https://www.w3.org/Submission/CBD/

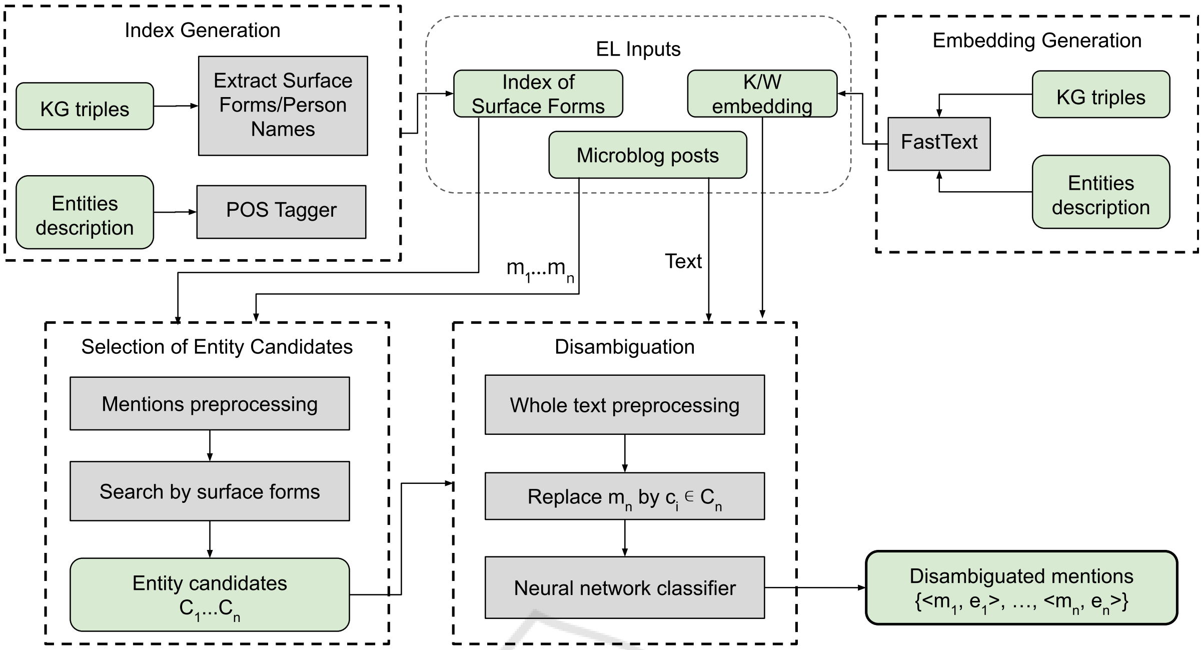

Figure 1: Overview of OPTIC Architecture and EL Process.

In this work, we only index surface forms, person names, and rare references to find entity candidates. These three indexes are implemented as a unified ElasticSearch index. Although acronyms could contribute to improving the performance of our proposal significantly, mainly because it is aimed at microblog posts, which usually contain many acronyms, we have not found any open and public acronym dataset yet. On the other hand, the use of a private dataset, as done by (Moussallem et al., 2017), would hinder the reproducibility of our experiments. The context index, by its turn, does not provide relevant results that justify its use, as microblog posts usually have little textual context surrounding the named entity mentions, and this context can contain a lot of noise.

Lastly, we take advantage of the ElasticSearch capabilities and add to each candidate its popularity. The popularity, also referred to as the probability of an entity e given a named entity mention m (i.e., $p(m|e)$), is a useful feature employed in several EL approaches (Moussallem et al., 2017; Kolitsas et al., 2018; Plu et al., 2019). We use the same popularity calculation proposed by Moussallem et al. (Moussallem et al., 2017), which is based on applying the PageRank algorithm to DBpedia.

3.2 Selection of Entity Candidates

As shown in Figure 1, the first step of the entity candidates selection stage is to preprocess the mentions $m_1, \ldots, m_n$, which were found in the texts by some named entity recognition tool. In microblog posts, a named entity mention can appear in one of three alternative forms: (*i*) normal text; (*ii*) mention to a user (e.g., @ShaneHelmsCom, @Twitter); or (*iii*) hashtag (e.g., #StarWars, #ForceAwakens). Therefore, we first determine the form of each mention to handle it properly. We remove the special character (@ or #, respectively) from each mention of the forms (*ii*) and (*iii*). Afterward, we use a regular expression to segment each mention that uses camel cases. For example, "TheForceAwakens" and "Star_Wars" are segmented into "The Force Awakens" and "Star Wars", respectively. Lastly, we ensure that only the first letter of each word of each mention is capitalized.

With the named entity mentions preprocessed, we query the ElasticSearch index for each entity mention m_i to produce its respective set of entity candidates C_i. We employ two types of queries simultaneously on ElasticSearch: exact/contain match and n-gram similarity. As ElasticSearch returns the candidates sorted by their similarity score, the candidates returned via n-gram similarity usually rank higher than the candidates returned via exact/contain match. Thus, if we only considered the m top-ranked candidates to be used in the disambiguation step, the correct entity, if returned by the exact/contain match, could be outside of this m top-ranked candidates. Therefore, we increase the score of the candidates returned by exact/contain b times, being b a parameter (real number) to be adjusted in experiments.

If the query does not return any candidate for a mention m_i composed of more than one word, we execute the procedure detailed in Algorithm 1 to derive a set of shorter mentions M'_i from each mention m_i, by removing each word from m_i at a time. We consider that a mention is a set of words $M = \{w_1, \ldots, w_k\}$. Algorithm 1 iterates over the k words of a mention m_i. For each word $w_j (1 \leq j \leq k)$ of m_i, the algorithm removes w_j from m_i and concatenates the remaining words in a simplified mention m'_j, without w_j, but preserving the order of the remaining words as in m_i. Each simplified mention m'_j is appended to the set M'_i. Notice that in the end of this procedure M'_i will contain k alternative simplified forms for the mention m_i, i.e., $|M_i| = k$, with each alternative form $m'_j \in M'_i$ excluding a word from the original mention m_i. To exemplify this procedure, consider that no entity candidate has been found for the mention "The Force Awakens". The alternative simplified mentions created from this 3-word mention are "Force Awakens", "The Awakens" and "The Force". Each simplified mention in M'_i is queried on the ElasticSearch index explained before, to look for entity candidates for m_i. This procedure is particularly important for microblog posts because some users may attach other words to their usernames as a way to distinguish themselves from other users.

Algorithm 1: Create Simplified Mentions for m_i.

Input: $m_i = w_1 \ldots w_k$ # mention m_i with $k \geq 1$ words

Output: $M'_i = \{m'_1, \ldots, m'_k\}$ # set of k simplified mentions

```
M'_i ← ∅; # Initially the set of simplified mentions is empty
if |m_i| > 1 then
    for w_j ∈ m_i do
        m'_j ← nil; # Empty simplified mention m_j
        for w_l ∈ m_i do
            if w_l ≠ w_j then
                append(m'_j, w_l);
            end if
        end for
        insert(M'_i, m'_j);
    end for
end if
```

Then, the set of candidates C_i found for each mention m_i (or its simplified mentions) are given as input to the disambiguation step, as detailed in Section 3.4.

3.3 Embedding Generation

As presented in Figure 1, the embedding generation in our current implementation is done by using fastText, which is available in Github[7]. KG triples and entity abstracts are used as inputs of the fastText to jointly train knowledge embeddings and word embeddings in the same vector space.

The KG used in this work is the English version of DBpedia. We have chosen DBpedia because it is the Linked Open Data (LOD) version of Wikipedia and, as presented in Section 2, Wikipedia has been adopted as the source of entity descriptions by most EL approaches. On top of this, the datasets used to evaluate our proposal (see Section 4) have pointers to DBpedia resources.

We used only the DBPedia triples of the high-quality version of the infobox data[8]. This decision has been made to produce more meaningful knowledge embeddings and in a faster way than by considering all the DBpedia triples. On the other hand, we used the long version of the DBpedia abstracts[9] to produce word embeddings. Each entity abstract was taken from the introductory text of each Wikipedia page about that entity. The long version of a DBpedia abstract encompasses the whole introductory text, while the short version includes only the first paragraph. Thus, useful information that can be encoded in word embeddings and help to disambiguate mentions could be lost if we had used only the first paragraphs of the introductory texts.

We have combined infobox data triples and long abstracts of entities in a single training file. This allows fastText to jointly produce the knowledge embeddings and word embeddings in the same vector space. The parameters for the fastText model training are detailed and discussed in Section 4.

3.4 Disambiguation

The first step of the disambiguation stage is to preprocess the microblog post texts. For this, we use the Part-of-Speech (PoS) Tagging functionality of the Tweet NLP (Gimpel et al., 2010; Owoputi et al., 2013) tool[10]. It attaches tags for the words present in the texts. Examples of these tags are user, hashtag, emoticon, URL, and garbage. Then, we categorize words tagged by Tweet NLP into two categories: words to be removed and words to be cleaned.

We consider words to be removed the ones tagged as an emoticon, URL, or garbage. These words do not help the EL task or constitute just noise that could

[7]https://github.com/facebookresearch/fastText

[8]http://wiki.dbpedia.org/services-resources/documentation/datasets#MappingbasedObjects

[9]http://wiki.dbpedia.org/services-resources/documentation/datasets#LongAbstracts

[10]http://www.cs.cmu.edu/ ark/TweetNLP/

hinder EL efficiency. Emoticons are useful for sentiment analysis but provide little if any contextual information for EL. URLs may be useful for the EL task since the contents pointed by them can provide extra contextual information. However, our approach focus on the context present in the post texts themselves. Moreover, URLs do not have an embedding representation. Lastly, the Tweet NLP attaches the tag "garbage" to words for which it could no infer a precise meaning. Examples of words tagged with this tag are "scoopz" and "smh", among others.

Meanwhile, words to be cleaned may provide useful contextual information, but have special characters or are presented in a particular way. We consider the words tagged as user or hashtag as words to be cleaned in microblog posts. Their cleaning follows the same preprocessing used for the selection of entity candidates detailed in Section 3.2.

Different from other approaches that handle the embedding of the textual contents separately from the embedding of the entity candidates, OPTIC handles them simultaneously. This is possible because we have trained word embeddings and knowledge embeddings together in a fastText model (Section 3.3) and, therefore, they are in the same vector space.

To exploit the embeddings concomitantly, we represent each post that has at least one mention with at least one entity candidate for EL by using both kinds of embeddings. Each ordinary word (that is not identified as a mention) of the post text is represented by its respective word embedding. Each mention m_i is replaced by the entity embedding of each one of its entity candidates $c_j \in C_i$, one candidate at a time. In other words, we generate an enriched semantic representation of each microblog post for each entity candidate $c^i_j \in C_i$ of each mention $m_i \in M$.

For each mention $m_i \in M$, we have a set of semantic enriched representations of the post $SE_i = \{se^i_1, \ldots, se^i_{ji}\}$, with each se^i_j being the embedded post representation corresponding to the entity candidate m_j and $|SE_i| = |C_i|$. Our disambiguation step aims to determine which enriched semantic representation $se^i_j \in SE_i$ makes more sense for the embedded context where m_i appears.

We consider the disambiguation of mentions as a binary classification problem, as shown in Figure 2. The binary classifier must decide correctly if an entity candidate (e.g., *dbr:Chicago_Bulls*) fits in the context that surrounds it (e.g., information about 2003, Michael Jordan and the number 23) or not. The positive case is labeled as 1, and the negative as 0. Our approach models a Bidirectional Long Short-Term Memory (Bi-LSTM), followed by a Feed-Forward Neural Network (FFNN) as a binary classifier. We adopt a neural network approach because it can capture non-linear interactions between embeddings.

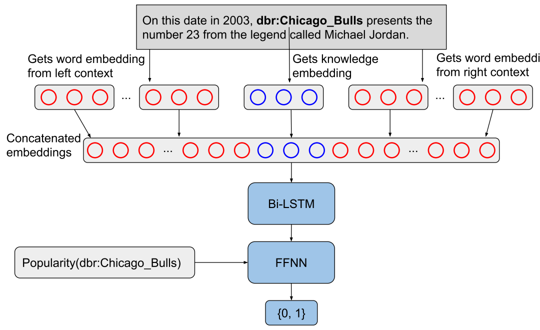

Figure 2: Bi-LSTM and FFN Neural Network as a Binary Classifier Considering Both Word and Knowledge Embeddings Simultaneously.

We model our DNN as a Bi-LSTM because it records long-term dependencies and takes into consideration the order of the input data, which is essential to interpret some textual contents properly. It is significantly important in our approach since we substitute the named entity mentions by their entity candidates. It allows us to properly capture the interactions between the entity candidates (represented as knowledge embeddings) and the context that surrounds them (represented as word embeddings). In addition, Bi-LSTM has been successfully employed for EL in long formal texts using word embeddings (Kolitsas et al., 2018; Wang and Iwaihara, 2019; Martins et al., 2019; Liu et al., 2019). The FFNN input is the bi-LSTM output, which is a sequence that represents the interactions between the embedding of the entity candidate and the embeddings of the words that surround it, and the popularity of the entity candidate. Therefore, the FFNN captures the interactions between the embeddings and the popularity of the entity candidate and classify if the entity candidate is correct or not.

Algorithm 2 depicts our disambiguation method. Its inputs are the enriched semantic representations SE_i of the microblog post for each mention $m_i \in M$, the popularity of the entity candidate, and a threshold value for the probability of an entity candidate being the correct one. For simplicity, we consider that the DNN is capable of getting the embeddings of both words and entity candidates in $se^i_j \in SE_i$. For each $se^i_j \in SE_i$, the DNN returns the probability (score) of se^i_j being the correct entity, which we append to a queue of highly scored entity candidates (lines 3 and 4). We decide which entity candidate $c_j \in C_i$ is the best to describe the mention $m_i \in M$ by taking the one with the highest score. In case there is no entity candidate with a sufficiently high probability of correctly describing m_i, we label this mention as "NIL".

Algorithm 2: Disambiguation of Entity Candidates.

Input: $SE_i = \{se^i_1, \ldots, se^i_{j^i}\}$ # instances of microblog posts with $c_j \in C_i$ replacing a mention $m_i \in M$
p_j # popularity of c_j
θ # score threshold

Output: e # correct entity candidate to describe mention m_i

1: $S = \emptyset$ # Set that will contain the disambiguation scores of the instances
2: **for** $se^i \in SE_i$ **do**
3: s = score(NN_Model(), se, p_j)
4: append(S, s)
5: **end for**
6: **if** $|S| = 1 \wedge S > \theta$ **then**
7: $e \leftarrow getCandidate(SE_i, s)$
8: **else if** $|S| = 1 \wedge S < \theta$ **then**
9: $e \leftarrow$"NIL"
10: **else**
11: $maxScore \leftarrow max(S)$
12: **if** $count(maxScore, S) > 1$ **then**
13: $e \leftarrow$"NIL"
14: **else**
15: **if** $maxScore < \theta$ **then**
16: $e \leftarrow$"NIL"
17: **else**
18: $e \leftarrow getCandidate(SE_i, maxScore)$
19: **end if**
20: **end if**
21: **end if**

For mentions that have more than one candidate, i.e., $|SE_i| = |S| > 1$, first we get the highest score from S (line 11). Then, we count in S how many times the highest score appears. If this count is bigger than one, our model is not capable of differentiating them, and we consider this case as "NIL" (lines 12 and 13). Lastly, if there is only one candidate with the highest score, we only need to check if its score is above or below the threshold.

4 EXPERIMENTS

This section reports the experiments performed to evaluate how well OPTIC disambiguates named entity mentions in microblog posts. We compare OPTIC results with those of state-of-the-art EL approaches in the literature. We use the F1 score as the comparison metric because it has been utilized as an evaluation metric for the disambiguation step of the EL task in several works (Moro et al., 2014; Moussallem et al., 2017; Sevgili et al., 2019; Wang and Iwaihara, 2019; Plu et al., 2019). The GERBIL framework calculates two versions of the F1 score: micro and macro. The micro F1 score calculation considers all true positives, false positives, and false negatives from all documents together, while the macro F1 score is the average of the F1 scores calculated for each document.

4.1 Experimental Setup

Our DNN model uses 200-dimensional embeddings. We apply dropout 0.5 on the embeddings before using them in the Bi-LSTM. The Bi-LSTM has a hidden size of 200, with two hidden layers. For the training of our model, we use Adam loss optimization (Kingma and Ba, 2014) with a learning rate of 0.001 and a batch size of 20. For disambiguation, we adopt a threshold of 0.7 for the probability of an entity candidate being the correct one.

For the embedding generation, we employ fastText with 500 epochs and context window size of 50. The remaining parameters are set to the default values presented in the fastText GitHub repository[11]. The embedding training dataset that we have used is the one described in Section 3.3.

We use the EL benchmark system GERBIL (Usbeck et al., 2015) to manage the experiments and the analysis of the result. As this work focus on the disambiguation step of the EL task, we use the *Disambiguation to KB* (D2KB) experiment type of GERBIL. In experiments of this type, GERBIL provides a text with the named entity mentions already recognized to the EL tools. Then, we only need to provide to GERBIL the named entity mentions disambiguated so that it can calculate performance measures such as macro F1 score and micro F1 score for each EL tool and generate the performance comparison reports.

As this work focus on microblog posts and for the sake of facilitating performance comparability, we use the following datasets that are integrated into GERBIL for the experiments: Microposts2014-Test; Microposts2015-Test; and Microposts2016-Test. These datasets are from the NEEL challenges of 2014 (Cano et al., 2014), 2015 (Rizzo et al., 2015) and, 2016 (Rizzo et al., 2016), respectively. Each one of these datasets contains a number of tweets with their named entity mentions recognized and linked to disambiguated resources of DBpedia. For simplicity, we call these datasets, respectively, as NEEL2014, NEEL2015, NEEL2016.

We use the dataset microposts2016-Training from the NEEL challenge 2016 for training the neural network model. This dataset consists of microblog posts

[11]https://github.com/facebookresearch/fastText

with 8665 instances of recognized mentions in their texts, of which 6374 points to DBpedia entities and 2291 point to "NIL". As we model our DNN as a binary classifier, our training dataset needs positive and negative instances. Therefore, we apply the following procedure on the microposts2016-Training dataset. For the mentions, we replace each mention that points to a DBpedia entity by the respective entity in the microblog text and labels that instance of the EL problem as a positive one (label 1). For each mention that points to "NIL", we apply the step *Selection of Entity Candidates* of our approach (Figure 1). Then, from the set of obtained entity candidates, we randomly select *two* candidates, replace the entity mention by the respective candidate, and label that instance of the EL problem as a negative one (label 0). Therefore, for each "NIL" mention, we generate two negative instances. Lastly, for each positive instance, we generate one negative one by replacing the correct entity with an incorrect one. In the end, our training dataset is composed of 16463 instances, being 6374 positive ones, and 10089 negative ones.

For the selection of entity candidates, the maximum number of candidates returned by ElasticSearch is 100. Moreover, we multiply by 5 the score of the candidates returned by exact/contain queries. Lastly, for the disambiguation, we consider the context window of size 3 and a threshold of 0.7. All these parameter values were obtained in preliminary experiments.

We have used blades of the Euler supercomputer[12] for embeddings generation and DNN training. The embedding generation was done on blades having just CPUs while the DDN training run on blades having also GPU. The first blades have 2 CPU Intel(R) Xeon(R) E5-2680v2 @ 2.8 GHz with 10 cores, and 128 GB DDR3 1866MHz RAM memory. The other blades have 2 CPU Intel(R) Xeon(R) E5-2650v4 @ 2.2 GHz with 12 cores, 128 GB DDR3 1866MHz RAM memory, 1 GPU Nvidia Tesla P100, 3584 Cuda cores and 16GB of memory. Afterwards, we run our EL process in another server with 2 CPU Intel(R) Xeon(R) E5-2620 v2 @ 2.10GHz with 6 core, and 128 GB DDR3 1600MHz RAM memory.

4.2 Results and Discussion

As our focus is on the disambiguation step of the EL task, we only employ the D2KB experiment of GERBIL. Table 1 presents the micro and macro F1 scores (lines *F1@micro* and *F1@macro*, respectively) of our proposal and of state-of-the-art approaches available on GERBIL. Notice that the macro F1 scores of most approaches are similar, even when there is a wide variation on the micro F1 scores. It happens especially with microposts2016 (column NEEL2015) because at least this dataset has several documents with no named entity mention. Therefore, we focus on the micro F1 scores on the following discussions.

Table 1: Macro and Micro F1 of the Approaches Tested on the GERBIL Benchmark System. The **highest Micro and Macro F1 Scores** for Each Dataset Are Highlighted in **bold**. the ERR Value Indicates That the Annotator Caused Too Many Single Errors on GERBIL. For ADEL, Only the F1@micro Score Is Available, from the Paper about the Approach.

F1@Micro F1@Macro	NEEL2014	NEEL2015	NEEL2016
ADEL	**0.591**	**0.783**	**0.801**
AGDISTIS/MAG	0.497 **0.701**	0.719 **0.768**	0.616 **0.964**
AIDA	0.412 0.588	0.414 0.439	0.183 0.919
Babelfy	0.475 0.623	0.341 0.384	0.157 0.917
DBpedia Spotlight	0.452 0.634	ERR	ERR
FOX	0.252 0.508	0.311 0.355	0.068 0.910
FREME NER	0.419 0.597	0.313 0.353	0.162 0.916
OpenTapioca	0.215 0.484	0.259 0.310	0.053 0.909
OPTIC	0.2906 0.5748	0.3362 0.4557	0.5089 0.9578

ADEL outperforms all approaches in all datasets in terms of F1 micro score, while AGDISTIS/MAG is always the winner in terms of the F1 micro score. Notwithstanding, OPTIC outperforms all the other approaches on the NEEL2016 dataset. OPTIC also stays competitive on the NEEL2015 dataset, while it only outperforms FOX and OpenTapioca on the NEEL2014 dataset.

OPTIC performs better on the NEEL2016 dataset because the training set of our neural network model is from that dataset. However, our model general-

[12]http://www.cemeai.icmc.usp.br/Euler/index.html

izes well enough to stay competitive on NEEL2015. We envision that this happens because the linguistic patterns and the popularity of the entity candidates present in the NEEL2015 dataset are more similar to the NEEL2016 dataset than to the NEEL2014 dataset. Conversely, other approaches, except ADEL and AGDISTIS/MAG, perform better on the NEEL2014 dataset than they do on NEEL2016. Unfortunately, we do not have the gold standard for both NEEL2015 and NEEL2014 to discuss these results further.

Both ADEL and AGDISTIS/MAG employ a more robust selection of entity candidates than OPTIC. As mentioned in Section 3.1, a good method for selecting entity candidates should narrow as much as possible the set of entity candidates for each named entity mention, but with the guarantee that the correct entity is in the set. While AGDISTIS/MAG employs more indexes than OPTIC, including an index for acronyms, ADEL optimizes the implementation of their index using several datasets, including the NEEL2014, NEEL2015, and NEEL2016.

We executed the training of the DNN and the OPTIC EL ten times to capture their running times. For the training of the DNN, the average running time is 2:58 hours. For OPTIC EL, the average running time, considering all datasets, is 2:51 hours. For the steps of the OPTIC EL, namely preprocessing, selection of entity candidates, and disambiguation, the average running times are, respectively: 3.024 seconds per tweet, 0.766 seconds per tweets, and 0.128 per tweet.

5 CONCLUSION

In this work, we have shown that the joint use of knowledge embeddings and word embeddings in our OPTIC proposal for doing EL in microblog posts can produce results comparable with those of state-of-the-art approaches from the literature. The DNN architecture of OPTIC is relatively simple if compared with other architectures. Moreover, our training set is smaller than the training set used by most works in the literature. Thus, OPTIC has the potential to produce better results with a more sophisticated DNN architecture and a more significant training set.

We plan as future work to consider the textual similarity between each mention and the surface names of the entity candidates as well as the type of the named entity mentions (e.g., organization, person, place) for better-selecting entity candidates, among other minor extensions to OPTIC. We also aim to improve our index of surface names of entity candidates, since such an index seems to have been decisive for the better performance of ADEL and AGDISTS/MAG. Moreover, we aim to propose and use better-preprocessing methods for microblog posts, since we envision that this could significantly improve the performance of our approach. Lastly, we intend to make our model interpretable by using current algorithms for interpreting black-box models and understanding how the model handles incorrect cases. This way, we can optimize our model to handle those cases better, improving its performance.

ACKNOWLEDGEMENTS

This study was financed in part by the Brazilian Agency for Higher Education (CAPES) - Finance Code 001, projects: 88881.189286/2018-01 of the PDSE program, 88881.121467/2016-01 of the Senior Internship program and PrInt CAPES-UFSC “Automation 4.0”. It was also supported by the Brazilian National Council for Scientific and Technological Development (CNPq) (grant number 385163/2015-0 of the CNPq/INCT-INCoD program). Experiments were carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0).

REFERENCES

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, Berlin, Heidelberg.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA. ACM.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Cano, A. E., Rizzo, G., Varga, A., Rowe, M., Stankovic, M., and Dadzie, A.-S. (2014). Making sense of microposts:(# microposts2014) named entity extraction & linking challenge. In *CEUR Workshop Proceedings*, volume 1141, pages 54–60.

Chen, H., Wei, B., Liu, Y., Li, Y., Yu, J., and Zhu, W. (2018). Bilinear joint learning of word and entity embeddings for entity linking. *Neurocomputing*, 294:12–18.

Chong, W.-H., Lim, E.-P., and Cohen, W. (2017). Collective entity linking in tweets over space and time. In *European Conf. on Information Retrieval*, pages 82–94, Berlin, Heidelberg. Springer.

Fabian, M., Gjergji, K., and Gerhard, W. (2007). Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *16th Intl. World Wide Web Conf., WWW*, pages 697–706.

Fang, W., Zhang, J., Wang, D., Chen, Z., and Li, M. (2016). Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 260–269.

Fang, Y. and Chang, M.-W. (2014). Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics*, 2:259–272.

Ganea, O.-E., Ganea, M., Lucchi, A., Eickhoff, C., and Hofmann, T. (2016). Probabilistic bag-of-hyperlinks model for entity linking. In *Proc. of the 25th Intl. Conf. on World Wide Web*, pages 927–938. Intl. World Wide Web Conf. Steering Committee.

Ganea, O.-E. and Hofmann, T. (2017). Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2010). Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

Guo, Y., Qin, B., Liu, T., and Li, S. (2013). Microblog entity linking by leveraging extra posts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 863–868.

Guo, Z. and Barbosa, D. (2014). Entity linking with a unified semantic representation. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 1305–1310. ACM.

Han, H., Viriyothai, P., Lim, S., Lameter, D., and Mussell, B. (2019). Yet another framework for tweet entity linking (yaftel). In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 258–263. IEEE.

Han, X., Sun, L., and Zhao, J. (2011). Collective entity linking in web text: a graph-based method. In *Proc. of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM.

Hua, W., Zheng, K., and Zhou, X. (2015). Microblog entity linking with social temporal context. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1761–1775. ACM.

Huang, H., Cao, Y., Huang, X., Ji, H., and Lin, C.-Y. (2014). Collective tweet wikification based on semi-supervised graph regularization. In *ACL (1)*, pages 380–390.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017a). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Joulin, A., Grave, E., Bojanowski, P., Nickel, M., and Mikolov, T. (2017b). Fast linear model for knowledge graph embeddings. *arXiv preprint arXiv:1710.10881*.

Kalloubi, F., Nfaoui, E. H., et al. (2016). Microblog semantic context retrieval system based on linked open data and graph-based theory. *Expert Systems with Applications*, 53:138–148.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kolitsas, N., Ganea, O.-E., and Hofmann, T. (2018). End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529. Association for Computational Linguistics.

Laender, A. H., Ribeiro-Neto, B. A., da Silva, A. S., and Teixeira, J. S. (2002). A brief survey of web data extraction tools. *ACM Sigmod Record*, 31(2):84–93.

Lehmann, J., Bizer, C., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). DBpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165.

Li, Y., Tan, S., Sun, H., Han, J., Roth, D., and Yan, X. (2016). Entity disambiguation with linkless knowledge bases. In *Proc. of the 25th Intl. Conf. on World Wide Web*, pages 1261–1270. Intl. World Wide Web Conf. Steering Committee.

Li, Y. and Yang, T. (2018). Word embedding for understanding natural language: a survey. In *Guide to Big Data Applications*, pages 83–104. Springer.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187.

Liu, C., Li, F., Sun, X., and Han, H. (2019). Attention-based joint entity linking with entity embedding. *Information*, 10(2):46.

Martins, P. H., Marinho, Z., and Martins, A. F. (2019). Joint learning of named entity recognition and entity linking. *arXiv preprint arXiv:1907.08243*.

Moreno, J. G., Besançon, R., Beaumont, R., D'hondt, E., Ligozat, A.-L., Rosset, S., Tannier, X., and Grau, B. (2017). Combining word and entity embeddings for entity linking. In *European Semantic Web Conference*, pages 337–352. Springer.

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

Moussallem, D., Usbeck, R., Röeder, M., and Ngomo, A.-C. N. (2017). Mag: A multilingual, knowledge-base agnostic and deterministic entity linking approach. In *Proceedings of the Knowledge Capture Conference*, page 9. ACM.

Nickel, M., Rosasco, L., Poggio, T. A., et al. (2016). Holographic embeddings of knowledge graphs. In *AAAI*, pages 1955–1961.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.

Parravicini, A., Patra, R., Bartolini, D. B., and Santambrogio, M. D. (2019). Fast and accurate entity linking via graph embedding. In *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, page 10. ACM.

Plu, J., Rizzo, G., and Troncy, R. (2019). Adel: Adaptable entity linking. *Semantic Web Journal*.

Rizzo, G., Basave, A. E. C., Pereira, B., Varga, A., Rowe, M., Stankovic, M., and Dadzie, A. (2015). Making sense of microposts (# microposts2015) named entity recognition and linking (neel) challenge. In *# MSM*, pages 44–53.

Rizzo, G., van Erp, M., Plu, J., and Troncy, R. (2016). Neel 2016: Named entity recognition & linking challenge report. In *6th International Workshop on Making Sense of Microposts*.

Sevgili, Ö., Panchenko, A., and Biemann, C. (2019). Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop*, pages 315–322.

Shen, W., Wang, J., and Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Shen, W., Wang, J., Luo, P., and Wang, M. (2013). Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 68–76. ACM.

Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70. Association for Computational Linguistics.

Trani, S., Lucchese, C., Perego, R., Losada, D. E., Ceccarelli, D., and Orlando, S. (2018). Sel: A unified algorithm for salient entity linking. *Computational Intelligence*, 34(1):2–29.

Usbeck, R., Röder, M., Ngonga Ngomo, A.-C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., et al. (2015). Gerbil: general entity annotator benchmarking framework. In *Proceedings of the 24th international conference on World Wide Web*, pages 1133–1143. International World Wide Web Conferences Steering Committee.

Wang, Q. and Iwaihara, M. (2019). Deep neural architectures for joint named entity recognition and disambiguation. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–4. IEEE.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014a). Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1591–1601.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014b). Knowledge graph embedding by translating on hyperplanes. In *AAAI*, volume 14, pages 1112–1119.

Wei, F., Nguyen, U. T., and Jiang, H. (2019). Dual-fofe-net neural models for entity linking with pagerank. *arXiv preprint arXiv:1907.12697*.

Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). 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.

Zhong, H., Zhang, J., Wang, Z., Wan, H., and Chen, Z. (2015). Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 267–272.