Contents lists available at ScienceDirect

Information Processing and Management

Learning to rank implicit entities on Twitter

Hawre Hosseini, Ebrahim Bagheri *

Laboratory for Systems, Software and Semantics (LS³) Ryerson University, Toronto, Canada

ARTICLE INFO

ABSTRACT

Linking textual content to entities from the knowledge graph has received increasing attention in the context of which surface form representations of entities, e.g., terms or phrases, are disambiguated and linked to appropriate entities. This allows textual content, e.g., social user-generated content, to be interpreted and reasoned on at a higher semantic level. However, recent research has shown that at least 15% of social user-generated content do not have explicit surface form representation of entities that they discuss. In other words, the subject of the content is only implied. For such cases, existing entity linking methods, known as explicit entity linking, cannot perform linking because entity surface form is missing. In this paper, we investigate how implicit entities within social content can be identified and linked. The contributions of our work include (1) modeling the problem of implicit entity linking as a learn to rank problem where knowledge graph entities are ranked based on their relevance to the input tweet, (2) the introduction and systematic classification of appropriate features for identifying implicit entities, (3) extensive evaluation of the proposed approach in comparison with existing state of the art as well as performing feature analysis over proposed features, and (4) the qualitative assessment of the root causes for mislabeled instances in our experiments and careful discussion on how mislabeled entity links can be addressed as a part of future work. In our experiments, we show that our proposed features are able to improve the state of the art over the standard Precision at 1 (P@1) metric.

1. Introduction

The task of recognizing mentions of entities in text and linking them to an appropriate corresponding entity of a knowledge graph, e.g., DBpedia, is referred to as entity linking, which is now extensively studied for textual content of various types (Shen, Wang, & Han, 2014; Zhao, Wu, Wang, & Li, 2016) and is an important building block in a variety of downstream applications (Dalton, Dietz, & Allan, 2014; Ensan & Al-Obeidat, 2019). The main objective of this task is to connect between the entities' surface form in the text, i.e., their explicit mentions, and their corresponding knowledge graph representations. The main premise of existing entity linking techniques is that a surface form of the entity is present in the textual content that is being examined. As such, an entity linking technique would connect the observed surface form representation with the most likely entity from the knowledge graph. As an example, let us consider the following tweet 'Also, one person asked how Linklater chose the family he wanted to follow. He did not even know how to answer it'. This tweet consists of one possible entity that is explicitly observed, namely Richard Linklater, which can be linked to its corresponding entity on DBpedia: dbr:Richard_Linklater. Such links to knowledge graph entities would allow for semantic level interpretation of content and reasoning about the text using the associated knowledge graph entities (Ling, Singh, & Weld, 2015; Vo & Bagheri, 2019).

---

* Corresponding author.
  E-mail addresses: hawre.hosseini@ryerson.ca (H. Hosseini), bagheri@ryerson.ca (E. Bagheri).

While entity linking techniques have shown strong performance for cases when some variation of the surface form of the entity is present in the textual content, there has been fewer work that focus on identifying entities for cases when the surface form of the entity is absent from the content. The process for linking textual content to unobserved yet implicitly indicated entities to the knowledge graph is known as implicit entity linking (Perera, Mendes, Alex, Sheth, & Thirunarayan, 2016). Consider the earlier tweet, which explicitly mentioned Richard Linklater. While it is important to determine that dbr:Richard_Linklater is mentioned in the tweet, the more important fact is that the tweet is about the Boyhood movie, which has been implied in the tweet but not explicitly mentioned. The objective of implicit entity linking is to relate this tweet with dbr:Boyhood_(film). According to Hosseini, Nguyen, Wu, and Bagheri (2019), on average, 15% of tweets contain implicit mentions and according to Perera et al. (2016), 21% of tweets in the domain of movies and 40% of tweets in the domain of books, contain implicit references to entities. This translates into a large amount of information-rich content that cannot be readily processed by existing entity linking techniques.

The objective of our work in this paper is to perform *implicit entity linking* by building on earlier works in entity linking that have successfully adopted a learn to rank framework for linking *explicitly* observed surface form of entities with their corresponding knowledge graph entities (Ceccarelli, Lucchese, Orlando, Perego, & Trani, 2013; Hasibi, Balog, & Bratsberg, 2017; Meij, Weerkamp, & De Rijke, 2012; Xiong, Liu, Callan, & Hovy, 2017). Such works systematically define features that effectively rank entities based on their suitability for an observed entity surface form. For instance, in their seminal paper, Meij et al. (2012) introduce four categories of features, namely N-gram features, Concept features, N-gram+Concept features and Tweet features, in order to learn the association between tweets and entities explicitly mentioned in them so that entities can be ranked within a learn to rank framework. Similarly, we adopt a learn to rank framework for our work; however, unlike existing works that are focused on explicit entity linking, the goal of our work will be to systematically define and classify features that would be most suitable for the task of implicit entity linking. Our work differentiates itself from existing literature in that it needs to consider types of features that can find association between tweets and implied entities without having access to surface form representation of the entity. This strong requirement makes existing discriminatory features that are widely used in explicit entity linking less applicable. For instance, Meij et al. found that the most effective feature for ranking relevant entities is the equivalence of the surface entity representation in the tweet and the title of the entity in the knowledge graph (e.g., linking '...how *Linklater* chose the family he...' to dbr:Richard_Linklater, which have terminological equivalence.). Such a feature would not even be applicable in implicit entity linking because surface forms of entities are not present for implicit entities (e.g. dbr:Boyhood_(film) is not mentioned in the tweet and hence terminological equivalence cannot be used). Our work is innovative in that it will introduce features that take *contextual clues* into account for determining implicit entities.

The concrete contributions of our work can be enumerated as follows:

1. We introduce and systematically classify features that can be used for linking tweets to their implicitly mentioned entities within a learn to rank framework;
2. The introduced features are examined in the context of both explicit and implicit entity linking tasks and their performances are compared and critically evaluated under the conditions of these two different tasks;
3. The outcomes of the experiments not only provide an assessment of the performance of the features, individually and collectively, but also offer an in-depth error analysis to understand the root causes of why certain types of features perform better (or worse) for the task of implicit entity linking.

Our work in this paper is impactful in that it (1) provides a systematic classification of features that can be used for implicit entity linking; (2) compares feature performances across two different entity linking (implicit vs explicit) tasks; (3) is performed on publicly available gold standard datasets for both tasks and is fully replicable and hence future research can be built based on its foundations to introduce new and more effective features for implicit entity linking; and finally, (4) does not resort to reporting quantitative performance evaluation of the features, but rather, offers insight into why features perform well or poorly for the task of implicit entity linking.

The rest of the paper is organized as follows. Section 2 provides a review of the related literature covering the related work in both explicit and implicit entity linking. In Section 3, appropriate features for entity linking are introduced and systematically classified. In Section 4, we present our experimental setup, datasets, evaluation results, as well as feature and error analysis. Finally, the paper is concluded with some final remarks and pointers to future work in Section 5.

2. Related work

We provide an overview of the related literature to entity linking by noting that most of the work in this space has been focused on explicit entity linking while a few more recent techniques have considered implicit entity linking for Twitter content.

2.1. Explicit entity linking

There is a rich line of research on the task of entity linking focusing on recognizing explicitly mentioned entities and linking them to knowledge graph entities (Derczynski et al., 2015; Sarmento, Kehlenbeck, Oliveira, & Ungar, 2009). Approaches for addressing this task often consist of two main steps, the first of which identifies the potential entity mentions that can be linked to some entity in the knowledge graph by performing tasks such as domain dictionary lookup (Tran, Tran, Asmelash, & Jäschke, 2015), term expansion (Zou, Sun, Sun, Liu, & Lin, 2014), and abbreviated form expansion (Charton, Meurs, Jean-Louis, & Gagnon, 2014). The subsequent step links each identified mention to a candidate entity by utilizing a set of features that measure the relevance of

the mention and the candidate entities. These features can be classified into two main categories namely *context-independent* and *context-dependent* features (Shen et al., 2014).

Context-independent features are only based on the surface form of the entity mentions (Hua, Zheng, & Zhou, 2015; Meij et al., 2012; Tran et al., 2015). These features overlook the context where the entity mentions appear. For example, Guo, Chang, and Kiciman (2013) have defined a popularity feature for each candidate entity by utilizing the Wikipedia page view statistics associated with each candidate entity. Liu et al. (2013) have considered if the entity mention contains the title of the Wikipedia page without looking at other surrounding terms. Meij et al. (2012) have also introduced various context-independent features such as n-gram and concept features. Hua et al. (2015) offer two additional context-independent features, namely entity popularity and entity recency for tweet entity linking. These features describe the freshness of an entity and are measured by assessing the extent to which a burst in the number of tweets related to that entity in a short period of time have happened. Similarly, Ibrahim, Amir Yosef, and Weikum (2014) have leveraged the temporal importance of an entity using page view statistics of Wikipedia articles.

Context-dependent features take context surrounding the entity mentions into consideration to build additional features (Huang, Cao, Huang, Ji, & Lin, 2014; Li, Tan et al., 2016; Zou et al., 2014). For example, as a context-dependent feature, Liu et al. (2013) have calculated the cosine similarity between the bag of words representation of a tweet where the entity mention appears and the whole Wikipedia entity page associated with a candidate entity. Habib and Van Keulen (2012) have assumed that the set of appropriate entities for mentions appearing in the same tweet are those that are related to each other in the knowledge graph. As such, they perform entity linking through an Agglomerative clustering technique where clusters of related entities are identified from the knowledge graph and used for performing entity linking. Similarly, TagMe (Ferragina & Scaiella, 2010) which has shown to perform reasonably well on different datasets and for various benchmarks (Cornolti, Ferragina, & Ciaramita, 2013) benefits from a measure of collective agreement between the entity associated with a mention and all of the other entities identified in the tweet.

There have been works, which show that it is possible to perform entity linking with little contextual information. For instance, Li, Tan et al. (2016) propose a generative model which relies only on immediate surrounding textual content to associate a mention to an entity in the context of a linkless knowledge graph; showing that it would be possible to perform entity linking even if links between entities on the knowledge graph are not taken into account. However, researchers have primarily been interested in ways through which additional contextual information can be taken into account to improve the performance on entity linking. The work by Zou et al. (2014) is among such techniques, which employs a belief propagation strategy over the entity candidates' common links and topic distributions, as additional contextual information, to compute the degree of coordination between the observed mentions and their candidate entities. Shen, Wang, Luo, and Wang (2013) also identify and benefit from additional contextual information by assuming that each individual user has an underlying topic interest distribution over various named entities. As such, they perform entity linking by considering all of the tweets posted by the user and according to the user's topic interest distribution. In a similar vein, Huang et al. (2014) have proposed a semi-supervised method which is based on a graph regularization model to collectively identify and at the same time disambiguate mentions within a tweet.

While a wide variety of contextual features have been leveraged to perform tweet entity linking, such as user interests (Shen et al., 2013), temporal popularity (Hua et al., 2015), and location information (Fang & Chang, 2014), just to name a few, the issues of scalability and efficiency are two common weaknesses of existing work in the literature (Feng, Zarrinkalam, Bagheri, Fani, & Al-Obeidat, 2018; Ran, Shen, & Wang, 2018). Recently, Feng et al. (2018) have proposed a method for optimizing the task of entity linking in tweets by narrowing down the candidate entities. Based on their hypothesis, only a subset of candidate entities need to be considered for disambiguation in a tweet since there are certain sets of entities that are likely to be discussed by the users on Twitter. They achieve a better performance on accuracy as well as a reduced execution time for performing entity linking. Further, Ran et al. (2018) have formalized the tweet entity linking problem as a factor graph model, which has shown to be both effective and efficient under different experimental settings. The authors show that their approach has a linear time complexity.

2.2. Implicit entity linking

While the majority of the work in the literature focuses on explicit entity linking, there are few recent works that address implicit entity linking. To the best of our knowledge, Perera et al. (2016) were the first to introduce this task. They prepared and publicly shared a dataset of tweets containing implicit entities in two domains, namely movies and books. The authors leveraged contextual and factual knowledge from the knowledge graph in order to address the problem of implicit entity linking and based their graph-oriented model heavily on contextual knowledge derived from pooled tweets with temporal affinity. More specifically, for each input tweet containing an implicit entity of a known domain, a graph denoted as entity model network is built based on explicitly observed entities and the relationships between those entities derived from DBpedia's triple relations. Furthermore, the graph is complemented by knowledge acquired from one thousand tweets posted closest to the time of the tweet of interest referred to as *tweet clues*. The tweet clues are exploited in order to generate uni-grams and weighted phrases to be used in the entity model in case they exist as Wikipedia anchor texts or page titles. Finally, implicit entity linking is performed in two steps: (1) candidate selection, and (2) candidate ranking. The initial candidate set in the candidate selection phase includes those entities which have at least one edge with matching clue nodes and tweet clues in the graph. The top-k entities with the highest relevance are selected to be passed onto the ranking phase. The candidate ranking phase is done as a learning to rank task with an SVM^{rank} model using a pairwise approach (the input includes pairs of candidate entities).

More recent work in implicit entity is that by Hosseini et al. (2019), which publicly shares a gold standard dataset of tweets with implicit entities and proposes an ad-hoc retrieval framework in order to address the problem of implicit entity linking. In their work, a Markov Random Field-based (MRF) framework is exploited in order to rank the entities using five features; four of which
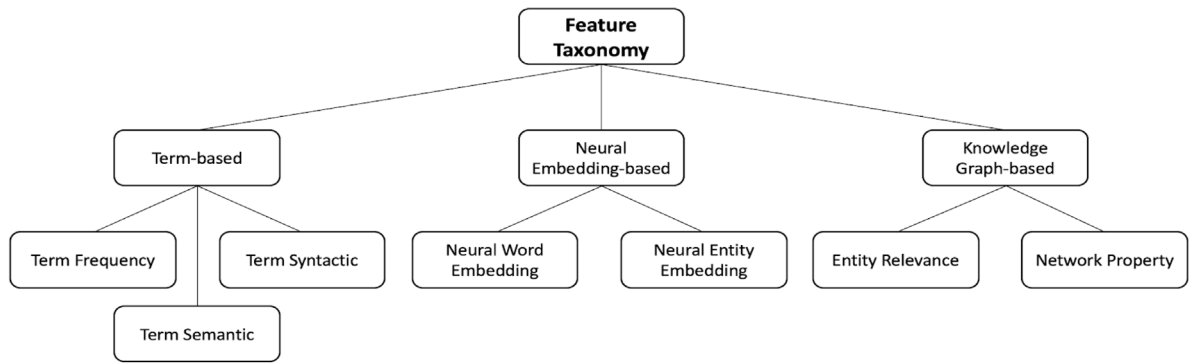
Fig. 1. Taxonomic view of the features introduced for implicit entity linking in our work.

are based on Sequential Dependence Modeling (SDM) (Metzler & Croft, 2005), and one of which is an embedding-based measure of similarity between entities. The dataset by Hosseini et al. consists of tweets from 6 domains, namely Person, Organization, Location, Event, Product (Device), and Work (Film and WrittenWork). This dataset is used in this work; further details can be found in the experiments section.

Seeking implicitly mentioned entities is being adopted more often in the academic community in order to address novel problems. The latest work to have used the notion of implicit entity linking is that of Huang, Yuan, Zhang, and Lu (2020). In this work, the authors aim to recognize illegal products from the large set of online products presented in an e-commerce platform. Such illegal products are advertised by the sellers through implicit references and with the help of 'camouflaged text'. While literature handles such cases as classification problems, the authors formulate the problem as one of implicit entity linking, where they endeavour to link a camouflaged product description to a known product. They do so by proposing a context representation model using BERT. The contextual information presented by their model includes three types. On this basis, they propose and exploit a symmetric metric for the calculation of the similarity score between a description and different products. The authors have evaluated their work on Hosseini et al.'s dataset and achieved better performance compared to standard baselines, i.e. Perera et al. (2016) and Hosseini et al. (2019).

In summary, Table 1 provides a structured review of the set of existing works that have been covered in this section for a clear depiction of the state of the art.

3. Research framework

This paper is inspired by strong work in explicit entity linking literature that adopt a learn to rank strategy to rank relevant entities for a given textual surface form representation (Hasibi et al., 2017; Meij et al., 2012). The core contribution of such work is the introduction and clear classification of features that would effectively identify relevant entities for a surface representation. In the context of our work, we are interested in features that would be able to rank relevant entities for a given tweet where the surface form of the entity has not been observed in the tweet. As explained in the related work section, the most effective features introduced in the related literature benefit from the association between the surface form representation of the entity and its representation on the knowledge graph. Clearly, while effective for explicit entity linking, these features are not applicable for implicit entity linking. As such, we introduce and systematically categorize three classes of features for entity linking that would be applicable to cases when the entity is implied. Fig. 1 provides an overview of our proposed set of features that are broadly categorized into (1) term-based, (2) neural embedding-based, and (3) knowledge graph-based features. Each of the feature categories are then broken down into finer-grained subcategories and introduced in the following subsections.

This work is performed in two major steps of Candidate Selection and Candidate Disambiguation. During the first, the set of relevant entities to an input tweet is retrieved. During the second, candidate disambiguation is performed leveraging the features that are introduced in the following subsections. Here, we start by explaining our candidate selection procedure for both implicit and explicit entity linking. Afterwards, the features exploited for the candidate disambiguation phase are elaborated on.

3.1. Candidate selection

This step is performed differently for implicit and explicit entity linking datasets. The reason is that in the explicit dataset, the task is to recognize mentions of the entities' titles through finding partial or complete matches. Whereas in implicit entity linking, the candidate set cannot be retrieved through textual matching since the target entity's title is not present inside the tweet. The candidate selection procedure for the implicit entity linking is adopted from the work of Hosseini et al. (2019), where candidate selection is performed as described in the following. With a Tweet *twt* as input, and the type of the sought entity's type ϑ, e.g., `wikidata:Film`, the candidate selection method retrieves a set of DBpedia entities f type ϑ which are relevant to *twt*. In doing so, the explicit entities present in the input tweet are extracted leveraging a standard entity tagger, e.g., TagME. Using the extracted explicit entities, the

Table 1

Summary of most relevant related work.

Work	Dataset	Baselines	Performance	Framework	Novelty	Strength/Weakness	Insights
Guo et al. (2013)	Annotated a dataset of approximately 1500 tweets.	- TagMe - Cucerzan, 2007	Outperforming baselines by 15% in absolute F1	Structural SVM algorithm	End-to-end entity linking, merging mention detection and entity disambiguation steps	- Efficient and simple disambiguation by linking to the most popular entity - Introducing entity popularity as a very powerful non-contextual feature	- Entity-related features such as entity popularity are crucial. - Combining entity-specific and mention-specific features results in a better model; mention specific features are not available in implicit entity linking.
Tran et al. (2015)	Collected a dataset with 6,965 Test$Entity, Hashtag) pairs	- TagMe - Wikimine - Meij - Kauri	Outperforming baselines by 17%–28% in P@5, P@15, and MAP	Linear combination of features	Leveraging Wikipedia temporal information, i.e., edit history and page view logs	- Dynamic annotation of trending hashtags - Leveraging Wikipedia edit history and page view logs for the first time	Trending hashtags are associated with an increased public attention to certain entities
Liu et al. (2013)	Meij	- Wikify! - Meij	Slightly outperforming baselines on precision, recall and F1	A greedy hill-climbing approach for training	Collective resolution of a set of entity mentions	- Integrating three similarity types: Mention-entry, Entry-entry, and Mention-mention - Using edit distance as a feature - Handling of OOV mentions with the help of other similar mentions	Collective inference handles the problems of limited length of tweets and rich entity mention variation
Hua et al. (2015)	Collected large scale datasets of tweets.	- TagMe - Shen et al. 2013	Outperforming baselines on efficiency and efficacy; accuracy metric used for effectiveness evaluation	Algorithms to detect user interest through followee-follower network properties	- Novel approaches to user interest and entity recency estimation - Measuring user interest by social interactions rather than tweet streams	Computationally more efficient model to address microblog properties of limited length, error-proneness, and informal language	Social and temporal contexts are significant in linking entities for microblog text.
Ibrahim et al. (2014)	Microposts2014 experimental corpus	AIDA	Outperforms baselines by 13% in precision	SVM used for learning parameters		Proposing AIDA-Social NEL system. Handling dynamic content of tweets, abbreviations, and tweets' insufficient context using three techniques: Mention normalization, Context expansion, and Temporal entity importance	A combination of all techniques yields best results, while the best of them when used individually is the Temporal Importance.
Ferragina and Scaiella (2010)	- Wiki-Disamb30 - WikiAnnot30 - IITB	Milne& Witten	Outperforming baselines on both precision and recall	Classifiers built with two features calculated collectively	Adapting techniques to work with microblog text.	Authors propose TagMe NEL system. This system can annotate short and poorly written text snippets on-the-fly.	The complexity increases with higher numbers of anchors to consider for collective linking; however, TagMe performs with reasonable time complexity for short texts.
Huang et al. (2014)	Meij	- TagMe - Meij	Outperforming baselines by 5% in absolute F1 gain	Semi-supervised graph regularization model	Proposing a collective approach for wikification of tweets through a semi-supervised graph regularization model.	- Rich Wikipedia lexicon using several resources - "Global evidence" is leveraged from multiple tweets which makes collective inference possible	Joint mention recognition and disambiguation can reduce annotation time complexity.
Fang and Chang (2014)	Developed a dataset of 1.8M tweets	Guo et al. 2013	Outperforming baselines by 10% in F1 score	Weak supervision to integrate spatio-temporal clues into existing linker	Utilizing spatial and temporal clues for linking entities in microblogs	- Offline context expansion - Evaluation of entity linking for both information extraction as well as information retrieval needs	Spatio-temporal properties play a key role in linking entities in microblogs.

DBpedia knowledge graph is queried for triples whose subject (or object) match one of the extracted explicit entities and the object (or subject) is of type `rdf:type` ϑ. The retrieved entities form a candidate entity set. This approach will identify entities related to the explicit entities within the input tweet and are of the specific type that we are looking for. The problem of *entity sparsity* happens often times due to the informal language of tweets as well as their short length, resulting in cases where only a few and in many cases only one entity is retrieved based on entity tagging a tweet and therefore, the search space for querying and recognizing candidate entities would be too narrow. To relieve this problem, a standard procedure called *context expansion* is performed where a set of relevant tweets to the input tweet are pooled and added as context. For the dataset used, Hosseini et al. (2019) report that there are as few as only 1.77 explicit entities related to each tweet, which was then increased to 19.02 as a result of this context expansion procedure. We use Twitter API to pool tweets with category-related keywords during time interval closely related to the input tweet. For searching such tweets, we look for Twitter posts that include the surface form of the explicit entities extracted from *twt* as well as a mention of the `dbp:label` for ϑ, e.g., `wikidata:Film dbp:label` Film. From among all the retrieved entities, we choose the top K entities sorted based on frequency of appearance.

As described earlier, the explicit entity linking portion of our work is inspired by Meij et al. (2012). In this work, concept linking is done at tweet level rather than entity disambiguation at mention level, hence very similar to our implicit entity linking setting. In their dataset, a tweet can be linked to more than one concept, which are mentioned in the tweet. For the purpose of the present study, we are interested in re-ranking of the candidate entity set and the differences between explicit as well as implicit entity linking with that regards, rather than high recall of the candidate selection phase. A simple lexical matching would return a list of entities whose titles are found in the input query. In this work, following the lexical matching procedure used in Meij et al. (2012), we extract all possible n-grams from the input tweet and calculate the Jaccard similarity between the n-grams and Wikipedia titles as well as the Wikipedia anchors. We retain top k retrieved entities in order to do re-ranking. For those n-grams whose constituent n-grams happen to be within the top-k, we retain the match between the n-grams with a higher number of n and discard the others. The re-ranking is done based on the feature vectors extracted from the knowledge graph contents and well as the input query's.

3.2. Term-based features

The first class of features that we present are based on how terms appear within a tweet. The underlying assumption behind these features is that while the surface form representation of the target entity is missing in implicit entity linking, it is likely that similar contextual term distributions are observed in the tweet and the representation of the target entity on the knowledge graph. For instance, referring back to the earlier example tweet about the Boyhood movie, although the movie itself is not mentioned but the director name is observed, which can be considered to be a strong discriminatory indicator for the movie. We define three subcategories of term-based features based on (1) term frequency, (2) term syntactics, and (3) term semantics. Table 2 provides the categorization and description of term-based features introduced in this paper.

3.2.1. Term frequency features

Existing works on explicit entity linking have shown that the frequency of a term observed in the surface form representation of an entity on the knowledge graph is a strong indicator for entity relevance (Meij et al., 2012). We expand this notion to consider any uni-gram, or ordered/unordered bi-gram observed in the tweet since the surface form representation does not exist in implicit entity linking. In addition to n-grams in the tweet, explicitly observed entities in the tweet are also taken into account. We define various measures of frequency over the unigrams, ordered/unordered bi-grams and explicit entities as shown in Table 2. The two most popular measures of frequency are Term Frequency (TF) and Term Frequency discounted with Inverse Document Frequency (TF-IDF). We additionally include the three features defined in the Sequential Dependence Model (SDM) by Metzler and Croft (Metzler & Croft, 2005) (referred to as potential functions in the authors' original paper).

Earlier research in explicit entity linking has shown that the presence of the title of the entity or its subset is a strong indicator for the relevance of the entity to the tweet (Hachey, Radford, Nothman, Honnibal, & Curran, 2013). As such, we define the TitleContainsTweet feature, which measures whether a substring of the tweet appears in the title of the candidate entity. This can be an effective feature in the case of explicit entity linking since surface forms of entities can appear in the text. We study the effectiveness of this feature in the context of implicit entity linking but hypothesize that it will not perform as effectively as it does in the explicit entity task, since implicit references do not contain surface forms of entity titles. Furthermore, several researchers have already suggested that when a tweet includes hyperlinks to external web pages, the content observed in those web pages become relevant for understanding the semantics of the tweet (Yu, Zheng, Yang, & Jin, 2014). As such, we hypothesize that the highly frequent explicitly observed entities on external web pages have a higher likelihood of appearing and being relevant to the tweet. Therefore, we define the URLEntityCount feature, which measures the frequency of explicitly observed entities on external web pages linked from the tweet. Such frequency will serve as a prior probability for the relevance of entities to the tweet. Based on a similar intuition, we define the EntFirstOccur feature to measure whether, and if so where, an explicitly observed entity in the tweet has appeared in the KG representation of a candidate entity. Our assumption is that the candidate entity is more likely to be relevant to the tweet if the explicit entities in the tweet are also observed in the KG representation of the candidate entity and are mentioned early in the description. We further introduce an additional feature similar to the EntFirstOccur feature, named Presence of Anchor in Candidate entity (PARC), which looks for anchor texts in the KG representation of the candidate entity. Wikipedia anchor texts are clickable phrases in Wikipedia which are linked to other Wikipedia pages. For instance, the phrase '44th president of the United States of America' is an anchor text linked to Barack Obama. The presence of an entity anchor within the tweet can be an indicator of its relevance to the entity referred to by the anchor.

In order to define features that can provide additional prior probability over knowledge graph entities, we adopt the finding by existing work that suggest entities with shorter titles have a lower likelihood of being used and adopted by users (Anastácio, Martins, Calado, et al., 2011). On this basis, we define two additional features, namely TitleCharLength and TitleTermCount. The first feature calculates the number of characters in the entity title and the second counts the number of terms in the entity title. These two features also form priors on the likelihood of the entity being relevant to the tweet given how users adopt entities depending on their title length. We hypothesize that such features would be more appropriate for explicit entity linking where the surface form of the entity needs to be present. It is less likely that the length of the entity title would impact the decision of the users to discuss the entity in a tweet when the entity is only implied.

Table 2

Description and categorization of the term-based features proposed for entity linking. Note, term refers to unigram, ordered/unordered bigram and explicitly observed entity in the tweet.

Sub-category	Name	Description
Frequency	TF	Frequency of tweet terms in the KG representation of candidate entity
	TF-IDF	Discounts tweet terms with its inverse frequency of tweet terms in the KG representation of candidate entity
	SDM	SDM-based potential functions defined over tweet terms and KG representation of candidate entity
	TitleContainsTweet	Shows if the tweet terms and the KG representation of the candidate entity have terms overlap
	EntFirstOccur	Relative place of the first occurrence of tweet explicit entity Within the KG representation of the candidate entity
	PARC	Presence of an anchor text referring to a candidate entity in the tweet
	URLEntityCount	Frequency of an explicit entity appearing on web pages linked within the tweet
	TitleTermCount	Number of terms in title of the KG representation of candidate entity
	TitleCharacLength	Character length of title of KG representation of candidate entity
Syntactic	EntDependent HeadCor	Correspondence of chunk root heads obtained from dependency parse of the tweet explicit entity compared to the occurrences of the entity inside candidate entity KG representation
	DependentHeadCor	Dependency parse similarity of the tweet to the dependency parse of the KG representation of the candidate entity (correspondence of chunk root heads)
	EntChunkingCor	Correspondence of the chunks where tweet's explicit entities appear in the tweet and the KG representation of the candidate entity (calculated using chunk root dependency)
	ChunkingCorFirst	Chunking similarity of the tweet to the first sentence of KG representation of the candidate entity (calculated using chunk root dependency)
	ChunkingCorAll	Chunking similarity of the tweet to the complete KG representation of the candidate entity (calculated using chunk root dependency)
Semantic	Synset_pathSim	Similarity of a tweet to the KG representation of the candidate entity based on the WordNet path similarity
	Synset_LCSim	Similarity of a tweet to the KG representation of the candidate entity based on the Leacock–Chodorow similarity
	Synset_WUPSim	Similarity of a tweet to the KG representation of the candidate entity based on the Wu–Palmer similarity
	CosineSimilarity	Cosine similarity of the tweet and KG representation of the candidate entity based on synsets

3.2.2. Term syntactic features

The second set of our proposed term-based features benefits from the syntactic structure of how content is presented on the KG representation of the candidate entity. We hypothesize that each entity has a higher likelihood to be adopted and used in similar syntactic roles within the sentences they appear in. Therefore, the association between the syntactic structure of the sentence in the tweet compared to the syntactic structure of the KG representation of the candidate entity can be considered to be a measure of relevance for the candidate entity to the tweet. In order to access syntactic structures, we adopt dependency parsing, which identifies head words and words that modify the head words in a sentence. Based on the dependency parse, we define two features, namely the EntDependentHeadCor and DependentHeadCor features. The purpose of the first feature is to find correspondence of chunk root heads between the tweet's explicitly observed entities and their occurrence in the KG representation of candidate entity. This is accomplished through correspondence of chunk root heads for chunks where the explicit entities happen in the tweet and the KG representation of candidate entity. The second feature adopts a similar strategy to the EntDependentHeadCor feature but this time it does so by parsing the whole tweet and comparing it to the dependency parsing of the candidate entity's dependency parse. As such, while the first feature relies on the presence of an explicit entity and hence, not applicable to the implicit entity linking task, the second feature can be applied for both implicit and explicit entity linking tasks.

We further define additional features based on chunking where we compute chunking output similarity of the tweet to that of the KG representation of the candidate entity through measuring correspondence of chunk root dependencies. These features include EntChunkingCor, ChunkingCorFirst, and ChunkingCorAll. The first feature, EntChunkingCor, is based on the similarity of

chunks where explicitly observed entities in the tweet appear in the KG representation of the candidate entity. The second and third features consider chunk root dependency similarity between all tweet chunks and chunking output of the KG representation of the candidate entity. In the second feature, ChunkingCorFirst, we only take the first sentence of the KG representation of the candidate into account while in the third feature, ChunkingCorAll, all sentences are considered. The reason we distinguish between the first sentence and the whole set of sentences in the KG representation is the fact that the first sentence of the KG representation often provides a concise description of the entity that compactly captures the essence of the entity and hence can avoid issues of potential topic drift (Audeh, Beaune, & Beigbeder, 2014) when longer sets of sentences are considered.

3.2.3. Term semantic features

The idea of the third set of features is to use the underlying semantics behind the term representation of the tweet and the KG representation of the candidate entity. In order to enable semantic comparison between the terms in the tweets, we use WordNet, which consists of sets of cognitive synonyms (synsets). We capitalize on synsets primarily because they allow us to identify similar content that have been expressed using different terminological representations. This is common especially when trying to associate content written informally on Twitter with formal text as communicated on encyclopedic knowledge that can be found on sources of the KG such as Wikipedia. We define three main features based on WordNet synsets, namely Synset_pathSim, Synset_LCSim, and Synset_WUPSim.

The idea behind the three features is that the similarity between the terms of a tweet and the KG representation of a candidate entity is an indication of the relevance of the candidate entity to the tweet. The use of WordNet synsets would overcome the vocabulary mismatch problem when computing the similarities. The first feature, Synset_pathSim, calculates the similarity of tweet and a candidate entity using the shortest path by which the senses in the hypernym–hyponym taxonomy are connected to each other. In the second feature, Synset_LCSim, the similarity between a tweet and the KG representation of the candidate entity is calculated based on Leacock–Chodorow (LC) Similarity (Leacock, Chodorow, & Miller, 1998). The LC similarity metric captures the similarity between senses based on the shortest path that connects the senses and the maximum taxonomy depth of where the senses occur. The third feature is quite similar except that it uses Wu–Palmer Similarity (Wu & Palmer, 1994), which determines similarity between senses based on the taxonomy depth of the senses and the taxonomy depth of their Least Common Subsumer (most specific ancestor node). These semantic features that are based on the similarity of the tweet and KG representation of the candidate entity will find relations between tweets and candidate entities that would not be identified based on term frequency or term syntactic features when vocabulary mismatch happens. Finally, we also measure the similarity between the tweet and the KG representation of the candidate entity based on the cosine similarity of their synset representation and use this as the last feature of the term semantic feature category.

3.3. Neural embedding-based features

The second set of features that we propose in this paper benefit from the characteristics of neural embedding representations. The information retrieval community has already exploited neural representations as they provide meaningful semantic relations between terms in the embedding space, which can be used to enhance retrieval effectiveness. Similar to our term semantic features whose goal is to capture some form of semantic relationship between tweets and candidate entities, we define neural embedding-based features to measure tweet and candidate entity associations beyond term co-occurrence and using the semantics of the terms that compose them. We define two subcategories of neural embedding-based features based on (1) word embeddings, and (2) entity embeddings. Table 3 provides the categorization and description of the neural embedding-based features introduced in our work.

3.3.1. Neural word embedding features

The first set of embedding features focuses on the representation of tweets and candidate entities based on the words that appear in each. We adopt the neural embedding representation of the words that are observed in the tweet and the KG representation of the candidate entity to measure degrees of similarity. In the first feature, WordSim, we obtain representations for the tweet and the candidate entity by averaging the embedding representation of the words that are observed in them. This would produce two vectors whose similarity can be computed through cosine similarity, as seen in the following equation:

$$WordSim = CosineSimilarity(\frac{1}{|D^t|}\sum_{i=1}^{|D^t|}\vec{v}_i^w, \frac{1}{|D^c|}\sum_{j=1}^{|D^c|}\vec{v}_j^w) \tag{1}$$

where D^t and D^c are tweet t and candidate entity c textual representations, $\vec{v}^w$ is the word vector obtained from a word embedding model.

We further define a variation of the WordSim feature, referred to as MentionSim. We hypothesize that when explicit entities are identified in a tweet, they play an essential role in communicating the subject matter expressed by the tweet. As such, in the MentionSim feature, we apply WordSim but only to the surface form of the explicit entities that have been observed in the tweet and the KG representation of the candidate entity, as in the following:

$$MentionSim = CosineSimilarity(\frac{1}{|E^t|}\sum_{i=1}^{|E^t|}\frac{1}{|E_i|}\sum_{j=1}^{|E_i|}\vec{v}_j^w, \frac{1}{|D^c|}\sum_{k=1}^{|D^c|}\vec{v}_k^w) \tag{2}$$

where E^t is the set of explicit entities observed in the tweet t.

Table 3

Description and categorization of the neural embedding-based features proposed for entity linking.

Sub-category	Name	Description
Word embedding	WordSim	Similarity of average of all word vectors in query and in document/title
	MentionSim	Similarity of average of mentions' word vectors (entity surface form) and average of document/title word vectors
	WordMov	Word movers distance between query text and document/title text
Entity embedding	AvgEntContent	Similarity of average of word vectors for explicit entities (entity title on KG) within the tweet and average of word vectors of explicit entities within the candidate entity KG content
	WMDEntContent	Similarity of word vectors for explicit entities within the tweet and the candidate entity based on their KG representation using word mover's distance
	AvgEntEmb	Similarity between the embeddings of explicitly observed entities in the tweet and the explicit entities observed within the representation of the candidate entity based on averaging of entity vectors
	WMDEntEmb	Similarity between the embeddings of explicitly observed entities in the tweet and the explicit entities observed within the representation of the candidate entity based on word mover's distance

While averaging the embedding representations of the words observed in a text for building a representation has been widely adopted in the literature (Bagheri, Ensan, & Al-Obeidat, 2018; Meij et al., 2012), there are other methods for computing similarity between sets of embeddings without having to perform averaging over the embeddings. One of the better known methods is the *word movers distance* method, which finds the optimal cost of moving all embeddings from one set to another within the embedding space. Based on word movers distance, we define a third word embedding feature, WordMov, which computes the similarity of the tweet and the KG representation of the candidate entity based on the word movers distance of the embeddings of their words.

3.3.2. Neural entity embedding features

The second set of neural embedding features are dedicated to implicit entity linking and are not applicable to explicit entity linking as they assume that explicit entities within the tweet have already been identified and appropriately linked. On this basis, the two features in this subcategory compute measures of similarity between the tweet and the candidate entity based on the embedding representation of the explicit entities that have been observed in the tweet.

In the first feature type of this category, we continue to adopt the word embedding model but in this case, we employ the KG representation of the explicit entity observed in the tweet with the KG representation of the candidate entity. This is intended to measure the possible association between the tweet and the candidate entity based on the similarity of the content representation of both explicit entities and the candidate entity on the KG. Similar to neural word embedding features, we define two features based on the average of the embeddings (AvgEntContent) as well as the word mover's distance (WMDEntContent). The difference between AvgEntContent and MentionSim is that the former (AvgEntContent) takes explicit entities inside the candidate KG documents into account to represent the candidate entities, while the latter (MentionSim) takes all of the words inside the documents to do so. Based on this, the formula for AvgEntContent is as follows:

$$AvgEntContent = CosineSimilarity(\frac{1}{|E^t|} \sum_{i=1}^{|E^t|} \frac{1}{|E_i|} \sum_{j=1}^{|E_i|} \vec{v}_j^w, \frac{1}{|E^c|} \sum_{k=1}^{|E^c|} \frac{1}{|E_k|} \sum_{l=1}^{|E_k|} \vec{v}_l^w) \qquad (3)$$

where E^c is the set of explicit entities observed in the KG representation of candidate entity c.

There have been impressive recent works that learn neural representations for knowledge graph entities based on the content similarity and structural association of entities on the knowledge graph (Wang, Mao, Wang, & Guo, 2017). While in the previous set of features, we adopt word embeddings, it is also possible to use KG entity embeddings to compute similarity between the tweet and the candidate entity. Similar to the previous two features, we adopt explicitly observed entities in the tweet. However, this time, we compare the explicitly observed entities in the tweet to the explicitly observed entities in the KG representation of the candidate entity based on entity embeddings (as opposed to the word embeddings of the entity content). We offer two features depending on how similarity between sets of embeddings are computed, namely AvgEntEmb for when average entity representations are used and WMDEntEmb for the case when word movers distance is employed. The calculation of AvgEntEmb is done based on the following:

$$AvgEntEmb = CosineSimilarity(\frac{1}{|E^t|} \sum_{i=1}^{|E^t|} \vec{v}_i^e, \frac{1}{|E^c|} \sum_{j=1}^{|E^c|} \vec{v}_j^e) \qquad (4)$$

where E^t denotes the set of explicit entities inside the tweet, E^c denotes the set of explicit entities inside the candidate entities' KG representation, and $\vec{v}^e$ stands for the entity embedding that is retrieved from entity embedding models.

Table 4

Description and categorization of the knowledge graph-based features proposed for entity linking.

Sub-category	Name	Description
Entity relevance	ViewCount	Number of times candidate entity was visited in a specific recent time frame
	ClickCount	The proportion of times users have navigated from the tweet's explicit entities to a candidate entity over the total number of visits to the candidate entity
	EntCoOccur	Co-occurrence of tweet explicit entities with the candidate entity on the KG
Network properties	Inbound	Number of entities on the KG linking to the candidate entity
	Outbound	Number of entities on the KG linked from the candidate entity
	Redirect	Number of redirect pages linking to the candidate entity on the KG
	Cat	Number of categories associated with the candidate entity on the KG
	Betweenness	The betweenness centrality of the candidate entity on the KG
	PageRank	The PageRank value of the candidate entity on the KG

3.4. Knowledge graph-based features

The objective of the third set of features is to extract actionable information from the content and structure of the knowledge graph to decide about the association of a candidate entity and a tweet. There are several aspects of the knowledge graph that can be used for defining features including its network structure, textual content describing the entities as well as provenance information. Much of this information can be used to learn prior probabilities for (1) the likelihood of two entities being related to each other based on how frequently they have been observed in similar context or how close they are on the knowledge graph structure, and (2) the likelihood of entities to serve as an implicit or explicit entity depending on external contextual information. We define two subcategories of knowledge graph-based features based on (1) entity relevance features, and (2) network property features. Table 4 provides the categorization and description of the knowledge graph-based features introduced in our work.

3.4.1. Entity relevance features

In the first set of features, we introduce features that compute a prior probability for the likelihood of an entity being a possible entity within a tweet. One of the underlying hypotheses for our features is based on the nature of how tweets are generated. In many cases, people tweet about a current event or a recent incident. Therefore, for such cases, those entities that have received more recent attention on the knowledge graph become more likely candidates. In order to formalize this idea of recency as a measure of likelihood, we exploit two sources of provenance information from the knowledge graph, namely entity view count and click count. We define two features based on this provenance information, i.e., ViewCount and ClickCount, which show how many times an entity's entry on the knowledge graph has been viewed by users and how many times users have navigated to this entity's entry from other entities, respectively. For the ClickCount feature, we add an additional constraint where we only count the proportion of the number of times that the candidate entity's entry has been visited from an explicit entity observed in the tweet of interest over the total number of navigations to this entity. This way, we ensure that we capture the degree of relevance of the candidate entity to the tweet in question.

We additionally define a measure of relevance between the tweet and candidate entity based on how frequently has the explicit entities in the tweet being concurrently observed with the candidate entity on the knowledge graph. We hypothesize that the more frequently a candidate entity is observed with the explicit entities of the tweet, the more likely it would be for that candidate entity to be relevant to the tweet. We specifically measure the number of times that the candidate entity has been observed with at least one of the explicit entities of the tweet over the total number of times that the candidate entity has been observed within the knowledge graph as a measure of relevance, denoted as EntCoOccur.

3.4.2. Network property features

In the second set of knowledge graph-based features, we adopt information from the structure of the knowledge graph so as to extract possible priors for the candidate entities. We hypothesize that the degree to which an entity is integrated within the knowledge graph is an indication of its importance to the context in which it appears in. To this end, we consider inbound and outbound links to/from the candidate entity's KG entry as a sign of how well the entity is integrated in its local neighborhood within the knowledge graph. A higher number of inbound and outbound links is an indication that the entity is considered more important to other surrounding entities. We additionally consider the concept of 'redirects' on the knowledge graph and define a

Table 5

Statistics for implicit entity linking gold standard dataset introduced by Hosseini et al. (2019).

Type	Implicit	Explicit	No Entity (NE)
Count	1,345	2,483	3,842
Average explicit entity per tweet	2.53	2.68	0
Average token per tweet	26.16	21.96	16.60

feature that measures how many redirect entities lead to the candidate entities. The set of redirect links indicates the possible set of different surface form representations that refer to the same entity. A large redirect set shows that a broader range of surface forms relate to the entity and hence increases the likelihood of an entity being relevant to a larger number of surface form representations. Similar to the redirect feature, we also benefit from knowledge classification systems on the knowledge graph (e.g., categories on Wikipedia) to see the diversity of contexts each entity can appear in. The higher the number of categories an entity belongs to, the higher likely it would be for the entity to appear in different situations.

While the four introduced features show how well an entity is integrated with local information surrounding the entity on the knowledge graph, they do not provide a global view of how the entity is integrated within the knowledge graph as a whole. Therefore, we adopt network centrality measures to define two additional features by employing PageRank and Betweenness Centrality. These two measures are selected to serve as features in our work as they are (1) widely adopted in the literature for analyzing knowledge graphs (Bagheri, Arabzadeh, Zarrinkalam, Jovanovic, & Al-Obeidat, 2020), and (2) able to expose an entity's influence on other knowledge graph entities, which is key for measuring the likelihood of an entity being relevant to different contexts.

4. Experiments

The main objective of our work in this paper is to propose and systematically classify features that could be appropriate for the task of implicit linking. As such, we will evaluate our work and compare it with existing baselines both quantitatively and qualitatively. We will also show how the proposed features perform on the explicit entity linking task and that the features that are suitable for each task can be different depending on the task due to the dissimilar nature of the two tasks.

4.1. Datasets

For the implicit entity linking, we adopt the gold standard dataset proposed in Hosseini et al. (2019), which has been specifically curated for this task. According to Hosseini et al. (2019), this dataset is inspired by traditional NERC tasks with a two-level fine and coarse grained hierarchy. This taxonomy includes 6 coarse-grained entity types, namely Person,[1] Organization,[2] Location,[3] Product/Device,[4] Event,[5] and Work.[6] These classes as well as the fine-grained classes associated with each are based on the DBpedia taxonomy. While the first level of the taxonomy, i.e., the coarse-grained classes, is designed to retain the elements of traditional NERC taxonomies, the second level is focused on specific entities. There are three major categories of tweets in this dataset: tweets including implicit entity mentions, denoted as *Implicit* tweets; tweets with explicit entity mentions, which are denoted as *Explicit* tweets; and tweets without either of the two mention types, denoted as *No Entity (NE)*. The statistics of this dataset is presented in Table 5. Tweets in the first category, i.e., implicit tweets, are also labeled with the list of explicit entities observed in the tweets. One of the limitations of this gold standard seems to be the fact that while some of the implicit tweets contain implicit references to more than one entity, only the main, core theme or implicitly referenced entity is tagged as the target. In this paper, the implicit tweets in this dataset (the Implicit column in Table 5) are used in our experiments.

According to Hosseini et al. (2019), the collection and manual tagging of tweets in the dataset used in our experiments was done as follows. First, Twitter API was used for performing repeated random sampling in order to arrive at a balanced ratio between the three tweet types. This is performed so that the dataset reflects the actual distribution of those three types in the real world. In doing so, 400 tweets were evaluated by three human judges with the following ratios: 35% Explicit, 15% Implicit, and 50% NE. In order to form the pool of tweets to be tagged, a large pool of tweets in a four-month time frame from October 2017 to January 2018 were collected. The choice of a four-month period was to avoid entity drift Masud et al. (2010). Three human annotators manually tagged the tweets for the three category types of Implicit, Explicit, and NE. The final tweets of the dataset were selected upon consensus among the human annotators. The dataset in the form of tweet IDs, user (tweeter) IDs, category, and the target entity labels is publicly available.[7] It is also noteworthy that this dataset includes the 327 implicit tweets that were introduced by Perera et al.'s gold standard dataset.

---

[1] http://mappings.dbpedia.org/server/ontology/classes/Person.
[2] http://mappings.dbpedia.org/server/ontology/classes/Organisation.
[3] http://mappings.dbpedia.org/server/ontology/classes/Place.
[4] http://mappings.dbpedia.org/server/ontology/classes/Device.
[5] http://mappings.dbpedia.org/server/ontology/classes/Event.
[6] http://mappings.dbpedia.org/server/ontology/classes/Work.
[7] https://github.com/HawreH/Implicit-Entity-Recognition-and-Linking-in-Tweets-Resources-and-Dataset.

Furthermore, given we are interested in analyzing the suitability of our proposed features for explicit entity linking, we adopt the gold standard dataset proposed by Meij et al. (2012). The choice of this dataset for explicit entity linking was primarily motivated by the fact that the work by Meij et al. is a seminal work that adopts a similar learn to rank strategy to our approach. Here, we describe the dataset and how it was collected. The authors performed random sampling to select users from the 'verified accounts' Twitter list so that they avoid collecting non-informative tweets. Then, 20 tweets were retrieved from each of the selected users. For manually annotating the tweets, the authors asked two human annotators to tag 562 tweets, in an annotation interface provided to them. The average term length of the tweets was 36.5. The annotators were asked to identify concepts 'contained in, meant by, or relevant to the tweet'. Of the initial 562 tweets, 419 were kept for further analysis; others were discarded due to being ambiguous or erroneous. On average, the 419 tweets received 2.17 concepts per tweet.

4.2. Experimental setup

Here, for the sake of reproducibility, we clearly describe the process for extracting the introduced features. For those features that require explicit entities within the tweet, we employ an entity linker that has shown to have a strong performance on tweets (Cornolti et al., 2013), namely TagMe (Ferragina & Scaiella, 2010), to annotate the tweets in the gold standard. For Wikipedia textual content, we extract entities by processing Wikipedia dumps. We have downloaded and worked with Wikipedia dump from December 20, 2018. In order to extract entity inlinks, outlinks, and redirects as well as the number of categories associated with each entity, we exploit the Wikipedia API. We measure betweenness centrality and PageRank from the knowledge graph based on the Wikipedia dump and using the networkx package in Python. We use the spaCy NLP framework for dependency parsing (Honnibal & Johnson, 2015) in features that need chunking as well as dependency parse of the textual inputs. The ViewCount feature considers the page view statistics pertinent to the month during which the tweet was posted. Finally, in order to extract the PARC feature, we build a mapping from anchors on Wikipedia to entities. We extract this mapping by processing the Wikipedia dump.

In the neural embedding-based features, we require access to both word and entity embeddings. For the word embeddings, we adopt the widely used Google News embeddings, which consists of 300-dimensional vectors for 3 million words and phrases. For entity embeddings, we use the entity embeddings trained by Li, Zheng et al. (2016) known as Hierarchical Concept Embeddings (HCE) made publicly available by the authors.

In order to train the rankers based on our proposed features, we exploit the SVMrank model. The choice of SVM^{rank} is motivated by the fact that it has been shown to perform well in ranking problems similar to ours (Hosseini, Nguyen, & Bagheri, 2018; Perera et al., 2016). The hyperparameter settings used in our work are as follows: linear kernel, 0:01 as the trade-off between training error and margin, and the loss function is the number of swapped pairs summed over all inputs.

4.3. Metrics and baselines

In order to evaluate the performance of our features and compare them against the baselines, we adopt a similar evaluation metric that is used in the related literature. The main evaluation metric is Precision at Rank 1 (P@1). The reason for this metric is that in the implicit entity linking task, each textual content, e.g., tweet, is referring to one implicit entity. As such, the objective is to identify the one entity that the textual is referring to. Therefore, when ranking entities based on their relevance to serve as an implicit entity, the entity ranked first would be considered to be the implicitly mentioned entity and hence the performance of the implicit entity linker is assessed based on the correctness of the entities appearing in the first position. Furthermore, Mean Reciprocal Rank (MRR) is used in order to identify the average ranking of the target entity when not at the first place.

In terms of baseline methods, we adopt two state of the art techniques by Perera et al. (2016), Hosseini et al. (2018), and Huang et al. (2020). The work by Perera et al. operates on the basic assumption that the implicit entity within a tweet can be inferred from the contextual clues offered by tweets that are posted in a similar time interval. In other words, this method hypothesizes that it is possible to explicitly observe the entity of interest if a sufficient number of temporally aligned tweets are pooled at the time when the tweet containing the implicit tweet is posted. On the other hand, Hosseini et al. define the problem of implicit entity linking as one of ad hoc retrieval. They view a tweet as a query to be posed to the knowledge graph for retrieving the relevant entity, which would serve as the implicit entity related to the tweet. Lastly, Huang et al. leverage the notion of implicit entity linking in order to reformulate the problem of identifying camouflaged products on e-commerce platforms.

4.4. Performance evaluation

We report the performance of the learn to rank model trained based on our proposed features against three baseline methods, namely Perera et al. Hosseini et al. and Huang et al. We provide performance statistics both overall as well as on a per category basis. The results based on P@1 are reported in Table 6. As seen in the table, our proposed features are able to outperform the baseline methods in all of the implicit entity categories (except Organization) and also over all of the gold standard dataset regardless of the category. This is a strong indication for the suitability of the features that we have defined for implicit entity linking. Furthermore, in order to show the performance of the proposed method not only for the top-rank retrieved entity (as in P@1) but also in other ranks, we report performance metrics based also on MRR in Table 7.

In order to observe the impact of various feature types on the performance of our approach, we report the performance of each feature sub-category on the task of implicit entity linking as reported in Table 8. Our experiments show that term-based features are the strongest category of features that are able to correctly identify implicit entity mentions. This indicates that while no surface form

Table 6

P@1 of our proposed approach for implicit entity linking compared to the baselines.

	Person	Organization	Location	Event	Product	WrittenWork	Film	Overall
Perera et al.	49.6	49	49.8	50.4	48.9	61.05	60.97	52.81
Hosseini et al.	59.82	61.23	58.25	54.09	67.63	72.97	76.43	64.34
Huang et al.	64.31	**66.45**	68.33	60.78	70.23	75.98	79.76	68.8
Our approach	**64.6**	63.13	**72.15**	**88.88**	**88.23**	**76**	**85.71**	**72.11**

Table 7

Performance of our proposed approach for implicit entity linking based on P@1 and mean reciprocal rank (MRR).

	Person	Organization	Location	Event	Product	WrittenWork	Film	Overall
P@1	64.6	63.13	72.15	88.88	88.23	76	85.71	72.11
MRR	71.54	72.67	81.62	90.12	91.17	82	90	79.01

Table 8

P@1 of implicit entity linking with different subsets of features.

	Term-based				Neural embedding-based			Knowledge graph-based			All
	Frequency	Syntactic	Semantic	All	Word	Entity	All	Entity Relevance	Network Properties	All	
Person	56.63	14.15	35.39	55.75	7.07	12.38	18.58	27.43	14.15	29.2	64.6
Organization	53.84	17.58	39.56	57.14	12.08	17.58	19.78	10.98	15.38	19.78	63.13
Location	65.82	29.11	48.1	67.08	22.78	35.44	34.17	29.11	16.45	30.37	72.15
Event	81.18	33.33	37.03	81.48	29.62	37.03	44.44	18.51	51.85	55.55	88.88
Product	58.82	5.88	52.94	64.7	11.76	2.9	11.76	20.62	5.88	23.52	80.23
WrittenWork	64	19.99	52	70	9.99	19.99	24	6	14	14	76
Film	76.78	35.71	62.5	78.57	16.07	16.07	16.07	17.85	25	37.5	85.71
All	62.79	21.95	44.77	64.65	14.07	20.21	23.33	19.73	18.26	28.18	72.11

representation is available for an implicit entity, the terms that appear in the tweet and their association with the KG representation of the candidate entities are strong signs for the relevance of the candidate entity to the tweet. Within term-based features, frequency features that measure the degree of association between the candidate entity and the tweet based on how tweet terms occur in the KG representation of the entity are the strongest features. We additionally find that features that are defined based on the semantics of terms according to WordNet are also quite competitive although not as strong as frequency-based features. This reinforces our hypothesis that it is possible to find entity and tweet association based on the underlying semantics of their content. We also observe that syntactic features based on dependency parsing are not strong features especially due to the ungrammatical nature of content on social platforms such as Twitter where the content is not guaranteed to respect any of the proper usage of grammatical rules. Finally, we observe that while syntactic and semantics features provide synergistic impact on frequency-based features and improve the performance of implicit entity linking when used in conjunction with frequency features, the improvement is not noticeable.

In the context of neural embedding-based features, entity features that consider entity representation explicitly show better performance for when word representations are used to determine the association between a tweet and candidate entities. This shows that when additional information about entity associations are taken into account, more discriminative information can be obtained for ranking entities. We however find that word and entity features have synergistic impact on each other and when used in tandem can show complementary performance leading to enhanced entity linking. The results in Table 8 provide similar observations for the knowledge graph-based features where both types of features, namely entity relevance and network properties, have synergistic performance and their combination shows improved performance compared to when they are used in isolation.

While neural embedding-based features and knowledge graph-based features do not have competitive performance with term-based features, our experiments show that they are able to correctly identify implicit entity instances that are not identifiable by term-based features alone. This reinforces a similar pattern in other related areas such as ad hoc retrieval where researchers had found that while semantic features are not as strong as term-based features, they are able to systematically improve the performance of term-based techniques for special cases such as when vocabulary mismatch exists (Ensan & Bagheri, 2017; Nikolaev & Kotov, 2020). Our work reports a similar observation in that it (1) finds term-based features to be the most effective set of features for implicit entity linking, and (2) reveals synergistic impact from less effective sets of features derived from neural embeddings and knowledge graphs on term-based features.

In Table 9, we further report the performance of the combination of different feature categories. The results show that while term-based features provide a significant portion of the performance, there are special cases that cannot be effectively identified by such features that are correctly identified and linked using knowledge graph and neural embedding features. Overall, the integration of neural embedding and knowledge graph based features with term-based features increases the performance of implicit entity linking from 64.65 on P@1 to 72.11, which is an increase of 11.53% that can be considered to be notable. To concretely show this impact, in Table 10, we report on example tweets that were incorrectly linked by term-based features when used in isolation but were later correctly linked to implicit entities when features from the other two feature categories were included.

We also report the performance of our proposed metrics on the explicit entity linking task in Table 11 where we compare the performance of our work with the performance reported by the seminal paper by Meij et al. (2012). Our proposed features show

Table 9

P@1 of implicit entity linking for different combinations of feature categories.

	Term + Neural embedding	Term + Knowledge graph	Neural embedding + Knowledge graph	All
Person	61.06	61.94	32.74	64.6
Organization	59.34	58.24	26.37	63.13
Location	69.62	70.88	41.77	72.15
Event	82.14	85.18	81.48	88.88
Product	70.58	76.48	26.42	80.23
WrittenWork	72	74	26	76
Film	80.35	82.14	48.21	85.71
All	67.69	68.796	37.07	72.11

Table 10

Sample tweets showing synergistic impact from neural embeddings and knowledge graphs on term-based features. The identified tweets were incorrectly linked by term-based features but then corrected using the additional features defined in the feature categories shown in each row.

	Sub-Category	Tweet	Target implicit entity	Domain
Neural embedding-based	Word embeddings	Wow, have you seen the new Stephen Hawking movie trailer? Inspiring, maybe cheesy, but INSPIRING http://t.co/XG0djtLSiG	The Theory of Everything (2014 film)	Film
	Entity embeddings	She succeeded in keeping her identity as Mary Westmacott unknown for almost 20 years which freed her from the expectations of her crime fans	Agatha Christie	Person
Knowledge graph-based	Entity relevance	His worry is @ArvindKejriwal formed new party instead of joining @BJP4India. He is mum on his students joined BJP.	Aam Aadmi Party	Organization
	Network properties	I loved it. Have you read the book it is based on, by Michel Faber? Also wonderful IMO.	Under the Skin (novel)	Written work

Table 11

Performance of our proposed approach for explicit entity linking based on P@1 and mean reciprocal rank (MRR), as compared to the baseline.

	P@1	MRR
Our approach	72.79	82.28
Meij et al.	67.8	76.76

improved performance compared to the baseline. We find that our proposed set of features for implicit entities, when applied to the task of explicit entity linking, can result in an overall P@1 score of 72.79 compared to the best performing model by Meij et al. that produced a P@1 of 67.80 on the same gold standard dataset. Furthermore, we show the break down of the feature category performances in Table 12. Similar to implicit entity linking, We make a similar observation that term-based features are the most effective type of features for identifying explicit entities. However, unlike implicit entity linking, when identifying explicit entities, neural embedding-based features are not as effective compared to knowledge graph-based features, which show to be very strong indicators of entity association to a tweet. We find that the second most effective feature subcategory after term frequency features is the network properties subcategory in knowledge graph-based features. Network property features measure local and global measures of importance of an entity within the knowledge graph. This shows that more central entities within the knowledge graph have a higher likelihood of acting as explicitly mentioned entities within a tweet. This was not the case for implicit entities. Finally based on the results reported in Table 13, the inclusion of neural embedding and knowledge graph based features to term-based features does not lead to any significant performance improvement. However, when both feature categories are added to the term-based features, this increases the performance of the explicit entity linking task from 68.38 on P@1 to 72.79, an increase of 6.45%.

4.5. Feature analysis

We evaluate the importance of the proposed features for entity linking in both implicit and explicit tasks based on the Gini score. We report feature importances in Fig. 2. The figure shows that for implicit entity linking, all ten features in the top-10 features belong to the term-based feature category and 8 out of these 10 belong to the frequency-based subcategory and are various variations of the TF-IDF, TF, or SDM features measured over bigrams, entities and unordered bigrams. There are in fact only 6 out of the top-20

Table 12

P@1 of explicit entity linking with different subsets of features.

	Term-based				Neural embedding-based			Knowledge graph-based			All
	Frequency	Syntactic	Semantic	All	Word	Entity	All	Entity relevance	Network properties	All	
Explicit linking	66.91	25	47.05	68.38	21.32	–[a]	21.32	19.85	55.88	55.88	72.79

[a]Note Entity Embedding features are not applicable to the task of explicit entity linking as they require explicit entities identified to measure the feature values.

Table 13

P@1 of explicit entity linking for different combinations of feature categories.

	Term + Neural embedding	Term + Knowledge graph	Neural embedding + Knowledge graph	All
Explicit	68.38	69.11	65.44	72.79
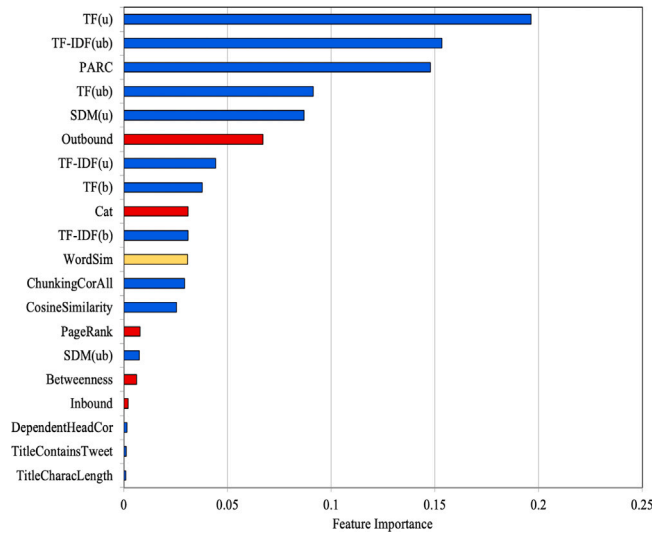
Fig. 2. Top-20 features ranked based on importance according to Gini score for implicit entity linking. The letters 'u', 'b','ub', and 'e' denote unigrams, bigrams, unordered bigrams and entities, respectively.

features that are not term-based features, which are all knowledge graph-based features. However, we note that feature importances should be interpreted within the context of feature interaction and their impact on overall performance.

We make two observations:

(1) While the top-20 features only include features from term-based and knowledge graph-based feature categories, the overall performance of the combination of features from these two categories does not significantly exceed the combination of term-based features with neural embedding features. This can be observed in Table 6 where the performance of the integration of term-based and knowledge graph-based features is 68.79 on P@1 compared to 67.69 when term-based features are integrated with neural embedding-based features.

(2) While not as discriminative based on Gini score, neural embedding-based features do have synergistic impact on both term-based and knowledge graph-based features as evidence in the final performance where the inclusion of neural embedding-based features increases the performance of implicit entity linking by 4.82% (from 68.79 to 72.11 on P@1).

We have similar observations regarding the importance of features for explicit entity linking as shown in Fig. 3. Likewise, 8 out of the top-10 features for explicit entity linking belong to term-based features from among which 7 are from the frequency-based features. The other 2 features belong to knowledge graph-based features. Different from implicit entity linking, one neural embedding-based feature, i.e., WordSim, can be seen in the top-20 features. In the context of explicit entity linking, we find analogous relations between neural embedding-based features with term-based and knowledge graph-based features to what we did in implicit entity linking where neural embedding features are generally not a strong as the other two categories overall in terms of Gini score but do lead to an increase of 5.32% on P@1 (increasing from 69.11 to 72.79).

In order to show consistency of feature importances over all entity domains as well as in each domain separately, we depict the percentage of number of features observed in the top-k features sorted based on Gini score in Fig. 4. This figure shows what percentage of features in the top-k belong to each feature category. For instance, looking at the overall chart, one can see that the
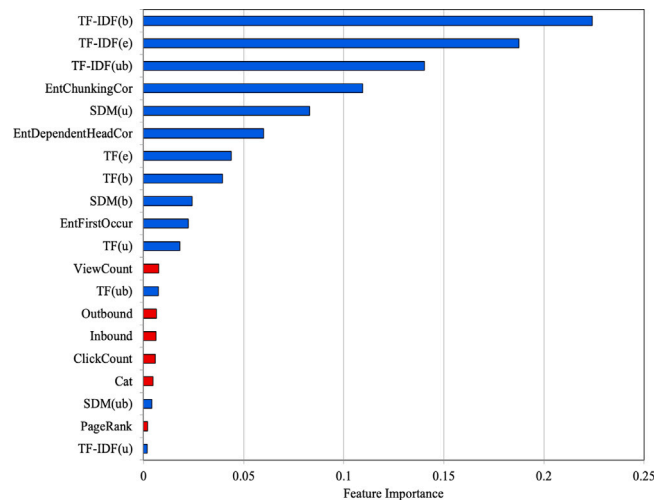
Fig. 3. Top-20 features ranked based on importance according to Gini score for explicit entity linking. The letters 'u', 'b','ub', and 'e' denote unigrams, bigrams, unordered bigrams and entities, respectively.

Table 14
Sample tweets representing each type of error made by our proposed approach.

Domain	Error type	Tweet text	Target Implicit entity	Incorrectly linked entity
Event	1	I really cannot believe that our president is so horrible that people are encouraging Oprah to run all because she gave a speech at an awards show. . .	Golden Globe Awards	Emmy Awards
Event	2	So the last really impactful hurricane season for the US was back in 2005... the last time the @FootballUGA won the @SEC Championship. Now 12 years later, a record active hurricane year in some regards, and the Georgia Bulldogs are again #SECChampions!! #Cray #ScientificTrends	Hurricane Katrina	SEC Championship Game
Person	3	Thank you the founder of WhatsApp! https://fb.me/3G4ZB5Wee	Jan Koum	Jeff Bezos
WrittenWork	4	it is a book series by Diana Gabaldon that was huge when I was in high school. It is either going to be amazing or crap	Outlander (novel)	Star Trek (novel)
Person	5	Inception, departed, shutter island, the beach, wolf of Wall Street, he do not do crap films TBH	Leonardo DiCaprio	Martin Scorsese
Person	6	40 Year Old Singer Shakara Split From Her Boyfriend Of Six Years. Rumours has emerged that Shakira and her... http://fb.me/76LupfvTf	Gerard Pique	Antonio de la Rúa

top-10 features are all from the term-based feature category, while some features from the knowledge graph-based category have also appeared in the top-20 features (30% equivalent to 6 features). The figure reinforces our earlier finding that term-based features show to be the most important features whether considered overall or on a per domain basis.

4.6. Qualitative error analysis

In order to provide a more in-depth understanding of the areas where the proposed features do not work well for identifying implicit entities, we carefully review the tweets that have been mislabeled by our proposed approach. For such instances, we determine the reason why our approach did not perform correctly and further classify them into *six error types*. For a more clear picture, we provide an example tweet that was mislabeled by our approach for each of the error types in Table 14. We also provide the detailed statistics of the percentage of error types per domain in the gold standard in Table 15. This table shows what percentage of mislabeled tweets were due to each of the error types. In the following, each type of error is elaborated in more detail:

Error Type 1: *Lack of closely related entities*: There are several errors that fall in this category: (a) Sometimes, off-the-shelf explicit entity linkers fail to correctly identify the explicit entity that is present in the tweet. This becomes an issue for those features that rely on the explicit entities that are present in the tweet. For the example in the sample shown in Table 14 for this error type, the entity linker that we employed, i.e.,TagMe, was not able to correctly identify and tag 'Oprah Winfrey'. In such instances, the features that rely on the explicit entities measure incorrect association between the tweet and the candidate implicit entities; and,
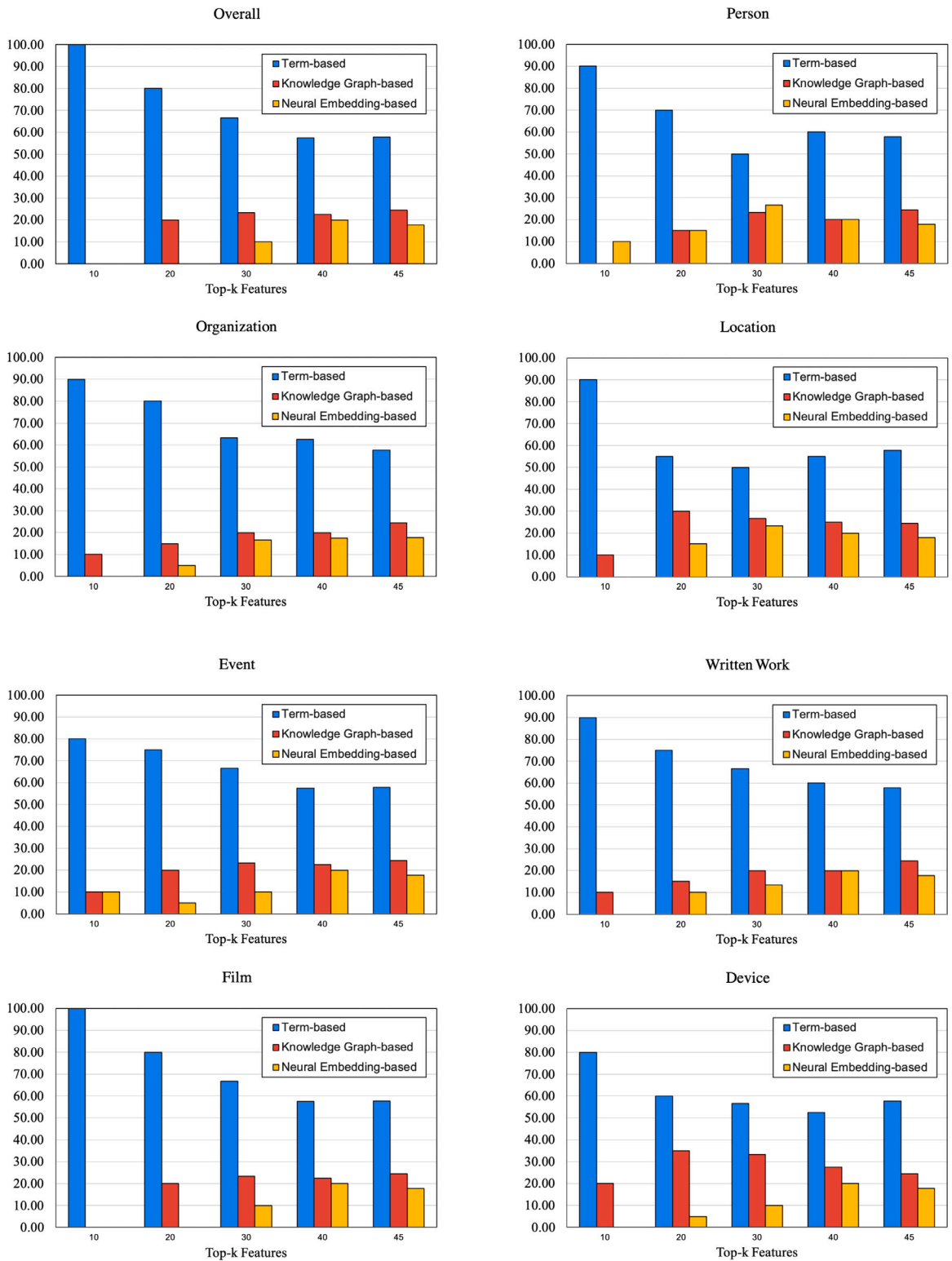
Fig. 4. Distribution of percentage of features from each category in the top-k features based on Gini score for implicit entity linking.

Table 15

Analysis of errors in implicit entity linking. Each cell shows the percentage of that error type observed in that specific domain.

Error type	Description	Person	Organization	Location	Event	Product	Written work	Film
1	Lack of closely related entities	25	–	21	–	60	–	27
2	Heterogeneity of references	7	22	9	50	–	15	–
3	Lack of contextual knowledge	14	22	9	33	–	15	9
4	Less frequent entities	14	12	16	17	20	15	27
5	Misleading explicit entities	33	22	25	–	–	40	37
6	Temporality sensitive reference	7	22	20	–	20	15	–

(b) There are also other cases, where the explicit entity is in fact correctly identified and linked by the explicit entity linker but the association between the identified explicit entity and the appropriate implicit entity is not properly captured on the knowledge graph. For instance, in the version of Wikipedia that we used for our experiments, Oprah Winfrey is not mentioned within the content related to the Golden Globe Awards Wikipedia entry; therefore, the association between Oprah Winfrey and Golden Globe Awards is not considered to be strong. The lack of closely related entities as captured in Error Type 1, either as a result of incorrect explicit entity linking or the lack of association between entities on the knowledge graph, can lead to incorrect feature values in our approach. While such errors are due to factors external to our work, we believe that performing implicit and explicit entity linking in tandem can minimize or control the impact of this error type. We are currently exploring (i) how content on the knowledge graph can be automatically entity linked using both explicit and implicit entities at the same time so that issues such as (b) can be addressed (identifying Oprah Winfrey as being implicitly relevant to Golden Globe Awards although she has not been explicitly mentioned), and (ii) jointly optimizing the likelihood of implicit and explicit entities appearing in the tweet as opposed to using explicit entities to determine the likelihood of implicit entities, which can potentially address issue (a).

Error Type 2: *Heterogeneity of References*: As indicated by other researchers (Zhao et al., 2011), given the short nature of tweets, they are primarily formed around a singular theme and convey one central message. Implicit entities in tweets are therefore related to the central theme of the tweet. However, even when the tweet is clear about the central theme, there can be strong references to other tangentially relevant information in the tweet that are only superficially relevant to the main topic. For instance, in the example provided in Table 14, while the tweet is primarily related to Hurricane Katrina, there are also strong references within the tweet that are not directly related to this central theme mentioning Southeastern Conference American football championship. Given the fact that many of our proposed features are dependent on the terms observed in the tweet or the explicit entities that are identified, such mentions can lead to the incorrect identification of the implicit entity. As such and as future work, it is important to introduce features that can prioritize term and entity relevance based on forms of topic precedence. This would ensure that terms and entities that are relevant to the central theme of the tweet receive higher importance and hence can potentially lead to the more accurate detection of the implicit entity.

Error Type 3: *Lack of Contextual Knowledge*: The other source of error for our proposed approach is related to cases when tweets are very short and hence do not include sufficient contextual information for our proposed features to identify relevant clues for determining the implicit entity. For instance, in the provided example tweet, i.e., ‘Thank you the founder of WhatsApp! https://fb.me/3G4ZB5Wee’, out of the six terms in the tweet four of them do not carry any indicative semantics and there are only two terms, namely ‘founder’ and ‘WhatsApp’, which are meaningful. In this specific example, given the lack of context in the tweet, our approach falls back to using features that rely heavily on prior likelihood of entity relevance to the tweet such as those in the knowledge graph-based feature category (e.g., view count). Due to the fact that Jeff Bezos is classified as a founder and has a high prior likelihood due to its position in the knowledge graph, our approach incorrectly picks Jeff Bezos as the implicit entity. Perera et al. (2016) have also identified the issue with lack of context and suggested to temporally pool related tweets to build some context for short tweets; however, given the nature of temporal pooling, this strategy would not be effective for cases when the tweet is not related to trending topics. In such cases, the pooled tweets will potentially have a negative impact as they can cause topic drift. As such, we propose that we need to consider features that are able to determine whether (1) a tweet relates to broader topics of the community and hence can be expanded through temporal pooling or (2) is very personal and requires user-specific expansion.

Error Type 4: *Less Frequent Entities*: We find that our proposed approach is more effective for mainstream entities in the knowledge graph. When dealing with explicit entity linking, researchers (Esquivel, Albakour, Martinez, Corney, & Moussa, 2017; Huang et al., 2018) have already identified the long tail problem of entities where a small set of entities are predominantly observed and used, while the large majority of entities are less frequent. As such, explicit entity linkers are often more effective for popular and frequent entities compared to for infrequent entities. We have a similar experience with our proposed approach where entities that are less mainstream and less frequently observed on the knowledge graph, have a higher likelihood of being mislabeled when performing implicit entity linking. When considering the importance of term-based features, and more specifically the frequency-based feature subcategory, we believe the reason for favorable bias towards frequent entities is due to their frequency across the knowledge graph. Therefore, we suggest that similar to Error Type 2, a reasonable alternative strategy for defining features would be to discount term and entity frequency information locally for each tweet in the context of their main theme. In other words, one would locally measure frequency information based on what topic the tweet is referring to. This way, long tail entities would not be disadvantaged compared to frequent entities.

Error Type 5: *Misleading Explicit Entities*: Unlike Error Type 2, which indicates the impact of the presence of irrelevant terms or entities within a tweet, this error type occurs when one or a set of explicit entities within the tweet are closely related to more than one entity on the knowledge graph. For instance, in the example tweet provided in Table 14, the implicit entity of interest is

Leonardo DiCaprio; however, there are multiple explicit entities available in the tweet who can be interpreted to refer to different entities even when considered in tandem. For instance, Martin Scorsese is the director of three of the movies that are mentioned in the tweet and can therefore be a strong contender to serve as the implicit entity. While type 2 error requires the consideration of topic precedence, this type of error can only be resolved through the aggregation of the semantics of the present entities or finding their common denominator. As such, features defined in the future would need to be cognizant of the interaction between explicit entities mentioned in the tweet for selecting the relevant implicit entity. This is because other features such as entity and term frequency, recency, centrality on the knowledge graph, among others would not be able to effectively discriminate between entities such as Leonardo DiCaprio and Martin Scorsese. It would only be the common denominator that can lead to the selection of DiCaprio over Scorsese.

Error Type 6: *Temporality-Sensitive Reference*: This category includes errors, which occur due to temporally evolving entities or time sensitive content that take time to get reflected on the knowledge graph. This happens mostly to trending topics that happen in real time and require some time before they are reliably mentioned on Wikipedia and then become a part of the knowledge graph. For such, time sensitive entity information, it would not be possible to measure the relevance of the tweet to the entity. For instance, the example provided for this error type discusses rumors about two celebrities that never get reflected on the knowledge graph and can very well be misinformation. For such types of content, it would not be possible to directly identify association between the tweet and the implicit entity unless other contextual information similar to Error Type 3 are identified and added through mechanisms such as temporal pooling.

4.7. Summary of findings

Based on our experiments and the analysis of the results, we can derive actionable findings that will be helpful for advancing the state of the art in implicit entity linking. We summarize our findings as follows. We find that:

(1) Term-based features are the most discriminative features for performing implicit entity linking. This is because those terms that appear in the input tweet have close resemblance to the text that appear in the textual representation of the target entity on the knowledge graph. From among term-based features, frequency-based features have the most significant contribution to the effectiveness of implicit entity linking;

(2) Knowledge graph-based relevance features are more effective for implicit entity linking as compared to explicit entity linking. This can be in part due to the fact that users often use implicit mentions when they believe their audience can understand the implicitly mentioned entity. Such identifiable entities would be those that have already become 'hot' in the social sphere or widely mentioned by the community. As such, knowledge graph-based relevance features that capture these characteristics are effective;

(3) In contrast, for explicit entity linking, network property-based features are more effective compared to relevance-based features. This can be explained by the fact that network measures determine the importance of entities that form effective priors for the likelihood of that entity being mentioned in text. When explicitly mentioned, these priors accurately estimate the likelihood of the entity to be mentioned. However, when discussing implicit mentions, these priors are not accurate but rather priors based on popularity of entities become better predictors;

(4) We find that relevance-based and network property-based features have a reinforcing effect on each other for implicit entity linking and as such, it is helpful to include features from both categories when building an implicit entity linker. On the other hand, these features have an overlapping effect on each other for explicit entity linking and as such the inclusion of only network property-based features to be sufficient and relevance-based features would be redundant for explicit entity linking;

(5) Neural embedding-based features show comparable performance with knowledge graph-based features for implicit entity linking in that they do not perform as well as term-based features. However they do have a reinforcing impact on term-based features. This is not true in case of explicit entity linking where knowledge graph-based features perform considerably better. This could, in part, be explained by the fact that entity embedding-based features subcategory is not applicable to explicit entity linking. This is especially important because entity embedding features show stronger performance compared to word embedding features for implicit entity linking;

(6) Finally, while poorer in performance compared to term-based features, neural embedding-based and knowledge graph-based features show synergistic and complementary performance to term-based features and hence contribute to an overall increase in performance for the implicit linking task.

5. Concluding remarks

In this paper, we have adopted a learning to rank approach for performing the task of implicit entity linking. We have systematically introduced three broad categories of features for this purpose, namely term-based, neural embedding-based and knowledge graph-based features. Term-based features incorporate aspects of term frequency, semantics and syntactics, while neural embedding-based features exploit the neural representation of words and entities. Furthermore, knowledge graph-based features benefit from external information such as those represented on DBpedia to measure entity characteristics such as their position within the knowledge graph network. Through our experiments, we show that our proposed features are able to collectively outperform three recent strong baseline methods for implicit entity linking. We find that while term-based features have the most significant

contribution towards the identification of correct entities, neural embedding and knowledge-graph-based features also add value to the implicit linking process. Finally, we argue and empirically show that legacy features defined specifically for explicit entity linking are not necessarily appropriate for implicit linking and hence the introduction of specific features for implicit entity linking is warranted. Finally, our work offered insight into six different error types that occurred in our implicit entity linking approach, looked into the reasons why they occurred and discussed how these can be addressed in future work.

On this basis, our future work will explore some areas as described in the following. One of the areas that we find to be quite important is to perform implicit and explicit entity linking in tandem. The main reason for this is that both implicit and explicit entity linking often struggle with minimal contextual information that is available on tweets. Therefore, identifying both types of entities at the same time will ensure identified entities are collectively disambiguated and relevant to each other. The other area that we will explore relates to how feature values can be computed specifically within different topical areas. This is particularly important for long-tail entities that are not as frequent as other entities. Given the fact that term frequency-based features are the most influential, they would by nature favor more frequent terms. As such, measuring frequency information both at the global level as well as domain (topic) specific level can potentially help long-tail entities. Finally, we are interested in exploring ways to mine entity representations from sources other than the knowledge graph for entities that do not have full or even partial presence on the knowledge graph. There are existing works in the literature (Feng et al., 2018) that mine entity representations for performing explicit entity linking on social content, but there is yet to be work on identifying entities to represent implicit mentions in an unsupervised way. This will be important to identify implicit entities for content that receive bursty attention on social networks without prior precedence.

CRediT authorship contribution statement

Hawre Hosseini: Conceptualization, Software, Validation, Investigation, Formal analysis, Writing - original draft. **Ebrahim Bagheri:** Funding acquisition, Conceptualization, Supervision, Methodology, Writing - review & editing.

Acknowledgment

References

Anastácio, I., Martins, B., Calado, P., et al. (2011). Supervised learning for linking named entities to knowledge base entries. In *Text analysis conference (TAC)*.

Audeh, B., Beaune, P., & Beigbeder, M. (2014). Exploring query reformulation for named entity expansion in information retrieval. In *Proceedings of the 29th annual ACM symposium on applied computing* (pp. 929–930).

Bagheri, E., Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., & Al-Obeidat, F. (2020). Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management, 57*(4), Article 102248.

Bagheri, E., Ensan, F., & Al-Obeidat, F. (2018). Neural word and entity embeddings for ad hoc retrieval. *Information Processing & Management, 54*(4), 657–673.

Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., & Trani, S. (2013). Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on information & knowledge management* (pp. 139–148).

Charton, E., Meurs, M.-J., Jean-Louis, L., & Gagnon, M. (2014). Improving entity linking using surface form refinement. In *LREC* (pp. 4609–4615).

Cornolti, M., Ferragina, P., & Ciaramita, M. (2013). A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on world wide web* (pp. 249–260).

Dalton, J., Dietz, L., & Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 365–374).

Derczynski, L., Maynard, D., Rizzo, G., Van Erp, M., Gorrell, G., Troncy, R., et al. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management, 51*(2), 32–49.

Ensan, F., & Al-Obeidat, F. (2019). Relevance-based entity selection for ad hoc retrieval. *Information Processing & Management, 56*(5), 1645–1666.

Ensan, F., & Bagheri, E. (2017). Document retrieval model through semantic linking. In *Proceedings of the tenth ACM international conference on web search and data mining* (pp. 181–190).

Esquivel, J., Albakour, D., Martinez, M., Corney, D., & Moussa, S. (2017). On the long-tail entities in news. In *European conference on information retrieval* (pp. 691–697). Springer.

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

Feng, Y., Zarrinkalam, F., Bagheri, E., Fani, H., & Al-Obeidat, F. (2018). Entity linking of tweets based on dominant entity candidates. *Social Network Analysis and Mining, 8*(1), 46.

Ferragina, P., & Scaiella, U. (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* (pp. 1625–1628). ACM.

Guo, S., Chang, M.-W., & Kiciman, E. (2013). To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 1020–1030).

Habib, M. B., & Van Keulen, M. (2012). Unsupervised improvement of named entity extraction in short informal context using disambiguation clues. In *SWAIE* (pp. 1–10).

Hachey, B., Radford, W., Nothman, J., Honnibal, M., & Curran, J. R. (2013). Evaluating entity linking with wikipedia. *Artificial Intelligence, 194*, 130–150.

Hasibi, F., Balog, K., & Bratsberg, S. E. (2017). Entity linking in queries: Efficiency vs. effectiveness. In *European conference on information retrieval* (pp. 40–53). Springer.

Honnibal, M., & Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1373–1378). Lisbon, Portugal: Association for Computational Linguistics, https://aclweb.org/anthology/D15/D15-1162.

Hosseini, H., Nguyen, T. T., & Bagheri, E. (2018). Implicit entity linking through ad-hoc retrieval. In *2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)* (pp. 326–329). IEEE.

Hosseini, H., Nguyen, T. T., Wu, J., & Bagheri, E. (2019). Implicit entity linking in tweets: An ad-hoc retrieval approach. *Applied Ontology*, *14*(4), 451–477.

Hua, W., Zheng, K., & Zhou, X. (2015). Microblog entity linking with social temporal context. In *ACM SIGMOD international conference on management of data* (pp. 1761–1775).

Huang, H., Cao, Y., Huang, X., Ji, H., & Lin, C.-Y. (2014). Collective tweet wikification based on semi-supervised graph regularization. In *Proceedings of the annual meeting of the association for computational linguistics* (pp. 380–390).

Huang, Z., Cautis, B., Cheng, R., Zheng, Y., Mamoulis, N., & Yan, J. (2018). Entity-based query recommendation for long-tail queries. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(6), 1–24.

Huang, L., Yuan, B., Zhang, R., & Lu, Q. (2020). Towards linking camouflaged descriptions to implicit products in E-commerce. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 901–910).

Ibrahim, Y., Amir Yosef, M., & Weikum, G. (2014). Aida-social: Entity linking on the social stream. In *Proceedings of the 7th international workshop on exploiting semantic annotations in information retrieval* (pp. 17–19).

Leacock, C., Chodorow, M., & Miller, G. A. (1998). Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, *24*(1), 147–165.

Li, Y., Tan, S., Sun, H., Han, J., Roth, D., & Yan, X. (2016). Entity disambiguation with linkless knowledge bases. In *25th international conference on world wide web* (pp. 1261–1270).

Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., & Sycara, K. (2016). Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*.

Ling, X., Singh, S., & Weld, D. S. (2015). Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, *3*, 315–328.

Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., & Lu, Y. (2013). Entity linking for tweets. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: long papers), vol. 1* (pp. 1304–1311).

Masud, M. M., Chen, Q., Khan, L., Aggarwal, C., Gao, J., Han, J., et al. (2010). Addressing concept-evolution in concept-drifting data streams. In *Data mining (ICDM), 2010 IEEE 10th international conference on* (pp. 929–934). IEEE.

Meij, E., Weerkamp, W., & De Rijke, M. (2012). *Proceedings of the fifth ACM international conference on web search and data mining* (pp. 563–572). ACM,

Metzler, D., & Croft, W. B. (2005). A Markov random field model for term dependencies.In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 472–479).

Nikolaev, F., & Kotov, A. (2020). Joint word and entity embeddings for entity retrieval from a knowledge graph. In *European conference on information retrieval* (pp. 141–155). Springer.

Perera, S., Mendes, P. N., Alex, A., Sheth, A. P., & Thirunarayan, K. (2016). Implicit entity linking in tweets. In *European semantic web conference* (pp. 118–132).

Ran, C., Shen, W., & Wang, J. (2018). An attention factor graph model for tweet entity linking. In *Proceedings of the 2018 world wide web conference* (pp. 1135–1144). International World Wide Web Conferences Steering Committee.

Sarmento, L., Kehlenbeck, A., Oliveira, E., & Ungar, L. (2009). An approach to web-scale named-entity disambiguation. In *International workshop on machine learning and data mining in pattern recognition* (pp. 689–703). Springer.

Shen, W., Wang, J., & Han, J. (2014). 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., & 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* (pp. 68–76). ACM.

Tran, T., Tran, N. K., Asmelash, T. H., & Jäschke, R. (2015). Semantic annotation for microblog topics using wikipedia temporal information. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 97–106).

Vo, D.-T., & Bagheri, E. (2019). Feature-enriched matrix factorization for relation extraction. *Information Processing & Management*, *56*(3), 424–444.

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

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. arXiv preprint arXiv:cmp-lg/9406033.

Xiong, C., Liu, Z., Callan, J., & Hovy, E. (2017). JointSem: Combining query entity linking and entity based document ranking. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 2391–2394).

Yu, Y., Zheng, L., Yang, K., & Jin, P. (2014). Named entity linking based on wikipedia. *International Journal of Database Theory and Application*, *7*(1), 1–10.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., et al. (2011). Comparing twitter and traditional media using topic models. In *European conference on information retrieval* (pp. 338–349). Springer.

Zhao, G., Wu, J., Wang, D., & Li, T. (2016). Entity disambiguation to wikipedia using collective ranking. *Information Processing & Management*, *52*(6), 1247–1257.

Zou, X., Sun, C., Sun, Y., Liu, B., & Lin, L. (2014). Linking entities in tweets to wikipedia knowledge base. In *Natural language processing and Chinese computing* (pp. 368–378). Springer.