From Explicit to Implicit Entity Linking: A Learn to Rank Framework

Hawre Hosseini(✉) and Ebrahim Bagheri(✉)

LS3 Laboratory, Ryerson University, Toronto, ON, Canada
{hawre.hosseini,bagheri}@ryerson.ca

Abstract. Implicit entity linking is the task of identifying an appropriate entity whose surface form is not explicitly mentioned in the text. Unlike explicit entity linking where an entity is linked to an observed phrase within the input text, implicit entity linking is concerned with determining specific yet implied entities. Existing work in the literature have already identified appropriate features that can be used for ranking relevant entities for explicit entity linking. In this paper, we (1) consider the applicability of such features for implicit entity linking, (2) introduce features that are suited for this task, (3) compare our work with the state of the art in implicit entity linking, and (4) and report on feature importance values and present their interpretations.

1 Introduction

When producing content on social media, such as Twitter, users often refer to people, places and things without explicitly mentioning them [2–4,8]. For instance, in the tweet '*and thats why he is the King of Pop, Duke of Dance, Master of The Moonwalk...but mostly the King of our Hearts for all eternity.*' Michael Jackson is the main person who is being referred to but without being mentioned. For such cases, traditional entity linking methods and named entity taggers cannot identify or link the content to an appropriate entity. According to [4], on average, 15% of tweets contain implicit mentions and according to [8], 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 number of information-rich content that cannot be readily processed by existing entity linking techniques. The task of *implicit entity linking* is concerned with identifying and linking such implied mentions. In this paper, we adopt a learning to rank approach for performing implicit entity linking. Our work's main motivation is that existing work on explicit entity linking have successfully adopted the learning to rank approach for identifying suitable entities [6]. These approaches rely on a collection of features that link the content space to the entity space. We systematically categorize and present features for the context of implicit entity linking. We will show that features showing strong performance on explicit linking do not necessarily have the same linking power for implicit linking. We identify most suitable features for implicit linking and show that when used in

C. Goutte and X. Zhu (Eds.): Canadian AI 2020, LNAI 12109, pp. 283–289, 2020.
https://doi.org/10.1007/978-3-030-47358-7_28

the context of a learning to rank approach, they can provide significantly better performance compared to the state of the art. Summarily, the contributions of our work are as follows: 1) We provide a framework for systematically categorizing features that can be used for explicit and implicit entity linking within the context of a learning to rank approach; and, 2) We examine the suitability of these features for both of the entity linking tasks and show their importance.

2 Proposed Approach

This paper rests on the foundation offered by work in the learning to rank literature, which rely on the definition of effective features for ranking relevant items for an input query. In our case, we are interested in ranking relevant entities from the knowledge graph for an input text, e.g., a tweet. Here, we introduce our different feature types and explain how they can be extracted. As seen in Table 1, the features are structured based on four main categories: term-based, string-based, graph-based (network properties) and graph-based (popularity properties). In the table, we additionally specify whether the feature is extracted based on input text (specified as T) or the target entity representation (specified as E) or both (denoted as TE). We further denote the *unit* of each feature as to the form by which the feature is extracted, which can include Unigrams (u), Bigrams (b), Unordered Bigrams (ub), entities identified using an explicit entity tagger (e), or anchor texts (a). The final column of the table indicates which features, and if so which variation, is not applicable for the task of explicit entity linking. In the following, we provide the details of each feature category.

2.1 Term-Based and String-Based Features

This category encompasses features that extract information from textual content of tweets and/or entities' textual representations. Features in this category are mainly Information Retrieval (IR)-based, such as, term frequency also discounted with inverse document frequency, sequential dependence model (SDM) [7] as well as textual similarity through cosine similarity. All of these features can be applied to both the tweet content and the entities' representations. We additionally extract three other term-based features, which are not based on IR methods, yet commonly used in entity linking: (1) *PARC* considers presence of anchors within the tweet. Presence of an entity anchor in the tweet can be an indicator for relevance to the anchor's pertinent entity. Also, it might serve as a textual reference to the target implicit mention; (2) *TitleContainsTweet*, investigates the presence of a substring of the tweet 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 implicit entity linking and we hypothesize that this feature will not perform as well, since implicit references do not contain surface forms of entities; and, (3) *URLEntityCount* depends on the URLs found within the tweet. This feature extracts and additionally considers the webpage content of URLs found within a tweet and counts the number of times a candidate entity appears in them.

Table 1. Description and categorization of features proposed for entity linking.

Category	Feature name	Description	Type	Unit					Not applicable in explicit linking
				u	b	ub	e	a	
Term-based	TF	Considers frequency of tweet term in the content of entity	TE	✓	✓	✓	✓	✓	TF(e)
	TF-IDF	Considers the inverse frequency of tweet term in entities' content	TE	✓	✓	✓	✓	✓	TF-IDF(e)
	SDM	SDM model with different feature functions	TE	✓	✓	✓	✓	✓	SDM(e)
	Cosine similarity	Cosine similarity of tweet text and entity representation	TE	✓	✓	✓	✓	✓	Cosine similarity(e)
	PARC	Presence of an Anchor Referring to a Candidate Entity inside tweet	TE	–	–	–	–	✓	–
	TitleContains tweet	If title of entity contains substring of the tweet	TE	–	–	–	–	–	–
	URLEntity count	Number of times entity appears on a webpage whose URL is in the tweet	T	–	–	–	–	–	–
	ECoocKB	Co-occurrence of tweet Explicit entities with Candidate entities on KB	E	–	–	–	✓	–	ECoocKB(e)
String-based	TitleCharLength	Character length of title of the entity	E	–	–	–	–	–	–
	TitleTermCount	Number of terms in title of entity	E	–	–	–	–	–	–
Graph-based (network properties)	InkinksKB	Number of entities on a KB linking to e	E	–	–	–	✓	–	–
	OutlinksKB	Number of KB articles linking from e	E	–	–	–	✓	–	–
	CatKB	Number of categories associated with e on a KB hierarchy	E	–	–	–	✓	–	–
	Redirect	Number of redirect pages linking to e on Wikipedia	E	–	–	–	✓	–	–
	Betweenness	Betweenness measure of each candidate entity in a constructed graph	E	–	–	–	✓	–	–
	PageRank	PageRank measure of each candidate entity in Wikipedia graph	E	–	–	–	✓	–	–
	EmbedEntSimilarity	Embedding-based Similarity Measure between the candidate entity and the entities in the tweet (Cosine similarity of embeddings)	TE	–	–	–	✓	–	EmbedEnt similarity(e)
Graph-based (popularity properties)	ViewCount	Number of times {e} was visited in a specific time frame	E	–	–	–	✓	–	–
	ClickStream	The number of times Wikipedia users have navigated from a tweet explicit entity to a candidate entity	TE	–	–	–	✓	–	ClickStream(e)

On the other hand, string-based features consists of two features, namely *TitleCharLength* and *TitleTermCount*, both of which are primarily syntactic features. 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 features can be important for the explicit entity linking task given significantly longer entity titles have a lower likelihood of appearing within a text.

2.2 Graph-Based Features

Network Properties. The most common form of network measures focus on centrality of nodes. In the context of a knowledge graph, node centrality can indicate the importance and/or relevance of the content represented by that node within the graph. As such, we adopt two widely used centrality measures, namely Betweenness Centrality and PageRank. Furthermore, we introduce additional locally defined features based on the neighborhood of an entity within the knowledge graph. For instance, we measure how many Wikipedia categories are associated with the entity, or how many inbound and outbound links are connected to the entity of interest. We consider such features as an indication of the extent to which a given entity is involved in relationships with other entities. We also employ entity representations that have been trained based on neural networks on the structure of the knowledge graph [9] to compute the similarity between the input tweet and the target entity from the knowledge graph. The neural representations of entities capture geometric relations between entities given their proximity and position to each other on the knowledge graph and can hence be an indication for the relevance of the tweet and the target entity.

Popularity Properties. Features in this category depend on the meta-data associated with the knowledge graph that are collected from external sources. We introduce two features in this category. *ViewCount* takes into account the number of times a specific entity was visited by viewers during a certain time frame. This feature aims at capturing hotness of entities in the real world at different times. Moreover, we introduce a novel feature denoted *ClickStream*, which captures the way users navigate on Wikipedia. This feature is extracted from the Wikipedia metadata and records how many times each linked entity on a specific entity's Wikipedia page has been clicked. This stream of clicks is hypothesized to show the different levels of relevance between an entity and other reachable entities. The feature takes explicit entities within the tweet and the target entity and calculates the click frequency between them.

3 Datasets and Experimental Setup

For implicit entity linking, we exploit the dataset introduced in [4], which contains $1,345$ tweets with implicit mentions. The dataset's taxonomy contains 6 coarse-grained entity types, namely Person, Organization, Location, Product, Event, and Work. In this dataset, every tweet is labelled with one target entity.

For explicit entity linking, we use the dataset provided by [6] consistinng 318 available tweets, with an average of 2.22 mentions per tweet. In our ranking problem, we consider each tweet-entity pair as one training instance resulting in a total of 707 samples. Here, for the sake of reproduciblity, we clearly describe the process for extracting the introduced features. For features requiring identification of explicit entities within the tweet, we employ TagMe entity tagger. For Wikipedia textual content, we extract entities by processing Wikipedia dumps. In order to extract entity inlinks, outlinks, and redirects and the number of categories associated with each entity, we exploit the Wikipedia API. For EmbedEntSimilarity, we use embeddings trained by Li et al. [5]. We extract Betweenness Centrality and PageRank from a graph that we construct from the DBpedia RDF dump. Finally, in order to extract PARC, we build a mapping from anchors on Wikipedia to entities, extracted by processing Wikipedia dump's textual content. To build the rankers, we exploit SVM$^{\text{rank}}$ model trained with features described in Table 1. The choice of SVM$^{\text{rank}}$ is motivated by the fact that SVM$^{\text{rank}}$ has been shown to perform well in ranking problems similar to ours [6]. The specification of our model, identical to the baseline, is 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.

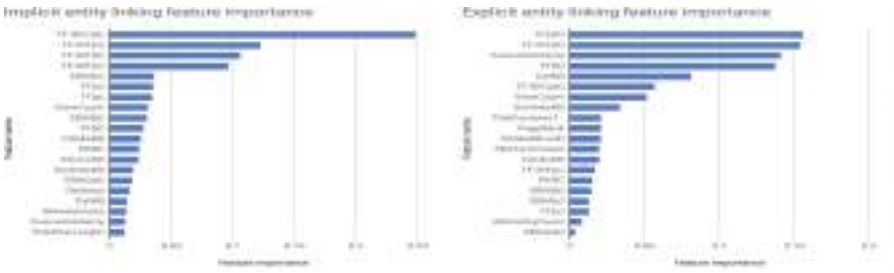

Fig. 1. Feature importance ranked by value for implicit and explicit entity linking.

Table 2. P@1 of this work for implicit entity linking as compared to baselines. The average is calculated with weighting of ratio of queries in each domain.

	Person	Organization	Location	Event	Product/Device	WrittenWork	Film	Average
This work (Implicit entity linking)	**66.37**	**62.6**	**62.02**	**77.77**	**70.58**	**74**	**76.78**	**67.53**
Hosseini et al. 2019-a	59.82	61.23	58.25	54.09	67.63	72.97	76.43	64.34
Perera et al. 2016	49.6	49	49.8	50.4	48.9	61.05	60.97	52.81

4 Results and Discussion

We report the overall results for the implicit entity linking task in Table 2 and compare performance of the introduced features against two state of the art

baselines. As seen, the proposed features when employed within the context of a learning to rank approach show improved performance compared to the two baselines over all entity domains. To evaluate the importance of features, we draw upon their Gini scores, as reported in Fig. 1. Literature on explicit entity linking report that graph-based features such as ViewCount alone are enough for successfully performing linking [1]. This point is reassured in our experiments as well; as seen in Fig. 1, CatKB, ViewCount, and OutlinksKB are among the top 10 best performing features for explicit entity linking. In case of implicit entity linking, however, only one of the graph-based features, i.e., ViewCount, is to be found among the top 10. A lower feature importance for Graph-based features for implicit entity linking as compared to explicit linking shows the difference between the two tasks. We further perform experiments with different groups of features as categorized in our work. We run our systems with features of the following four groups: Term-based and String-based (we combine these two categories as string-based features alone do not produce any noticeable results), Graph-based (popularity-based), Graph-based (network-based), and Graph-based (combined), i.e., the combination of popularity-based and network-based features; results are reported in Table 3. As seen, there are significant differences between feature performance for implicit and explicit entity linking tasks. For implicit linking there is significant difference between the performance of term-based features and that of the graph-based features. However, such a difference is not noticeable for explicit linking. Most specifically, 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 text, e.g. tweet, have close resemblance to the textual representation of the target entity. While strong features, these term-based features are not as effective for explicit entity linking; (2) Graph-based popularity features are quite effective for implicit 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 implicit reference. Such identifiable entities are often those which have become 'hot' in the social sphere or widely mentioned by the community. As such, graph-based popularity features that capture these characteristics are effective. On the other hand, these features are not useful for explicit entity linking at all. (3) On the contrary, graph-based network features are quite effective for explicit entity linking. 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

Table 3. P@1 of implicit and explicit entity linking with subsets of features.

	Term and string-based	Graph-based (popularity)	Graph-based (network)	KB (combined)
Implicit linking	70.58	41.17	5.88	47.05
Explicit linking	31.62	7.26	25.21	27.35

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 are more accurate; and, (4) Finally, we find that popularity and network features have 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-based features seems to be a better strategy.

References

1. Guo, Y., Che, W., Liu, T., Li, S.: A graph-based method for entity linking. In: IJCNLP, pp. 1010–1018 (2011)
2. Hosseini, H.: Implicit entity recognition, classification and linking in tweets. In: The 42nd International ACM SIGIR, pp. 1448–1448 (2019)
3. Hosseini, H., Nguyen, T.T., Bagheri, E.: Implicit entity linking through ad-hoc retrieval. In: ASONAM, pp. 326–329. IEEE (2018)
4. Hosseini, H., Nguyen, T.T., Wu, J., Bagheri, E.: Implicit entity linking in tweets: an ad-hoc retrieval approach. Appl. Ontol. (Preprint) **14**, 1–27 (2019)
5. Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., Sycara, K.: Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In: Proceedings of COLING (2016)
6. Meij, E., Weerkamp, W., De Rijke, M.: Adding semantics to microblog posts. In: Proceedings of WSDM, pp. 563–572. ACM (2012)
7. Metzler, D., Croft, W.B.: Latent concept expansion using markov random fields. ACM SIGIR **2007**, 311–318 (2007)
8. Perera, S., Mendes, P.N., Alex, A., Sheth, A.P., Thirunarayan, K.: Implicit entity linking in tweets. In: European Semantic Web Conference, pp. 118–132 (2016)
9. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. TKDE **29**(12), 2724–2743 (2017)