



HAL
open science

A Novel Path-based Entity Relatedness Measure for Efficient Collective Entity Linking

Cheikh Brahim El Vaigh, François Goasdoué, Guillaume Gravier, Pascale Sébillot

► **To cite this version:**

Cheikh Brahim El Vaigh, François Goasdoué, Guillaume Gravier, Pascale Sébillot. A Novel Path-based Entity Relatedness Measure for Efficient Collective Entity Linking. International Semantic Web Conference (ISWC), Nov 2020, Athens, Greece. hal-02937580

HAL Id: hal-02937580

<https://inria.hal.science/hal-02937580v1>

Submitted on 14 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Novel Path-based Entity Relatedness Measure for Efficient Collective Entity Linking

Cheikh Brahim El Vaigh¹, François Goasdoué², Guillaume Gravier³, and Pascale Sébillot⁴

¹ Inria, IRISA, Rennes, France cheikh-brahim.el-vaigh@inria.fr

² Univ. Rennes, IRISA, Lannion, France fg@irisa.fr

³ CNRS, IRISA, Rennes, France guig@irisa.fr

⁴ INSA Rennes, IRISA, Rennes, France pascale.sebillot@irisa.fr

Abstract. Collective entity linking is a core natural language processing task, which consists in jointly identifying the entities of a knowledge base (KB) that are mentioned in a text exploiting existing relations between entities within the KB. State-of-the-art methods typically combine local scores accounting for the similarity between mentions and entities, with a global score measuring the coherence of the set of selected entities. The latter relies on the structure of a KB: the hyperlink graph of Wikipedia in most cases or the graph of an RDF KB, e.g., BaseKB or Yago, to benefit from the precise semantics of relationships between entities. In this paper, we devise a novel RDF-based entity relatedness measure for global scores with important properties: (*i*) it has a clear semantics, (*ii*) it can be calculated at reasonable computational cost, and (*iii*) it accounts for the transitive aspects of entity relatedness through existing (bounded length) property paths between entities in an RDF KB. Further, we experimentally show on the TAC-KBP2017 dataset, both with BaseKB and Yago, that it provides significant improvement over state-of-the-art entity relatedness measures for the collective entity linking task.

Keywords: Collective entity linking, entity relatedness measure, RDF KBs

1 Introduction

Entity linking is a crucial task for textual document engineering in both natural language processing and information retrieval, with applications such as semantic search [2] and information extraction [16]. It aims at identifying the mentions of entities in a document and linking each mention to a unique referential such as a URI in Wikipedia or in an RDF knowledge base (KB). Entity linking is thus instrumental for semantic search and retrieval [2,16].

The literature on entity linking distinguishes two main approaches, depending on whether mentions within a single document are linked to entities independently one from another [20,10,12] or collectively [13,11,27,5,24,19]. The former uses the KB at hand to generate and select candidate entities independently for each entity mention in the text, while collective linking further uses the KB to select the best global mapping between mentions and candidate entities based on the entity interrelationships recorded

in the KB. State-of-the-art methods for this collective linking step typically combine within a classifier a local score accounting for the string similarity between the mention and an entity’s name, with a global score that measures the coherence of the set of selected entities. In particular, the cornerstone of global score computation is a *measure of relatedness between two entities* that indicates to which extent these entities may co-occur in a document.

In this paper, we focus on improving collective entity linking performance by devising a novel entity relatedness measure. Notably, we advocate that, in addition to showing significant performance improvement on standard benchmarks w.r.t. state-of-the-art competitors, a *well-founded* measure should meet the following three requirements to the extent possible: **(R1)** it must have a *clear semantics* so that linking decisions can be easily understood or explained, in particular it must build on a knowledge base with formal semantics (e.g., an RDF or OWL one, as opposed to Wikipedia) and avoid tuning parameters or knobs that are hard to set by end-users, **(R2)** it must be calculated at *reasonable computational cost* to be of practical interest and **(R3)** it must consider relatedness as a *transitive relation*, to capture that entities may be related within the KB either directly or indirectly, i.e., through *paths*. The last requirement **(R3)** is crucial as it allows encoding implicit links between entities. For instance, if X *worksFor* Y and Y *isLocatedIn* Z then, the path from X to Z implicitly encodes X *worksIn* Z , which is an information not stored in the KB that can be captured by measures meeting **(R3)**.

To the best of our knowledge, no entity relatedness measure in the literature meets all three requirements. Approaches making use of Wikipedia, e.g., [1,4,6,13,11,24,17], consider Wikipedia’s web page URIs as entities, web pages as textual entity descriptions, and hyperlinks between web pages as generic relations between entities. It is worth noting that, although a hyperlink from an entity to another states a direct relation between them, it carries very loose semantics: it solely indicates that the target entity somehow occurs in the description of the source one, be it central to this description or unimportant. Hence, Wikipedia-based entity relatedness measures do not meet **(R1)**, at least. A few other approaches [26,14,22,15] rely on RDF KBs, like BaseKB, DBpedia or Yago, instead of Wikipedia. Such KBs encode in a formal knowledge graph model, the precise semantics of entities (e.g., types) and of their direct relations called *properties* (e.g., property names and cardinalities). While the Ref measure [1,18] just provides a binary indicator of whether or not a relation exists between two entities in the RDF KB, the recent WSRM measure [9], which can be viewed as an extension of Ref, further considers the amount of relations between two entities. Though they both have (simple) clear semantics **(R1)** and are cheap to calculate with edge lookups **(R2)**, they only consider properties between entities to compute relatedness, thus do not meet **(R3)**. By contrast, the rel_{Excl}^k relatedness measure [15] exploits the top- k property paths between two entities (more details in Sec. 2), hence meets **(R3)**. However, it does not fully meet **(R1)** because though its definition has a clear semantics, it relies on user-defined constants that are non-trivial to set due to their unforeseeable consequences on the measure results. Also, rel_{Excl}^k does not meet **(R2)** because it requires computing all paths between entity pairs so as to select the top- k ones; this is not feasible in general in the setting of entity linking, which relies on large encyclopedic RDF KBs. Finally, the cosine similarity is used as an entity relatedness measure in approaches based on RDF

KB embeddings [3,25], i.e., when entities are mapped into coordinates of a multidimensional space. Though the cosine similarity itself has a clear semantics (**R1**) and is not costly to compute (**R2**), the machine learning-based computation of embeddings cannot guarantee that cosine similar entities within the multidimensional space are indeed related in the KB, hence does not meet (**R3**).

Our main contribution is the novel $ASRMP_m$ entity relatedness measure for RDF KBs, which satisfies the three requirements stated above. In particular, for two entities e_1 and e_2 , it uses the fuzzy logic AND and OR operators [7] to compute, respectively, the score of every e_1 -to- e_2 path of length up to m within the KB, by aggregating the WSRM values between the entity pairs found along the path, the final measure being obtained by aggregating over all paths of length m between e_1 and e_2 . In particular, $ASRMP_1$ boils down to WSRM. Importantly, $ASRMP_m$ is not tied to WSRM (i.e., another measure could have been used like Ref). We adopt it here because, in addition to satisfying (**R1**) and (**R2**), it is currently the relatedness measure showing best performance for collective entity linking in the literature [9]. Our fuzzy logic-based aggregation scheme allows $ASRMP_m$ to inherit both (**R1**) and (**R2**) from WSRM. Further, while computing the paths of length up to m between entities rapidly becomes unfeasible as m grows, (**R3**) is met by the need for considering low m values only. Indeed, it has been widely observed (e.g., in [15] for rel_{Excl}^k that also consider paths) that the longer the path between two entities, the less significant the relation it encodes. To evaluate $ASRMP_m$ for entity linking, we first define a collective entity linking system within which we experimentally show on the TAC-KBP2017 dataset, both with the BaseKB and Yago RDF KBs, that $ASRMP_m$ with $m > 1$ improves linking performance w.r.t. the above-mentioned relatedness measures from the literature. We also show significant improvement over popular collective linking techniques using standard entity linking benchmarks.⁵

The paper is organized as follows. We first present in Sec. 2 the main entity relatedness measures used for collective entity linking. In Sec. 3, we define the $ASRMP_m$ entity relatedness measure. In Sec. 4, we describe our collective entity linking system with which $ASRMP_m$ is experimentally compared to state-of-the-art competitors in Sec. 5. Finally, we conclude and discuss perspectives in Sec. 6.

2 Related Work

Most of the entity relatedness measures proposed so far in the context of collective entity linking rely on Wikipedia’s hyperlink structure [1,4,6,13,11,24,17]. As pointed out above, such hyperlinks do not encode the precise semantics of the relations between entities they model, hence can hardly be used within well-founded entity relatedness measures, i.e., that meet the three requirements stated above.

A handful of measures rely on RDF KBs [1,18,15,3,25,9]. Such KBs model both data (facts) and knowledge (ontological description of the application domain) using explicit and implicit triples; the latter can be derived through reasoning based on an RDF-specific consequence relation, a.k.a. entailment. In particular, within RDF KBs,

⁵ <https://gitlab.inria.fr/celvaigh/celasrmp>.

the precise relation (a.k.a. *property*) r that directly relates an entity e_i to another entity e_j is encoded by the triple (e_i, r, e_j) . The use of RDF KBs can therefore be seen as an important step towards devising well-founded entity relatedness measures. We recall below the few relatedness measures that use RDF KBs, and discuss to which extent they meet the three requirements of well-foundedness introduced above: **(R1)**, **(R2)** and **(R3)**.

The binary indicator Ref [1,18] is defined between two entities e_i and e_j as:

$$\text{Ref}(e_i, e_j) = \begin{cases} 1 & \exists r \text{ s.t. } (e_i, r, e_j) \in \text{KB}; \\ 0 & \text{otherwise.} \end{cases}$$

The above definition shows that Ref has a clear semantics **(R1)** and a low computational cost **(R2)** since it can be computed using edge lookups. We however remark that, though clear, its semantics is *very simple*: it does not take into account the various properties between e_i and e_j , nor those that e_i and e_j may have with other entities. Further, Ref does not allow entities to be related through a property path within the RDF KB, hence does not meet **(R3)**: they can only be related through a *single* property, i.e., a single edge or triple.

The Weighted Semantic Relatedness Measure WSRM [9] improves on Ref by not only accounting for the existence of some property between two entities using a Boolean value, but also by *weighting* how related they are in the $[0,1]$ interval, assuming that the more properties between them, the stronger their relatedness. Formally, WSRM is defined between two entities e_i and e_j as

$$\text{WSRM}(e_i, e_j) = \frac{|\{r \mid (e_i, r, e_j) \in \text{KB}\}|}{\sum_{e' \in E} |\{r' \mid (e_i, r', e') \in \text{KB}\}|}, \quad (1)$$

where E denotes the set of entities in the KB and $|S|$ the cardinality of the set S .

In spirit, WSRM is comparable to the *Wikipedia popularity* often used in local entity linking scores, e.g., [8,10], as the probability that a mention m is used as the text (anchor) of a hyperlink referring to an entity e . WSRM is however conceptually different, being applied between two entities rather than between a mention and an entity. It can be interpreted as the probability that e_i is directly related to e_j through some property.

The above definition shows that WSRM has a clear and more fine-grained semantics than Ref **(R1)**. Also, clearly, it can be computed at low computational cost **(R2)** based on edge lookups. However, like Ref, it does not allow entities to be related through property paths within the RDF KB, hence does not meet **(R3)**.

The path-based semantic relatedness measure [15] between two entities, denoted $rel_{E_{xcl}}^{(k)}$, is an aggregation of *path weights* for the top- k paths with highest weights between those entities; path weights are computed using the so-called *exclusivity* measure

$$\text{exclusivity}(x \xrightarrow{\tau} y) = \frac{1}{|x \xrightarrow{\tau} *| + |* \xrightarrow{\tau} y| - 1}, \quad (2)$$

Measure	(R1)	(R2)	(R3)	Measure	(R1)	(R2)	(R3)
Ref [1,18]	×	×		$rel_{Excl}^{(k)}$ [15]	~		×
WSRM [9]	×	×		cosine [3,25]	×	×	

Table 1: Entity relatedness measures in the light of well-foundedness requirements: × indicates the requirement is met, while ~ indicates it is only partially met.

where $|x \xrightarrow{\tau} *|$ is the number of outgoing τ relations for x , while $|* \xrightarrow{\tau} y|$ is the number of incoming τ relations for y ; 1 is subtracted to avoid counting the relation $|x \xrightarrow{\tau} y|$ twice. Given a path $P = x_1 \xrightarrow{\tau_1} x_2 \xrightarrow{\tau_2} \dots \xrightarrow{\tau_{k-1}} x_k$ within the KB, its weight is

$$\text{weight}(P) = \frac{1}{\sum_{i=1}^{k-1} 1/\text{exclusivity}(x_i \xrightarrow{\tau_i} x_{i+1})} . \quad (3)$$

Finally $rel_{Excl}^{(k)}$ is defined as the weighted sum of the top-k paths with highest weight between x and y

$$rel_{Excl}^{(k)}(x, y) = \sum_{P \in P_{xy}^k} \alpha^{\text{length}(P)} \text{weight}(P) \quad (4)$$

where P_{xy}^k denotes the top-k paths with highest weight between x and y , and $\alpha \in [0, 1]$ is a constant length decay factor introduced to give preference to shorter paths.

We remark that the above definition relies on paths between entities to measure their relatedness **(R3)**. However, we note that the semantics of $rel_{Excl}^{(k)}$ is controlled with parameters whose "good" values are hard to guess, though $k = 5$ and $\alpha = 0.25$ are recommended default values based on empirical observations. Thus $rel_{Excl}^{(k)}$ hardly meets **(R1)**. Further, the above definition requires to compute all the paths within the KB, which may not be computationally feasible since in large KBs, like the encyclopedic ones used for entity linking, the number of paths blows up as the considered path length increases; hence $rel_{Excl}^{(k)}$ does not meet **(R2)**.

Cosine similarity [3,25] is used to measure the semantic relatedness between two entities in entity linking systems based on embeddings, e.g., [22,24,19,5]: entities are mapped into coordinates of a multidimensional space, in which the closer two entities are, the more related they are. Several kernels exist for computing such embeddings, e.g., [3,25,23]. While the cosine similarity itself has a clear semantics **(R1)** and is not costly to compute **(R2)**, the machine learning-based construction of the entity embeddings cannot guarantee that cosine similar entities are indeed somehow related through some path in the KB, hence does not meet **(R3)**.

Tab. 1 recaps the above discussion and highlights that none of the entity relatedness measures used so far in the entity linking literature meets the three requirements of well-foundedness. Devising a measure that meets them all is a contribution of this paper, which we present next.

3 The Path-based Weighted Semantic Relatedness Measure

Our approach to define a novel entity relatedness measure that meets all the well-foundedness requirements extends a measure from the literature that only considers

properties (direct relations) between entities, to a measure that considers paths between entities. In the sequel, we chose to rely on WSRM to capitalize (i) on properties **(R1)** and **(R2)** that WSRM verifies and (ii) on its state-of-the-art performance for collective entity linking, in particular w.r.t. Ref [9].

A straightforward extension of WSRM to take into account paths between entities would consist in counting the paths between the entities e_i, e_j and e_i, e' , instead of the properties r and r' respectively in Eq. 1. However, the resulting measure would loose **(R2)** as it would require to compute all the paths between the entities in the KB. To circumvent this issue and retain **(R2)**, one may be tempted to only count paths up to some typically small length, as it is well-known (e.g., [15]) that the longer a path between two entities, the weaker the semantics of the relation it encodes. Still, in this case, though clear, the semantics of the resulting measure is poor as it does not account for the strength of the paths between entities.

Instead, in addition to bounding the length of the paths we consider, we do aggregate the WSRM values of the successive entity pairs found along a path between two entities, so that the resulting value reflects how related these entities are through this particular path. Further, since many paths (with same or different lengths) may relate two entities, we also aggregate the individual relatedness values of these paths into a final entity relatedness score. Hereafter, the aggregation operator for the WSRM values found along a path is denoted \otimes , while the one for path scores is denoted \oplus . Tough typical candidate operators for \otimes and \oplus are either min and max, or product and sum, we chose fuzzy logic operators (discussed shortly) modeling the counterparts of the Boolean logical AND and OR operators in the $[0,1]$ interval (recall that WSRM values are also within this interval). We now discuss three strategies to combine path relatedness values, yielding a family of entity relatedness measures.

The first strategy consists in aggregating all paths of length m separately, and aims at showing the contribution of paths with different lengths when considered separately. Formally, we define the weighted semantic relatedness measure for path of length m between entities e_i and e_j as

$$\text{ASRMP}_m^a(e_i, e_j) = \oplus_{p \in e_i \rightsquigarrow e_j, |p|=m} \otimes_{k=1}^{|p|-1} \text{WSRM}(p_k, p_{k+1}) \quad , \quad (5)$$

where $e_i \rightsquigarrow e_j$ denotes the set of paths between e_i and e_j , here limited to paths of length m , and p_k is the k^{th} entity along path p (hence $p_1 = e_i$ and $p_{|p|} = e_j$). The inner \otimes operator aggregates the WSRM scores along the edges of a given path; the outer \oplus operator aggregates scores obtained for different paths of length m between the two entities. The cost of the different aggregations is low, so $\text{ASRMP}_m^a(e_i, e_j)$ meets both **(R2)** and **(R3)**. It however only roughly meets **(R1)**, because the semantics is deteriorated by combining separately the paths of different lengths at a subsequent stage, e.g., in the entity linking process.

A second strategy consists in aggregating all paths of length less or equal m , as opposed to limiting ourselves to paths of a given length, extending Eq. 5 as

$$\text{ASRMP}_m^b(e_i, e_j) = \oplus_{p \in e_i \rightsquigarrow e_j, |p| \leq m} \otimes_{k=1}^{|p|-1} \text{WSRM}(p_k, p_{k+1}) \quad . \quad (6)$$

This measure provides a first approach to combining paths of different lengths, however assuming equal weight for all of them. This assumption seems unrealistic: intuitively,

direct relations are expected to account for strong relations, while indirect ones are weaker, where the longer the path, the weaker the relation. We thus introduce a weight depending on the path length according to

$$\text{ASRMP}_m^c(e_i, e_j) = \sum_{l=1}^m \sum_{p \in e_i \rightsquigarrow e_j, |p|=l} w_l \otimes_{k=0}^{|p|-1} \text{WSRM}(e_k, e_{k+1}) \quad , \quad (7)$$

where w_l is a length-dependent weight roughly corresponding to the percentage of useful paths of length l and optimized by grid search. Thus, $\text{ASRMP}_m^b(e_i, e_j)$ meets the three requirements while $\text{ASRMP}_m^c(e_i, e_j)$ does not meet **(R1)**, because the semantics is once again deteriorated by the introduced weight.

Finally, all measures are made symmetrical according to

$$\psi_m^x(e_i, e_j) = \frac{1}{2} (\text{ASRMP}_m^x(e_i, e_j) + \text{ASRMP}_m^x(e_j, e_i)) \quad x \in \{a, b, c\} \quad . \quad (8)$$

The rationale for symmetrization is that in an RDF KB, if a triple (e_i, r, e_j) exists, the symmetric triple (e_j, r^-, e_i) may not exist at the same time, e.g., for r, r^- the symmetric properties 'hasWritten', 'writtenBy' respectively. This depends on the modeling choices adopted for the KB at design time.

Aggregating scores with fuzzy logic The score aggregators used in the definition of ASRMP_m^x are crucial: they have to be chosen so as to preserve the semantics of the relations between entities without introducing noise, i.e., semantic drift. The longer a path between two entities, the smaller should be the relatedness value because the link between the entities may become meaningless. Typically, a product of WSRM values along a path will quickly decrease, resulting into useless scores; the average score can be noisy. For two given entities with a direct link and indirect links, the average can also result in scores for paths of length $m > 1$ larger than the score for the direct link, which we assume to be semantically incorrect. Hence we advocate for fuzzy logic operators which provide a wide range of aggregators, such as the equivalent of the AND/OR logic operators for real values in the $[0, 1]$ interval. The semantics of the fuzzy operators is also important because it allows to explain the linking decisions and ensures **(R1)**.

Fuzzy logic, especially triangular norm fuzzy logic (*t-norm*) which guarantees triangular inequality in probabilistic spaces, generalizes intersection in a lattice and conjunction in logic, offering many aggregation operators to define conjunction for values within $[0, 1]$. Each t-norm operator is associated with an s-norm (t-conorm) with respect to De Morgan's law: $S(x, y) = 1 - T(1 - x, 1 - y)$. The t-norm is the standard semantics for conjunction in fuzzy logic and thus the couple t-norm/s-norm acts as AND/OR operators on real values in $[0, 1]$. Thus using fuzzy logic to define our relatedness measure allows to ensure its transitivity by definition and avoids the introduction of arbitrary weighting parameters like in $rel_{Excl}^{(k)}$.

As $\text{WSRM}(e, e') \in [0, 1]$, any t-norm/s-norm couple can be used to aggregate values along one path of length m and across all paths between two entities. We experimented with several couples of fuzzy operators: beside the classical min/max, we also

consider the family of Hamacher t-norms (Hamacher product) defined for $\lambda \geq 0$ as

$$T_{H,\lambda}(x, y) = \frac{xy}{\lambda + (1 - \lambda)(x + y - xy)} \quad , \quad (9)$$

the family of Yager t-norms defined for $\lambda > 0$ as

$$T_{Y,\lambda}(x, y) = \max \begin{cases} 0 \\ 1 - \sqrt[\lambda]{(1-x)^\lambda + (1-y)^\lambda} \end{cases} \quad (10)$$

and the Einstein sum

$$T_E(x, y) = \frac{xy}{1 + (1-x)(1-y)} \quad . \quad (11)$$

The two families of t-norm used here are not exhaustive but generalize many t-norms: one can easily see that $T_{H,2}(x, y) = T_E(x, y)$; $T_{H,1}(x, y)$ is known as the product t-norm; $T_{Y,1}(x, y)$ is the Łukasiewicz t-norm. We studied a large body of those operators and chose the one maximizing the accuracy of the collective linking system described hereunder.

4 Linking with entity relatedness measure

We study the interest of the entity relatedness measure in the context of entity linking. In a general collective entity linking pipeline, semantic relatedness measures between entities are used at the end of the process to globally select the best candidate entity for each mention. They are typically used within a classifier along with features describing the mapping between the mention and the entity, to predict whether an entity is a good match (1) for a mention or not (0). The classifier operates independently on each mention-entity pair, and allows an ensemble of local classifications based on the relatedness of the entity to candidate entities of other mentions.

We briefly review the entity linking pipeline that we adopted. As in many previous pieces of work, e.g., [9,11,22,24,5], we do not consider the initial named entity recognition step, assuming perfect entity mention detection. The next step is the candidate entity generation stage, which consists in determining for each mention a reduced set of plausible entities that the mention could refer to. The final stage is the candidate selection stage, a.k.a. disambiguation, in which the best candidate is selected for each mention taking into account possible relations to candidates from other mentions.

In the remainder of this section, a document D is represented by its set of entity mentions, $D = (m_1, \dots, m_n)$. For each mention m_i , $C(m_i) = (e_{i1}, \dots, e_{ik})$ denotes the set of its candidate entities.

4.1 Knowledge Base

In this paper, we focus on two RDF KBs, namely Yago⁶ and BaseKB⁷, but however make use of Wikipedia for candidate generation for practical reasons, since the names

⁶ <https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago>

⁷ <http://basekb.com/>

of Wikipedia pages are meaningful unique identifiers unlike entities' labels in KB. BaseKB, derived from Freebase, contains over one billion facts (i.e., triples) about more than 40 millions subjects. Yago, derived from Wikipedia, WordNet and GeoNames, currently has knowledge of more than 10 million subjects and contains more than 120 million facts. Within those two KBs, interrelationships between entities bear precise semantics as specified by their schema. Contrary to Yago, BaseKB is saturated, i.e., all facts are made explicit with property instances thus circumventing the need for reasoning mechanisms. As, for practical reasons, we take advantage of Wikipedia in the candidate generation step, a mapping between Wikipedia and Yago or BaseKB entities is maintained. We also limit ourselves to entities appearing both in Wikipedia and in the RDF KB, resulting in approximately 2.5M entities in BaseKB and 3M entities in Yago.

Note that while BaseKB and Yago are used in this paper, there are no conceptual limitations to those KBs, ASRMP_m being able to account for any RDF KB schema.

4.2 Candidate Entity Generation

The generation of the candidate entities e_{ij} for each mention m_i in a document relies on Cross-Wiki, a dictionary computed from a Google crawl of the web that stores the frequency with which a mention links to a particular entity within Wikipedia. We used the same Cross-Wiki dictionary as in [12]. Each entry of the dictionary corresponds to a possible entity mention and provides a list of Wikipedia entities to which the mention points to, along with popularity scores. This list is directly used for candidate generation whenever a mention appears in the dictionary. The dictionary entries are normalized by removing all punctuation marks and converting to lower case. For mentions absent from Cross-Wiki, a query on Wikipedia was performed using the text of the mention, and the resulting Wikipedia pages were collected as the candidate entities.

4.3 Supervised Entity Selection

To select the best candidate entity $e_{i\hat{j}}$ for each entity mention m_i in a document in a collective manner, we adopted a supervised approach similar to [9,22,27], where a classifier is trained to predict whether a mention and a candidate entity are related (1) or not (0). We used a binary logistic regression, denoted $\text{logreg}()$, applied independently on each mention-candidate entity pair, selecting for a mention m_i the candidate entity with the highest response from the classifier, i.e., $\hat{j} = \arg \max_j \text{logreg}(m_i, e_{ij})$. We also experimented with different classifiers—see Sec. 5.3 for details—and the choice of a binary logistic regression is motivated by its simplicity and the fact that it turned out the best classification strategy. In our collective setting, the classifier relies on features describing the similarity between the mention and the entity on the one hand, and, on the other hand, the relatedness of the candidate entity under consideration with the candidate entities from other mentions in the document. The latter accounts for the context and ensures the collective aspect of the linking.

For the similarity between the mention and the candidate entity, we considered two features namely the cosine similarity between the vectors representations of the mention and of the entity name within Wikipedia, as obtained with word2vec [21], and the Wikipedia popularity as provided by Cross-Wiki.

For the relatedness of the candidate entity e_{ij} with candidate entities from other mentions, i.e., e_{kl} with $k \neq i$, we relied on an aggregation of the scores $\phi(e_{ij}, e_{kl})$ over the set of candidate entities $\cup_{k \neq i} C(m_k)$, where $\phi()$ is an entity relatedness measure (e.g., $rel_{Excl}^{(k)}$, WSRM, ASRMP), thus providing a global measure of how e_{ij} relates to other entity propositions in D . This aggregation is different from the one used to design our relatedness measure. We used sum and maximum aggregation, which has proven efficient in previous work. Formally, considering an entity relatedness $\phi()$, we define the sum aggregator as

$$S(e_{ij}; D) = \sum_{k=1, k \neq i}^n \sum_{e \in C(m_k)} \phi(e_{ij}, e) , \quad (12)$$

and the maximum aggregators as

$$M^k(e_{ij}; D) = \max_{k=1, k \neq i}^n \max_{e \in C(m_k)} \phi(e_{ij}, e) \quad (13)$$

where $\max@k$ is an operator returning the k^{th} highest value. Note that the two aggregators are complementary: the sum provides a global averaged view while the max values emphasize good matches. We observed that retaining the sum, $\max@1$, $\max@2$ and $\max@3$ aggregators as global features for the logistic regression worked best for the relatedness measure $\psi_1^a()$. We therefore retained the same strategy for $\psi_2^a()$, and $\psi_3^a()$ resulting in a total of 12 global features—namely S_m , $M_m^{(1)}$, $M_m^{(2)}$ and $M_m^{(3)}$ for $m = 1, 2, 3$ —to represent the relatedness of a candidate entity with other possible entities in D . Experiments with $\psi_m^x()$ with $x \in \{b, c\}$, i.e., where different path lengths are already aggregated within $ASRMP_m^x$, involve only 4 global features, i.e., sum, $\max@1$, $\max@2$ and $\max@3$. Thus $ASRMP_m^a$ leverages 12 global features while $ASRMP_m^b$ and $ASRMP_m^c$ only use 4.

5 Experiments

In the remainder of the paper, we report on a set of experiments conducted to assess the benefit of our entity relatedness measure in a collective entity linking task. We are using different entity relatedness measures, within the same collective entity linking pipeline as described per Sec. 4. Experiments are mostly carried out on the TAC-KBP Entity Discovery and Linking (EDL) 2016-2017 datasets, two newswire and forum-discussion sets of documents originally collected for the TAC Knowledge Base Population Entity Discovery and Linking 2016 and 2017 international evaluation campaigns [16], which constitute the reference for the task of entity linking. Results are reported in terms of F1-score, where precision $P = \frac{|G \cap S|}{|S|}$ and recall $R = \frac{|G \cap S|}{|G|}$ are calculated between the linking in the gold-standard (G) and the linking given by a system (S). The 2016 version was used to learn the classifiers while the 2017 one served as test set. As the collective entity linking system is trained while only changing the entity relatedness measure, the linking accuracy can be used to evaluate the quality of the entity relatedness measure.

After providing implementation details in Sec. 5.1, selecting the best fuzzy aggregator in Sec. 5.2 and the best classification strategy in Sec. 5.3, we compare in Sec 5.4

the various flavors of $ASRMP_m$ seeking for the best one. The latter is compared to the entity relatedness measures used for entity linking in the literature in Sec. 5.5. Finally, we compare in Sec. 5.6 our collective entity linking system to a series of competing systems.

5.1 Implementation Details

Computing *all* the paths of length m between *every* pair of entities in the KB can be computationally expensive. For instance, in BaseKB, and after data cleansing, there are approximately 13M paths of length one and 46B paths of length two. We designed an efficient way of doing so, taking advantage of a relational database management system—which offers today much more tuning opportunities than RDF data management systems, e.g., various indices, clustered tables, etc.—to store edges and their semantic relatedness weights.

In PostgreSQL 11.2⁸, a table `edges(e_1, e_2, v)` is used to store the pairs of entities (e_1, e_2) directly connected through some property in the KB, along with the corresponding WSRM value v . This table is dictionary-encoded (entity names are replaced by integers) to save space and speed up value comparisons, indexed by (e_1, e_2) and (e_2, e_1) values to offer many options to the PostgreSQL optimizer. Limiting ourselves to path lengths $m \leq 4$, the four tables `path1(e_1, e_2, v_1)`, `path2(e_1, e_2, v_1, v_2)`, `path3(e_1, e_2, v_1, v_2, v_3)` and `path4($e_1, e_2, v_1, v_2, v_3, v_4$)` are efficiently created from the edge table using SQL queries, to represent paths of length 1, 2, 3 and 4 respectively. The entities e_1 and e_2 are restricted to the candidate entities for the entity mentions found in the TAC-KBP2016-2017 datasets: entities along the paths may however not be candidate entities. The values v_i are the WSRM values along the path.

In BaseKB, we obtained approximately 53K one-, 11M two- and 2B three-edges paths, from which we computed the various $ASRMP_m$, relatedness values. We were not able to compute paths of length four, as the number of paths exploded. The same process was applied to Yago and we obtained approximately 28K one-, 845K two-, 25M three- and 679M four-edges paths. Paths of length four could be computed due to the cleanliness and the higher structure of Yago.

5.2 Comparing Fuzzy Logic Aggregators

One crucial issue for paths of length $m > 1$ lies in the aggregation of the semantic relatedness measure of each edge along the path and of the relatedness measure over multiple paths between two entities. $ASRMP_m$ reflects entity relatedness in the KB at hand: obviously, an aggregation of its values should reflect similar properties. Moreover, and in order to avoid a semantic drift, the resulting value of the aggregation for one path of length m must be smaller than that of a path of length $m - 1$ since the latter bears stronger semantics. Finally, because there can be many paths between two entities, one needs also to aggregate the values of the different paths connecting two given entities.

Experimental results (not reported here for lack of space) show that $T_{h,0}(x, y)$ is the best aggregator with the collective linking setting in this paper. We however experimentally observed only minor differences between the Hammacher and Yager t-norms and

⁸ <https://www.postgresql.org>

Approach	BaseKB					Yago				Yago+Saturation					
	KNN	DT	GB	SVM	REG	KNN	DT	GB	SVM	REG	KNN	DT	GB	SVM	REG
ASRMP ₁	49.58	47.71	79.59	79.19	80.03	49.64	47.51	79.67	79.75	79.88	50.46	47.58	80.05	79.60	79.94
ASRMP ₂ ^a	50.00	47.10	79.75	79.82	80.79	49.13	47.02	80.93	79.52	80.71	49.46	46.99	79.09	80.15	80.78
ASRMP ₃ ^a	50.02	47.24	80.20	80.12	80.60	49.48	46.79	80.36	79.66	80.40	50.33	46.74	79.42	79.62	80.67
ASRMP ₄ ^a	-	-	-	-	-	50.20	46.78	78.56	80.40	80.98	49.43	46.79	80.51	80.78	81.34

Table 2: F1-scores for various classifiers within the entity linking system for TAC-KBP.

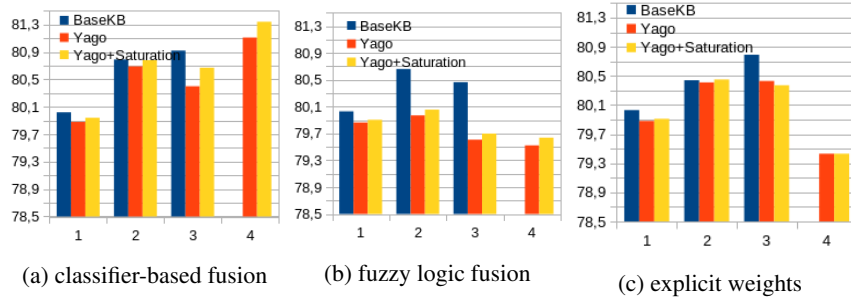


Fig. 1: Linking F1-score for various aggregation strategies.

various values of λ . In the remainder, $T_{h,0}(x, y)$ with its associated s-norm is used for the aggregation of paths of length $m \in \{2, 3, 4\}$ between two entities.

5.3 Comparing Classifiers

We compared several classifiers within our collective entity linking system. In addition to popular classification techniques such as k-nearest neighbours (KNN), decision trees (DT), logistic regression (REG) or support vector machines (SVM), we also experimented with gradient boosting (GB). The latter was used in previous work on entity relatedness for entity linking [27,28]. Results reported in Tab. 2 for ASRMP_m^a, $m \in \{1, 2, 3, 4\}$, on the TAC-KBP dataset using either BaseKB or (saturated) Yago as KB, clearly show that the logistic regression classification strategy turns out to be the best option overall, in particular when considering paths of length 2 or more.

5.4 Comparing Aggregation Strategies

We also compared the aggregation strategies described in Sec. 3, reporting in Fig. 1 the F1-score as a function of m for the various strategies: distinct ASRMP_m^a measures for each value of m (including length four for Yago) aggregated by the classifier; aggregation with fuzzy logic as defined by ASRMP_m^b; explicit weighting as in ASRMP_m^c optimized by grid search. In most cases, better performance is achieved for $m = 2$, diminishing for $m > 2$, which confirms that paths longer than 2 mostly bring noise because of a semantic drift. This is particularly visible in Fig. 1b. Classifier-based fusion, Fig. 1a, however seems to keep increasing for $m = 3$ on BaseKB, but the gain is only minimal between $m = 2$ and $m = 3$ and is counterbalanced by the computational cost (see Sec. 5.5), specially for BaseKB. Interestingly, for explicit weighting, the weights w_l can be seen as the strength of the paths with length l . We found that the optimal values of w_l decrease when l increases, i.e., $w_2 = 1$, $w_3 = 0.1$ and $w_4 = 0.1$ for Yago.

Features	BaseKB	Yago	Yago+Saturation
local (no collective)	78.72	78.72	78.72
local+cosine similarity(rdf2vec)	78.58	78.58	78.58
local+cosine similarity(TransE)	79.39	79.39	79.39
local+Ref	79.70	79.81	79.82
local+ $rel_{Eycl}^{(5)}$	80.54	80.49	79.27
$ASRMP_1 = \text{local} + S_1 + M_1^{(k)}$	80.03	79.88	79.94
$ASRMP_1 + S_2$	80.02	80.02	80.12
$ASRMP_1 + M_2^{(k)}$	80.68	80.69	80.78
$ASRMP_2^a = ASRMP_1 + S_2 + M_2^{(k)}$	80.79	80.71	80.78
$ASRMP_2^a + S_3$	80.92	80.77	80.77
$ASRMP_2^a + M_3^{(k)}$	80.55	80.35	80.76
$ASRMP_3^a = ASRMP_2^a + S_3 + M_3^{(k)}$	80.60	80.40	80.67
local + $S_2 + M_2^{(k)}$	80.16	80.60	80.52
local + $S_3 + M_3^{(k)}$	80.42	79.46	79.27

Table 3: Linking F1-score on the TAC-KBP2017 dataset. Popularity and cosine similarity are the local mention-entity scores; the sum (S_m) and max ($M_m^{(k)}$) global features are defined in Eq. 12 and Eq. 13 resp.

These different aggregation studies show that fuzzy aggregator (Fig. 1b) and explicit weights (Fig. 1c) are more robust for combining paths of different lengths, while the classifier-based fusion (Fig. 1a) is more accurate though it introduces noise for paths of length > 2 . For example, in both Fig. 1b and Fig. 1c paths of length four are always adding noise, when considered with Yago and Yago saturated. With respect to the entity linking task, $ASRMP_m^a$ with classifier-based fusion appears the best strategy. In all generality and contrary to $ASRMP_m^b$, this strategy only loosely verifies **(R1)** as classifier-based fusion can be difficult to interpret. In this regard, logistic regression nevertheless offers interesting properties, with coefficients and intercepts that can be interpreted to some extent.

5.5 Comparisons of entity relatedness measures

We now concentrate on the study of (the different components of) $ASRMP_m^a$, $m > 1$, with classifier-based fusion, and how it compares with other relatedness measures, namely WSRM [9], cosine similarity [25,3] and Ref [1,18]. All measures are used within the same collective entity linking system as input features to the classifier, thus providing fair comparison of the entity relatedness measures. Results are gathered in Tab. 3 for BaseKB, Yago and Yago saturated, reporting linking accuracy (F1-score). The different measures compared are:

- Local performs linking using only the two local features depicting the adequacy of the mention and the entity—see Sec. 4.3—thus not considering entity relatedness
- Cosine similarity(kernel), the kernel being either rdf2vec [25] or TransE [3], measures entity relatedness as the cosine similarity between the entities embedded in a high-dimension space with the given kernel

	<i>TransE</i>	$rel_{Excl}^{(5)}$	<i>Ref</i>	ASRMP ₁ = WSRM	ASRMP ₂ ^a	ASRMP ₃ ^a
BaseKB	15.29	1680	13.33	13.85	20,94	418,85
Yago	0.84	507	0.58	0.59	6.75	9.17
Yago+Saturation	0.79	403	0.57	0.69	6.14	8.60

Table 4: Time in (min.) for different entity relatedness measures.

- *Ref* [1,18] considers the *Ref* entity relatedness measure as defined in Eq 1
- $Rel_{Excl}^{(5)}$ [15] uses entity relatedness as defined in Eq 4 with $k = 5$
- WSRM [9], which is equivalent to ASRMP₁, where only direct paths are used to measure entity relatedness
- ASRMP_m^a which embed basic reasoning mechanisms accounting for paths of length $m > 1$

Adding paths of length 2 allows a slight increase of the linking accuracy, where the best score for ASRMP₂^a is obtained using both S_2 and $M_2^{(k)}$ for $k = 1, 2, 3$ (row ASRMP₂^a). Looking separately at the benefit of the aggregators S_2 and $M_2^{(k)}$ across couples of candidate entities, we see that considering only the maximum increases the accuracy of the ASRMP₁ system but, as it reflects the predominant topic, mentions that are far from that general topic can be incorrectly linked. Meanwhile, using S_2 can be slightly worse than ASRMP₁ only (e.g., on BaseKB, not on Yago) because this aggregator reflects choosing the mean topic which can be very vague. Combining both seems to be a compromise between the two extreme cases. On the other hand, ASRMP₂^a is better than both WSRM [9] and $Rel_{Excl}^{(5)}$ [15] showing the interest of using a well founded entity relatedness measure along with property paths.

Paths of length 3 can further be successfully combined with the features used for ASRMP₂^a when S_3 is considered; while using $M_3^{(k)}$, either alone or with S_3 , seems to introduce noise in the linking decision. This counter-intuitive result can be explained by the fact that introducing path of length three adds limited relevant semantics into the relatedness measure. As an outcome, considering the predominant entities only (max aggregators) tends to take strong linking decision and can be more drastic than adding vague links, mostly for entities that were not linked with the aggregation of ASRMP₁ and ASRMP₂^a.

From the complexity point of view, relatedness measures are computed offline for a static KB (a given version of Yago or BaseKB). Meanwhile ASRMP_m^x can easily be computed for lower values of m making it tractable and more suitable for dynamic scenarii where entities are added to or removed from the KB, unlike rel_{Excl}^k where top-k path has to be computed, or cosine similarity where the kernel embedding has to be retrained. Tab. 4 shows the computation time for the different entity relatedness measures, including the offline part. For small values of m , which are required in practice, *Ref*, ASRMP_m^a, and *TransE* have low computation cost, while rel_{Excl}^k has high computation cost due to the need to compute top-k best paths. Thus we can conclude that ASRMP_m^a meets **(R2)**, and more generally that ASRMP_m^x with $x \in \{a, b, c\}$ meets **(R2)**. They indeed have similar computation times: most of the time is spent in computing paths of length up to m , while aggregating path scores is very fast.

Approach	AIDA-A	AIDA-B	Reuters128	RSS-500
NCEL [5]	79.0	80.0	-	-
AIDA [13]	74.3	76.5	56.6	65.5
PBoH [11]	79.4	80.0	68.3	55.3
CEL-ASRMP ₁ =CEL-WSRM	90.6	87.7	76.6	76.4
CEL-ASRMP ₂ ^a	93.8	91.0	77.5	76.6
CEL-ASRMP ₃ ^a	93.4	90.6	78.5	76.6
CEL-ASRMP ₄ ^a	93.1	90.3	76.6	74.6

Table 5: Micro-averaged F1 score for different collective entity linking systems on four standard datasets.

We also studied the impact of the saturation of the KB using Yago. As shown in Tab. 3 (columns 3 and 4) and in Fig. 1 (red and yellow bars), the gain is very limited in the case of TAC-KBP2017 dataset. In practice, this result saves the explicit computation of the implicit triples in the RDF KB.

5.6 Comparison of entity linking systems

We finally compared the collective entity linking system based on ASRMP_m^a with prominent state-of-the-art methods over standard benchmarks: NCEL [5], AIDA [13], PBoH [11] and CEL-WSRM [9]. All follow the classical three stage architecture for collective entity linking. CEL-WSRM [9] is based on the WSRM entity relatedness measure (Eq. 1), equivalent to ASRMP₁. Results of the entity linking process, evaluated in terms of micro-averaged F1 classification scores, are reported in Tab. 5. These results were obtained with the Yago KB that allows considering paths of length up to 4. Similar results are obtained when the Yago KB is saturated. On all four datasets, the proposed method CEL-ASRMP_m^a, $m \in \{2, 3, 4\}$, does outperform the NCEL, AIDA and PBoH collective linking approaches by a large margin. The proposed method is better than CEL-WSRM on the four datasets, with small improvement on the RSS-500 dataset. Moreover, we observe the same conclusion as before: paths of length two improve the accuracy of the linking, while longer paths may add noise.

6 Conclusions

In summary, we extended previous measures of entity relatedness within a knowledge base to account for indirect relations between entities through the consideration of property paths. The measure that we proposed is the first to satisfy the three good properties that such measures should have: clear semantics, reasonable computational cost and transitivity. We experimentally showed its benefit in a collective entity linking task, where paths of length 2 and 3 bring improvement over the state of the art in collective entity linking, using either only direct connections between entities [9] or previous work on path-based relatedness measures [15]. In theory, the scalability of ASRMP_m varies in inverse proportion with the length of the paths. We however proved it to be still tractable for reasonable sized datasets with paths of length up to 3, which is sufficient in practice as longer paths add noise.

This contribution opens up new horizons towards fully exploiting the semantics of RDF knowledge bases for entity linking, when only relatedness measures are used. Taking a historical perspective, this task was first conventionally addressed leveraging entity relatedness measures based on Wikipedia hyperlinks counts between two pages and on the presence of one relation between two entities in the KB. WSRM (= ASRMP₁) made use of KB semantics by weighting the relatedness between entities exploiting the basic properties within the KB. The ASRMP_m extension proposed here further introduces (basic) reasoning mechanisms that exploit the graph-structure of the KB alongside robust aggregators for paths of arbitrary length. In this work, all paths were considered regardless of their precise semantics. In specific application contexts, this could be improved by selecting paths between two entities that are semantically meaningful in this context, e.g., using ontological knowledge and reasoning.

References

1. Alhelbawy, A., Gaizauskas, R.: Graph ranking for collective named entity disambiguation. In: 52nd Annual Meeting of the Association for Computational Linguistics. vol. 2, pp. 75–80. Baltimore, Maryland, USA (2014)
2. Blanco, R., Ottaviano, G., Meij, E.: Fast and space-efficient entity linking for queries. In: 8th ACM International Conference on Web Search and Data Mining. pp. 179–188. Shanghai, China (2015)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: 27th Advances in Neural Information Processing Systems. pp. 2787–2795. Lake Tahoe, Nevada, USA (2013)
4. Bunescu, R., Paşca, M.: Using encyclopedic knowledge for named entity disambiguation. In: 11th Conference of the European Chapter of the Association for Computational Linguistics. pp. 9–16. Trento, Italy (2006)
5. Cao, Y., Hou, L., Li, J., Liu, Z.: Neural collective entity linking. In: 27th International Conference on Computational Linguistics. pp. 675–686. Santa Fe, New Mexico, USA (2018)
6. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 708–716. Prague, Czech Republic (2007)
7. Detyniecki, M.: Mathematical aggregation operators and their application to video querying. Ph.D. thesis, Univ. Paris 6 (2000)
8. Durrett, G., Klein, D.: A joint model for entity analysis: coreference, typing, and linking. *Transactions of the Association for Computational Linguistics* **2**, 477–490 (2014)
9. El Vaigh, C.B., Goasdoué, F., Gravier, G., Sébillot, P.: Using knowledge base semantics in context-aware entity linking. In: ACM Symposium on Document Engineering 2019. pp. 8:1–8:10. Berlin, Germany (2019)
10. Francis-Landau, M., Durrett, G., Klein, D.: Capturing semantic similarity for entity linking with convolutional neural networks. In: 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1256–1261. San Diego, CA, USA (2016)
11. Ganea, O.E., Ganea, M., Lucchi, A., Eickhoff, C., Hofmann, T.: Probabilistic bag-of-hyperlinks model for entity linking. In: 25th International Conference on World Wide Web. pp. 927–938. Montréal, Québec, Canada (2016)
12. Gupta, N., Singh, S., Roth, D.: Entity linking via joint encoding of types, descriptions, and context. In: 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2681–2690. Copenhagen, Denmark (2017)

13. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenu, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: 2011 Conference on Empirical Methods in Natural Language Processing. pp. 782–792. Edinburgh, Scotland, UK (2011)
14. Huang, H., Heck, L., Ji, H.: Leveraging deep neural networks and knowledge graphs for entity disambiguation. arXiv preprint arXiv:1504.07678 (2015)
15. Hulpuş, I., Prangnawarat, N., Hayes, C.: Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In: 14th International Semantic Web Conference. pp. 442–457. Bethlehem, PA, USA (2015)
16. Ji, H., Pan, X., Zhang, B., Nothman, J., Mayfield, J., McNamee, P., Costello, C.: Overview of TAC-KBP2017 13 languages entity discovery and linking. In: Text Analysis Conference. Gaithersburg, Maryland, USA (2017)
17. Le, P., Titov, I.: Improving entity linking by modeling latent relations between mentions. In: 56th Annual Meeting of the Association for Computational Linguistics. pp. 1595–1604. Melbourne, Australia (2018)
18. Ling, X., Singh, S., Weld, D.S.: Design challenges for entity linking. *Transactions of the Association for Computational Linguistics* **3**, 315–328 (2015)
19. Liu, M., Gong, G., Qin, B., Liu, T.: A multi-view-based collective entity linking method. *ACM Transactions on Information Systems* **37**(2), 23:1–23:29 (2019)
20. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: 7th International Conference on Semantic Systems. pp. 1–8. Graz, Austria (2011)
21. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*. pp. 3111–3119. Lake Tahoe, Nevada, USA (2013)
22. Moreno, J.G., Besançon, R., Beaumont, R., D’hondt, E., Ligozat, A.L., Rosset, S., Tannier, X., Grau, B.: Combining word and entity embeddings for entity linking. In: 14th European Semantic Web Conference. pp. 337–352. Portorož, Slovenia (2017)
23. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: 2014 Conference on Empirical Methods in Natural Language Processing. pp. 1532–1543. Doha, Qatar (2014)
24. Phan, M.C., Sun, A., Tay, Y., Han, J., Li, C.: Pair-linking for collective entity disambiguation: Two could be better than all. *IEEE Transactions on Knowledge and Data Engineering* (2018)
25. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: Rdf2vec: Rdf graph embeddings and their applications. *Semantic Web* **10**(4), 721–752 (2019)
26. Wang, H., Zheng, J.G., Ma, X., Fox, P., Ji, H.: Language and domain independent entity linking with quantified collective validation. In: 2015 Conference on Empirical Methods in Natural Language Processing. pp. 695–704. Lisbon, Portugal (2015)
27. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. In: 20th SIGNLL Conference on Computational Natural Language Learning. pp. 250–259. Berlin, Germany (2016)
28. Yang, Y., İrsoy, O., Rahman, K.S.: Collective entity disambiguation with structured gradient tree boosting. In: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 777–786. New Orleans, Louisiana, USA (2018)