Entity Reference Resolution via Spreading Activation on RDF-Graphs

Joachim Kleb and Andreas Abecker

FZI Research Center for Information Technology Karlsruhe
Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
surname@fzi.de

Abstract. The use of natural language identifiers as reference for ontology elements—in addition to the URIs required by the Semantic Web standards—is of utmost importance because of their predominance in the human everyday life, *i.e.* speech or print media. Depending on the context, different names can be chosen for one and the same element, and the same element can be referenced by different names. Here homonymy and synonymy are the main cause of ambiguity in perceiving which concrete unique ontology element ought to be referenced by a specific natural language identifier describing an entity. We propose a novel method to resolve entity references under the aspect of ambiguity which explores only formal background knowledge represented in RDF graph structures. The key idea of our domain independent approach is to build an entity network with the most likely referenced ontology elements by constructing steiner graphs based on spreading activation. In addition to exploiting complex graph structures, we devise a new ranking technique that characterises the likelihood of entities in this network, *i.e.* interpretation contexts. Experiments in a highly polysemic domain show the ability of the algorithm to retrieve the correct ontology elements in almost all cases.

1 Introduction

The World Wide Web provides access to content produced by people all over the world including people of very different cultural and local backgrounds. Hence a large variety of content providers create articles using different writing styles—based on their different background—that contain the same or similar information. Thus the comparison of information becomes harder as different terms for equal or similar information are common involving the problem of ambiguity. For example, the same person can be called "John Doe" in one and "John D." or "Johnny" in another source.

In ontology-based applications, one has to associate additional literal relations to concepts and instances, in order to cover their different identifiers used in input media. This raises the problem of ontology-element identification concerning the retrieval of information in knowledge bases (KBs). Here a specific ontology element cannot necessarily be determined uniquely in a KB, solely based on its natural language identifier (NLI, identifiers using natural language names), due to their ambiguity. As opposed to URIs, NLIs do not guarantee uniqueness. For example, the DBPedia ontology[1]

[1] http://dbpedia.org

L. Aroyo et al. (Eds.): ESWC 2010, Part I, LNCS 6088, pp. 152–166, 2010.

includes six different ontology elements sharing the same NLI "George Bush". But exact reference resolution in case of ambiguity can benefit from exploiting co-occurrence information. In particular, the relational network between entities in an ontology graph includes information that can enable the unique identification of entities. For instance, only one of the above mentioned "George Bush"s has *a farm in Texas*.

Current approaches typically use this co-occurrence information as importance measures for nodes [7] or use heuristics on the graph structure [11]. Also, many try to transfer NLP approaches to this domain [26,14,9], mostly focusing on a specific domain, using domain-specific measures. So far, in the field of ontology-based entity disambiguation, domain-independent complex structures in semantic graphs have not been exploited. Only certain aspects of entity-networks have been used, *e.g.* relations of a certain type. Also no ranking measures for entity networks have been proposed.

We present a novel approach for entity disambiguation, applied to RDF(S)-ontologies, which determines the most likely references for a given NLI. By exploiting the structure of graphs with spreading activation (SA), we identify a steiner graph covering only the most likely ontology elements which denote reasonable references for the given NLIs, from the set of all elements referable by an NLI (surrogates). Our approach does not need pre-learned knowledge and is also applicable to huge entity-networks. Its weighting and prominence scheme allow for a valued consideration of ontology elements and includes a new ranking model. It uses the activation value of a connector node, characterising the semantic coherence of a steiner graph which includes the result of the disambiguation process in form of the most likely surrogates.

This paper is structured as follows: After a short background section 2, Sect. 3 defines the problem of entity-reference resolution and devises our algorithm. Section 4 contains the used measures and the ranking procedure, whereas related work is surveyed in Sect. 5. Section 6 describes the evaluation of our approach in detail, based on a polysemic domain. The paper concludes with a summary and outlook in Sect. 7.

2 Background

Representation of Names in the Semantic Web. For representing knowledge in a KB, we refer to the Semantic Web standard ontology languages RDF(S) and OWL. Those specify the identification of an ontology element by its $rdf{:}ID$ tag. This must be a URI guaranteeing a nearly unique identification of an element[2]. According to the standard, $rdfs{:}label$ specifies a literal in form of a natural language name which identifies an ontology element, *i.e.* a natural language identifier (NLI). Here, without loss of generality, we disregard the language tagging facility of RDF(S) literals. A label does not guarantee for a unique identification, as multiple elements can share the same label but are not equal. It has been designed in order to enable access to an element based on human language expressions[3]. Hence, the search for ontology elements via their NLIs implies

[2] Two RDF URI references are equal if they compare as equal.

[3] NLIs can be associated via other property relations as well. For example, SKOS (`http://www.w3.org/2004/02/skos/`) defines the relations `hasPrefLabel` and `hasAlterLabel` for this purpose.

the problem of ambiguity, *e.g.* "Karlsruhe" in the geo-ontology is associated to a city, an administrative region in Germany and a spring in Africa.

Ambiguity. According to the Encyclopedia Britannica, ambiguity is a *"factual, explanatory prose,* [...]*"* and *"*[...] *considered an error in reasoning or diction"*. It leads to a non-exclusive connection between ontology elements. In an ontology, the following types of ambiguity can be found: *(a) Multi-reference ambiguity within instances of one concept* and *(b) Multi-reference ambiguity across concepts* (cf. [26]). Both include the problem of synonymy, *i.e.* several NLIs are associated to one element and can be used interchangeably. As identifiers are not unique, it is also possible that an NLI refers to an element outside of the examined ontology domain, and thus there is no corresponding element included in the ontology at all.

Co-occurrence. In order to overcome this problem, we make use of the co-occurrence of entities in input data, *e.g.* a text document, and try to reproduce this co-occurrence on an RDF(S)-graph. Focusing on text, co-occurrence means the joined appearance of entities in a document. Regarding RDF(S)-graphs, co-occurrence means the possibility to retrieve paths between the ontology elements.

Spreading Activation. Like Quillian's original spreading-activation idea [20], almost all later extensions (cf. [5,1]) transfer an initial activation to a selection of nodes in a network. Source nodes fire and transfer (spread) their activation to their adjacent nodes. This is done iteratively until no node is left which is allowed to fire. Whenever a node gets activations via multiple paths, this results in a high overall activation of this node. Often such nodes are important for the result sets of the algorithms. The activation values are often decreased in each iteration with a decay factor.

3 Entity Reference Resolution Based on Entity Identifiers

In order to retrieve **the** ontology element that stands for an ambiguous identifier found in the context of a document, we first collect all ontology elements that can be referred by this identifier (NLI) in the ontology, *e.g.* "John Doe" leads to the elements A, B and C (cf. Fig. 1). This set of elements is called a surrogate set for this identifier. In order to obtain the **reference** for an ontology element (equivalent to the meant element for this identifier in the text), we use the connections between elements of different surrogate sets, *e.g.* between a surrogate for identifier *John Doe, Karl Foo* and identifier *Fritz Hall*, means between A, D and E for example. Our **hypothesis** is, that identifiers co-occurring in a text must be also related in an ontology and thus can represented by connected elements. These connections are included in a spanning graph. A spanning graph between a subset of elements of the overall graph is called **steiner graph** We search for a steiner graph connecting at least one surrogate of each surrogate set to each other and thus each identifier to the others[4]. As this is a combinatorial problem, multiple of such steiner graphs can be found in the same ontology graph (cf. Fig. 1, here two possible graphs are shown). Thus there must be a ranking method evaluating these steiner graphs in order to obtain the graph representing the references and thus the ontology elements that stand for each identifier originally found in the text.

[4] There may occur several members of a surrogate set, if they have been equally weighted.

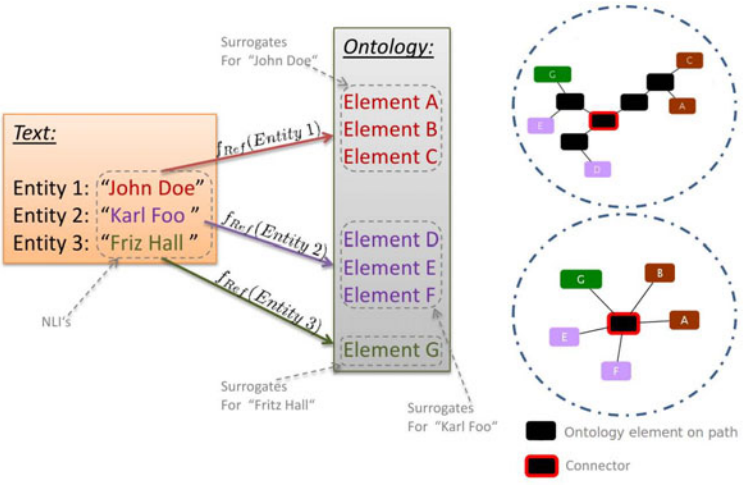

Fig. 1. Left: NLI based surrogate retrieval; Right: Examples for steiner graphs

More formally: An RDF(S)-ontology can be represented as triple list $G \subseteq V \times E \times V$, where the nodes $v \in V$ represent classes, instances and literals while the edges $e \in E$ denote to object-properties. Any ordered triple $\langle u, e, v \rangle$ states that node u is related to node v via edge e. We assume that edges are undirected, following the design principle that edges in a semantic graph imply a semantic meaning to both adjacent nodes and thus are navigable in both directions[5], *e.g.* $u \xrightarrow{lives\ in} v$ implies $u \xleftarrow{accommodate} v$.

Let I be the set of all **given** initial entity NLIs. $L_v := \{l_1^v, l_2^v, \ldots, l_p^v\}$ denotes the set of possible identifiers for a node v, whereas an entity n, *e.g.* a named entity in a text document, is identified by its textual identifier $i \in I$. The set of ontology surrogates S_i for a given entity identifier i corresponds to the nodes with an associated label equal to the entity identifier, formally given as follows

$$v \in S_i \text{ iff } \exists\, l^v \in L_v,\ l^v = i \tag{1}$$

The references of an entity n—with an associated identifier i—are initially indicated by its surrogates in the graph given by

$$f_{Ref}(n) = S_i \tag{2}$$

We **search** for (multiple) steiner graphs Z, formally defined by

$$Z := \{\langle v, e, u \rangle \in (V^Z, E^Z, V^Z) |\, V^Z \subseteq V \text{ s.t. } V^Z \cap S_i \neq \emptyset \text{ for each } S_i \in S\} \tag{3}$$

Each of them includes for each entity n in the set of entities N and their corresponding identifiers $i \in I$ at least one surrogate node $v \in S_i$, for each $S_i \in S$. S states the union

[5] OWL includes the $owl{:}inverse$ property to allow the definition of inverse relations and $owl{:}symmetricProperty$ to express symmetry.

of all surrogate sets $\bigcup_{i \in I} S_i$. Steiner points are represented by $V^Z \notin S$. The edge set E^Z of Z is a subset or equal to E.

Spreading Activation for Steiner Graph generation. In order to retrieve a graph including at least one surrogate node out of each surrogate set, we explore an ontology graph using *spreading activation*. Our approach (cf. Alg. 1) explores possible paths among entity-representing nodes and allows for a weighted selection of the next path steps. It permits the ranking of constructed steiner graphs (see above). The ranking of the steiner graphs is done by their activation values (see Sect. 4).

In the following we explain our algorithm in detail: At first—method INITIALISATION —all surrogate nodes $v \in S_i$ are retrieved for each given entity identifier i and inserted in queue Q according to their activation values. The nodes included in the surrogate sets $S_i \in S$ build the processing foundation as we aim for a construction of result graphs including at least one node out of each set. In every iteration step, the node v with the highest activation value is selected from Q and explored. Hence the most important node per iteration step is explored. The overall activation a_v for node v results from the sum of activations per identifier $a_{v,i}$ and the $nodePrestige(v)$.

After selecting the most activated node, we explore its vicinity. Nodes adjacent to v are retrieved by the method GETPATHSTEPSOFINTEREST(v). We consider only edges with a $degree(e) \leq deg_{max}$. This allows us a restriction to edges of a certain degree that implies that these edges represent rather important connections than connections to multiple nodes by the same edge type. We denote the first as semantically more expressive (see Sect. 4). We also consider the presence of adjacent nodes in Q or X (X includes the already analysed nodes). The queues contain important nodes, since all of them hold connections to at least one surrogate.

Each returned $u, e \in pathSteps$ is analysed in combination with node v in the method ANALYSECONNECTION(u, e, v). Each node v has associated sets $P_{v,i}$[6], each indicating the best path(s) to a surrogate for an identifier i. For each path only the first node p per path is included. If a node is a surrogate node itself, a reflexive auxiliary relation is used. In this method all available best path connections of node v are considered if they denote better surrogate connections for node u via e. The decision concerning a best path is based on the path distances, $dist_{v,i}$ resp. $dist_{u,i}$. In case u has no connection to a member of surrogate set S_i or its distance to a member of this surrogate set is longer than the distance via v, v is used as best parent. If the distances are equal, the node with a greater overall activation is used. If the analysis denotes equal paths, the one via v is also used since they are equally important. In case of a retrieved connection to a new surrogate set, u may denote a full-Connector. The method (IS-FULLCONNECTOR(u)) examines if a node u holds connections to all surrogate sets, whereas the parents are not allowed to be equal for all paths. In this case it would be the "root" of a sub-graph Z and thus inserted into the result set R.

For all identifier connections of v the activations to spread from v to u are calculated. If the value of an identifier i is higher than the stored activation for this identifier $a_{u,i}$, the activation value is updated. If the parent association $P_{u,i}$ or the activation value $a_{u,i}$ of a node u changes, this change is back-propagated to all already

[6] If there are multiple equally valued paths then $|P_{v,i}| > 1$.

Algorithm 1.

Initialisation $Q \leftarrow S$; $X = \emptyset$; $R = \emptyset$; $\forall u \in S : depth_u = 0$;;
$\forall i, \forall u \in S$: *if* $u \in S_i$ *then* $dist_{u,i} \leftarrow 0, P_{u,i} \leftarrow u$ *else* $dist_{u,i} \leftarrow \infty, P_{u,i} \leftarrow \emptyset$;
while Q *is non-empty* **do**
  Retrieve node v, with highest overall activation, from Q and insert in X;
  if IS-FULLCONNECTOR(v) **then** insert v in R;
  foreach $(u, e) \in$ GETPATHSTEPSOFINTEREST(v) **do**
    ANALYSECONNECTION(u, e, v);
    if $((u \notin X)\, and\, (a_u > a_{min})\, and\, (depth_u < depth_{max}))$ **then** insert it into Q with $depth_v + 1$
  end
end
Func GETPATHSTEPSOFINTEREST(v)
  $pathSteps \leftarrow \emptyset$;
  foreach $(u, e) \in incoming(v) \bigcup outgoing(v)$ **do**
    if $((degree(e) \leq deg_{max}) || (u \in (Q \cup X)))$ **then**
    $insert(e, u)$ into $pathSteps$;
  end
  return $pathSteps$;
end
Func ANALYSECONNECTION(u, e, v)
  foreach *identifier* $i \in I$ **do**
    if $dist_{v,i} + 1 \leq dist_{u,i}$ **then**
      if $dist_{v,i} + 1 < dist_{u,i}$ **then**
        $P_{u,i} \leftarrow \emptyset$;
        $dist_{u,i} = dist_{v,i} + 1$;
      end
      else
        foreach $p \in P_{u,i}$ **do**
          if $a_v > a_p$ **then** $p \leftarrow \emptyset$;
        end
        add v to $P_{u,i}$;
      end
      if IS-FULLCONNECTOR(u) **then** insert u in R;
      COSTUPDATE(u, i);
    end
    if v *spreads more activation to* u *from* t_i **then**
      update $a_{u,i}$ with this new activation;
      ACTIVATIONUPDATE(u, i);
    end
  end
end

analysed adjacent nodes of u (COSTUPDATE(u, i) for parent and distance update, ACTIVATIONUPDATE(u, i) for activation update), since their activation value or parent relation for the identifier in question may also need to be changed. Each node affected by the back-propagation is analysed according to a possible optimisation of its parent associations or activation values.

After the connection analysis the node u is considered for further exploration of its vicinity. If its activation exceeds the minimum activation threshold and its depth $depth_u$ is lower than the maximum depth, it is inserted in Q on condition that the node is not already a member of Q or X. The parameter $depth_{max}$ specifies the maximum depth allowed for a node in the graph—which is used as a performance optimiser. The output of our algorithm is a set of full-connectors in R, sorted according to their activation value. A graph can be reconstructed by an iterative exploration of the parent nodes associated to each node. The surrogates per identifier within this graph represent the final result of our algorithm.

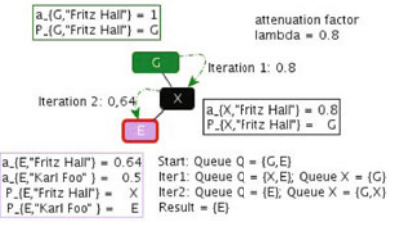

Fig. 2. Example

Fig. 2 illustrates our algorithm[7]. The nodes E, G are connected through node X. E is a surrogate node for identifier "Karl Foo" while G is a surrogate node for "Fritz Hall". The figure represents the status of the algorithm after two iterations without back-propagation and shows the retrieval of the connector node E. The example includes two iterations; the status of the queues is shown in the lower right corner.

4 Measures and Ranking

The activation of a node represents its prestige and its connectivity to keyword surrogates. The higher the activation, the more keyword surrogates of different identifiers are connected and the closer they are.

Initial Activation. The initial activation of keyword surrogates is calculated by

$$a_{u,i} = \frac{nodePrestige(u)}{|S_i|}) \times proximity(u,i) \tag{4}$$

As suggested in [3,13], we use the function $nodePrestige(u)$ in order to express the importance of u in an ontology graph. Common measures are $indegree$ or $outdegree$ of u (for an overview of $nodePrestige$-Measures, see [22]). We introduce the multiplication with $proximity(u,i)$ that denotes to the reliability of the identifier i in context of node u, *e.g.* due to non-exact recognition of the identifier in the pre-analysis-phase. A typical example for pre-analysis is the Levensthein-distance in text recognition.

Activation Spreading. The calculation of the spreading activation from a node u to v via $e_{u,v}$ is calculated per identifier i

$$\begin{aligned} a_{v,i} &= a_{u,i} \times \lambda \times degree(e_{u,v}) \\ degree(e_{u,v}) &:= \frac{|e_{u,v}|}{|e_u|} \end{aligned} \tag{5}$$

[7] Note: This is an excerpt from the example in Fig. 1. For the sake of brevity, here only two initial keywords "Karl Foo" and "Fritz Hall" are considered.

The variable λ is an attenuation factor for decreasing activation over the path length and thus per iteration. The $degree(e_{u,v})$ represents the structural coherence between two adjacent nodes. The higher the value, the tighter the connection between two nodes depending on the edge type. This is illustrated in Fig. 3.

(a) $Boston$ point of reference (Edge $e_{\text{Boston,USA}}$)

(b) USA point of reference (Edge $e_{\text{USA,Boston}}$)

Fig. 3. Example of Edge Degree

Figure 3 shows a relation of type $rdf{:}inCountry$ between two sets of instances. From the perspective of $Boston$, the degree of the edge to USA, $e_{\text{Boston,USA}}$ is 1, since there is one edge in total (Fig. 3a). From the perspective of USA, there are n edges in total and all of the same type, but only one of them is associated to $Boston$. Hence the edge degree for $e_{\text{USA,Boston}}$ is $\frac{1}{n}$ (Fig. 3b). The point of view is important, since the degree represents the semantic expressivity during exploration. In consequence, it characterises the importance for $Boston$ to be of type USA, and for USA having an associated city $Boston$ based on the degree of relations that are of the same type.

Overall Activation. The overall activation of a node v is calculated as the sum of the identifier-dependent activations and the prestige of v.

$$a_v = \sum_{i \in I} a_{v,i} + nodePrestige(v) \tag{6}$$

The overall activation is used to estimate the processing order in queue Q. The node with the highest activation value is considered as node for exploration. A $nodePrestige$ is not included in the spreading phase (only if surrogate), but considered to be of importance to the ranking measure.

Ranking. Based on the activation functions above, the calculation of importance is done during the execution of the algorithm. Apart from the ranking position in queue Q the activation value is also considered as ranking value for the generated result graphs. As stated in Sect. 3, a connecting node is the root r of a generated result graph Z and thus holds connections to at least one surrogate of each identifier. The quality of the connections is expressed by the identifier activation $a_{r,i}$. The prestige of a node as well as the proximity of each identifier is included in each identifier-dependent activation value. We consider the overall activation value of a_r, also including $nodePrestige$ of $r \in R$, as a valid measure for the quality of a result graph. In our algorithm the constructed result graphs are ordered according to their overall activation value. In consequence the top k results in R, defined by the parameter Top_k, are consistent with the best k results for the entity resolution process.

Steiner Graphs and Semantic Coherence. A steiner graph Z denotes an excerpt of the ontology graph including at least one surrogate node for each given NLI (cf. Eq. 3). The significance of Z is given through the quality of its semantic coherence. We adapt the notion of *semantic coherence* (SC) from [7] to the problem of entity resolution and to the used algorithm, by defining it as the *cohesiveness* and *expressivity* of a steiner graph Z including surrogates for given NLIs. **Cohesiveness:** The information existing between every two entities can be accessed by the exploration of their mutual relations in an ontology graph. Each Z includes a subset of this information that can be qualified first by the amount of included entities. Further, the shape of a graph Z, defined by the quality of the included relations between the entities (from non-existent till very tight), constitutes an important measure for the cohesiveness. **Expressivity:** The individual quality of an element in a graph Z is specified by an element-specific quality measure, *i.e.* the *overall activation*. This value is derived by the quality and amount of connected keyword surrogates as well as their individual *initial activations*. Further, the overall activation is computed by the *activation spreading* influenced by the shape of the paths between the nodes. The quality of each individual node is recursively calculated by the quality of its surrounding nodes. This peaks out in the quality of the connector node, the first node with associated paths to all entities and the representative of a graph Z. The overall activation of the connector node constitutes the significance of a steiner graph Z and its *ranking* value.

5 Related Work

Related research fields are word-sense disambiguation on graphs, ontology-element spotting and disambiguation, as well as keyword and entity search on graphs.

In the context of **graph based algorithms**, Veronis and Ide [25] used SA to disambiguate the senses of given words in a thesaurus. A gloss text associated to each word is used for disambiguation. During the spreading phase each word entailed in the gloss activates its associated sense while the sense itself has an associated gloss again. Recently, Tsatsaroni *et al.* [24] applied this approach to WordNet [6], including a modification that allows to use direct associations between senses. Rada *et al.* [21,22] build graphs from textual context information and use them for disambiguation based on co-occurring context words. Bhattacharya *et al.* [4] also represent textual co-references using hypergraphs. They merge different graphs based on their similarities in order to identify the exact entities. Mailaise *et al.* [15] used an SA-based approach for disambiguation between annotations used for TV programme description, including human feedback to increase the precision. This class of approaches builds upon the linguistic domain. They focus either on lexicons or on measures given through natural-language analysis. The use of graphs does not include the specifics of ontologies. Mostly background knowledge is not explored independently from the domain.

In the context of **ontology-element disambiguation** Banek *et al.* [2] recently associated a WordNet Synset to each ontology element. The disambiguation utilises the synonyms for identifying similar elements. A similar approach by Nguyen *et al.* [17] *et al.* uses an associated bag-of-words vector to each entity including certain nouns, co-occurring entities and further Wikipedia knowledge. Nguyen *et al.* also used the

KIM-ontology for disambiguation [16]. They used the textual distance between two entities in a learning corpus and preferred entities of the same concept affiliation. Garcia *et al.* [7] used a modified PageRank-Algorithm [18]; here, the textual co-occurrence of entities is used to calculate the relevance of a possible surrogate without including other ontological knowledge. Garcia *et al.* presented a modification of the algorithm including Wikipedia information [8]. Hassel *et al.* [11] presented an approach based on the DBLP-ontology which disambiguates authors occurring in mails published in the DBLP-mailing list. They used ontology relations of length one or two, in particular the co-authorship and the areas of interest. The approach by Volz *et al.* [26] used contextual information for detecting the concept affiliation of entities. Kleb *et al.* [14] used concept-dependant text patterns for the disambiguation of text information. Gruhl *et al.* [9] trained an SVM classifier in order to spot ontology entities. Here, many common ideas from information retrieval (IR) have been transferred to this domain. However, we neither require training data to learn classifiers for recognition, nor focus on the analysis of textual knowledge. Our approach could use this information later, in order to improve its weighting scheme (cf. 7). But currently, we focus on exploring complex structural background knowledge. Apart from disambiguation, Hasan [10] used spreading activation with user feedback for enabling user-driven search for information items, *i.e.* the user is allowed in an interactive approach to increase/decrease the activation of a node in order to influence the retrieval of connected graphs between entity references.

Keyword search on graphs also retrieves connected graphs including the searched keywords, but does not focus primarily on disambiguation. Bhalotia *et al.* [3] presented the BANKS system which performs an iterated search starting from all surrogates, with a best-first expansion based on the exploration of backward-edges. This approach has been optimised by Kacholia *et al.* [13] regarding a spreading-activation based forward-backward exploration search using two iterators. We took up the idea of spreading activation; however, we use other measures for calculating the activation and we discard the distinction between edge directions. We also consider multiple best surrogate connections and disregard nodes with the same parent for all identifiers. Further, Kacholia's algorithm terminates with the first result found, while our algorithm finds all possible results. The recent approach BLINKS [12], based on [13], selects clusters according to the smallest cardinality. Thanh *et al.* [23] presented a search algorithm for RDF-Graphs based on one iterator per surrogate set. Recently, Kasneci *et al.* [12] proposed an approximation algorithm for Steiner trees that allows for a fast retrieval of relationship graphs based on minimal weights. In contrast to keyword search, our input information often covers more than 2-3 keywords (average for keyword search) by regarding all entity identifiers in a text. Also synonymy of input strings is not given in this field. Keyword search focuses for the retrieval of short paths following the assumption of a tight relation between the input words. This is a-priori not given for entity disambiguation.

We focus on the relational structure and specific properties of an ontology and propose a generic algorithm for disambiguation using the semantic relations between entities. We allow for an adaption of our weighting scheme by the use of preanalysis tasks.

6 Evaluation

6.1 Ontology and Input Data

Nearly all ontologies contain NLIs for ontology elements. This often implies ambiguity—but for the evaluation of our algorithm we need a *highly* polysemic domain. Regarding existing benchmarks, entity- disambiguation approaches either do not use ontologies (cf. Trec-conferences[8] or [27,4]) or do not provide their data-set publicly [7,8,16,17]. As classical entity disambiguation approaches do not include an ontology they can not be used for comparison. For the data-set [11], the NLP process for retrieving surrogates is unclear. Thus we had to construct our own evaluation scenario. In order to reflect the ambiguity of ontology elements we used a highly ambiguous **geography ontology** which is a refinement from [26,14]. It currently contains 132,087,082 RDF(S) triples, including 18 classes, 50 relations, and instance data collected from Geonames.org (information from NGA, GNIS and 36 additional sources[9]). Each instance and concept of the ontology has an associated `hasName` and `hasAlterName` relation, equivalent to the SKOS relations `hasPrefLabel` and `hasAlterLabel`. An excerpt of the ontology is shown in Fig. 4. The property `hasGeographicFeature` has eight associated subproperties defining the specific relations between the subclasses of `GeographicFeature`.

The ontology includes 6,085,125 different ontology-element identifiers. 5,109,884 of them are associated to exactly one element and thus unique. 975,241 are associated to multiple elements (in average to 4.44 elements). This results in an overall average ambiguity of 1.55 elements per identifier. The most ambiguous identifier is "First Baptist Church" with 2,085 associated elements.

As **input data** we use news articles (cf. [26,14]) crawled from the European Media Monitor (EMM [19]) which collects news of European newspapers and clusters them topic-wise. We chose the topic *natural disasters* in order to guarantee for documents including geographic entities. The included entities have been manually annotated with their exact ontology surrogate. For our evaluation, we used 46 documents. Within the corpus there are 353 mentions of identifiers pointing to 237 entities. 74 of them are unique; the most ambiguous identifier with 1,739 surrogates is "San Antonio". In average a document includes 7.67 entities. Of those 2 refer to exactly one surrogate node ($|S_i| = 1$) leaving 37.06 surrogates for each of the remainder entities. The maximum of included entities in a document was 35, and the highest overall ambiguity, the accumulated sum of entity ambiguities in a document, was 1,914.

6.2 Evaluation Process

The process describes an ontology based reference resolution for given entity identifiers. At first all entity identifiers occurring in a document are collected and used as input data. Based on the identifiers we calculate possible steiner graphs including ontology surrogates. As stated in Sect. 4 we use as result the selection of surrogates entailed in the graph with the highest activated connector node. The quality of our algorithm is

[8] `http://trec.nist.gov/`

[9] `http://www.geonames.org/about.html`

evaluated by comparing the entity-associated annotation in the text (including the best ontology reference based on human choice) and the reference retrieved by our algorithm. We construct steiner graphs including the searched entity-representing surrogates based on the annotations in the reference corpus. The top_1 ranked graph includes the surrogates used as result references for evaluation. We calculate the standard IR measures. The *recall* ($R_Z := \frac{|relevant\ entities \cap retrieved\ entities|}{|relevant\ entities|}$) is the ratio between the amount of correctly identified entities in the result list and all relevant entities that should be included in the result list. *Precision* ($P_Z := \frac{|relevant\ entities \cap retrieved\ entities|}{|retrieved\ entities|}$) is the ratio of the number of correctly identified entities in the result list and all retrieved entities. The *F-measure* ($f_{\text{measure}} := \frac{2 * R_Z * P_Z}{R_Z + P_Z}$) is the harmonic mean between recall and precision.

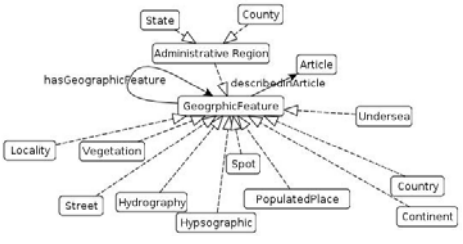

Fig. 4. Geoname Ontology Excerpt

(a) Connector: Israel **(b)** Connector: Gaza

Fig. 5. Example of Steiner Graphs

We used the following algorithm thresholds: The maximum depth ($depth_{max}$) was set to 4 as this resulted in an acceptable processing time. Every possible activation value was considered using 0 as minimum activation (a_{min}). From another set of experiments we could conclude that a minimum value for activation is advantageous in combination with a high maximum depth. We restricted the edge degree (deg_{max}) to 1 and thus only considered the strongest relations in the graph. Only the highest ranked result graph, Top_1, has been used for evaluation. An example is given in Fig. 5. Here, two result graphs for the NLIs *Gaza*, *Sederot*, *Zaitoun* and *Israel* are given. Graph 5a is higher ranked than graph 5b indicated by a higher activated connecting node (see Sect. 4). In consequence only graph 5a would be considered for the result analysis.

6.3 Evaluation Results

Table 1 shows the evaluation results. In addition to the two included identifiers, each referring to exactly one ontology element, our algorithm retrieves 1.9268 additional exact

surrogate association in average. Thus, in nearly four out of 7.67 identifiers, the algorithm returns exactly one surrogate as reference for each of them. In each case these surrogates were the correct references by comparison to the annotations. The algorithm achieved a recall of 97.73%[10]. Two different causes for the loss of recall can be identified here. The first kind of documents includes entities which references are widespread in the ontology graph. Visually these geographic places are distributed over several continents. This turns out as problematic by the use of the current metric. A perfect result graph in this case would include long path distances between the surrogates. However, there is often one element in a surrogate set that could be accessed via a shorter distance. This is due to the fact that spreading is influenced by an attenuation factor and thus decreased over path length. In the latter case not all entities are distributed. Here often one local cluster of entities is accompanied by a far distant entity. Here we also have been able to retrieve a surrogate accessible via a shorter distance. Result sets R,

Table 1. Evaluation Results

Method	Recall	Precision	F-measure
SA algorithm	97.731	48.342	64.687
Random ($\|R_i\| = 1 \| \forall R_i$)	38.596	38.596	38.596
Random ($\|R_i\| = \eta \| \forall R_i$)	64.286	5.028	9.322

containing η surrogates randomly selected from the set of possible surrogates S_i for each identifier, have been used for comparison with the result of our algorithm. Tab. 1, row 3, shows the results based on $S_i = 1$ whereas Tab. 1, row 4, contains the result for a random subset size η. The average of 10 evaluations is shown for both. Our approach performed much better than these baselines. 97.7% recall indicates that almost all entity references have been retrieved. 48% precision states that for almost every second entity the correct reference has been retrieved as first result. This was done in a domain with 37.40 possible references per identifier. The information regarded for disambiguation was restricted to the entity names. Except for edge degree, there was no use of further information or heuristics, like conceptual closeness. During the evaluation, the algorithm discovered at least 10 annotation mistakes in the test corpus, resulting from the geographic domain and the fact that it is not easy for a human annotator to manually annotate places, he has never been to, with the correct identifier.

The evaluation confirms our hypothesis to resolve the correct entity-representing ontology elements by the use of co-occurrence information, *i.e.* a steiner graph. Here 97% of all searched ontology elements have been retrieved. Based on this foundation a further improvement of precision is possible, as the selection within the retrieved surrogates must be improved and not so much the retrieval itself.

7 Conclusion and Outlook

We presented a novel and general applicable reference resolution algorithm under the aspect of ambiguity that allows the determination of corresponding surrogates of

[10] The represented precision and recall values are based on *all* identifiers found in a document.

entity identifiers based on their co-occurrence in RDF(S)-graphs. Starting from natural language identifiers, we exploit the graph representation of an ontology in order to construct spanning graphs between ontology-element surrogates. The graph itself is generated by exploration, starting from the most activated node until a connector between the elements of the different surrogate sets is retrieved. We search for all possible connectors and rank them according to their activation values. The resulting set includes the URIs of the most probable ontology elements per identifier.

We examined the quality of our approach by retrieving referencing ontology elements based on NLIs in a highly polysemic domain. We achieved 97.73% recall and 48.34% precision. As a next step, we will extend our evaluation by the annotation of further data. We will include other domains in order to verify our first promising results on other data-sets and in other domains.

Our current approach does not regard concept identifiers. The consideration of schema information would allow for the retrieval of surrogates with a certain nearness according to their instance-to-concept relation. Here the concept identifiers in the input data as well as the conceptual relations in the graph could be considered, *e.g.* surrogates with the same concept affiliation are considered to be more likely. Also relational identifiers can be used. The design of our algorithm allows this already by now.

We did not to focus on linguistic preprocessing here, as our focus was on structural processing and measures. However, parameters like the distance between identifiers in text that could be an indicator for path length between surrogates, can be included. Also phrases like "lies in" can be used to indicate certain concept affiliation or patterns between identifiers that could be mapped to an ontology graph.

Currently our approach is restricted to the identification of full-connectors, *i.e.* connectors related to one surrogate per given identifier at least. Using partial connectors in order to retrieve local clusters and thus tighter connected graphs is a next step. We will also modify the spreading between nodes and the consideration of best parents in order to allow for a more fine-grained distinction between best-parents. This will decrease the amount of possible surrogates per set and thus improve the precision of our algorithm.

References

1. Anderson, J.R.: A spreading activation theory of memory. Journal of Verbal Learning and Verbal Behavior (1983)
2. Banek, M., Vrdoljak, B., Tjoa, A.M.: Word sense disambiguation as the primary step of ontology integration. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 65–72. Springer, Heidelberg (2008)
3. Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., Sudarshan, S.: Keyword searching and browsing in databases using banks. In: Proc. ICDE. IEEE Computer Society, Los Alamitos (2002)
4. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. IEEE Data Eng. Bull. 29(2) (2006)
5. Collins, A.M., Loftus, E.F.: A spreading-activation theory of semantic processing. Psychological Review 82(6) (1975)
6. Fellbaum, C. (ed.): WordNet An Electronic Lexical Database. The MIT Press, Cambridge (1998)

7. García, N.F., del Toro, J.M.B., Sánchez, L., Bernardi, A.: Identityrank: Named entity disambiguation in the context of the news project. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 640–654. Springer, Heidelberg (2007)
8. García, N.F., del Toro, J.M.B., Sánchez, L., Centeno, V.L.: Semantic annotation of web resources using identityrank and wikipedia. In: Proc. AWIC (2007)
9. Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., Sheth, A.P.: Context and domain knowledge enhanced entity spotting in informal text. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 260–276. Springer, Heidelberg (2009)
10. Hasan, M.M.: A spreading activation framework for ontology-enhanced adaptive information access within organisations. In: van Elst, L., Dignum, V., Abecker, A. (eds.) AMKM 2003. LNCS (LNAI), vol. 2926, pp. 288–296. Springer, Heidelberg (2004)
11. Hassell, J., Aleman-Meza, B., Arpinar, I.B.: Ontology-driven automatic entity disambiguation in unstructured text. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 44–57. Springer, Heidelberg (2006)
12. He, H., Wang, H., Yang, J., Yu, P.S.: Blinks: ranked keyword searches on graphs. In: Proc. SIGMOD. ACM Press, New York (2007)
13. Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Desai, R., Karambelkar, H.: Bidirectional expansion for keyword search on graph databases. In: Proc. VLDB (2005)
14. Kleb, J., Volz, R.: Ontology based entity disambiguation with natural language patterns. In: Proc. ICDIM (2009)
15. Malais, V., Gazendam, L., Brugman, H.: Disambiguating automatic semantic annotation based on a thesaurus structure. In: Proc. TALN (2007)
16. Nguyen, H.T., Cao, T.H.: A knowledge-based approach to named entity disambiguation in news articles. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 619–624. Springer, Heidelberg (2007)
17. Nguyen, H.T., Cao, T.H.: Named entity disambiguation on an ontology enriched by wikipedia. In: Proc RIVF 2008 (2008)
18. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
19. Pouliquen, B., Steinberger, R., Ignat, C., Käsper, E., Temnikova, I.: Multilingual and cross-lingual news topic tracking. In: Proc. COLING. ACL (2004)
20. Quillian, M.R.: A revised design for an understanding machine. Machine Translation (1975)
21. Rada, M.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: Proc. HLT. ACL (2005)
22. Sinha, R., Rada, M.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: Proc. ICSC. IEEE Computer Society, Los Alamitos (2007)
23. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query candidates for efficient keyword search on graph-shaped RDF data. In: Proc. ICDE. IEEE, Los Alamitos (2009)
24. Tsatsaronis, G., Vazirgiannis, M., Androutsopoulos, I.: Word sense disambiguation with spreading activation networks generated from thesauri. In: Proc. IJCAI (2007)
25. Veronis, J., Ide, N.M.: Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In: Proc. COLING. ACL (1990)
26. Volz, R., Kleb, J., Mueller, W.: Towards ontology-based disambiguation of geographical identifiers. In: Proc. WWW Workshop I^3 (2007)
27. Wick, M.L., Culotta, A., Rohanimanesh, K., McCallum, A.: An entity based model for coreference resolution. In: Proc. SIAM (2009)