

Disambiguating Entity References Within an Ontological Model

Joachim Kleb
FZI - Research Institute for Information
Technology
Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
kleb@fzi.de

Andreas Abecker
FZI - Research Institute for Information
Technology
Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
abecker@fzi.de

ABSTRACT

In our everyday conversations, entities (persons, companies, etc.) are referred to by natural language identifiers (NLIs). Humans employ personal experience and situational context to interpret such identifiers. However, due to ambiguity, even humans run the risk of misinterpretations. In our prior work, we presented a *novel method to resolve entity references in texts under the aspect of ambiguity*. We explore ontological background knowledge represented in an RDF(S) graph. The different interpretation possibilities lead to different subgraphs of the underlying ontology, each subgraph describing one consistent, non-ambiguous interpretation of the ambiguous NLIs within the ontological knowledge base. Our domain-independent approach is *based on spreading activation and uses a semantic relational ranking*. In this paper, we suggest three extensions to our original algorithm. First, we process in a two-step interpretation—instead of the whole original input text—at first hand smaller text windows in order to get more precise reference interpretations through a *smaller local text context*. Second, we extend the spreading-activation algorithm within the RDF(S) graph towards a *bidirectional exploration of edges* which shall speed-up the algorithm. Third we use *reinforcement learning* in order to take advantage of re-occurring information. We present first experimental results with these algorithmic extensions and derive directions for future work.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Information Search and Retrieval—*Selection process*; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Graph and tree search strategies*

General Terms

Algorithms, Experimentation

Keywords

Named Entity, Spreading Activation, Graph

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIMS '11, May 25-27, 2011 Sogndal, Norway
Copyright 2011 ACM 978-1-4503-0148-0/11/05 ...\$10.00.

1. INTRODUCTION

In order to still the information need of humans a multiplicity of automatic computer programs gather huge amounts of data from the web. Most of this Web Mining tools are restricted to the unstructured part of knowledge available in the web, *i.e.* text. However in order to take a maximum advantage from the web sources it would be also beneficial to combine unstructured information and structured knowledge as provided by the wealth of Linked (Open) Data. This includes the crucial requirement to map natural-language expressions in free texts to their respective correspondents in formal ontologies (and in the optimum case, this mapping should even benefit from the already available formal knowledge in the Web). All practical experience says that this problem will – in the general case – not in the near or mid-term future (and maybe never) be completely solved by approaches like extensive lexical layers associated to ontologies or simple information extraction or text classification methods. It can neither be solved by more rigorous URI re-use or ideas like this (as it is fostered in the LOD initiative), because the problem actually arises if we start from natural-language texts, not from formal representations. And natural language is always and inevitably characterised by a significant amount of ambiguity – the phenomenon of words that are equally written but bear different meanings (for example, “bow” could identify the stick for a music instrument as well as the front of a ship). In fact, the problem is even worse because it is not only and exclusively caused by the use of natural language, but often the real world itself offers ambiguous or contradictory information. For instance, one can easily find out, that alone in the US, there are at the moment at least 921 persons living who bear the name “Robert Gates”, the same name as the current secretary of defence in the US. So, in an interpretation process purely based on lexical information, at least those 921 people would, at first hand, be literally *correct* individuals who could be denoted with that name. But, of course, in a concrete text, normally exactly one of those persons is really meant. Humans are often not even aware that a word is ambiguous. They tend to listen subconsciously to the context of the conversation and use this context information in order to identify the correct meaning of an ambiguous word. In this paper we present a similar approach that first establishes lexical links between texts and formally represented background knowledge, and then explores different possible knowledge contexts in this background knowledge in order to identify the most probable consistent formal interpretation for the informal text elements. In our current version of our approach (method and algorithms), we focus on texts containing natural language identifiers for *named entities* which shall unambiguously be identified in the background ontology, as named entities are often the most important concrete crystallisation point for assessing the

This paper is structured as follows: After the definition of named entities and the ambiguity related to a conceptual model in Sect. 2, Sect. 3 defines the problem of entity-reference resolution and illustrates the idea of our spreading-activation algorithm using an example. Sect. 4 explains the three algorithmic extensions mentioned above. Sect. 5 contains the used measures and the ranking procedure. Related work is surveyed in Sect. 6. Sect. 7 describes the evaluation of our approach in detail, within a highly polysemous example domain. The paper concludes with a summary and outlook in Sect. 8.

In order to allow for a refined problem specification this section introduces the term “named entity” (2.1) and describes the correlation between ambiguity and the ontological model (2.2).

Jurafski *et al.* [19] described a *named* entity by stating: “By *named entity*, we simply mean anything that can be referred to with a *proper name*”. This definition includes the simple distinction between general entities and named entities by the assertion of a proper name. However, this name allows for an entity identification. Regrettably, this implies not the *unique* identification of an entity by this name as names can be shared.

2.2 Ambiguity in Ontologies

Ambiguity can be subdivided in the categories of *polysemy* as well as *homonymy*. The literature agrees with Klavans that polysemy occurs when a single word has two or more meanings, but in contrast to homonymy, the different meanings are somehow related to each other, *e.g.* the term “wood” refers either to a piece of a tree or to a geographical area with many trees. In both cases “tree” is the related item for both meanings. By contrast, “bow” can denote a wooden tree necessary for certain music instruments or the front of a ship. Both are not related and therefore dedicated as homonyms. As polysemy and homonymy are the main causes for ambiguity, Klavans’ definition must be extended by homonymy, because both describe the phenomenon of words that can refer to different meanings.

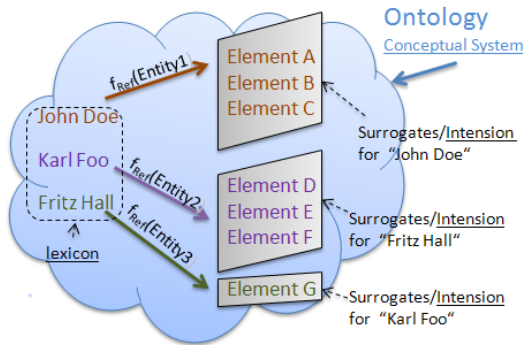
polysemy, respectively. The former describes the kind of polysemy that includes a relational binding between all different meanings of a polysemous word; see the above example of “wood”. Here “wood” as tree has a relation to “wood” as forest and vice versa and thus all meanings depend on each other. In the latter, one cannot derive rules from the relationship between the interpretations of words, e.g. “glass” as optical aid or “glass” as container. Here a relation from “glass” as material to “glass” as optical aid as well as to “class” as container is given. However, no relation is given between “glass” as optical aid and “glass” as container. In almost all cases it is unclear how to differentiate between non-systematic polysemy and homonymy¹. In case of non-systematic polysemy, another entity within the contextual surrounding may allow for a disambiguation. This phenomenon is called *conceptual shift*. In case of systematic polysemy we expect a problematic situation for our algorithm. In such a situation very tight relational bindings between all possible referents of the same ambiguous term are common. This results in close distances between entities representing the same name which may lead to false disambiguation results.

[illegible]

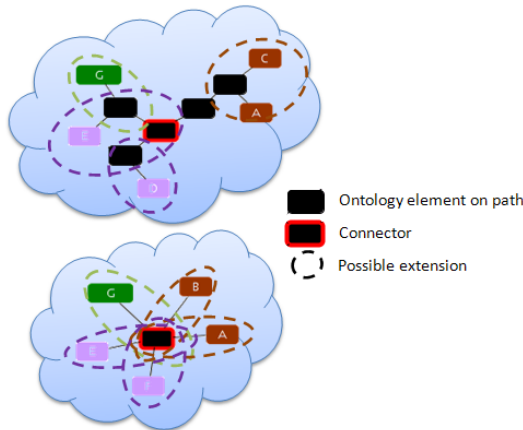
Semantic Web Model. The general model can be transferred to the model of Semantic Web knowledge bases. We define the conceptual system by ontologies known in the Semantic Web, *i.e.* ontologies represented in the languages OWL and RDF(S). According to the language standard, *rdfs:label* specifies a literal in form of a natural language name which identifies an ontology element. The labels can be interpreted as lexicon entries. Hence for ontologies the lexicon is part of the conceptual system. A label does not guarantee for an unique identification, as multiple elements may share the same label though not being identical. It has been designed

¹Many authors claim that both is actually the same, *e.g.* [9, 21]

in order to enable access to an element based on human language expressions². Thus the labels represent the lexicon entry for words in our model, and the elements identified by a label are the addresses given by the address generator. This denotes that each URI-identified ontology element can be referenced by a natural language name which is also part of a lexicon. Thus, an ontology represents a conceptual system that includes a lexicon as mentioned above. The extension of an ontology element is given by its semantic relations. Thus all semantic relations defined in RDF(S), respectively OWL, can be applied (Fig. 2).



(a) Lexicon and Intensions



(b) Extensions and possible Steiner-graphs

Figure 2: Example model for entity identifier polysemy

The differentiation between systematic and non-systematic polysemy can be transferred to our model using Semantic Web ontologies. The former refers to the existence of at least one relation between the different elements in the ontology retrieved by one and the same word. On the one hand, this applies if a word refers to different instances of the same concept. Here the instances share the same terminological knowledge, but can differ according to their assertions. On the other hand, this can apply for two concepts that share the same super-concept. Depending on the interpretation of the term “relation” also more complex semantic relations between those elements could be considered as a systematic coherence. However, non-systematic polysemy can only be resolved by the use of contextual information. Here contextual words and their

²NLIs 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.

extensions are important parameters for the retrieval of the referent. Resolution can be defined as the process of referent retrieval, *i.e.* the element actually referred to by the signifier out of the signified elements. Each word can be associated to several entities via the address generator³. Based on the ontological description each entity has associated semantic properties.

From a conceptual point of view the retrieval of a referent is only possible by the evaluation of the importance of one extension against another (based on the different addresses generated for a given word). We focus on the relation between different extensions of different words, *i.e.* we make use of the co-occurrence of words (as a specific definition of context), *e.g.* in a text document, and try to reproduce this co-occurrence in the conceptual system. In particular, we try to find the semantic extensions of word *a* that is related with a semantic extension of word *b*. Thereby the term “relation” is not further specified, *e.g.* this could be an intersection between an extension of *a* and an extension of *b*, as well as a path between the extensions.

Our approach is restricted to the problem of named entity disambiguation. We use their identifiers as words and refer via their according lexicon entries to their representations in the conceptual system. Here, the conceptual system is given by an RDF(S)-ontology and we utilise the connecting paths between the different addresses of co-occurring entities.

3. DISAMBIGUATION OF ENTITY REFERENCES WITH SPREADING ACTIVATION

The last section highlighted the complexity of problems that come along with the existence of ambiguity. The model of polysemy in Fig. 1 emphasizes the importance of a conceptual system for the disambiguation and thus for a clear indication in which situational context a specific variant of an ambiguous term has to be chosen. The situational context of an entity in respect to the ontology model is given by its embedding in the relational network of the ontology. In the rest of this section we recall the algorithm and the motivation behind our approach. Please refer to [23] for a detailed description of the basis algorithm.

A typical information source with ambiguous identifiers are documents like newspaper articles. Our approach enables the disambiguation of the contained entities within such a document by using a background ontology which includes information about the domain described in the article. Such ontological information can be retrieved from sources like DBpedia or other elements of the Linked Open Data Cloud⁴. A typical example are names of entities which are of type person or location. Assume there are “Andreas” and “Karlsruhe” mentioned in one article. Considering an ontology like DBpedia these identifiers are most probably not unique. Thereby two different characteristics concerning ambiguity can be identified. 1. a label can be associated to multiple instances of a concept and 2. a label can be associated to multiple instances across concepts. In the above example, the term “Karlsruhe” identifies a fountain in South Africa as well as four other places, in addition to the town in Germany⁵. Humans refer to their experiences and try to establish links between entities with these names. This is often done subconsciously. However, this relational analysis is important in order to select the correct meaning. Using ontologies, this

³The address generator retrieves all entities within the ontology whose label is equivalent to the given input word (cf. f_{Ref} -Function in Fig. 2a).

⁴For an overview exemplary possible sources refer to <http://esw.w3.org/DataSetRDFDumps>.

⁵See <http://www.geonames.org> for further details.

analysis can be performed by the use of the ontological network. Here the semantic relation between entities are represented by the path within the ontology which connects them. Thus the ontology graph characterises the relational descriptions between the entities in question. However in order to identify the entities originally meant in the text, the possible paths between those entities must be evaluated and ranked and thus the ontology subgraph representing the relational dependencies between the entities.

In the following we present the necessary foundations in order to provide an understanding for the algorithm. The algorithm itself is described in detail in [23]. In the remainder of this section we introduce the data model of an ontology and the Steiner tree model. The subsequent example includes the main steps of the algorithm. In order to get a detailed understanding of the extensions in relation to the base-algorithm, please refer to [23].

3.1 Data Model

The standard representation languages describing ontologies are RDF and OWL. RDF Ontologies can be represented as directed labelled graph, where edges represent a named link between two resources, represented as graph nodes (cf. [17]). OWL-ontologies can be materialized into RDF- ontologies if the materialisation is restricted, e.g., to atomic classes. We split the ontology graph in a *model graph* and a *data graph*. The nodes within the model graph represent the concepts (classes in RDF) of an ontology. Edges are used for the relationship between classes (object-properties) as well as data dependent relations (data-properties). In contrast to the data model, the edges in the object model include *rdfs:typeOf*, *rdfs:subclass*, *rdfs:range*, *rdfs:domain* etc. relations that are exclusively part of a conceptual description of an ontology. Concerning domain specific relations, like *x example:knows b*, there exists an overlap between object and data model. Ontology elements need to have an id - URI - which uniquely describes them. Also they can be identified by non-unique labels (e.g. *rdfs:label*).

DEFINITION 1. *The data graph is given by $O_d = \langle V_d, E_d, L_E, L_V, l \rangle$. V is a finite set of vertices while E is a finite set of edges. $L_V = \{l_1, \dots, l_o\}$ and $L_E = \{e_1, \dots, e_p\}$ describe the set of node labels respectively edge labels. The labelling function for vertices is given by $l : V \rightarrow 2^{L_V}$. $l(v) = \{l_1, \dots, l_m\}$ describes the set of labels asserted to node v . $l : E \rightarrow 2^{L_E}$ describes the set of labels of an edge analogously. A set of labelled edges is described by $E \subseteq V \times L_E \times V$ where *source* : $E \rightarrow V$ describes the subject of an triple while *target* : $E \rightarrow V$ describes the object node. The model graph is given by $O_m = \langle V_m, E_m, L_E, L_V, l \rangle$ and thus defined analogously to the data graph above. $V = V_d \cup V_m$ describes the set of all nodes within the ontology. The set of all edges is given by $E = E_d \cup E_m$. $E_d \cap E_m \neq \emptyset$ while $V_d \cap V_m = \emptyset$.*

In our approach we focus on the data model, nonetheless is our method applicable for the object graph as well as for the union of object and data graph. Thus we refer to V_d and E_d in the remaining paper. As illustrated in Fig. 2, each node v has an associated set of labels $l(v)$. These labels include natural language identifiers. In consequence the set of nodes S_i includes all the nodes that are associated with the identifier i , i.e. $S_i = \{v | \exists l \in l(v) \text{ s.t. } l = i\}$, are used as possible references for this identifier. Fig. 2a shows an excerpt of an ontology lexicon. The lexicon itself includes all node labels, L_v . For example the ambiguous identifier “John Doe” identifies the set of nodes $S_{JohnDoe} = \{v_A, v_B, v_C\}$.

3.2 Steiner Trees

Prior to the disambiguation process the above mentioned identification of the set of possible entities is necessary. The disambiguation process itself tries to find a subgraph of the ontology which includes at least one entity for each given entity identifier, e.g. the entity labels entailed in a text (see Fig.2b). Such a subgraph is analysed from the perspective of a central node (cf. Sect. 3.3). This analysis results in a tree describing the entities mentioned in a document by following definition:

DEFINITION 2. *The tree Steiner Tree Z is defined as a subtree of O_d that is reduced in respect to $S = \{S_1, \dots, S_m\}$ the set of all Surrogate Sets for the amount of m given entity identifiers. In the subtree Z there exists a sequence of m nodes, $\langle v_1, \dots, v_m \rangle$ where $v_i \in V(Z)$ and such a node is a surrogate node $v_i \in S_j$ for $1 \leq i \leq m, 1 \leq j \leq m$, such that the leaves of Z can only come from those nodes, i.e. $leaves(Z) \subseteq \{v_1, \dots, v_m\}$. In case the root of Z has only one child this root must be within the set of those nodes⁶, i.e. $root(Z) \in \{v_i, \dots, v_m\}$.*

Here the resulting Steiner Tree includes at least one surrogate from each surrogate set, i.e. for each given identifier there is at least one element representing the reference ontology element for this identifier.

Consider the example given in Fig. 2. The m initial identifiers are given by “John Doe”, “Karl Foo” and “Fritz Hall”. The corresponding Surrogate sets are: $S_{John Doe} = \{v_A, v_B, v_C\}$, $S_{Karl Foo} = \{v_D, v_E, v_F\}$ and $S_{Fritz Hall} = \{v_G\}$. The lower subtree $\mathcal{H} = \{v_G, v_E, v_F, v_A, v_B, v_{blank}\}$ in Fig. 2b forms a Steiner tree according to the above definition, where v_{blank} is the root of the tree. The root has multiple children, so it has not to be a node within a surrogate set $S_i \in S = \{S_{John Doe}, S_{Karl Foo}, S_{Fritz Hall}\}$. All leaf nodes $leaf(\mathcal{H}) = \{v_G, v_E, v_F, v_A, v_B\}$ are members of $S_i \in S$.

If there is only one surrogate node for each given identifier and this node describes the correct reference, the disambiguation of this element had been performed in an optimal way. Steiner Trees follow the mathematical description of minimum spanning trees. However, until now, it is not clear what “minimum” shall mean in the context of disambiguation. This question is answered below.

3.3 Spreading Activation for Subgraph generation explained with the help of an example

As described in [23] we use spreading activation as underlying idea of our approach. Spreading activation was originally defined by Quillian [31]. Here as well as in the later approaches and extensions (cf. [8], [1]), an initial set of nodes is activated with an initial activation value. In every iteration the nodes bearing activation fire, i.e. they transfer their activation to their neighbours, based on the relations in the graph. Whenever a node receives activation from multiple other nodes, this activation is aggregated. A high activation value indicates an important node for the algorithm. Via spreading activation we explore possible paths between the given nodes. The activation allows us to weight each path between any two nodes.

In order to provide a better understanding of the extensions presented in the next section 4, we first present an example program execution for a given document. This example includes the execution steps of the algorithm. However for a detailed description of the algorithm itself, please refer to [23].

⁶It is not exclusively to one child. In the case of multiple children the root can be also a member of a Surrogate set.

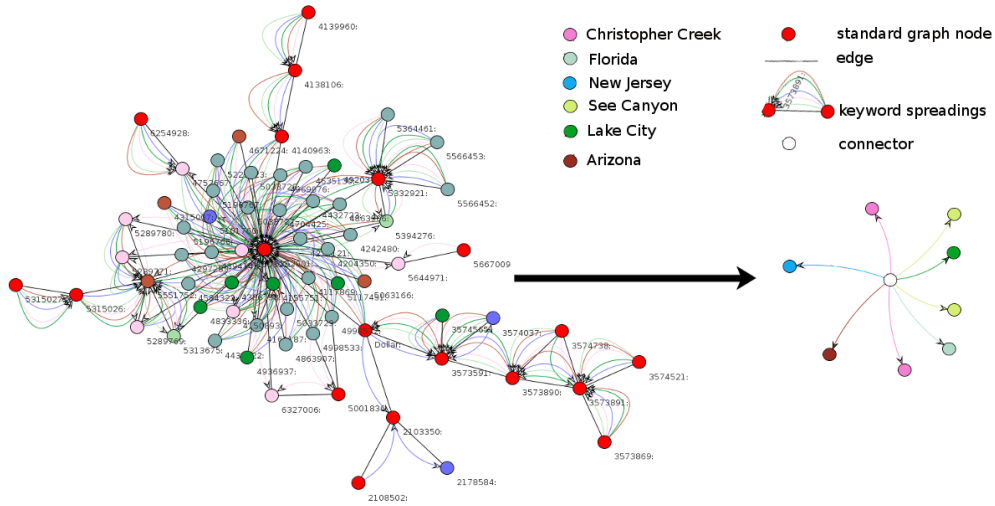


Figure 3: Pruned⁶ Processing Graph (numbers refer to URIs)

Example: “A wildfire in northern Arizona [...]. a fire north of Lake City in Florida. Flames remained about a mile from the community of Christopher Creek. The community is south of See Canyon [...]. Elsewhere New Jersey [...]”

The algorithm retrieves the entity identifiers entailed in the text document (underlined words). Subsequently a surrogate set S_i for each identifier is generated including the nodes that can be described by the names in question (cf. Sect. 3.2). After the surrogate sets have been retrieved the included nodes are initially activated (cf. Sect. 5). The initial activation reflects the amount of nodes within a surrogate set and decreases with the size⁷. In the above textual example New Jersey is the least ambiguous identifier - 3 nodes bear this name - while Florida the most ambiguous - 249 nodes. All nodes within the surrogate sets are added to the main processing queue Q . As the queue is ordered according to the activation of the nodes⁸, the surrogates of New Jersey are at the top position. Each node holds informations about its surrounding surrogates. This implies associated activation and distance values for each reachable surrogate. In the first step the most activated node spreads his information to the nodes in the near vicinity. As the surrogates of “New Jersey” bear the highest activation they concentrate the exploration in their vicinities in the beginning of the algorithm. The nodes are selected iteratively from Q and each one spreads its information to its vicinity. Each target node verifies if it contains already information about the spread identifier activations. If this is the case it checks if the new information contains a higher activation value for the identifier in question, it changes its information concerning this identifier. Each node spreads the information for each assigned identifier separately. Each node is only explored once. Afterwards the information exchange is done via back-propagation. If an already explored node changes an identifier-specific value this value is spread via back-propagation to all adjacent nodes in order to allow an update throughout the whole explored network. Throughout the algorithm each node is checked if it includes already information about all identifiers, i.e. at least one surrogate for each surrogate set. If this is the case the node is called connector. After the execution of the algorithm a set of connectors has been retrieved.

⁷If node prestige and proximity are equal for all nodes (cf. Sect. 5).

⁸The queue is reordered if activation values change.

Each of them represents another subgraph of the overall ontology graph. They are ranked according to their overall activation values, i.e. the sum of their identifier dependent activation values, and the one with the highest activation is used as result.

In the example, there is no close connection between the surrogates of “New Jersey” and the surrogates of the other identifiers (beside the obvious fact that all of them are in the USA). This phenomenon can occur if all identifiers are used concurrently as input for the algorithm. The exploration of the graph is shown in Fig. 3⁹.

After the initial focus on New Jersey the algorithm changes its centrality. This is due to the aggregation of the multiple identifier-dependent activation values in nodes that are connected with multiple surrogate nodes. Here the distance also influences the amount of activation. The graph in Fig. 3 concentrates around the node representing the USA. This is due to the fact that all towns mentioned in the text are located within the USA. The multiple result graphs are ranked according to the activation of the connector and the Steiner Tree out of the best ranked graph is shown on the right side of the figure. However, this connector was not able to reduce the ambiguity for Christopher Creek as well as for See Canyon to one.

4. ALGORITHMIC EXTENSIONS

In this paper we introduce three independent extensions of the base form of the algorithm presented in the prior section. First we added the variant *context-based* that splits the document in contextual subparts which are independently explored. The second variant *bidirectional* extends the so far backward based exploration in an exploration considering the situation from the perspective of both nodes within a triple. The third modified algorithm variant *reinforcement learning* considers the preceding results while processing a set of documents. The different variants are explained in the remainder of this section.

4.1 Context-based

By analysing human conversations, movies and films one common characteristic is apparent. Humans tend to cluster related in-

⁹This graph is already pruned to the closer vicinity of the connector node as otherwise the graph would be too big.

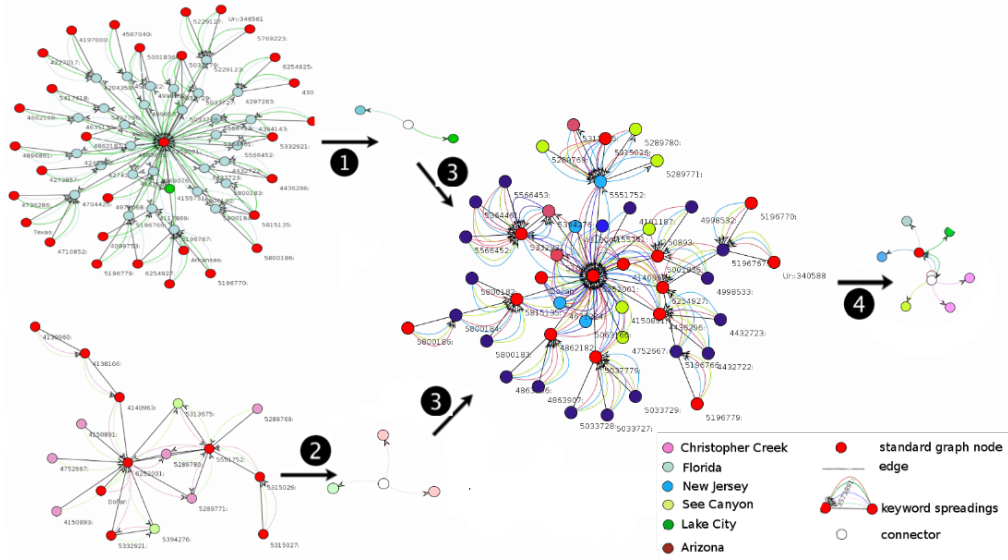


Figure 4: Processing - contexts

formation within a certain time-frame of the conversation or a certain part of a document. This allows humans to separate one aspect of the conversation from another. Even if a later one refers to one of these aspects, crucial information still is given in the prior mentioned context or in context of the reference. We take advantage of this phenomenon by translating it to our problem by splitting the complete text given by a document in different contextual parts. This is done by a segment-specific separation, means the text itself is split into different segments. All identifiers within a segment form a *context*. If multiple contexts overlap with respect to the entailed identifiers these segments are merged into one context. Each context is analysed using the already described base-form of the algorithm. This results in segment-specific Steiner trees as result of the respective analysis. After the individual algorithm execution per segment, a final overall algorithm execution is performed. By “overall” we refer to the use of all identifiers within the text as input of this algorithm. This is equal to the normal analysis by the base-form of the algorithm. However, we re-weight the nodes in the segment-specific results by setting their initial activation $a_{u,i}$ equal to the connector node’s overall activation a_v . This increases the activation of these nodes against the other nodes in the surrogate sets and results in a preferable selection within the algorithm execution, but still allows the selection of other surrogate nodes not included in the context-specific result but useful in the overall algorithm situation.

Example:

1. “A wildfire in northern *Arizona*” (context 1)
2. “[...] A fire north of *Lake City* in *Florida*. Flames remained about a mile (context 2)
3. from the community of *Christopher Creek*. The community is south of *See Canyon* (context 3)
4. [...] *Elsewhere New Jersey* [...]” (context 4).

In the following we refer to the same sample document used in Sect.3.3. For this type of algorithmic variant we split the document into multiple segments. The size of the segment is given by the algorithm. The splitting results in 4 segments. As none of the segments shares identifiers with the other segments the segments are directly transferred to unique contexts. After the identification of

the different contexts each context is analysed separately in a own algorithm execution. This is also shown in Fig.4. In the first context the identifier “Arizona” is included. As the algorithm cannot reduce the ambiguity of an identifier without at least another given identifier this segment is not analysed in a own process. The second segment includes “Lake City” and “Florida”. The graph (shown in the upper left corner of the figure) shows this segment specific analysis. The number 1 indicates the result graph for this context. Here it was already possible to disambiguate both identifiers. The next context-specific exploration concerning “Christopher Creek” and “See Canyon” is shown in the lower left corner. The number 2 indicates the context-specific result. Here, still two “Christopher Creek’s” are included. In the next step of this variant-specific analysis, all leaf nodes in the result graphs get a new initial activation value. This information is regarded in the next graph generation using all surrogates for all identifiers, but with specific initial activations (shown by 3). In the end, this last analysis delivers the final result graph (4). Unfortunately, “Christopher Creek” could not be further refined.

4.2 Bidirectional analysis

Our algorithm explanation in Sect. 3 contains the description of the base-algorithm representing the exploration based on the comparison of nodes. This is currently done by forward exploration, i.e. a triple $\langle u, e, v \rangle$ as given input parameter is analysed only from one perspective. Only the algorithm-dependent parameters for u are evaluated and eventually adapted in the case of a possible optimisation of its values. Fig. 5 shows this default forward exploration method. Starting from the given three nodes USA, Boston and Kansas, the algorithm selects USA based on the highest initial activation of 0.5. In the first processing step USA spreads his information to the nodes Boston and Kansas ($USA = u$; Boston, Kansas = v). Thus it spreads an activation value of 0.13^{10} to Boston (1) and 0.4 to Kansas (2). After this step the nodes Boston and Kansas have information about USA.

This is the case as each relation can also be traversed in the op-

¹⁰the edge degree value is lower (see Sect. 5) relations of the same type the decayed spreading value of 0.4 is divided by 3

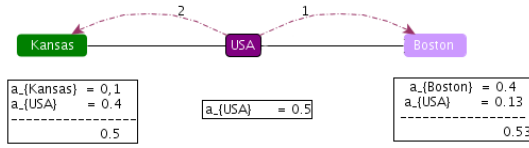


Figure 5: Forward expansion

posite direction. This ensures a “backward” exploration from u to v by the analysis of the triple $\langle v, e, u \rangle$. However these exploration steps do not coincide with each other as the exploration of a node depends on its activation value and thus on its position in Q . This results in large temporal distances between these two exploration steps. Getting back to the example Boston and Kansas do not know of each other, up to now. In the next step, Boston would be selected for spreading as its overall activation value is higher than the one of Kansas. In consequence, it takes for the exploration of each individual node until all three nodes get activation values concerning the initial activations of the other ones.

Our proposed modification changes the exploration by enabling a coincide exploration of both nodes in a triple. This is based on a subsequent mutual analysis. This means that after the triple $\langle u, e, v \rangle$ has been analysed, the triple $\langle v, e, u \rangle$ is analysed in the next step. Thus the algorithm has to be adapted to ensure the directly consecutive execution of the two exploration steps. This modification takes action in the algorithm as it also pre-explores a certain path(s) of node v before node v and its proximity itself is going to be explored¹¹. Thus the application flow of exploration is changed. Also the selection process for edges changes as now both perspective are considered. This implies also that another edge value can be chosen based on this change (cf. Sect. 5). This is shown in Fig. 6 (1) USA spreads his value to Boston (USA 0.13) and (2) immediately Boston spreads his value back to USA (Boston 0.11). Now the next triple adapted to USA is retrieved and used for spreading, thus (3) USA spreads his as well as his value of Boston to Kansas (USA 0.4, Boston 0.08). In consequence of the bidirectional exploration (4) Kansas spreads his information back to USA (Kansas 0.08) and due to back-propagation (5) USA spreads his value of Kansas to Boston (Kansas 0.02). All information has been exchanged during the analysis of USA. In the end Kansas will be used next as his overall activation value is higher than the one of Boston.

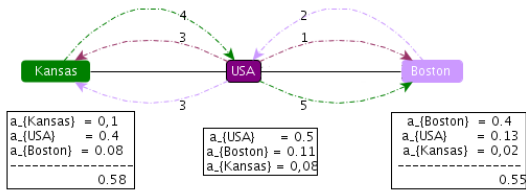


Figure 6: Bidirectional expansion

This modification has two effects. First, each explored node possesses the maximum of actual available information in the graph. By the use of this kind of spreading each pair of nodes exchanges their complete knowledge and this information is propagated in the next step to all adjacent nodes of both nodes. Thus each explored

¹¹If v has been explored already, this information is propagated.

node is notified by an update. Second, expansion of the graph according to its total size is slower, as the additional spreading takes more time. Third the order of exploration can be changed and thus the differences in activation are also reflected in the selection order from queue Q .

4.3 Reinforcement learning

The base algorithm as well as the context-based and bidirectional modifications are completely self-contained. Of course, they refer to measures, an ontology and textual information. However, no kind of learning functions are used by these variants. This includes no kind of external feedback like user-relevance feedback or use of annotated material. For realizing this kind of algorithmic extension, we focus on the use of reinforcement learning.

Reinforcement learning is a machine-learning technique where an agent learns based on prior executed actions and uses this knowledge in order to evaluate and adapt its upcoming actions enabling him to take optimal decisions.

Following this principle, this method makes use of formerly processed information in order to improve upcoming algorithm executions. Thereby this method applies this principle based on the fact that the result of each disambiguation process represents a set of entities of variable size which are most probable for a given identifier. Subsequently, this information is verified and remembered. The stored information includes the surrogate result as well as the processing number of the document. This variant makes use of all prior stored information for the next algorithm execution. If an identifier occurs which has already been used before – and, in consequence, information about the former referent is available – this information is used for weighting in the current process. The weighting is done by a change of the initial activation value of a node. This weighting is based on the amount of occurrences of an entity in the former result sets. In consequence these surrogates are focused during the graph exploration.

5. MEASUREMENTS

Initial Activation. As mentioned before, each node included in the initial set has an associated activation value. This activation value is spread (transferred) to adjacent nodes through the algorithm. We calculate this initial activation (see [6, 20, 30]) via

$$a_{u,i} = \frac{nodePrestige(u)}{|S_i|} \times proximity(u,i) \quad (1)$$

This specific initial activation for u depends on the entity identifier (i). Usually a node represents only one keyword, however this method allows also the representation of multiple identifiers by the same node. The variable *nodePrestige* allows for the association of a node-specific importance value, e.g. this allows to prefer instances of a certain concept by giving them a higher *nodePrestige*. Here additional information computed by other algorithms can be used to influence the spreading, e.g. based on pre-executed machine-learning algorithms. The factor *proximity*(u, i) can be used to evaluate identifier specific informations for the node u , e.g. based a linguistic analysis.

After the initial activation, an activated node spreads its activation to its vicinity. The calculation of the spread activation from node u to node v is done via $a_{v,i} = a_{u,i} \times \lambda \times degree(e_{u,v})$. As in the case of initial activation, the spreading of activation is done identifier dependent. Each spreading is decreased by the attenuation factor λ . Thus the importance of the activation value decreases with growing distance. The function *degree*($e_{u,v}$) allows for a triple-specific ranking of the relation between u and v . This function allows a domain-specific adaptation. We currently use an

instance-specific measure. It is similar to SemRank [2] which employs concept-specific measures for the degree of the relation between two nodes. SemRank basically uses the number of relations of type c between concept a and concept b in order to calculate the degree for a relation of type c between an instance of a to an instance of b . We perform an instance-specific degree calculation. We use the number of relations of a certain type between two specific instances in order to calculate its importance. This is shown in Fig. 7.

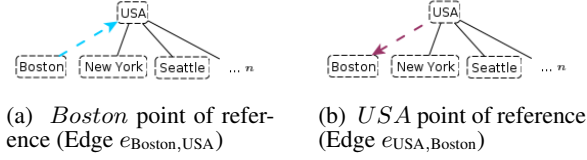


Figure 7: Example of Edge Degree

On the left-hand side, Boston is related to USA via an *inCountry* relation. On the right-hand side, USA is related via edge *liesInCountry* to Boston. In the first case, the instance Boston has only one associated relation of this type, i.e. $\frac{1}{1}$. In the second case, USA possesses a large amount (n) of relations of the type *liesInCountry*, i.e. $\frac{1}{n}$. In consequence, an activation spread from Boston to USA is less decreased than an activation from USA to Boston¹².

Ranking. The summation of all identifier-dependent activation values and the nodePrestige ($a_v = \sum_{i \in I} a_{v,i} + \text{nodePrestige}(v)$) results in the overall activation value. This value is used as selection criterion for the node selection form Q in the algorithm. Within the algorithm it is repeatedly checked if a node has activation values for *all* entity identifiers. In the case this is true for a node this node is declared as a *connector node* representing a resulting Steiner Tree as proposed above. The *importance* of this tree is given by the overall activation value of the *connector node*. The *connector node* with the highest activation is used as result of the algorithm.

6. RELATED WORK

Related work can be found in the areas of (i) word-sense disambiguation on graphs, (ii) ontology-element spotting and disambiguation, as well as (iii) keyword and entity search on graphs.

Word-sense disambiguation on graphs: Veronis and Ide [37] 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 in the gloss activates its associated sense while the sense itself has an associated gloss again. Tsatsaroni *et al.* [36] applied this approach to WordNet, including a modification that allows to use direct associations between senses. Rada *et al.* [32, 34] build graphs from textual context information and use them for disambiguation based on co-occurring context words. Bhattacharya *et al.* [7] also represent textual co-references using hyper-graphs. They merge different graphs based on their similarities in order to identify the exact entities. Mailaise *et al.* [25] used an SA-based approach for disambiguation between annotations of TV programme descriptions, including human feedback to increase the precision. This class of approaches builds upon linguistics. They focus either on lexicons or on measures achieved by natural-language analysis. They do not exploit specific ontological knowledge. Background knowledge in these approaches is mostly domain-specific.

¹²Depending on the edge type.

Ontology-element disambiguation: Banek *et al.* [4] associate a WordNet Synset to each ontology element. The disambiguation uses the synonyms for identifying similar elements. A similar approach by Nguyen *et al.* [27] uses an associated bag-of-words vector to each entity—including certain nouns, co-occurring entities and further Wikipedia knowledge. Nguyen *et al.* also use the KIM-ontology for disambiguation [26]. They use the textual distance between two entities in a learning corpus and preferred entities of the same concept affiliation. Garcia *et al.* [11] use a modified PageRank algorithm; 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.* present a modification of the algorithm including Wikipedia information [12]. Hassel *et al.* [16] present an approach based on the DBLP-ontology which disambiguates authors occurring in mails published in the DBLP-mailing list. They use ontology relations of length one or two, in particular the co-authorship and the areas of interest. Volz *et al.* [38] use contextual information for detecting the concept affiliation of entities. Kleb *et al.* [24] use concept-dependent text patterns for the disambiguation of text information. Gruhl *et al.* [14] train an SVM classifier in order to spot ontology entities.

In our algorithm, many common ideas from information retrieval (IR) have been adopted. In contrast to other approaches, we neither require training data to learn classifiers for recognition, nor do we focus on the analysis of textual knowledge. Our approach may use this information later, in order to improve its weighting scheme (cf. 8). But currently, we focus on exploring complex structural background knowledge. Apart from disambiguation, Hasan [15] 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.* [6] present 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 optimized by Kacholia *et al.* [20] 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. 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. BLINKS [18], based on [20], selects clusters according to the smallest cardinality. Thanh *et al.* [35] present a search algorithm for RDF-Graphs based on one iterator per surrogate set. Recently, Kasneci *et al.* [18] proposed an approximation algorithm for Steiner trees that allows for a fast retrieval of relationship graphs based on minimal weights. Bao *et al.* [5] proposed a new ranking measurement for keyword search on RDF-graphs. This also involves the aspect of ambiguity which is tackled by this heuristic.

In contrast, 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 also allow for an adaption of our

weighting scheme by the use of pre-analysis tasks.

7. EVALUATION

Our approach enables the disambiguation of ambiguous identifiers according to entities defined in an ontology. By ambiguity we describe the fact that multiple ontology elements can be specified by the same label. The unique identification of ontology elements and thus a disambiguation of their identifiers enable a multiplicity of possibilities for later applications. In especially the additional stored semantic associations to an identified ontology object can be used for further processing.

Two conditions are necessary before the algorithm can be executed. On the one hand we need an ontology defining the considered background knowledge. On the other hand we need also a set of entity identifiers to disambiguate. While these requirements should be easy to fulfill our search for an open available evaluation set was not successful. So to the best of our knowledge there is no public available pre-annotated dataset out there, that could have been used. In consequence we generated our own evaluation dataset which has also been used in our prior evaluations [38, 24, 23].

Nowadays many available data can be related with ontologies in the web. Especially as the amount of publicly available ontologies increases from day to day. We referred to the Linking Open Project¹³. This project includes various interesting open datasets on the web, e.g. Wikipedia in from of DBpedia which includes a multiplicity of entities and their identifiers. We referred to another member of this dataset called Geonames¹⁴. This dataset describes geographical facts, e.g. that “Berlin” is the capital of “Germany”. The ontology includes 6,085,125 different geographical element identifiers. The most ambiguous identifiers is “First Baptist Church” which is related to 2,085 different entities in the ontology. As source of input data containing identifiers we choose newspaper articles. We crawled articles from the European Media Monitor[29], a website collecting news from a selection of European newspapers. We choose the category *natural disasters* in order to focus on articles containing geographic references. The articles denote to a specific event as well as to multiple events. In average 6.55 entities are included in an article and an article identifier can be referred to 34.28 possible entities in the ontology. In theory 248.79 combinations can be done in average.

By using an ontology from the LOD and newspaper texts as input source we refer to a real world scenario that can be transferred to multiple other domains. Thus no artificial evaluation dataset had been chosen.

7.1 Evaluation Process

In order to evaluate the results of our dataset analysis we refer the standard methods in information retrieval *precision* and *recall*. However both measure take an available annotated evaluation dataset as granted. Therefore the set of newspaper articles have been manually annotated¹⁵. In consequence each document includes a set of annotations denoting to the most probable ontology entity.

As stated is the result of our algorithm given by a Steiner Tree including at least one surrogate per identifier. The algorithm chooses the graph represented by the most activated connector node. If multiple surrogates of an identifier are included in the result graph only

¹³<http://www.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

¹⁴<http://www.geonames.org>

¹⁵The human annotation process by non domain experts runs the risk of wrong annotations. This cannot be excluded in the dataset evaluation.

the surrogate with the highest activation is selected as result for the identifier in question. If there are multiple nodes sharing the highest activation this set is chosen as result.

The *recall* ($R_Z := \frac{|relevant\ entities \cap retrieved\ entities|}{|retrieved\ entities|}$) 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* ($f_{measure} := \frac{2 * R_Z * P_Z}{R_Z + P_Z}$) is the harmonic mean between recall and precision.

For a fast execution of our algorithm we used following thresholds. The maximum exploration depth was set to 4 nodes. Minimum activation for spreading was 0. The edge degree was restricted to 1 for forward exploration.

7.2 Evaluation Results

A total of 200 newspaper articles out of the category “Natural Disasters” from EMM have been used for evaluation. The current version of the algorithm disambiguates the denoted ontology elements referenced by the included identifiers. The algorithm only allow for a node set optimisation in case at least two identifiers are given. Otherwise only the whole surrogate set associated to the detached identifier is returned. As consequence we excluded from this set of documents all documents including only 1 identifier. Due to this restriction 169 documents out of the set of 200 could have been used.

Additional to the *base* form of the algorithm we also evaluated the proposed algorithmic extension *context-based*, *bidirectional* and *inductive learning*.

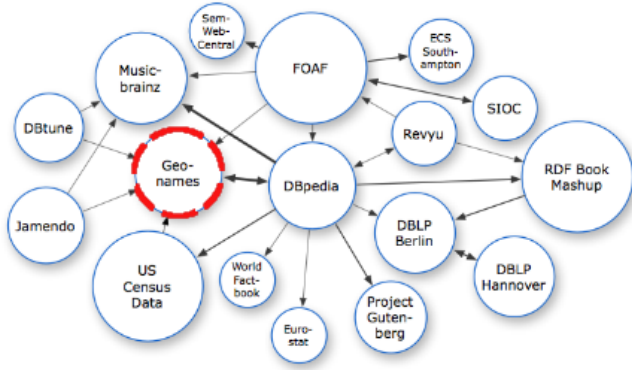
variant	recall	precision	f-measure
base	76.03	68.71	71.09
context	75.03	69.89	71.70
reinforcement	77.63	72.46	74.11
bidirectional	78.05	73.14	75.05

Table 1: Evaluation Results sorted by variant (with $degree(e_{u,v})$)

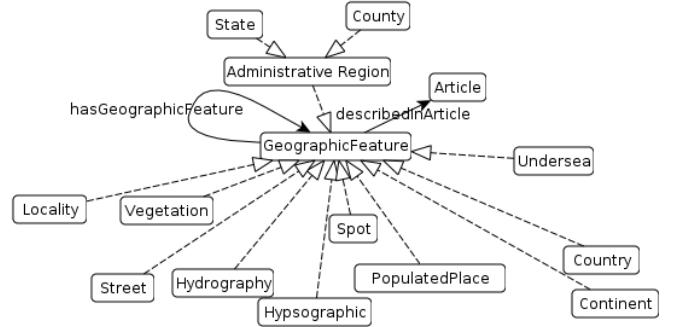
The result included in Tab. 1 showed that bidirectional performed best with an f-measure of 75% directly followed by the variant using reinforcement learning. All extension variants performed better than the base variant. Thus the additional benefit by using the variants becomes clear. The current evaluation result of the context variant does not meet the initial expectations. A deeper analysis made clear that the selection of the context size is crucial for the success of this method. Further evaluations will show if this method can gain an additional value in respect to the base algorithm. Reinforcement learning improved the results by using the nodes within the results of the evaluated prior algorithm executions. This method takes advantage from the current dataset which includes articles about the same topic by different newspapers. This is due to the specification of EMM which collects articles from different European newspaper. Thus different newspaper articles about the same topic occur in the dataset. Bidirectional exploration improved the edge selection as both directions are considered. In combination with the edge degree measure this variant achieved best evaluation results.

base	context	reinforcement	bidirectional
0%	9%	21.89%	15.38%

Table 2: Best results per variant (with $degree(e_{u,v})$)



(a) LOD-cloud (excerpt Mai 2007)



(b) Used Ontology excerpt

Figure 8: Geonames Ontology

Tab. 2 shows percentage of documents for each variant in which the variant achieved better results than any of the other extensions. Reinforcement reached the highest percentage of clear improvements, however also missed the optimal result in other cases. Here bidirectional method was the second best result.

variant	recall	precision	f-measure
base	76.83	54.98	61.2
context	76.94	58.12	63.64
reinforcement	79.07	62.15	66.79
bidirectional	78.28	56.72	62.6

Table 3: Evaluation Results sorted by variant (without $degree(e_{u,v})$)

Table 3 entails the variant specific results in case no degree measure is considered. All variants resulted in worse results than by the use of the suggested measure. This is due to the misleading edge selection and a non-optimal Steiner Tree generation for the disambiguation process. Considering all edges as equally important leads to an inclusion of a majority of surrogates out of each set in the result graph. The higher amount of surrogates in the result increases the recall but also includes a large set of false positives and thus a low precision value.

In general the result ordering of the approaches according precision and recall stays the same for both variants. The influence by weighting the edges is minor opposed to the influence of the different algorithmic variants. The use of prior results due to reinforcement learning confirms the assumption of (partly) re-occurring entity constellations. This phenomenon of repetitions in corpora is also known in the NLP-Research. The use of bidirectional spreading enables a better result than the base variant. However it comes along with a slower execution time (factor 3:2). The use of contexts results in a lower recall and a higher precision than the base variant (with degree). The quality of the disambiguation depends on the entities entailed in a context window. A retrieval of wrong candidates based on a context disambiguation leads to a higher activation value for those candidates in the disambiguation of the whole text.

In the later the wrong prior initial activation estimations are hard to overcome and thus the correct referents are not included in the result.

8. SUMMARY AND OUTLOOK

In this paper we presented three novel extensions for an algorithm allowing the automatic reference resolution (within an ontology) for a given set of (typically ambiguous) entity identifiers (typically occurring within text documents). The three presented extensions comprise (1) a method for contextual splitting of documents in order to create sets of more closely connected co-referenced entities; (2) a method for the modification of the activation spreading that enables a bidirectional information exchange in the graph; and (3) a method based on reinforcement learning taking advantage of re-occurring entities.

We examined the effect of these extensions on the algorithm's overall result quality through an evaluation based on the widely known information retrieval measures precision and recall. For the extension by contextual splitting, the evaluation results in an improvement for some specific documents in the evaluation set, but not for the whole evaluated dataset. The use of bidirectional spreading and reinforcement learning in the evaluation scenario resulted in a better quality than without.

As next steps, we are going to investigate the effect of contextual windows in more detail. The retrieval of an optimal context size per document would allow for a better aggregation of entities per context window and thus a better spreading in the context graphs. Here also linguistic methods can be used in order to detect the linguistic references between identifiers. Concerning the bidirectional spreading, we will further examine the modification via specific degree measures and possible vicinity parameters that may influence the use of this spreading based on specific situation parameters.

Also the use of conceptual information via spreading in the concept graph in relation to the instance graph can be used. We also investigate the use of concept-specific measures.

Acknowledgment: This work has been partially financed by the German Federal State of Baden-Württemberg and by the Federal German Ministry of Economics and Technology under the research programme THESEUS ("New Technologies for the Internet of Services"), as part of CTC WP3 "Ontology Management".

9. REFERENCES

- [1] J. R. Anderson. A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior*, 22:261–295, 1983.
- [2] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 117–127, New York, NY, USA, 2005. ACM.
- [3] J. Apresjan. Regular polysemy. *Linguistics*, 142:5–32, 1973.
- [4] M. Banek, B. Vrdoljak, and A. M. Tjoa. Word Sense Disambiguation as the Primary Step of Ontology Integration. In *Proc. DEXA*. Springer-Verlag, 2008.
- [5] Z. Bao, T. W. Ling, B. Chen, and J. Lu. Effective xml keyword search with relevance oriented ranking. In *ICDE*, pages 517–528, 2009.
- [6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In *Proc. ICDE*. IEEE Computer Society, 2002.
- [7] I. Bhattacharya and L. Getoor. Collective Entity Resolution In Relational Data. *IEEE Data Eng. Bull.*, 29(2), 2006.
- [8] A. M. Collins and E. F. Loftus. A Spreading-Activation Theory of Semantic Processing. *Psychological Review*, 82(6), 1975.
- [9] A. Copestake and T. Briscoe. Semi-productive polysemy and sense extension. *Journal of Semantics*, 12:15–67, 1995.
- [10] F. De Saussure, C. Bally, A. Sechehayé, R. Harris, and A. Riedlinger. *Course in general linguistics*. Open Court, 1986.
- [11] N. F. García, J. M. B. del Toro, L. Sánchez, and A. Bernardi. IdentityRank: Named Entity Disambiguation in the Context of the NEWS Project. In *Proc. ESWC*, 2007.
- [12] N. F. García, J. M. B. del Toro, L. Sánchez, and V. L. Centeno. Semantic Annotation of Web Resources Using IdentityRank and Wikipedia. In *Proc. AWIC*, 2007.
- [13] R. Grishman and B. Sundheim. Message Understanding Conference - 6: A Brief History. In *Proc. Int. Conference on Computational Linguistics*, 1996.
- [14] D. Gruhl, M. Nagarajan, J. Pieper, C. Robson, and A. P. Sheth. Context and Domain Knowledge Enhanced Entity Spotting in Informal Text. In *Proc ISWC*, 2009.
- [15] M. M. Hasan. A Spreading Activation Framework for Ontology-Enhanced Adaptive Information Access within Organisations. In L. van Elst, V. Dignum, and A. Abecker, editors, *Proc. AMKM*, volume 2926 of *LNCS*. Springer, 2004.
- [16] J. Hassell, B. Aleman-Meza, and I. B. Arpinar. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In *Proc. ISWC*, 2006.
- [17] J. Hayes and C. Gutiérrez. Bipartite graphs as intermediate model for rdf. In *International Semantic Web Conference*, pages 47–61, 2004.
- [18] H. He, H. Wang, J. Yang, and P. S. Yu. BLINKS: Ranked Keyword Searches on Graphs. In *Proc. SIGMOD*, New York, NY, USA, 2007. ACM Press.
- [19] D. Jurafski and J. H. Martin. *Speech and Language Processing*. Prentice Hall, 2000.
- [20] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional Expansion for Keyword Search on Graph Databases. In *Proc. VLDB*, 2005.
- [21] A. Kilgariff. *Polysemy*. PhD thesis, University of Sussex, UK, 1992.
- [22] J. Klavans. Polysemy, Ambiguity and Generativity. Technical report, AAAI Symposium, 1995.
- [23] J. Kleb and A. Abecker. Entity Reference Resolution via Spreading Activation on RDF-Graphs. In *Proc. ESWC 2010, Part I*, pages 152–166, 2010.
- [24] J. Kleb and R. Volz. Ontology based Entity Disambiguation with Natural Language Patterns. In *Proc. ICDIM*, 2009.
- [25] V. Malais, L. Gazendam, and H. Brugman. Disambiguating Automatic Semantic Annotation Based on a Thesaurus Structure. In *Proc. TALN*, 2007.
- [26] H. T. Nguyen and T. H. Cao. A Knowledge-Based Approach to Named Entity Disambiguation in News Articles. In *Proc. AI*, 2007.
- [27] H. T. Nguyen and T. H. Cao. Named Entity Disambiguation on an Ontology Enriched by Wikipedia. In *Proc RIVF 2008*, 2008.
- [28] G. Pethö. What Is Polysemy? A Survey of Current Research and Results. In K. Bibok and E. Németh, editors, *Pragmatics and the Flexibility of Word Meaning*, pages 175–224, Amsterdam, 2001. Elsevier.
- [29] B. Pouliquen, R. Steinberger, C. Ignat, E. Käsper, and I. Temnikova. Multilingual and Cross-lingual News Topic Tracking. In *Proc. COLING. ACL*, 2004.
- [30] S. Preece. *A Spreading Activation Network Model for Information Retrieval*. PhD thesis, University of Illinois, Urbana-Champaign, 1985.
- [31] M. Quillian. A Revised Design for an Understanding Machine. *Mechanical Translation*, 7(1):17–29, 1962.
- [32] M. Rada. Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. In *Proc. HLT. ACL*, 2005.
- [33] S. Sekine and H. Isahara. IREX: IR and IE Evaluation project in Japanese. In *Conference on Language Resources and Evaluation*, 2000.
- [34] R. Sinha and M. Rada. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In *Proc. ICSC*. IEEE Computer Society, 2007.
- [35] T. Tran, H. Wang, S. Rudolph, and P. Cimiano. Top-k Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped RDF Data. In *Proc. ICDE*. IEEE, 2009.
- [36] G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In *Proc. IJCAI*, 2007.
- [37] J. Veronis and N. M. Ide. Word Sense Disambiguation With Very Large Neural Networks Extracted From Machine Readable Dictionaries. In *Proc. COLING. ACL*, 1990.
- [38] R. Volz, J. Kleb, and W. Müller. Towards Ontology-based Disambiguation of Geographical Identifiers. In *Proc. WWW Workshop I³*, 2007.