Ontology Querying Support in Semantic Annotation Process

Miha Grčar[1], Vid Podpečan[1], Borut Sluban[1], Igor Mozetič[1]

[1] Jožef Stefan Institute, Jamova c. 39, 1000 Ljubljana, Slovenia

{miha.grcar, vid.podpecan, borut.sluban, igor.mozetic}@ijs.si

Abstract. In this paper, we present an approach to ontology querying for the purpose of supporting the semantic annotation process. We present and evaluate two algorithms, (i) a baseline algorithm and (ii) a graph-based algorithm based on the bag-of-words text representation and PageRank. We evaluate the two approaches on a set of semantically annotated geospatial Web services. We show that the graph-based algorithm significantly outperforms the baseline algorithm. The devised solution is implemented in Visual OntoBridge, a tool that provides an interface and functionality for supporting the user in the semantic annotation task. The improvement over the baseline is also reflected in practice.

1 Introduction and Motivation

Semantic annotations are formal, machine-readable descriptions that enable efficient search and browse through resources and efficient composition and execution of Web services. It this work, the semantic annotation is defined as a set of interlinked domain-ontology elements associated with the resource being annotated. For example, let us assume that our resource is a database table. We want to annotate its fields in order to provide compatibility with databases from other systems. Further on, let us assume that this table has a field called "employee_name" that contains employee names (as given in Fig. 1, left side). On the other hand, we have a domain ontology containing knowledge and vocabulary about companies (an excerpt is given in Fig. 1, right side). In order to state that our table field in fact contains employee names, we first create a variable for the domain-ontology concept *Name* and associate it with the field. We then create a variable for an instance of *Person* and link it to the variable for *Name* via the *hasName* relation. Finally, we create a variable for an instance of *Company* and link it to the variable for *Person* via the *hasEmployee* relation. Such annotation (shown in the middle in Fig. 1) indeed holds the desired semantics: the annotated field contains names of people which some company employs (i.e., names of employees).

Note that it is possible to instantiate any of the variables with an actual instance representing a real-world entity. For example, the variable *?c* could be replaced with

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

an instance representing an actual company such as, for example, *Microsoft* ∈ *Company*. The annotation would then refer to "names of people employed at Microsoft".

The annotation of a resource is a process in which the user (i.e., the domain expert) creates and interlinks domain-ontology instances and variables in order to create a semantic description for the resource in question. Formulating annotations in one of the ontology-description languages (e.g., WSML [1]) is not a trivial task and requires specific expertise.

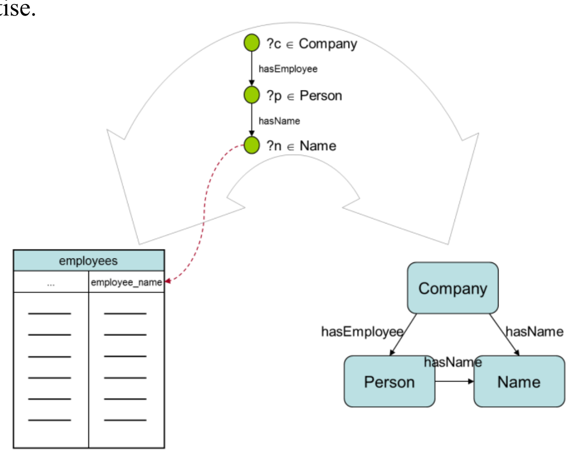

Fig. 1. Annotation as a "bridge" between a resource and the domain ontology.

In the European projects SWING[1] and ENVISION[2], we have for this reason developed Visual OntoBridge (VOB) [2], a system that provides a graphical user interface and a set of machine learning algorithms that support the user in the annotation task. VOB allows the user to (i) visualize the resource and the domain ontology (much like this is done in Fig. 1), (ii) create variables by clicking on the domain-ontology concepts, and (iii) interlink the variables and/or instances by "drawing" relations between them.

In addition, the user is able to enter a set of natural-language (Google-like) queries, according to which the system provides a set of "building blocks" that can be used for defining the annotation.

The main purpose of this paper is to present and evaluate the developed ontology querying facilities implemented in VOB. The paper is organized as follows. In Section 2, we present two approaches to ontology querying, the baseline approach and the proposed graph-based approach. We evaluate these two approaches in Section 3 and present our conclusions in Section 4. We discuss some related work in Section 5.

---

1 Semantic Web Services Interoperability for Geospatial Decision Making (FP6-26514), http://138.232.65.156/swing/

2 Environmental Services Infrastructure with Ontologies (FP7-249120), http://www.envision-project.eu/

2 Ontology Querying Approach

Establishing annotations manually is not a trivial task, especially if the domain ontology contains a large number of entities and/or the user is not fully familiar with the conceptualizations in the ontology. VOB provides functionality for querying the domain ontology with the purpose of finding the appropriate concepts and triples. A triple in this context represents two interlinked instance variables (e.g., *?Company* hasEmployee *?Person*) and serves as a more complex building block for defining semantic annotations.

The process of querying the domain ontology is as follows. The user enters a set of Google-like natural-language queries. The system then provides the user with two lists—the list of proposed concepts and the list of proposed triples. The user inspects the two lists, from top to bottom, and selects the relevant entities. If the required concepts/triples are not found at the top of the list, the user should consider reformulating the queries. The selected concepts and triples are transferred to the graphical annotation editor, where the user is able to revise and extend the annotation as required.

VOB employs text mining techniques, the PageRank algorithm, and consults a Web search engine to populate the two lists of recommended building blocks. In the following sections, we first present two important technical aspects of our ontology querying approach and later on discuss the developed ontology querying algorithms in more details.

2.1 Text Preprocessing and PageRank

In this section, we present two important aspects of our ontology querying procedure, i.e., the bag-of-words vector representation of documents and the (Personalized) PageRank algorithm.

Bag-of-Words Vectors and Cosine Similarity.
Most text mining approaches rely on the bag-of-words vector representation of documents. To convert text documents into their bag-of-words representation, the documents are first tokenized, stop words are removed, and the word tokens are stemmed [4, 5]. N-grams (i.e., word sequences of up to a particular length) can be considered in addition to unigrams [17]. Infrequent words and terms (n-grams) are removed from the vocabulary (regularization). The remaining words and terms represent the dimensions in a high-dimensional bag-of-words space in which every document is represented with a (sparse) vector. The words and terms in this vector are weighted according to the TF-IDF weighting scheme (TF-IDF stands for "term frequency—inverse document frequency"; see http://en.wikipedia.org/wiki/Tf*idf). A TF-IDF weight increases proportionally to the number of times a word appears in the document, but is decreased according to the frequency of the word in the entire document corpus.

Cosine similarity is normally used to compare bag-of-words vectors. It measures the cosine of the angle between two vectors. Note that cosine similarity is equal to dot product, provided that the two vectors are normalized in the Euclidean sense.

PageRank and Personalized PageRank.
PageRank [6] is a link analysis algorithm that assigns a numerical weighting to each element of a set of interlinked documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm can be applied to any collection of entities with reciprocal references.

PageRank effectively computes the importance of a vertex in the graph with respect to a set of source vertices. The way PageRank is typically used, every vertex is considered a source vertex. This means that we are interested in the importance of vertices in general (i.e., with respect to the entire set of vertices). Note that it is possible to modify the algorithm to limit the set of source vertices to several selected vertices or only one vertex. This variant of PageRank is called Personalized PageRank (P-PR). "Personalized" in this context refers to using a predefined set of nodes as the starting nodes for random walks. At each node, the random walker decides whether to teleport back to the source node (this is done with the probability $1 - d$ where d is the so-called damping factor) or to continue the walk along one of the edges. The probability of choosing a certain edge is proportional to the edge's weight compared to the weights of the other edges connected to the node. In effect, for a selected source node i in a given graph, P-PR computes a vector of probabilities with components $PR_i(j)$, where j is one of the nodes in the graph. $PR_i(j)$ is the probability that a random walker starting from node i will be observed at node j at an arbitrary point in time.

2.2 Baseline Ontology Querying Algorithm

In [3], we presented and evaluated several term matching techniques that serve as the basis for automating the annotation process. To produce the two lists of recommendations as discussed in the previous section, it is possible to directly apply the term matching techniques. The algorithm is as follows:

1. Each concept and each possible domain-relation-range triple in the domain ontology is grounded through a Web search engine [3]. Grounding a term means collecting a set of documents and assigning them to the term. In our case, the terms are the concept and relation labels in the domain ontology. With the ontology being grounded, it is possible to compare a natural-language query to the grounded domain-ontology entities. To ground a concept, the search engine[3] is queried with the corresponding concept label. To ground a triple, on the other hand, the search engine is queried with the search term created by concatenating the label of the relation domain, the label of the relation, and the label of the relation range, respectively.
2. The groundings are converted into TF-IDF bag-of-words vectors [5]. Each vector is labeled with the corresponding domain ontology entity (either the concept or the triple label). These vectors constitute the training set (i.e., the set of labeled examples).

---

[3] We use the Yahoo search engine through their API.

3. The training set is used to train the centroid classifier [7]. Each centroid is computed as the l^2-normalized sum of the corresponding TF-IDF vectors.
4. The set of queries, provided by the user, is first grounded through a Web search engine. For each query, the corresponding centroid TF-IDF vector is computed. These TF-IDF vectors constitute the test set (i.e., the set of unlabeled examples).
5. Given a bag-of-words vector from the test set, the centroid classifier is employed to assign a classification score to each target class, that is, to each ontology entity. These scores are aggregated over the entire set of query vectors.

Given the set of bag-of-words vectors representing the user's queries, the classifier is thus able to sort the domain ontology concepts and triples according to the relevance to the queries. This gives us the two required lists of annotation building blocks: the list of concepts and the list of triples.

2.3 Incorporating Ontology Structure: The OntoBridge Approach

To establish the baseline discussed in the previous section, we treated the domain ontology as a flat list of entities. What we did not take into account is that these entities are in fact interlinked. This means that the domain ontology can be represented as a graph in which vertices are entities and edges represent links. In this section, we show how we can couple text similarity assessments with PageRank to exploit the ontology structure for determining relevant ontology entities.

To employ PageRank, the domain ontology is first represented as a graph. This can be done in numerous different ways. Naively, we could represent each concept with a vertex and interconnect two vertices with an undirected edge if there would exist at least one domain-relation-range definition involving the two concepts. However, since only the concepts would then be represented with vertices and would thus be the only entities able to "accumulate" rank, we would not be able to rank the triples. With a slightly more sophisticated transformation approach, it is possible to include the triples as well. This type of transformation is illustrated in Fig. 2. We create additional vertices (i.e., vertices representing triples; drawn as squares and triangles in the figure) to "characterize" all possible relations between the two concepts. We also include vertices representing triples based on inverse relations (drawn as triangles in the figure) even though they are not explicitly defined in the domain ontology. The reason for this is that we do not want the random walker to reach a triple vertex and then head back again; we want it to reach the other concept through a pair of directed edges.

In more details, the proposed ontology-to-graph transformation process is as follows:

1. Represent each concept with a vertex.
2. Represent each triple c_1-r-$c_2 \in \mathbf{T}$, where $\mathbf{T}$ is the set of triples in the domain ontology, with two vertices: one representing c_1-r-c_2 and one representing the corresponding inverse relation, c_2-r^{-1}-c_1.
3. For each pair of concepts c_1, c_2 and for each relation r such that c_1-r-$c_2 \in \mathbf{T}$, do the following:

- Connect the vertex representing c_1 to the vertex representing c_1-r-c_2 with a directed edge and weight it with $C(\mathbf{Q}, c_1\text{-}r\text{-}c_2)$. Here, $\mathbf{Q} = \{q_1, q_2, q_3\ldots\}$ is a set of natural-language queries and $C(\mathbf{Q}, c_1\text{-}r\text{-}c_2)$ is computed as $\Sigma_{q\in\mathbf{Q}}C(q, c_1\text{-}r\text{-}c_2)$. $C(a, b)$ refers to cosine similarity between the centroid of groundings of concept/relation a and the centroid of groundings of concept/relation b. A centroid is computed by first converting the corresponding groundings to TF-IDF feature vectors and then computing the l^2-normalized sum of these feature vectors.
- Connect the vertex representing c_1-r-c_2 to the vertex representing c_2 with a directed edge and weight it with 1.
- Connect the vertex representing c_2 to the vertex representing c_2-r^{-1}-c_1 with a directed edge and weight it with $C(\mathbf{Q}, c_1\text{-}r\text{-}c_2)$.
- Connect the vertex representing c_2-r^{-1}-c_1 to the vertex representing c_1 with a directed edge and weight it with 1. Note that since this is the only edge going out of this vertex, its weight can in fact be an arbitrary positive value. This is because PageRank normalizes the weights of the outgoing edges at each vertex so that they sum up to 1.

4. Represent each bag-of-words vector q_i, representing the test set $\mathbf{Q} = \{q_1, q_2, q_3\ldots\}$, with a vertex. Note that the test set represents the user queries.
5. For each bag-of-words vector q_i representing the query and each concept c_j, if $C(q_i, c_j) > 0$, create a directed edge from q_i to c_j and weight it with $C(q_i, c_j)$.

This process is illustrated in Fig. 2 where w_i represent the weights computed in Step 3 of the presented ontology-to-graph transformation process.

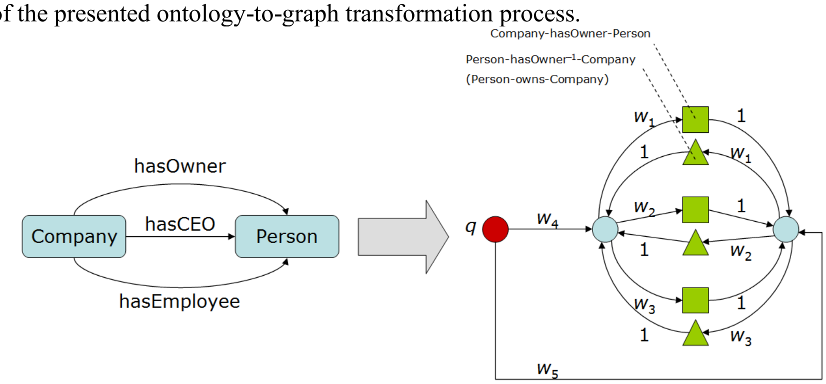

Fig. 2. Representing ontologies as graphs.

When the graph is created and properly weighted, we run PageRank to rank vertices (i.e., concepts and triples) according to the relevance to the query. The vertices representing the query are therefore used as the source vertices for PageRank. Note that a triple c_1-r-$c_2 \in \mathbf{T}$ "accumulates" the ranking score in two different vertices: in the vertex representing c_1-r-c_2 and in the vertex representing c_2-r^{-1}-c_1. It is thus necessary to sum the ranking scores of these two vertices to obtain the ranking score of the corresponding triple.

With the discussed procedure, every concept and every triple is ranked by PageRank. We can therefore populate the two lists of annotation building blocks and present these to the user.

3 Evaluation of the Ontology Querying Algorithms

We evaluated our approach to ontology querying in the context of a project committed to develop an infrastructure for handling geospatial Web services. In the devised system, geo-data is served by a database of spatial-information objects through standardized interfaces. One of such standard interfaces, defined by the Open Geospatial Consortium (OGC), is the Web Feature Service (WFS) [16]. WFS are required to describe the objects that they provide (e.g., rivers, roads, mountains…). These objects are also termed "features". Each feature is described with a set of attributes (e.g., water bodies can have depth, temperature, water flow…).

We used a set of WFS schemas, enriched with the golden-standard annotations and user queries, to evaluate the developed ontology querying algorithms. The evaluation process and the evaluation results are presented in the following sections.

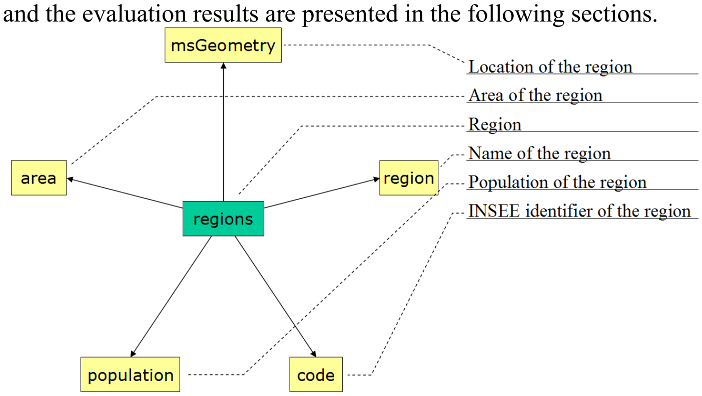

Fig. 3. The golden-standard acquisition form for the feature type "regions". The feature type (green box) with all its attributes (yellow boxes) is visualized in the left-hand side, the corresponding queries, provided by one of the participants, can be seen in the right-hand side.

3.1 Golden Standard

For the experiments, we acquired a set of Web Feature Services (WFS). Each WFS was accompanied with the corresponding semantic annotation and several sets of user queries. The service schemas were annotated with the SWING ontology (available at http://first-vm2.ijs.si/envision/res/swing.n3). It contains 332 concepts, 141 relations, and 4,362 domain-relation-range triples (taking the basic triple-inference rules into account). We asked the domain experts at Bureau of Geological and Mining Research (BRGM, France) to provide us with natural-language queries with which they would hope to retrieve relevant building blocks for the annotations. For this purpose, we gave each of the participating domain experts a set of forms presenting the WFS schemas. A participant had to describe each feature type with a set of English queries, one query per attribute and one additional query for the feature type itself. Fig. 3 shows one of such golden-standard acquisition forms.

We received input from 3 domain experts, each assigning queries to 7 feature types (41 queries altogether by each of the participants). We have identified 114 concepts and 96 triples (unique in the context of the same feature type) relevant for annotating the feature types involved in the golden-standard acquisition process. Since the acquired golden standard thus contained both, the queries and the corresponding building blocks, we were able to assess the quality of the ontology querying algorithms by "measuring" the amount of golden-standard building blocks discovered in the domain ontology, given a particular set of queries. We measured the area under the Receiver Operating Characteristic (ROC) curve to evaluate the lists produced by the algorithm. We discuss the evaluation process and present the results in the following section.

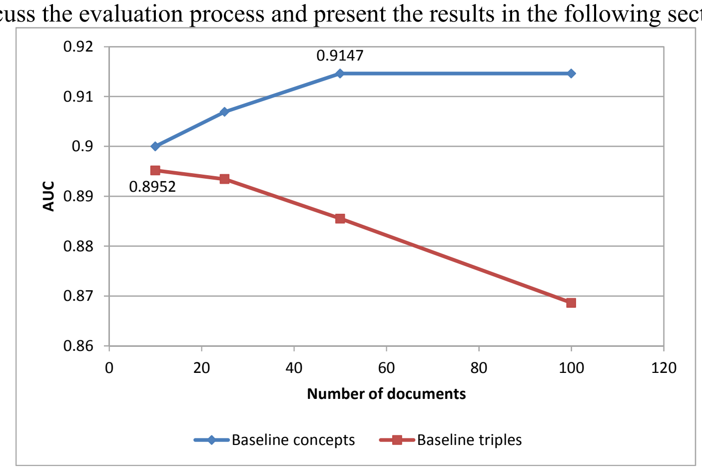

Fig. 4. Evaluation results for the baseline methods.

3.2 Evaluation of the Baseline Algorithm

In this section, we establish the baselines and determine the setting in which the baseline algorithm, presented in Section 2.2, performs best. Through the evaluation, we determined the number of search-result snippets used for grounding domain ontology entities and user queries. We experimented with 10, 25, 50, and 100 documents per grounding.

The results are shown in Fig. 4. The chart in the figure presents the evaluation results for the list of proposed concepts and the list of proposed triples. Both series show the average area under the ROC curve (y axis) with respect to the number of documents per grounding (x axis).

From the results, we can conclude that the concepts—as well as the queries when used for ranking the concepts—should be grounded with at least 50 documents (91.47% AUC). As we can see from the chart, at around 50 documents, all available useful information is already contained in the collected documents. On the other hand, the triples—as well as the queries when used for ranking the triples—should be grounded with only around 10 documents (89.52% AUC). We believe this is because the triples are more precisely defined than the concepts (i.e., the corresponding search terms contain more words), which results in a smaller number of high-quality search results.

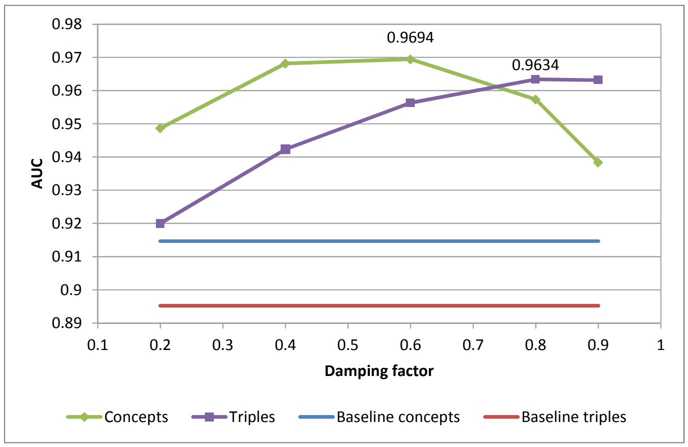

Fig. 5. Evaluation results for the graph-based methods.

3.3 Evaluation of the Graph-Based Algorithm

When evaluating the graph-based algorithm, the most important parameter to tune is the PageRank damping factor. We experimented with the damping factor values of 0.2, 0.4, 0.6, 0.8, and 0.9. The results are presented in Fig. 5. The chart in the figure presents the evaluations result for the list of proposed concepts and the list of proposed triples. Both series represent the average area under the ROC curve (y axis) with respect to the value of the damping factor (x axis). The chart also shows the baselines (see the previous section).

When evaluating the baseline algorithm, we learned that the concepts should be grounded with 50 documents each, so should the queries when used to rank the concepts. On the other hand, the triples should be grounded with only 10 documents each, so should the queries when used to rank the triples. The evaluation of the graph-based algorithms fully confirms these findings at low damping factor values. This is expected because low damping factor values mean putting less emphasis on the structure—the random walker "gets tired" after only a few steps and "jumps" back to a source vertex. However, as we increase the damping factor towards the values at which the graph-based algorithms perform best, we achieve better results when simply grounding concepts, triples, and queries with 50 documents each. Note that this grounding size setting was also used for computing the results in Fig. 5.

The damping factor should be set to 0.6 for the concepts and 0.8 for the triples. This means that we can either run PageRank twice or set the damping factor to 0.7 to increase the speed at the slight expense of quality on both sides. The rewarding fact is that we managed to significantly beat the baselines. We have increased the average AUC for 5.48% on the concepts and for 6.82% on the triples. This presents a big difference. For example, if the correct triples were distributed amongst the top 597 of 4,362 suggestions with the baseline algorithm, they will now be distributed amongst the top 319 suggestions (almost half less). Also, we believe that the user will have to inspect far less than 319 items to find the required building blocks as he will be able to interact with the system (i.e., reformulate queries to direct the search). To support this claim, we computed the average AUC by taking, for each annotation, only the most successful annotator into account (i.e., the annotator that formulated the query resulting in the highest AUC). In this modified setting, the average AUC on the triples rose to 98.15%. This reduces the number of items that need to be inspected from 319 to 161 (of 4,362). The improvement over the baseline method is also reflected in practice. Several user queries and the corresponding retrieved concepts (top 10 according to the relevance) from the SWING ontology are shown in Table 1.

Table 1. Several user queries and the corresponding retrieved concepts.

User query	Retrieved concepts
"rare birds, sequoia forests, natural parks"	ImportantBirdArea, AviFauna, Bird, NationalProgram, QuarrySite, Lake, NationalCoordinator, ProtectedArea, NaturalSite, Mammal…
"protected fauna and flora in France"	WildFauna, WildFlora, Fauna, Flora, ProtectedArea, AviFauna, Znieff [4], ZnieffTypeI, ZnieffTypeII, NaturalSite…
"locaiton[5] of open-pit mine"	AllowedMiningDepth, QuarrySite, Legislation, QuarryAdministration, QuarrySiteManagement, MineralResource, QuarryLocation, MineralProperty, ConstructionApplication, Location…

4 Conclusions

Semantic annotations are formal, machine-readable descriptions that enable efficient search and browse through resources and efficient composition and execution of Web services. Formulating annotations in one of the ontology-description languages is not a trivial task and requires specific expertise. In this paper, we presented an approach to ontology querying for the purpose of supporting the semantic annotation process. In the approach, we use the bag-of-words text representation, employ a Web search

[4] ZNIEFF stands for "Zone naturelle d'intérêt écologique, faunistique et floristique" and denotes protected natural areas in France. This example demonstrates that we are able to use descriptive queries to discover concepts labeled with acronyms.

[5] This example shows that our ontology-querying approach is even resilient to typos.

engine to ground ontology objects and user queries with documents, and run PageRank to take the ontology structure into account.

We evaluated the approach in the context of geospatial Web services. In the evaluation process, we used a set of Web Feature Service (WFS) schemas, enriched with the golden-standard annotations and user queries. In the experiments, we varied the number of grounding documents and PageRank damping factor. We concluded that it is best to ground the concepts, triples, and user queries with 50 documents each and to set the damping factor to 0.7. The achieved AUC for retrieving concepts was 96.94% and for retrieving triples, 96.34%. With these results, we managed to achieve a significant improvement over the baselines.

5 Related Work

The main contribution of this paper is a novel ontology querying algorithm. In this section, we overview several techniques that can be used to assess the relevance of an object (with respect to another object or a query) or the similarity between two objects in a network. Some of these techniques are: spreading activation [8], hubs and authorities (HITS) [9], PageRank and Personalized PageRank [6], SimRank [10], and diffusion kernels [11]. These methods are extensively used in information-retrieval systems. The general idea is to propagate “authority” from “query nodes” into the rest of the graph or heterogeneous network, assigning higher ranks to more relevant objects.

ObjectRank [12] employs global PageRank (importance) and Personalized PageRank (relevance) to enhance keyword search in databases. Specifically, the authors convert a relational database of scientific papers into a graph by constructing the data graph (interrelated instances) and the schema graph (concepts and relations). To speed up the querying process, they precompute Personalized PageRank vectors for all possible query words. HubRank [13] is an improvement of ObjectRank in terms of space and time complexity without compromising the accuracy. It examines query logs to compute several hubs for which PPVs are precomputed. In addition, instead of precomputing full-blown PPVs, they compute fingerprints [14] which are a set of Monte Carlo random walks associated with a node.

Stoyanovich et al. [15] present a ranking method called EntityAuthority which defines a graph-based data model that combines Web pages, extracted (named) entities, and ontological structure in order to improve the quality of keyword-based retrieval of either pages or entities. The authors evaluate three conceptually different methods for determining relevant pages and/or entities in such graphs. One of the methods is based on mutual reinforcement between pages and entities, while the other two approaches are based on PageRank and HITS, respectively.

Acknowledgements

This work has been partially funded by the European Commission in the context of the FP7 project ENVISION (Environmental Services Infrastructure with Ontologies) under the Grant Agreement No. 249120.

References

1. Roman, D., de Bruijn, J., Mocan, A., Lausen, H., Domingue, J., Bussler, C., Fensel, D.: WWW: WSMO, WSML, and WSMX in a Nutshell. In The Semantic Web – ASWC 2006, LNCS, vol. 4185, pp. 516–522 (2006)
2. Grcar, M., Mladenic, D.: Visual OntoBridge: Semi-Automatic Semantic Annotation Software. Lecture Notes in Computer Science, LNAI 5782, pp. 726–729 (2009)
3. Grcar, M., Klien, E., Novak, B.: Using Term-Matching Algorithms for the Annotation of Geo-Services. Knowledge Discovery Enhanced with Semantic and Social Information, vol. 220, pp. 127–143 (2009)
4. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, Cambridge, England and New York, USA (2006)
5. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Longman Publishing, Boston, USA (1989)
6. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66. Stanford InfoLab, Stanford, USA (1999)
7. Cardoso-Cachopo, A. Oliveira, A. L.: Empirical Evaluation of Centroid-Based Models for Single-Label Text Categorization. Technical Report 7/2006. Instituto Superior Tecnico, Lisbon, Portugal (2006)
8. Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval. Artificial Intelligence Review, vol. 11, pp. 453–482 (1997)
9. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. Journal of the Association for Computing Machinery, vol. 46, pp. 604–632 (1999)
10. Jeh, G., Widom, J.: SimRank: A Measure of Structural Context Similarity. Proceedings of KDD '02, pp. 538–543 (2002)
11. Kondor, R.I., Lafferty, J.: Diffusion Kernels on Graphs and Other Discrete Structures. Proceedings of ICML '02, pp. 315–322 (2002)
12. Balmin, A., Hristidis, V., Papakonstantinou, Y.: ObjectRank: Authority-based Keyword Search in Databases. Proceedings of VLDB '04, pp. 564–575 (2004)
13. Chakrabarti, S.: Dynamic Personalized PageRank in Entity-Relation Graphs. In Proceedings of WWW 2007, pp. 571–580 (2007)
14. Fogaras, D., Rácz, B.: Towards Scaling Fully Personalized PageRank. In Proceedings of the Workshop on Algorithms and Models for the Web-graph (WAW 2004), pp. 105–117 (2004)
15. Stoyanovich, J., Bedathur, S., Berberich, K., Weikum, G.: EntityAuthority: Semantically Enriched Graph-based Authority Propagation. In Proceedings of the 10th International Workshop on Web and Databases (2007)
16. Open Geospatial Consortium: Web Feature Service Implementation Specification, Version 1.0.0 (OGC Implementation Specification 02-058) (2002)
17. Mladenic, D.: Feature Subset Selection in Text Learning. In Proceedings of the 10th European Conference on Machine Learning (ECML-98), Chemnitz, Germany, 21–23 April, pp. 95–100. Springer-Verlag, Berlin (1998)