Search and Navigation in Semantically Integrated Document Collections

Saša Nešić, Fabio Crestani, Mehdi Jazayeri
Faculty of Informatics,
University of Lugano
Lugano, Switzerland
{*sasa.nesic, fabio.crestani, mehdi.jazayeri*}*@usi.ch*

Dragan Gašević
School of Computing and Information System,
Athabasca University
Athabasca, Canada
dgasevic@acm.org

***Abstract*—The paper presents a novel approach to semantic search and navigation in office-like document collections. The approach is based on a semantic document model that we have developed to enable unique identification, semantic annotation, and semantic linking of document units of office-like documents. In order to semantically annotate document units and to link semantically related document units, we first conceptualize document units' semantics and represent them by vectors of ontological concepts and their corresponding weight vectors. In the semantic search, we represent a user query by a query's concept vector, which is generated in the same way as document units' concept vectors, and then determine the search results by measuring the similarity between the query's and the document units' concept weight vectors. After the search, by following the semantic links of a selected document unit, the user can navigate through the document collection and discover semantically related document units. Results of the preliminary evaluation, conducted with a prototype implementation, are promising. We present a brief analysis of these results.**

***Keywords*-semantic search, semantic linking and navigation;**

I. INTRODUCTION

During the last decade a considerable number of ontology-driven information retrieval approaches [1], [2], [3], [4] has been developed to enhance the search and retrieval by making use of available semantic annotations and their underlining ontologies. Central to the ontology-driven information retrieval is the problem of having substantial amount of accurate semantic annotations. Most existing semantic annotation approaches [5] are based on the syntactic matching of ontological concept descriptions against document content. In spite of advanced data mining and NLP techniques applied in these approaches, usually poor and ambiguous concept descriptions lead to insufficient and inaccurate semantic annotation. Few approaches, such as [2], try to enhance the semantic annotation by extending the set of syntactic matches with related concepts from the ontology, discovered by utilizing formal ontological semantics. Such concepts are usually referred to as semantic matches. The combination of the syntactic and semantic matching can increase the amount of semantic annotations, but it opens the problem of the concept relevance [6]. Therefore, one of the most important issues in this scenario is how to assess the relevance weight of the discovered semantic matches and to use only the most relevant of them.

In this paper we present a unified solution that should enable efficient semantic search and navigation in document collections holding semantically related data. The solution is based on the novel document representation model, namely semantic document model (SDM) [7] which comprises the publishing document data in RDF, the semantic annotation and indexing of document data by weighted ontological annotations and the semantic linking of related data within the document collections. By the weighted annotations, which we calculate based on the semantic distances between concepts in the annotation ontology, we intend to improve the semantic search in document collections. By the publishing document data in RDF and the semantic linking of document data, we intend to enable client applications to easily navigate between documents and to discover semantically related data.

The rest of the paper is organized as follows. In Section II we outline main characteristics of the SDM model. In Section III we describe our approach to concepts discovery in document units, especially focusing on a novel, concept exploration algorithm that we apply in the semantic matching. Section IV explains the way we use the discovered concepts for the semantic annotation, indexing and linking of document units of a given document collection. In Section V we present the semantic search and navigation services, which utilize the semantic annotations and links to search and navigate in semantically integrated document collections. In Section VI we discuss the results of the preliminary evaluation that we conducted as a proof of concept. We conclude the paper with Section VII, giving some final remarks and discussing our plans for the future work.

II. SEMANTIC DOCUMENT MODEL

We have created a novel document representation model, namely semantic document model (SDM) [7], aiming to provide the infrastructure for the unique identification, the semantic annotation and the semantic linking of fine-grained units of document data. SDM represents document data as RDF [8] linked data, providing an RDF node for each document unit. Document units are

 ISBN: 978-1-61208-104-5

uniquely identified by the means of HTTP dereferencable URI of their RDF nodes, semantically annotated by ontological concepts from domain ontologies, and linked to other document units by RDF links that model hierarchical, structural and semantic relationships among them. SDM is formally described by the *smd ontology* [7], which specifies possible types of document units (e.g., `sdm:paragraph`, `sdm:section`, `sdm:table` and `sdm:illustration`), types of hierarchical and structural relationships among document units (e.g., `sdm:hasPart`, `sdm:isPartOf` and `sdm:belongsTo`), and the semantic annotation and the semantic linking interfaces.

The semantic annotation interface consists of the `sdm:Annotation` entity with its two properties: the `sdm:annotationConcept` property that holds a reference to the concept from an annotation ontology and the `sdm:conceptWeight` property that determines the relevance of the annotation concept for the document unit it annotates. The semantic linking interface consists of the `sdm:SemanticLink` entity and the following properties: the `sdm:unitOne` and `sdm:unitTwo`, which hold the document units to be linked, the `sdm:relationshipConcept` property that holds the reference to the ontological concept that annotates both units and determines the type of the semantic relationship, and the `sdm:linkStrength` property that determines the strength of the semantic relationship between the document units. As we can see from the specification of the semantic annotation and the semantic linking interfaces, both of them require the concepts from domain ontologies that conceptualize human-readable information stored in document units. Therefore, the concept discovery represents the foundation of the semantic annotation and linking in SDM.

III. Concepts Discovery in Document Units

The concept discovery that we propose, combines the syntactic matching of lexically expanded concept descriptions with the semantic matching by applying the concept exploration algorithm. In the rest of the section we first describe the main characteristics of the proposed syntactic and the semantic matching and then give detailed description of the concept exploration algorithm.

A. Syntactic and Semantic Matching

Any domain ontology can be represented as a graph $O := (\mathbb{C}, \mathbb{R}, H^C, H^R)$ where $\mathbb{C} = \{c_1, c_2, c_3, ..., c_n\}$ is a set of concepts, $\mathbb{R} = \{r_1, r_2, ..., r_m\}$ is a set of relations and H^C, H^R are hierarchies defining a partial order over concepts and relations respectively. Moreover, each concept is described with a set of labels. For example, the set of labels of the concept c_i is $\mathbb{L}_i = \{l_{i1}, l_{i2}, ..., l_{im}\}$. In practice, however, ontology engineers provide only one label for each ontology concept or even neglect to label concepts considering human readable parts of concept URIs to be concept labels [5]. In order to cope with this problem, which can lead to inefficient syntactic matching, prior to the syntactic matching we perform the lexical expansion of the concept descriptions with related terms from lexical dictionaries such as WordNet [9].

The objective of the syntactic matching is to analyze the content of a document unit (*DU*) and to check if some of the concept labels, appear in it. For the concepts whose labels appear in the *DU*, we calculate the concept weight by taking into account the following: 1) the labels' origin factor that makes distinction between original concept labels and those from the lexical expansion, 2) the labels' frequency of occurrence in the *DU* and 3) the inverse *DU*' frequency in a document collection. The result of the syntactic matching is the concept vector of the *DU* and the corresponding concept weight vector. For example, if we have a document unit d that is being annotated, after the syntactic matching we got the following concept vector: $\overrightarrow{d} = [c_1, c_2, ..., c_r]; \quad c_i \in \mathbb{C}$ and the corresponding concept weight vector $\overrightarrow{W_C}(d) = [w_{c_1}, w_{c_2}, ..., w_{c_r}]$, where w_{c_i} is the relevance weight of the concept c_i for the document unit d.

The objective of the semantic matching is to extend the concept vector $\overrightarrow{d}$, which is formed as a result of the syntactic matching, with semantically related concepts from the annotation ontology. By applying the concept exploration algorithm, which we explain in detail in the following section, to each of the document unit's syntactic matches, we discover the document unit's semantic matches and form the expanded concept vector $\overrightarrow{d}^e = [c_1, c_2, ..., c_r, c_{e1}, ..., c_{em}]$. For each of the semantic matches c_{ej} the algorithm calculates the semantic distance $SDist^c(c_{ej}, c_i)$ from the initial syntactic match $c_i \in \overrightarrow{d}$. The weight $w_{c_{ej}}$ of the semantic match c_{ej} for the document unit d is then calculated by the following formula:

$$w_{c_{ej}} = w_{c_i} * \beta^{-SDist^c(c_{ej}, c_i)}; \quad \beta > 1 \qquad (1)$$

where w_{c_i} is the weight of the syntactic match c_i and β is a generic coefficient. We devised the formula (1) so that it satisfies boundary conditions regardless of the value of coefficient β. For the first boundary condition $SDist^c(c_{ej}, c_i) = 0$, meaning that the concepts c_{ej} and c_i are semantically identical, $w_{c_{ej}} = w_{c_i}$, that is, the weight of the semantic match is the same as the weight of the initial syntactic match. For the second boundary condition $SDist^c(c_{ej}, c_i) \to \infty$, meaning that the concepts c_{ej} and c_i are semantically unrelated, $w_{c_{ei}} \to 0$, that is, the weight of the semantic match tends towards zero. For $SDist^c(c_{ej}, c_i) \in (0, \infty)$, the optimal value of coefficient β has to be experimentally determined. For the evaluation, which results we discuss in Section VI, we used the exponential constant e as the value of coefficient β thus making (1) belongs to the family of negative exponential functions.

 ISBN: 978-1-61208-104-5

B. Concept Exploration Algorithm

The main assumption on which the concept exploration algorithm runs is the possibility to associate numerical values to ontological relations in the annotation ontology and to form the weighted ontology graph. We refer to these values as *the relation semantic distances ($SDist^r$)*. Moreover, we distinguish between two types of the relation semantic distance: 1) $SDist^r_{\mathcal{D}\rightarrow\mathcal{R}}(r)$ determining semantic distance of the concepts belonging to the domain ($\mathcal{D}$) of the relation r from the concepts belonging to the range ($\mathcal{R}$) of r, and 2) $SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r)$ determining the semantic distance of the concepts belonging to the range of r from the concepts belonging to the domain of r. In general, the values of the relational semantic distances can be: 1) specified at design time of the ontology by the domain experts, 2) experimentally devised by using a controlled knowledge/data base and 3) learned over time by exploiting the ontology in real world applications within the ontology domain. Based on our experience the choice between these three strategies is strongly domain-dependent. A combination of the strategies is also valid.

The general idea of the algorithm (see Algorithm 1) is to explore the ontology graph starting from the input concept to find all concepts which satisfy the given semantic distance constraint (SD_c) and the given path length constraint (PL_c). SD_c is the maximum allowed semantic distance between the input and target concepts. PL_c is the maximum number of hops (i.e., ontology relations) allowed to belong to a path between the input and target concepts. The algorithm takes the following input: the weighted ontology graph O_w formed by associating values of the relation semantic distances to the ontology relations, the input concept c, the semantic distance constraint SD_c, and the path length constraint PL_c. The output consists of a vector of discovered related concepts $\overrightarrow{C_e}$ and a vector of the semantic distances $\overrightarrow{SD_e}$ between the discovered concepts and the input concept. The algorithm starts by the $Paths1(O_w, c, PL_c)$ function (line 3) which constructs a set of all possible acyclic paths $\mathbb{P}$, starting from the input concept c and whose length is less than PL_c. Next, (line 4) the $Concepts(\mathbb{P})$ function extracts all concepts from the set of paths $\mathbb{P}$ and forms a distinct set of extracted concepts $\mathbb{C}$. Next, (line 6) for each concept $c_i \in \mathbb{C}$ function $Paths2(c, c_i, \mathbb{P})$ returns a set of paths $\mathbb{P}_i$ ($\mathbb{P}_i \subseteq \mathbb{P}$) which start in concept c and end in concept c_i. Next, (line 8) for each path $p_{ij} \in \mathbb{P}_i$ between c and c_i, function $SDist^p(p_{ij})$ calculates the semantic distance of the path that we refer to as *the path semantic distance ($SDist^p$)*. The function actually sums the relation semantic distances of relations that make the path. For those relations $r_k \in p_{ij}$ with the same direction as a direction $c \rightarrow c_i$, the function takes $SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r_k)$ while for r_k with the direction $c \leftarrow c_i$, the function takes $SDist^r_{\mathcal{D}\rightarrow\mathcal{R}}(r_k)$.

Algorithm 1 Concept Exploration Algorithm

1: INPUT O_w, c, SD_c, PL_c
2: OUTPUT $\overrightarrow{C_e}$, $\overrightarrow{SD_e}$
3: $\mathbb{P} = Paths1(O_w, c, PL_c) = \{p_1, ..., p_m\}$ {finds all paths from c with a length $\leq PL_c$}
4: $\mathbb{C} = Concepts(\mathbb{P}) = \{c_1, ..., c_n\}$ {extracts all concepts from the set of paths $\mathbb{P}$}
5: **for all** c_i such that $c_i \in \mathbb{C}$ **do**
6: $\quad\mathbb{P}_i = Paths2(c, c_i, \mathbb{P}) = \{p_{i1}, ..., p_{ik}\}$ {finds a set of acyclic paths $\mathbb{P}_i \subset \mathbb{P}$ between c to c_i}
7: $\quad$**for all** p_{ij} such that $p_{ij} \in \mathbb{P}_i$ **do**
8: $\quad\quad SDist^p(p_{ij})$ {calculates the semantic distance of path p_{ij}}
9: $\quad$**end for**
10: $\quad SDist^c(c_i, c)$ {calculates the semantic distance of the concept c_i from c}
11: **end for**
12: $\overrightarrow{C_e} = [c_1, ..., c_p]$, $c_i \in \mathbb{C}$ and $SDist^c(c_i, c) \leq SD_c$
13: $\overrightarrow{SD_e} = [SDist^c(c_1, c), .., SDist^c(c_p, c)]$

After the algorithm calculates the path semantic distances of all paths $\mathbb{P}_i$, it calculates the semantic distance of concept c_i from the input concept c by applying function (2). We call this semantic distance *the concept semantic distance* ($SDist^c$). $SDist^c(c_i, c)$ can be also considered as the relation semantic distance $SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r(c, c_i))$ of a new single relation $r(c, c_i)$ from the concept c to c_i.

$$SDist^c(c_i, c) = SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r(c, c_i)) = \frac{1}{\sum_{j=1}^{k} \frac{1}{SDist^p(P_{ij})}} \quad (2)$$

We designed function (2) so that it prioritizes the impact of those paths with the small path semantic distance, in determining the concept semantic distance. Finally, the algorithm discards all concepts from the set $\mathbb{C}$ which do not satisfy the SD_c constraint, and forms the output vector of the discovered related concepts $\overrightarrow{C_e}$, along with the vector of their semantic distances $\overrightarrow{SD_e}$ from the input concept c.

IV. Semantic Documents Annotation, Linking and Indexing

The semantic annotation of document units defined by SDM refers to the process of linking the discovered ontological concepts and their weights to document units' RDF nodes, via the SDM annotation interface. If documents of a given document collection are annotated by concepts from the same, shared domain ontology, then implicit semantic relationships between their document units can easily be identified and made explicit. For example, if two document units are annotated by the same ontological concept, it means that they share some semantics and there is an implicit semantic relationship between them. By setting up explicit

RDF links between RDF nodes of document units, based on the SDM linking interface, we bring the collection's data into an integrated information space. Following the semantic links, the user can easily navigate in this information space and discover semantically related data. Moreover, by exposing the SPARQL HTTP endpoints [8] to RDF repositories of the document collections, we can enable the integration of distant document collections. In this way, the user can navigate through semantically related data belonging to different document collections as well.

Finally, besides the semantic annotation and linking, the discovered concepts are also used for the concept indexing of the document units. The concept index contains a list of concepts (i.e., concept identifiers) from the annotation ontologies, each of which is assigned a list of the document units it annotates. For each document unit in the concept's list, the index also stores the weight of the concept for the document unit. The concept index plays the key role in the semantic search that we discuss in the next section.

V. Semantic Search and Navigation

In this section we describe the semantic search and the semantic navigation services, which we have developed as parts of a broader, service oriented architecture called the Semantic Document Architecture - SDArch [10]. These two services provide the mechanisms for the semantic search and navigation in the collections of documents represented by SDM. In order to provide the user interface to the SDArch services, we have developed a set of tools called 'SemanticDoc' and integrated them into MS Office as add-ins. Further information about the SDArch services and SemanticDoc tools can be found on our project web page [11].

The search process normally starts with the user constructing a query that reflects her information needs. As the initial form of the user query, the semantic search service takes a free text query. The service then models the semantic meaning of the query by forming a weighted query concept vector, which we refer to as a semantic query. The search service actually applies the syntactic and the semantic matching to discover ontological concepts, which conceptualize the semantic meaning of the query. For each discovered concept, the service calculates its relevance weight to the query and forms a weighted query concept vector. The way the search service forms semantic queries is quite similar to the process of the concepts discovery (Section III) in document units.

Having both the document units and the user query represented in the same way, by their weighted concept vectors, the rest of the search process proceeds as follows. From the concept index of the selected document collection, the search service discovers document units which are indexed by the concepts from the semantic query. Then, the service calculates the similarity between the discovered document units and the semantic query, by computing the similarity between the document units' concept vectors and the query's concept vector. At the end, the service ranks the document units based on the calculated similarity and retrieves the ranked list of the document units.

The semantic navigation service enables users to traverse document collections by navigating along the semantic links. The navigation process assumes the existence of an exploratory interface through which the users interact with the semantic navigation service (i.e., SemanticDoc [11] tools). The navigation process starts by the user browsing the initial document unit and clicking on one of the unit's annotation concepts (i.e., concept label). This activates the navigation service, which takes the URIs of the document unit and the concept, forms a SPARQL query (see Fig. 1), and executes it against the collection's RDF repository. Since the initial document unit can be linked to many document units via the same semantic link, thus the query can return multiple document units. The query orders the return document units by the strength of the semantic links between them and the initial document unit. After the query execution, the navigation service sends the list of the document units to the browse in which the user browses their details.

```
PREFIX sdm: <http://www.semanticdoc.org/sdm.owl#>
SELECT ?targetUnit ?strenght
WHERE{?link sdm:relationshipConcept concept_a32c154
      ?link sdm:unitOne unit_b42c177
      ?link sdm:unitTwo ?targetUnit
      ?link sdm:linkStrength ?strength }
ORDER BY ?strength
```

Figure 1. Example of a SPARQL query executed by the navigation service

VI. Evaluation

In order to evaluate the usability of the semantic search and navigation services we conducted a usability study in which we considered the user effectiveness, efficiency and satisfaction in using the services, while preparing a course material. Both, the quantitative measures (e.g., task execution time, number of window switches and number of mouse clicks) and the users' subjective feedback, collected by the evaluation questionnaire and the series of interviews, showed positive results. The detailed discussion on the usability study can be found in [12]. In this paper, however, our focus is on the experimental evaluation of the semantic annotation (i.e., the concept discovery) and the semantic search.

The experimental evaluation that we discuss hereafter, was designed more as a proof of concept; it was not meant to address issues of scalability or efficiency. The document collection that we used in the experiments was composed of 170 Word documents (2735 paragraphs - document units of interest for these experiments) containing records for steel, aluminum, copper, titanium, and other metals. We optioned the collection from KEY-to-METALS [13] company, which

 ISBN: 978-1-61208-104-5

Semantic relation	Representation	$SDist^r_{\mathcal{R}\to\mathcal{D}}(r)$	$SDist^r_{\mathcal{D}\to\mathcal{R}}(r)$
hypernym	$skos : broader$	$1-\delta_{hyper}=0,53$	$1-\delta_{hypo}=0,16$
hyponym	$skos : narrower$	$1-\delta_{hypo}=0,16$	$1-\delta_{hyper}=0,53$
holonym	$skos : relatedPartOf$	$1-\delta_{holo}=0,88$	$1-\delta_{mero}=0,84$
meronym	$skos : relatedHasPart$	$1-\delta_{mero}=0,84$	$1-\delta_{holo}=0,88$
synonym	$owl : equivalentClass$	$1-\delta_{syn}=0,30$	$1-\delta_{syn}=0,30$
identical	$owl : sameAs$	0	0

Table I
RELATION SEMANTIC DISTANCES IN METALS ONTOLOGY

Strategy	Number of concepts	Number of syn. matches	Number of sem. matches	Avg. weight of syn. match.	Avg. weight of sem. match.
S_1	211	1524	-	2.56	-
S_2	343	3182	-	3.62	-
S_3	672	3182	6714	3.62	2.43
S_4	795	3182	11102	3.62	1.12
S_5	924	3182	23716	3.62	0.27

Table II
CONCEPT DISCOVERY RESULTS FOR STRATEGIES (S_1-S_5)

maintains one of the world's most comprehensive metals database. As the annotation ontology we used the *Metals* ontology, which we also got from the same company. The ontology contains over $3,500$ concepts about metals and their applications. It is an OWL ontology which conforms to the SKOS specification [6]. SKOS defines a family of relations such as $skos : narrower$, $skos : broader$ and $skos : related$ for expressing simple relationships between concepts within an ontology.

Table I shows a subset of semantic relations in the *Metals* ontology, along with their SKOS and OWL representations and values of the relation semantic distances. The values of the relation semantic distances were assessed based on the results of the experimental studies [14]. In these studies the authors measured the semantic similarity/relatedness between terms in WordNet, connected via the *hypernymy*, *hyponymy*, *holonymy*, *meronymy* and *synonymy* relations, and produced the following values: $\delta_{hyper}=0.47$, $\delta_{hypo}=0.84$, $\delta_{holo}=0.12$, $\delta_{mero}=0.16$ and $\delta_{syn}=0.70$. Value $\delta_r=0$ means that two terms are semantically unrelated via relation r, and $\delta_r=1$ that the terms are semantically identical. We calculate the values of the relation semantic distances as $1-\delta_r$ and take into account the fact that *hypernymy* and *hyponymy* as well as *holonymy* and *meronymy* are mutually inverse relations. Moreover, the *Metals* ontology contains the $owl : sameAs$ relation which links two semantically identical concepts/individuals, so that both of the relation semantic distances have been assessed as zero.

In order to evaluate the semantic annotation, we have transformed the document collection by applying five different concept discovery strategies: S_1 - simple syntactic matching, S_2 - lexically expanded syntactic matching, and S_3, S_4, S_5 - lexically expanded syntactic matching and the semantic matching with $SD_C = 1, 2, 3$ respectively. The last three strategies comprise all the features (i.e., lexical expansion, syntactic matching and semantic matching) of our concept discovery approach. They only differ in the value of the SD_C (semantic distance constraint) parameter of the concept exploration algorithm (Section III-B). The value of the path length constraint is fixed at $PL_c=3$ for these evaluation tests.

As a result of the transformation we obtained five semantic document collections, each of which having the corresponding concept index. Table II shows for each of the concept discovery strategy: 1) the distinct number of concepts from the annotation ontology that have been involved, 2) the total number of syntactic and semantic matches, that is, the number of document units in which the concepts have been discovered by the syntactic and the semantic matching respectively and 3) the average weights of the syntactic and semantic matches calculated based on 20 randomly chosen document units. Comparing results of S_1 and S_2 which both implement only syntactic matching, we can see that the lexical expansion of concept descriptions increases the number of discovered concepts from 211 to 343 and the total number of syntactic matches from 1524 to 3182 but also the average weight of syntactic matches from 2.56 to 3.62. In other words, these increases show that the lexical expansion improves both the quantity and quality of the annotation. The next three strategies $S_3 - S_5$ produce the same number of syntactic matches as S_2 (i.e., 3182), since the syntactic matching stays intact, but they increase the number of the semantic matches (i.e., 6714; 11102; 23716). On contrary, the average weight of the added semantic matches decreases (i.e., 2.43; 1.12; 0.27). This shows that with higher values of the semantic distance constraint (SD_c) we can get more, but less relevant semantic matches.

To evaluate the performance of the proposed semantic search we formed five queries related to the data of the evaluation document collection and asked three KEY-to-

 ISBN: 978-1-61208-104-5

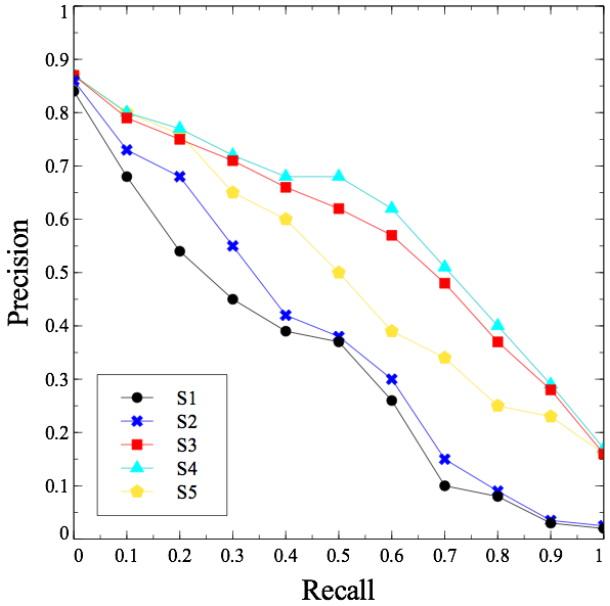

Figure 2. Interpolated precision of S_1-S_5 at standard recall points

METALS engineers to assess the relevance of document units (i.e., paragraphs) of the collection to the queries. The queries were then executed against each of the five semantic document collections. Fig. 2 shows interpolated precision at standard recall points. Comparing the P-R curves of S_1 and S_2 we can see that the lexically expanded syntactic matching outperforms from the simple syntactic matching in both recall and precision. Moreover, all three strategies (i.e, S_3, S_4, S_5) which include the semantic matching, further increase overall precision and recall. Comparing their P-R curves and by knowing that they differ only in the value of the semantic distance constraint (i.e., $SD_c = 1$, $SD_c = 2$, $SD_c = 3$) we can observe that there is an optimal value for the concept semantic distance ($SDist^c$) with regard to optimal precision and recall. It means that the semantic matches, which concept semantic distance is higher than the optimal one, reduce retrieval performances. In our evaluation the optimal value of the concept semantic distance falls in a range between 2 and 3, since the precision of S_4 is higher than of S_3 but then it drops for S_5.

The results of the preliminary evaluation indicate that the proposed concept discovery approach has potential to enlarge the amount of semantic annotations and to improve the performances of DU_s search and retrieval, not just in terms of better recall, but also in terms of better precision.

VII. Conclusions

In this paper we present an ontology-driven approach to semantic search and navigation in semantically integrated document collections. The semantic integration of document collections is achieved by the novel semantic document representation that comprises the publishing document data in RDF, the semantic annotation and indexing of document data with weighted ontological annotations and the semantic linking of related data. The results of both, the usability study and the experimental evaluation of the semantic annotation and search are promising. In the future work, we plan to continue with the evaluation of our approach, addressing issues such as the scalability, efficiency and applicability of the approach to document collections of different domains.

References

[1] J. F. Song, W. M. Zhang, W. Xiao, G. hui Li, and Z. ning Xu, "Ontology-Based Information Retrieval Model for the Semantic Web," in *Proc. of the Int. Conf. on e-Technology, e-Commerce and e-Service*, 2005, pp. 152–155.

[2] C. Rocha, D. Schwabe, and M. P. de Aragão, "A hybrid approach for searching in the semantic web," in *Proc. of the 13th international conference on World Wide Web*, 2004, pp. 374–383.

[3] A. Kiryakov, B. Popov, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *J. Web Sem.*, vol. 2, no. 1, pp. 49–79, 2004.

[4] D. Vallet, M. Fernández, and P. Castells, "An Ontology-Based Information Retrieval Model," in *Proc. of the ESWC*, 2005, pp. 455–470.

[5] V. Uren and et al., "Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art," *J. Web Sem.*, vol. 4, no. 1, pp. 14–28, 2006.

[6] J. Tuominen, M. Frosterus, and E. Hyvönen, "ONKI SKOS Server for Publishing and Utilizing SKOS Vocabularies and Ontologies as Services," in *Proc. of the European Semantic Web Conference*, 2009, pp. 768–780.

[7] S. Nešić, "Semantic Document Model to Enhance Data and Knowledge Interoperability," *Annals of Information Systems, Springer*, vol. 6, pp. 135–162, 2009.

[8] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The story so far," *Int. Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[9] WordNet, "http://wordnet.princeton.edu/."

[10] S. Nešić, M. Jazayeri, and D. Gašević, "Semantic Document Architecture for Desktop Data Integration and Management," in *Proc. of the 22nd International Conference on Software Engineering and Knowledge Engineering*, 2010, pp. 73–78.

[11] Semantic Documents, "http://www.semanticdoc.org/."

[12] S. Nešić, M. Jazayeri, M. Landoni, and D. Gašević, "Using semantic documents and social networking in authoring of course material: An empirical study," in *Proc. of the 10th IEEE International Conference on Advanced Learning Technologies*, 2010, pp. 666–670.

[13] KeyToMetals, "http://www.keytometals.com/."

[14] Z. Gong, C. W. Cheang, and L. H. U, "Multi-term Web Query Expansion Using WordNet," in *Proc. of the 17th Database and Expert Systems Applications Conference, DEXA*, 2006, pp. 379–388.

 ISBN: 978-1-61208-104-5