A Fast and Simple Path Index Based Retrieval Approach for Graph Based Semantic Descriptions

Mathias Lux, Michael Granitzer

[1]Institute for Knowledge Management and Visualization
Graz University of Technology and
Know-Center Graz, Austria
mathias.lux@tugraz.at

Know-Center Graz, Austria
[2]Competence Centre for Knowledge Based Applications and Systems R&D
mgrani@know-center.at

Abstract. The Semantic Web is a quite controversial concept, which is discussed extensively. Nevertheless most discussions showed that ways handling semantic metadata are needed urgently. Semantic metadata allows the storage of information which not only includes a set of concepts, but also the relations between the concepts in computable way. Within this paper an indexing and retrieval technique for semantic metadata is presented. The paper includes the discussion of a graph based data model for MPEG-7 based semantic metadata and an indexing technique for this model is outlined. Details on the implementation are given and a preliminary evaluation of the retrieval performance is included. As a last chapter related work is compared to our approach and pointers to related projects are given.

Introduction

While low level metadata of multimedia documents can be extracted automatically, high level and semantic metadata has to be created manually or semi-automatically. Current research projects aim to optimize the extraction of high level metadata and so to bridge the *semantic gap*, which identifies the inability of machines to gain understanding of the meaning of the data without human help (see also [DelBimbo1999], [Smeulders2000]). At present, semantic descriptions have to be created, at least in parts, manually. Human computer interaction (HCI) methods and information retrieval methods exist, that support the user in the annotation task. Different formats and approaches for creation and storage of semantic descriptions are currently discussed and in use, wherefrom MPEG-7 is one of them.

Given that the high level descriptions already exist, retrieval mechanisms for searching and browsing the multimedia content based on the high level descriptions have to be found. In case of textual descriptions and keywords the task can be simplified to text and keyword retrieval. Semantic descriptions, as they are used in

this paper, consist of concepts and relations, which allow the expression of complex issues like for instance the fact “Mathias Lux is in Graz, which is in Austria” in a machine interpretable manner, leaving no room for misinterpretation. Semantic descriptions in our understanding belong to the area of knowledge representation and ontologies, which is described in short in the context of the Semantic Web in [Daconta 2003].

MPEG-7, called the *Multimedia Content Description Interface*, is, unlike MPEG-1, 2 and 4, no video or audio coding standard but a XML based standardized way to store annotations for multimedia documents (see e.g. [Martinez 2003]). MPEG-7 documents are built from descriptors, which are organized in descriptor schemes. One specific part of MPEG-7, the *Semantic Descriptor Scheme* (Semantic DS), allows the creation of semantic annotations based on a standardized and extendable ontology for annotation of multimedia content. Within the Semantic DS different *Semantic Descriptors* (Semantic Ds) are specified, which represent for instance agents, locations, events, time points or periods or concept. These Semantic Ds are interconnected pair wise with *Semantic Relations*, which are defined within the MPEG-7 standard (e.g. agentOf, patientOf, locationOf, etc.). The Semantic DS allows describing e.g. the content of a scene or an image like specifying: “The location of Mathias Lux is Graz, which in Austria”. An example of a visual representation of a semantic description is given in Fig. 4. MPEG-7 can be expressed either in XML or a binary representation optimized for transmission and storage. A comprehensive description of the Semantic DS and its usage within an application are given in [Martinez 2003]. The MPEG-7 standard and its applications are described in detail in [Kosch 2003].

While the MPEG-7 standard defines how semantic descriptions have to be stored and coded and how these definitions can be adopted and extended, it fails to define how to retrieve multimedia content using these semantic descriptions. Applying Semantic Web[1] ideas and methods the MPEG-7 based semantic annotations can be converted to RDF[2] (see [Hunter 2001] for details) and stored within a triple database, which allows querying the semantic descriptions with a RDF specific query language like SPARQL[3]. This allows data retrieval of the semantic descriptions (For more information on the Semantic Web see [Daconta 2003]). To realize information retrieval on semantic descriptions, featuring for instance retrieval and ranking of partially matching semantic annotation, above method is not applicable (for a detailed discussion on differences between information and data retrieval see [Baeza-Yates 1999]). Although with SPARQL a precise query using a complex query language can be defined, partial matches or result sorting based on relevance functions, which allow the ranking of the results of the query, are not supported. Ongoing work on searching ontologies and RDF based information is described at the end of this paper.

Within this paper an approach for an extended retrieval mechanism, based on standard information retrieval techniques, for MPEG-7 based semantic descriptions is given. The approach has been implemented within a prototype and extends a previous approach (see [Lux 2005]) in precision and recall. In the next chapter the method is

[1] http://www.w3.org/2001/sw/
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/

described. The proposed method has been implemented within an open source project[4], details are given in section 3. The evaluation in section 4 is followed by a conclusion in section 5. An outlook on future developments and research task is given to close the paper.

Architecture

The main goal of this approach is to overcome the restrictions of inference used for searching semantic metadata by merging graph matching methods with current state of the art text methods. Thereby the benefits of semantic enriched descriptions and the performance of current retrieval information retrieval methods is combined. TF*IDF, which is short for Term Frequency - Inverse Document Frequency, is a method for term weighting in text document indices to enhance retrieval performance while inverted lists or files allow fast term based retrieval of text documents (see [Baeza-Yates 1999] or chapter 14 in [Hand 2001] for details).

The input for the retrieval process is a semantic description, given by the user. The output lists all relevant semantic descriptions in the database sorted by their relevance compared to the query. To achieve these goals a mathematical and data model for the semantic descriptions had been built and a retrieval strategy had been created.

The Model of the MPEG-7 Semantic Description Scheme

All semantic descriptions consist of nodes, which are semantic descriptors extended from the semantic base descriptor, and relations, which interconnect two different nodes. The MPEG-7 Semantic DS can be seen as directed graph, whereas the nodes are the vertices and the relations are the directed edges. The graph is not necessarily connected, as relations are not mandatory. As all the nodes and relations are identified by descriptors, a semantic description is a labeled graph, whereas the MPEG-7 descriptors are the labels for the edges and vertices. Screenshots of visual representations are given in Figure 1 and Figure 5.

For the definitions of graphs, directed graphs (digraphs) and labelled graphs see [Diestel 2000] or [Tittmann 2003]. For the sake of simplicity two nodes cannot be connected through two different relations and a relation cannot have one single node as start and end node. In graphs based on semantic DS no two nodes can have the same label (the same descriptor), so the node labels are unique. Each directed edge can be inverted as there exists an inverse of each MPEG-7 based semantic relation.

[4] Caliph & Emir is available at sourceforge.net: http://caliph-emir.sourceforge.net

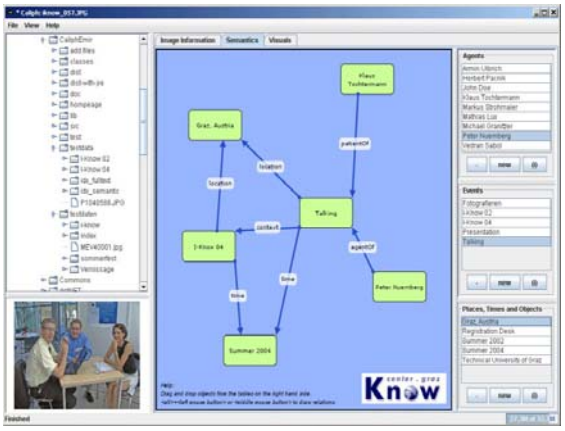

Fig. 1. Example of the representation of a MPEG-7 based semantic description. The green boxes represent the semantic objects, while the blue labeled arrows symbolize the relations of the description.

The Term Space for the MPEG-7 Semantic DS

Based on the identified constraints of a directed labelled graph with unique node labels and edges, that can be inverted, an efficient retrieval strategy can be designed as follows: The idea of fast retrieval of graph based structures is not new, as the contribution of Simmons in 1966 (see [Simmons 1966]) shows. Common retrieval techniques for graphs are the usage of metrics like the maximum common subgraph metric, or the graph edit distance. A straight forward implementation using this distance measures results in search time *O(n)*, whereas *n* defines the number of graphs in the database. Please note that the distance or similarity calculation between two graphs is NP-hard in respect to the number of nodes and edges of the graphs to compare (see e.g. [Valiente 2002]). Another approach is the filtering of the database with a fast (less than linear search time) algorithm and the ranking of the results with a slower metric which is described in chapter 12 in [Baeza-Yates 1999]. This method has been successfully used for graphs e.g. in [Fonseca 2004] for clipart retrieval by using graph eigenvalues as filters like in [Shokoufandeh 1999]. A more promising approach for MPEG-7 semantic DS, if the usage of an existing text search engine is constraint, is the usage of a path index (see e.g. the description of GraphGrep or the Daylight database in [Shasha 2002]). A path index allows the fast retrieval of graphs based on paths (sequences of nodes connected by edges, whereas the number of edges defines the length o the path) of different lengths extracted from the graphs. The extracted paths can be interpreted as index terms for a graph.

The graph can be expressed using all paths, which are sequences of nodes interconnected by edges, of chosen length. Paths of length 0 are the nodes themselves while paths of length 1 are triples as used in RDF. Paths of length 0 and length 1 have unique string representations for MPEG-7 based semantic descriptions as shown in

[Lux 2005]. To allow the usage of wildcard nodes at least the paths of length 2 have to be used, for which a unique string representation can be defined as shown below. The graph can be stored using the paths of length 0, 1 and 2 as index terms. Using a query graph all paths of the query graph are extracted and used as search terms. The ranking is done by TF*IDF on the index terms, which are the paths of the graphs.

Implementation

For the implementation an existing open source tool, called Emir (from Experimental Metadata based Image Retrieval), has been extended. Together with the annotation tool Caliph (Common And LIght-weight PHoto annotation) Emir allows the organisation, annotation and retrieval of digital photos based on MPEG-7 documents. Emir features content based and keyword based retrieval of images annotated by Caliph. Based on the open source retrieval engine *Lucene*[5] an index for node descriptors has been implemented in a previous project (see [Lux 2005]), where the string representations of paths of length 0 and 1 also have been implemented. As the query graph can consist of query strings for the node values, query expansion based on the node descriptors is used as described in [Lux 2005]. All path representations are constructed from node IDs, which identify a unique node descriptor in the index, and relation names or wildcards for nodes or relations. For the usage for terms within Lucene the path representations were adopted: all white spaces were replaced by '_' and all paths start with a leading '_'. The leading '_' allows the usage of wildcards at the start of a path expression.

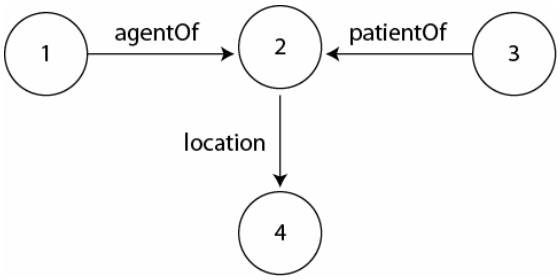

Fig. 2. Example for a graph following the model of MPEG-7 semantic DS graph. Node descriptors are already substituted with node IDs. Only the node IDs, which point to the node descriptors, are shown.

For the graph given in Figure 2 the terms for paths of length 0 and 1 would for example be:

[5] http://lucene.apache.org

Table 1. Extracted path terms of length 0 and 1 from graph shown in Fig 1. Note that the path _locationOf_4_2 has been inverted. This is done to *normalize* the edge directions in the index.

Term	Path length
_1	0
_2	0
_3	0
_4	0
_agentOf_1_2	1
_locationOf_4_2	1
_patientOf_3_2	1

For the creation of terms from paths of length 2 following method has been introduced: The input of the method is either a graph representing a semantic DS or a query graph. In a first step all paths of length 2 are extracted from the graph (see [Valiente 2002] for details on the algorithms). For each of these extracted paths the unique string representation has to be created as follows:

1. Compare the start node of the path with the end node of the path
2. If the start node is bigger than the end node reverse the path:
    a. Switch end and start node
    b. Switch and invert first and second relation
3. Create string in order: start node – first relation – middle node – second relation – end node with '_' as separator.
4. Prepend '_' to the string.

This results for the graph shown in Fig. 1 in following additional path terms:

Table 2. This table shows all available extracted path terms of length 2 from the graph shown in Figure 2.

Term	Path length
_1_agentOf_2_patient_3	2
_1_agentOf_2_location_4	2
_3_patientOf_2_location_4	2

All these above shown terms are used to index the semantic description with Lucene, all terms are used as Lucene tokens without stemming or other pre-processing. For queries the terms are constructed in a similar manner with one exception: Wildcards for nodes and relations can be used. For relations the adoption is straightforward: As Lucene supports wildcard queries for a wildcard relation the String '*' is inserted instead of the relation name, e.g. _*_1_2 instead of _agentOf_1_2. To support undirected wildcard relations two relation query terms are constructed and combined with a Boolean OR, like (_*_1_2 OR _*_2_1) . For paths of length 2 only the '*' is inserted instead of the relation name as the order of the path only depends on the start and end node.

For nodes in paths of length 0 the query string is omitted. For paths of length 1 and middle nodes in paths of length 2 the node ID is replaced with a '*'. For start and end

nodes in paths of length 2 a Boolean query clause has to be constructed as the order of the start and end node cannot be used to identify the term representation, e.g. (_*_patientOf_2_location_4 OR 4_locationOf_2_patient_*). Note that the relations have to be inverted in this case.

A simple example for a wildcard query would be: "Find all semantic descriptions where *Mathias Lux* is *doing something* at the *I-Know*". In a first step possible candidates for nodes are identified to construct the query graphs. Assuming that for *Mathias Lux* the node with ID 28 has been found, while for *I-Know* the node with ID 93 has been found, the query graph would look like "[28] [93] [*] [agentOf 1 3] [locationOf 3 2]". The numbers within the relations reference the node using their position in the node list. Such a query would result in a query like "_28 _93 _agentOf_28_* _locationOf_*_93 _28_agentOf_*_locationOf_93".

Note that arbitrary methods, which are not restricted to text retrieval, could be used to identify candidate nodes in this approach. Methods different from term based text retrieval were not implemented in this state of the project but possible mechanisms include multimedia retrieval like content based image retrieval, or Latent Semantic Indexing.

Evaluation

The data repository, which was used for evaluation, consists of 85 different semantic descriptions of digital photos from two different scientific conferences, the I-Know 02 and the I-Know 04.

Table 3. This table summarizes the size of the graphs of the test data set. In the description 46 different semantic objects (names, locations, events, etc.) were used.

	Min	**Max**	**Median**
Nodes	3	11	5.5
Relations	2	12	5.6

All descriptions consist of a minimum of 3 nodes up to 11 nodes with a median of 5.5 nodes and 2 to 12 relations with a median of 5.6. Each of these descriptions was taken as query input for the evaluation and the average precision at 11 standard recall levels was calculated. The test set was generated by taking the query graph, and ranking all graphs based on the maximum common subgraph distance. The maximum common subgraph distance, formally defined and proved as metric in [Bunke 1998], was chosen, because it is a metric that takes structure as well as content into account and it is a good representative for a group of similar metrics including the graph edit distance and the minimum common supergraph metric The main idea of this metric is to compare the size of the maximum common subgraph *mcs(G1, G2)* to the maximum of the size of the Graphs *G1* and *G2* as shown in (1) (see also [Bunke 1997]).

$$similarity(G_1, G_2) = \frac{|mcs(G_1, G_2)|}{\max(|G_1|, |G_2|)} \quad (1)$$

The best ten ranked documents made up the test set and precision and recall for the query results based on the path index and the full text index were calculated. For the full text index search the node IDs were converted back to the node labels, which were concatenated to a query string. The full text index itself contains all non-tag strings of the MPEG-7 XML document, including all semantic object labels, descriptions, etc.

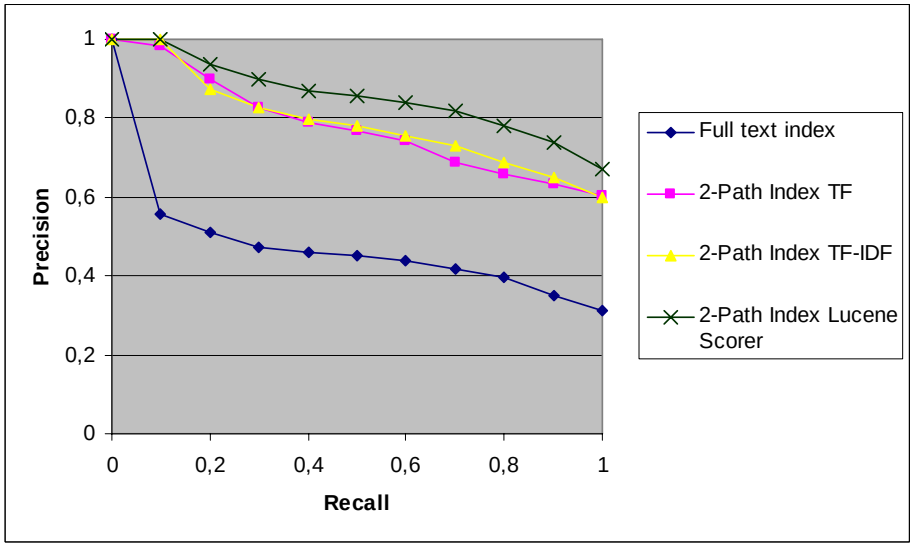

Fig. 3. Average precision at 11 standard recall levels using a test set generated with the maximum common subgraph metric. The lowest graph shows the performance of the full text search compared maximum common subgraph metric, while the other three graphs show the performance of path index based method using different term weighting schemes.

As can be seen easily the usage of the path index gives a better retrieval performance compared to the full text index for a test set generated with the maximum common subgraph distance. This can be easily explained as the full text index cannot take the structure of the semantic descriptions with its relations and paths into account, while the path index, to a limited account, can do this.

Experiments with different relevance functions, which are called “scoring functions” in Lucene, revealed that there is an obvious difference between the built-in Lucene Scorer and a straight forward classical TF*IDF implementation.

$$score(q,d) = \sum_{t \in q} TF(t,d) \cdot IDF(t) \cdot b(t.field,d) \cdot lNorm(t.field,d) \cdot coord(q,d) \cdot qNorm(q) \quad (2)$$

The Lucene scoring function, shown in (2), can be explained as follows: *TF(t, d)* calculates the term frequency of term *t* in document *d* by using the square root of the term count, *IDF(t)* the inverse document frequency of the term defined by *log(numDocs/(docFreq+1)) + 1*. Function *b(t.field, d)* takes the boost factor for the specific field into account, while *lNorm(t.field, d)* penalizes longer fields with much content. With the function *coord(q, d)* documents matching more terms of query *q* are scored higher, while *qNorm(q)* tries to normalize the score based on the length of the query.

The tested term weighting schemes used in the valuation were taken from the Lucene implementation. For the classical TF*IDF weighting the score function shown in (2) was modified by removing all factors inside the sum apart from *TF(t, d)* and *IDF(t)*, for the TF weigthing scheme the *IDF(t)* factor was removed too.

The Lucene scoring function outperforms the classical TF*IDF implementation and the term frequency scoring function. We assume that the *coord(q, d)* factor is the reason for the different performance of the classical TF*IDF and the Lucene score function by reflecting the denominator of the maximum common distance metric. Between classical TF*IDF implementation and the term frequency scoring function only slight differences in retrieval performance can be identified.

Conclusion

Comparing the path index with the full text index based approach and taking a look at the precision and recall it can be easily seen, that the resulting relevance function is more similar to the maximum common subgraph metric than the relevance function based on terms in the full text index, which allows the retrieval of descriptions that have a common sub-description with the query description. The adopted TF*IDF based scoring function of Lucene proves as best choice for calculating relevance of matching documents. The approach based on triples of 1-path indices stored in a file for retrieval with regular expressions, as presented in [Lux 2005], has in comparison to the method presented in this paper only limited use. While the 2-path index based technique allows the retrieval of model graphs, which have a common subgraph with the query graph, the approach in [Lux 2005] only allows retrieval of model graphs, where the query is a subgraph of the model graph.

One of the benefits of the presented method is the expected increased runtime performance. This is because of the term index based retrieval compared to the linear search although the worst case, where a term (or path in this case) is part of each graph in the database, also has runtime linear in the number of graphs. Linear search using the maximum common subgraph metric is rather slow as the problem of subgraph isomorphism is in NP (see e.g. [Valiente 2002]). Additional benefit is the increased focus on the graph structure of the semantic descriptors, which increases the

precision compared to the full text index approach. The support for precise search with wildcard nodes is limited to one wildcard node in a 2-path:

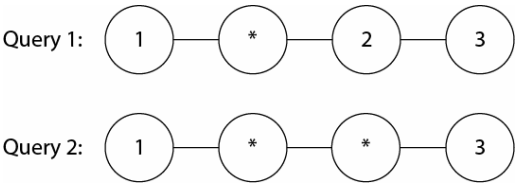

Fig. 4. Two examples for a graphical query, where '*' denotes a wildcard node. In Query 1 there is only one wildcard, while in Query 2 two wildcards in a sequence occur.

Following example shows the problem with two wildcards in a row based on the queries shown in Figure 4: For Query 1 in the retrieval of structures similar to the one shown above can be guaranteed, as the constructed query terms "_1_relation_*_relation_2" and "_*_relation_2_relation_3" define Query 1 unmistakeable. For Query 2 the terms "_1_relation_*_relation_*" and "_*_relation_*_relation_3" could also match a graph with only three nodes like the one identified by term "_1_relation_2_relation_3".

Due to the usage of Lucene as search engine the retrieval is fast and stable and the implementation effort could be minimized as TF*IDF and inverted lists were not implemented for this project.

The usage of the query expansion mechanism for generation of the query graphs resulted in performance problems with very unspecific queries as reported in [Lux 2005]. Within this extension a solution based on query refinement was integrated: For each node defined by a query string a list of possible candidate nodes is shown. A selection of one candidate node is possible to reduce the number of expanded query graphs.

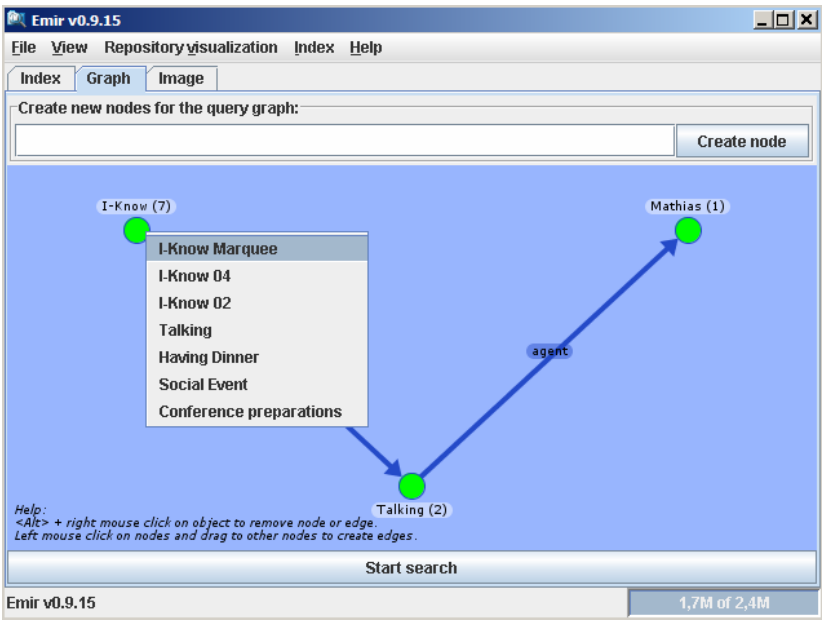

Fig. 5. Integration of the selection of possible candidates for a node to reduce the number of expanded query graphs. With a mouse click a context menu appears which allows the selection of a specific node.

While the original query of Figure 5 would result after expansion in 14 query graphs, a selection of one of the nodes in the context menu would reduce the number of query graphs to 2.

Future Work

Above evaluation has shown that the usage of a 2-path index improves retrieval performance. Another interesting aspect to evaluate is the general performance for differing sizes of paths, longer than 2 edges. Another evaluation task will be to create test sets based on wildcard node queries and define result sets with an error correcting maximum common subgraph (mcs) distance measure as described in [Berretti 2004]. This error correcting maximum common subgraph distance allows the integration of node and edge distances and optimizes the distance between two graphs to a minimum. This method is currently used in the area of content based image retrieval, but can be adapted to the retrieval of semantic metadata easily provided that node and edge distance functions exist.

The technique introduced within this paper can also be applied to other graph based representations of semantic information like it is done in the Semantic Web

standards RDF and OWL[6]. Further adoptions to standards and scenarios that aren't based on MPEG-7 are currently issue of research.

Another promising idea is the usage of the suffix tree retrieval model (see [Meyer zu Eissen 2005] for details) instead of the vector space model. Retrieval with the suffix tree model using the paths of a graph would allow the efficient search for graphs, which have paths in common. This allows the combination of content and structure, like it is done by sophisticated graph matching algorithms like error correcting maximum common subgraph distance of Berretti et. al.

State of the Art Graph based and Semantic Web Retrieval

In contrast to the presented approach inference has been identified in [Carroll 2002] as main tool for retrieval in the Semantic Web. Examples for inference based queries are "Show me every photo showing a person is living in Graz" or "Play a rock concert recorded in Graz, where at least one rock band member has long hair". All facts are investigated and all matching true facts fulfilling all given constraints are returned. This is a Boolean concept as there is no grade of falseness or truth. Although it was tried to show that distance calculation between graphs based on graph isomorphism, which is also called graph matching, does not prove useful for RDF, recent work shows possible applications for MPEG-7 (see Lux and Granitzer in [Lux 2005]) and conceptual graphs (see [Zhong 2002], [Zhu 2002] and [Yang 1993]). As Corby, Dieng and Hubert showed in [Corby 2000] that there is a bijective transformation between conceptual graphs and the RDF/XML serialization of the RDF Model the results of Zhong et. al. can be applied to RDF as well. Arguments against inference in semantic search engines are summarized in [Alesso 2004]. Main points of critique are the incompleteness and the halting problem for large ontologies.

Search engines for the Semantic Web, which do not rely on inference, have been already created using different approaches. In [Rocha 2004] a retrieval system, which allows the retrieval of resources in an ontology by spreading activation, was presented. The main flaw of this approach is that an overall ontology has to be built, which implies that many small heterogeneous ontologies have to be mediated. The OntoSeek system, described in [Guarino 1999], implements query expansion of search queries based on ontologies. The actual retrieval process does no take relations into account, but relies on terms identifying concepts. The ranking of the results is done using a graph similarity metric. The metadata search engine Swoogle (see [Ding 2004]) harvests and indexes RDF based metadata and ontologies and allows retrieval based on the literals used in the indexed RDF data. Ranking of the results is based on the in and out degrees of the found nodes and concepts, similar to the PageRank algorithm (for details on PageRank see e.g. [Rogers 2002]).

In [Stojanovic 2003] a ranking mechanism for matching expressions resulting from inference is given. The query is formulated in an ontology query language, matching instances within an ontology driven knowledge base are returned. The approach is based on node distance in hierarchical structures or in other words on a tree edit

[6] http://www.w3.org/TR/owl-features/

distance measure. To create a query a user has to acquire knowledge upon the underlying ontology. As though this approach could be used to solve a part of the problems solved with our method several differences can be stated: In comparison to the technique presented in this paper the query formulation heavily depends on the ontological definition of the conceptual hierarchy of the knowledge base. The querying mechanism focuses on the retrieval of concept fulfilling different aspects defined in the query, while the technique presented in this paper aims to retrieve graph based structures similar to the one a user specified. Unlike the approach in of Stojanovic et. al. the approach presented in this paper allows the retrieval using a not connected graph (a non empty graph G is called *connected* if any two vertices are linked by a path in G). Furthermore our approach is performance optimized as it uses adopted standard approved text retrieval methods.

In [Shasha 2002] *GraphGrep* was introduced, an approach for indexing undirected graphs with node labels, which are unique within a single graphs. The approach can be easily adopted to directed graphs with labelled nodes and edges and uses a path index of variable length, which means in this case that the maximum length of the indexed paths can be given at indexing time. However this approach does not allow wildcard queries, was only used for database filtering and does not give a relevance function for the retrieved graphs. It has not been tested upon retrieval performance and size of the index.

[Yan 2004] introduces a system for the retrieval of chemical structures, which is capable of indexing graphs based on their paths. The decision upon which paths are integrated in the index is made using a graph structure frequency measure. This frequency measure implements a TF*IDF variant and allows the indexing of paths, which are longer and more significant instead of many less significant paths, which are omitted for indexing. Unfortunately the authors did not evaluate the retrieval performance using precision and recall, although their approach is compared to GraphGrep. However a comparison with our approach would be inapproprioate: The use case of retrieving chemical structures is quite different to the use case of retrieval of MPEG-7 based semantic descriptions: Within molecules node labels are not unique, e.g. paths of different lengths consisting of multiple carbon atoms can occur, for instance in benzene hexachloride.

Acknowledgements

The Know-Center is a Competence Center funded within the Austrian Competence Center program K plus under the auspices of the Austrian Ministry of Transport, Innovation and Technology (www.kplus.at).

References

[Alesso 2004] Alesso, H. Peter, “Semantic Search Technology”, SIGSEMIS: Semantic Web and Information Systems, http://www.sigsemis.org/columns/swsearch/SSE1104, last visited 15th July 2005

[Baeza-Yates 1999] Baeza-Yates, R. , Ribeiro-Neto, B., “Modern Information Retrieval”, ACM Press and Addision-Wesley, 1999

[Berretti 2004] Berretti, S., Del Bimbo, A., Pala, P. A, “Graph Edit Distance Based on Node Merging”, in Proceedings Image and Video Retrieval: Third International Conference, CIVR, Springer, LNCS 3115, pp. 464-472, 2004

[Bunke 1997] Bunke, Horst, “On a relation between graph edit distance and maximum common subgraph”, Pattern Recognition Letters, Vol. 18, Num. 9, 1997, pp. 689-694

[Bunke 1998] Bunke, Horst, Shearer, Kim, “A graph distance metric based on the maximal common subgraph”, Pattern Recognition Letters, Elsevier Science Inc., Vol. 19, 1998, pp. 255-259

[Carroll 2002] Carroll, Jeremy J., “Matching RDF Graphs”, International Semantic Web Conference 2002 ISWC, Springer Lecture Notes in Computer Science, 2002, 2342, pp. 5-15

[Corby 2000] Corby, Olivier, Dieng, Rose and Hebert, Cedric “A Conceptual Graph Model for W3C Resource Description Framework”, 8th International Conference on Conceptual Structures ICCS 2000, Springer, 2000, pp. 468-482

[Daconta 2003] Daconta, Michael C., Obrst, Leo J., Smith, Kevin T., “The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management”, John Wiley & Sons, 2003

[DelBimbo 1999] Del Bimbo, Alberto, "Visual Information Retrieval", Morgan Kaufmann Publishers, 1999

[Diestel 2000] Diestel, R., “Graph Theory, 2nd Edition”, Springer, 2000

[Ding 2004] Ding, Li, Finin, Tim, Joshi, Anupam, Pan, Rong, Cost, R. Scott, Peng, Yun, Reddivari, Pavan, Doshi, Vishal and Sachs, Joel “Swoogle: a search and metadata engine for the semantic web”, CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management, ACM Press, 2004 , 652-659

[Fonseca 2004] Fonseca, Manuel, “Sketch-Based Retrieval in Large Sets of Drawings”, PhD thesis, Universidade Tecnica de Lisboa, 2004

[Guarino 1999] Guarino, Nicola, Masolo, Claudio and Vetere, Guido, “OntoSeek: Content-Based Access to the Web”, IEEE Intelligent Systems, IEEE Educational Activities Department, 14, 1999, pp. 70-80

[Hand 2001] Hand, David, Mannila, Heikki, Smyth, Padhraic, “Principles of Data Mining”, MIT Press, 2001

[Hunter 2001] Hunter, Jane, “Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology”, First Semantic Web Working Symposium (SWWS), Stanford, USA, 2001, pp. 261-281

[Kosch 2003] Kosch, Harald, "Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21", CRC Press, November 2003

[Lux 2005] Mathias Lux and Michael Granitzer, "Retrieval of MPEG-7 based Semantic Descriptions", BTW-Workshop "WebDB Meets IR" in context of the "11. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web", March 1st 2005, University of Karlsruhe, Germany. URL: http://caliph-emir.sourceforge.net/docs.html#publications

[Martínez 2003] Martínez, José M., "MPEG-7 Overview", Moving Picture Expert Group MPEG, Pattaya, March 2003, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, last visited 15th July 2005

[Meyer zu Eissen 2005] Meyer zu Eissen, Sven, Stein, Benno and Potthast, Martin, “The Suffix Tree Document Model Revisited”, I-KNOW 05: 5th Intl. Conference on Knowledge Management, Graz, Austria, 2005, pp. 596-603

[Rocha 2004] Rocha, Cristiano, Schwabe, Daniel and Aragao, Marcus Poggi, “A hybrid approach for searching in the semantic web”, WWW '04: Proceedings of the 13th international conference on World Wide Web, ACM Press, 2004, pp. 374-383

[Rogers 2002] Rogers, Ian, “The Google Pagerank Algorithm and How It Works”, IPR Computing 2002, http://www.iprcom.com/papers/pagerank/, last visited 15th Juli 2005

[Shasha 2002] Shasha, Dennis, Wang, Jason T. L., Giugno, Rosalba, “Algorithmics and applications of tree and graph searching”, in Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM Press, pp. 39-52, 2002

[Shokoufandeh 1999] Shokoufandeh, A., Dickinson, S.J., Siddiqi, K., Zucker, S.W., “Indexing using a spectral encoding of topological structure”, in Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999

[Simmons 1966] Simmons, R. F., “Storage and retrieval of aspects of meaning in directed graph structures”, Communications of the ACM, ACM Press, 9, pp. 211-215, 1966

[Smeulders 2000] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R. "Content-based image retrieval at the end of the early years", Pattern Analysis and Ma-chine Intelligence, IEEE Transactions on, Vol. 22, No. 12, pp. 1349-1380, December 2000

[Stojanovic 2003] Stojanovic, Nenad, Studer, Rudi, Stojanovic, Ljiljana, “An Approach for the Ranking of Query Results in the Semantic Web”, in Proceedings International Semantic Web Conference 2003, pp. 500-516, 2003

[Tittmann 2003] Tittmann, Peter, “Graphentheorie”, Hanser Fachbuchverlag, ISBN: 3-446-22343-6, September 2003

[Valiente 2002] Valiente, Gabriel, “Algorithms on Trees and Graphs” Springer, ISBN 2-540-43550-6, 2002

[Yan 2004] Yan, Xifeng, Yu, Philip S. and Han, Jiawei, “Graph indexing: a frequent structure-based approach”, SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, ACM Press, 2004, pp. 335-346

[Yang 1993] Yang, Gi-Chul and Oh, Jonathan, “Knowledge acquisition and retrieval based on conceptual graphs”, SAC '93: Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing, ACM Press, 1993, pp. 476-481

[Zhu 2002] Zhu, Haiping, Zhong, Jiwei, Li, Jianming and Yu, Yong, “An Approach for Semantic Search by Matching RDF Graphs”, Fifteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2002, pp. 450-454

[Zhong 2002] Zhong, Jiwei, Zhu, Haiping, Li, Jianming and Yu, Yong, “Conceptual Graph Matching for Semantic Search”, ICCS '02: Proceedings of the 10th International Conference on Conceptual Structures, Springer, 2002, pp. 92-196