The SemSets Model for Ad-hoc Semantic List Search

Marek Ciglan
Institute of Informatics
Slovak Academy of Sciences
Bratislava, Slovakia
marek.ciglan@savba.sk

Kjetil Nørvåg
Dept. of Computer and
Information Science
Norwegian University of
Science and Technology
Trondheim, Norway
kjetil.norvag@idi.ntnu.no

Ladislav Hluchy
Institute of Informatics
Slovak Academy of Sciences
Bratislava, Slovakia
ladislav.hluchy@savba.sk

ABSTRACT

The amount of semantic data on the web has been growing rapidly in recent years. One of the key challenges triggered by this growth is the ad-hoc querying, i.e., the ability to retrieve answers from semantic resources using natural language queries. This facilitates interaction with semantic resources for the users so they can benefit from the knowledge covered by semantic data without the complexities of semantic query languages. In this paper, we focus on semantic queries, where the aim is to retrieve objects belonging to a set of semantically related entities. An example of such an adhoc *type query* is "Apollo astronauts who walked on the Moon". In order to address the task, we propose the *SemSets retrieval model* that exploits and combines traditional document-based information retrieval, link structure of the semantic data and entity membership in semantic sets, in order to provide the answers. The novelty of the approach lies in the utilization of *semantic sets*, i.e., groups of semantically related entities. We propose two approaches to identify such semantic sets from the knowledge bases; the first one requires involvement of an expert user knowledgeable of the data set structure, the second one is fully automatic and provides results that are comparable with those delivered by the expert users. As demonstrated in the experimental evaluation, the proposed model has the state-of-the-art performance on the SemSearch2011 data set, which has been designed especially for the semantic list search evaluation.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

semantic search, retrieval model, semantic sets

1. INTRODUCTION

The recent boom in the amount of available semantic data increases the already high interest of the research community in the semantic technologies. This growth can be largely contributed to the emerging web of Open Linked Data[1]. More and more sources publish, along with the traditional human-readable content, structured and linked metadata records in formats such as RDF, RDFa and microformats[2]. The increasing amount of semantic data brings along important technical challenges. Except the basic need for the ability to store and access those data, the prevailing challenge is the retrieval from semantic data. While there exist languages for querying semantic data, such as SPARQL for RDF data sets, they require a certain level of technical skills to formulate queries, as well as the knowledge of the data representation in the underlying knowledge bases. It is only natural that the research community targets the question whether the natural language queries could be used instead of traditional structured query languages to tap the knowledge stored in semantic data sources.

The ad-hoc semantic search targets the challenge of answering keyword queries from structured knowledge bases. While in information retrieval, the result for a keyword query is the ranked list of documents from a collection, in semantic search, the result comprise ranked list of entities, resources from the queried knowledge base. As shown recently by Pound et al. [16], already today search engine users submit many queries that would be suitable for a semantic search; for example search for specific entities, sets of entities or entity attributes. Based on this observation, Pound et al. described ad-hoc object retrieval for semantic search where a user formulates queries using keywords, much like in the web search, and they propose a classification of semantic ad-hoc queries into five categories: *entity queries, type queries, attribute queries, relation queries and other keyword queries*. In this work, we focus on semantic type queries. The task of answering semantic type query is, given an unstructured keyword query in natural language with the intent of retrieving objects of a give type and a semantic graph (knowledge base), to find objects/entities of the desired type. For example, the following keyword queries are examples of semantic type queries: "Apollo astronauts who walked on the Moon", or "Arab states of the Persian Gulf". To continue the example, the correct answer for the first query, using DBpedia as a knowledge base, would comprise 'dbpedia:Neil Armstrong', 'dbpedia:Buzz Aldrin' and ten more astronauts represented by DBpedia entities.

In this paper, we propose a retrieval model for ad-hoc type queries called the *SemSets model*. The approach can be described as trying to mimic the behaviour of a human trying to answer the query using a web search engine. A human user would probably enter such a query to a web search engine and inspect several *top-k* results. The user would search the text of the inspected documents to find desired set of entities. Then, the user could rank the entities based on the quality of the information in retrieved documents, the query target and the user's knowledge and confidence.

Similarly, in our approach, we first search the documents con-

[1] http://www.linkeddata.org/

[2] http://microformats.org/

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

structed for resources of a knowledge base and we use the spreading activation technique to identify additional relevant entities in the knowledge base. This corresponds to a user performing a web search and retrieving the top-k results. Then, we check the membership of candidate resources in semantic sets constructed from the knowledge base. This corresponds to the user inspection of the retrieved documents. Finally, we evaluate the relevance of identified semantic sets to a given query and rank the members of semantic sets accordingly. The final step mimics user evaluation of the results, based on his/her knowledge. The proposed approach combines information retrieval techniques, activation spreading over the link structure of a knowledge base and information about entity membership in semantic sets, defined by the knowledge base. The idea of combining the text information retrieval with activation spreading is well known (e.g. [17]). The main innovation of the proposed approach is the utilization of semantic sets in the process. We propose two approaches for construction of semantic sets, groups of semantically related entities. One approaches requires an expert user for the task, the second one is fully automatic.

The challenge of answering ad-hoc keyword type queries from knowledge bases is a new task, unexplored for the most part. The novelty of the presented work is in the retrieval model itself, which exploits information about entity membership in semantic sets and in proposed methods for construction of semantic sets from a given knowledge base. The main contributions of the paper are:

- **Retrieval model** for the ad-hoc semantic type queries, with the goal of answering keyword queries for semantic list search from semantic data. It combines information retrieval techniques, link evidence and information on the membership of entities in semantic sets to produce the results.

- **Methods for semantic sets construction.** As shown in the evaluation section, the use of the information on the membership in semantic sets brings significant increase in the precision for the proposed retrieval model. We show how an expert user, knowledgeable about the data set, can define such semantic sets; in addition we propose a fully automatic method for the construction of semantic sets. When used in the proposed SemSets retrieval model, semantic sets constructed by both methods have very similar positive effects on the precision of the results.

- **Evaluation results** provide evidence of the method's efficiency. Moreover, we use publicly available data sets in the evaluation, which makes our results reproducible and allows direct, head-to-head, comparison with other approaches to the semantic type retrieval task.

The organization of the rest of the paper is as follows. We briefly discuss related work in Section 2, and in Section 3 we state the problem and describe the required preliminaries. We propose the *SemSets model* in Section 4 and approaches to the SemSets construction in Section 5. In Section 6 we describe our experimental setup and present the results of the evaluation. Finally, we discuss future work in Section 7 before we conclude in Section 8.

2. RELATED WORK

The problem of providing natural language interfaces to semantic data, the area also addressed by this work, is currently in the centre of research attention, with several prototype systems already in existence. Some were built for a specific domain (e.g. [5]), others are domain independent [9]. The dominant approach is to transform a user's keyword query to a formal semantic query by matching query segments to triples from the knowledge base [3, 9, 18]). One of the main challenges is the task of mapping segments of a free-text keyword query to entities of a given knowledge base. Different strategies to this task can be found in the literature, from pattern-matching, bag-of-words and gazetteer approaches to deep linguistic analysis [19]. Work related to the problem of keyword queries analysis includes also annotation of the free text with resources from a knowledge base [15], segmentation of the keyword queries [7], as well as semantic query suggestion [12].

Learning from the user interaction can be employed in order to improve performance of the semantic ad-hoc retrieval system, cf. Lopez et al. [11] and later by Damljanovic et al. [4], who extend work from [18] by allowing user feedback, query refinement, and query expansion. In contrast to the most of other systems, PowerAqua [10] (building on [11]) is able to work with multiple heterogeneous ontologies. It transforms the input keyword query into the intermediate triple form, similarly to the principle of other approaches, the intermediate format is than mapped to the candidate entities in distinct ontologies. The authors in [13] and [16] study real query logs of a major search engine from a semantic search point of view; their study allowed for classification of semantic query types and for the definition of the ad-hoc object retrieval from semantic data. Their classification comprise also the ad-hoc semantic type search that is the main focus of this work and is largely uncovered by the previous work. A methodology for the evaluation of ad-hoc retrieval is discussed in detail in [8].

The ad-hoc semantic type queries were the centre of attention in the list search track of the SemSearch 2011 challenge[3]. The systems that have participated in the challenge addressed the problem of answering type queries and are the most related works to the approach presented in this paper. An implementation of the SemSet model, presented in this paper, has been one of the participating systems and has won the challenge, achieving the highest precision scores in the evaluation. The other notable approaches participating in the challenge were a) BM25MF model that is a modified version of BM25F model adopted for semantic data, allowing the fields to have multiple values and b) the usage of the NLP parser to analyse the queries, with the goal of identifying the type of entities targeted by the query; only entities of the specified type are returned in the result set. The description of the challenge and individual approaches can be found in [1].

3. PROBLEM STATEMENT

In this section, we formulate the problem of ad-hoc object retrieval from a knowledge base. First, we formalize the property graph data model that we use in the paper as for knowledge base representation. The property graph is a formalism allowing us to reason about a knowledge base in terms of entities (vertices in the graph) that have attributes and are connected via labelled edges to other entities. We then define the task of ad-hoc object retrieval from a property graph knowledge base.

3.1 Property Graph Data Model

Informally, the property graph data model is a multigraph data structure in which vertices and edges can have properties with values. We can define the property graph as a tuple:

$$G = (V, A, P, D, L, \eta, \epsilon)$$

where V is a set of vertices, A is a multiset of directed edges (ordered pairs of vertices), P is the domain of properties, D is the domain of allowed property values for nodes, L is a domain of

[3] http://semsearch.yahoo.com/

allowed property values for edges, $\eta : V \times P \rightarrow \mathcal{P}(D)$ is the function that maps nodes properties to their values ($\mathcal{P}(D)$ being the power set of D), and $\epsilon : A \times P \rightarrow L$ is the function that maps edge properties to their values.

As the dominant approach to represent semantic data is through the RDF triples, we first discuss how to map triples to the property graph data structure. To model triple statements, we assume that P contains at least two attributes: *URI*–specifying resource identifier and *predicate_type* specifying the type of the relation. From graph theory point of view, an RDF model is equivalent to a directed and labelled multigraph. For our purposes, we need only one attribute on the edges of the property graph, which denotes a label of the relation. Thus, for simplicity, in the following text we omit the attribute type and use the relation $\epsilon' : A \rightarrow L$, instead of the original ϵ. In order to illustrate how triples are represented using the property graph model, consider the following example using DBpedia data (for simplicity, we use prefix 'dbpedia:' instead of the whole URI of the resource) of two triples:
dbpedia:Trondheim , dbpedia-owl:populationTotal , 170936
dbpedia:Trondheim , dbpprop:city_of , dbpedia:NTNU
The triples would translate into a property graph with two nodes $V = \{v_1, v_2\}$, one edge $A = \{(v_1, v_2)\}$, properties *P* = { *URI*, *predicate*, *dbpedia-owl:populationTotal* }, with $\eta(v_1, URI) ='$ *dbpedia:Trondheim*$'$, $\eta(v_2, URI) =$ '*dbpedia:dbpedia:NTNU*', $\eta(v_1,$ *dbpedia-owl:populationTotal*)=*170936* and $\epsilon'(v_1, v_2) ='$ *dbpprop:city_of*$'$.

3.2 Ad-hoc Object Retrieval for Type Queries

We adopt the definition of the ad-hoc object retrieval task from [16], where the authors specify the task in terms of inputs, outputs and evaluation.

Input: Unstructured keyword query q and a property graph G.
Output: Ranked list of resource identifiers $o = (o_1, \ldots, o_k)$, where resources are equivalent to the nodes of multigraph: $\forall o_i \in o$ $\exists v_j \in V : \eta(v_j, URI) = o_i$.
Evaluation: All the resources in o are labeled by an independent judge, knowledgeable about q and about all the necessary information on resources in o.

In our work we consider the task of ad-hoc object retrieval for semantic type queries, where a user's intents is to find members of a particular set of entities.

4. SEMSETS RETRIEVAL MODEL

This section presents the SemSets retrieval model for semantic type queries. Our approach consists of several successive phases and we structure the section accordingly. First, we discuss the query analysis, which is a preprocessing step. We then describe three scoring functions that contribute to the final scoring function of the model. Where appropriate, we also discuss secondary data structures used by the model, derived from the underlying knowledge base represented as a property graph. To facilitate the reading, we provide an example query, and for each step of the model, we provide intermediate results of the model scoring for the example query and the DBpedia data set. The example query that we use through the rest of the section is: 'Apollo astronauts who walked on the Moon'.

4.1 Query Analysis

The goal of the preprocessing phase is to analyse the keyword query and relate its segments to resources in the knowledge base. The aim of the analysis is to identify the query segments and the principal entity of the query that belongs to the property graph. The analysis can be exploited for refinement of the query, using query segmentation in the document retrieval model used in Section 4.2 and the usage of identified principal entity is described in Section 4.4. Let $pent(q, G) = v : v \in V$ be the principal entity of the query, given the property graph G. For the sake of generality, we do not impose any restriction on the method for computation of the $pent(q, G)$ function, any suitable method can be used. In our experiments, we rely on a dictionary based annotation that links text fragments to the entities in a knowledge base [15].

Example of intermediate results. Given the example query, the query analysis phase identified following entities mentioned in the query: '[[List of Apollo astronauts | Apollo astronauts]] who [[Base on balls | walked]] on [[Moon | the Moon]]'. The query segments identified as related to the entities of the knowledge base (DBpedia) are enclosed by symbols '[' and ']', where the string segment following after the symbol '|' is the original text of the query and string segment before the '|' symbol refers to the entity of the DBpedia knowledge base. In this case, the query analysis procedure identified correctly entities 'List of Apollo astronauts' and 'Moon', the third identified entity is incorrect. The used query annotation method also assigns confidence values for the annotated entities[15]; the scores for the our example were: conf('List of Apollo astronauts') = 0.962; conf('Base on balls')=0.619 and conf('Moon')=0.547. Thus, the entity 'List of Apollo astronauts' has been identified as the principal entity of the query.

4.2 Candidate Entities Score

The first approximation of the result ranking is done by the document retrieval method combined with the expansion of the entities based on the activation spreading over the edges of the knowledge base. As we have vertices with properties instead of documents as the input, we first discuss the construction of documents for entities of the knowledge base. Then, we describe the identification of candidate entities for the query answer, the first approximation of the result.

Entity documents. We deal with unstructured keyword queries, it would be convenient to take advantage of the information retrieval models that are able to handle such queries. However, information retrieval models are designed for the document retrieval, whereas we have a knowledge base composed of vertices with properties and edges (relations) between the vertices. Our approach is to construct entity documents, i.e., a textual representations of entities (vertices) in the given property graph knowledge base. We construct documents by concatenating a defined subset of properties of each vertex in the knowledge base. Let $P' \subset P$ be the subset of properties selected for documents construction ($P' := P$ would be the default, general setting). We can define a document for a vertex $v \in V$ as a union of selected properties:

$$doc(v, G) = \bigcup \eta(v, p) : p \in P'$$

We can then define the similarity between an entity in a knowledge base and a keyword query as:

$$vsim(v, q, G) = sim(doc(v, G), q)$$

where $sim(doc(v, G), q)$ can be substituted by a suitable information retrieval model, e.g., probabilistic or vector-space model. We define the rank of a vertex v for a given query q and a property graph G as:

$$rank(v, q, G) = |\{v_i \in V; vsim(v_i, q, G) > vsim(v, q, G)\}|$$

Candidate entity score. In order to identify entities that are likely to be a part of the query answer we compute candidate scores for entities of the knowledge base. We exploit the document retrieval model and combine it with the link evidence stored in the

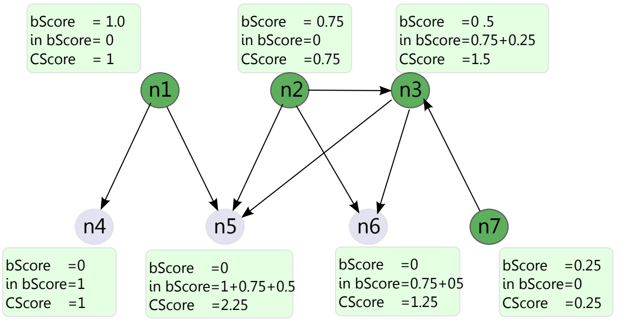

Figure 1: Example of S_C computation for a simple graph. Parameter $k = 4$, the nodes n1,n2,n3,n7 (in this order) are the top k matches for document similarity to the query.

property graph, exploiting the principle of the activation spreading. Informally, we take the top-k vertices with the highest similarity to the query and expand from those vertices by the edges, as defined in the property graph knowledge base G. Let $top(k, q, G) = \{v \in V : rank(v, q, G) < k\}$ be the top-k items from G with highest similarity values with respect to the query. A vertex $v \in V$ is assigned a score equal to the sum of scores of top k items that link to it. Activation is spreading one hop from the vertices retrieved by the information retrieval part. We restrict the spreading only to one hop from the given vertices because of the mathematical properties of the network. The semantic network used in our experiments is a small-world network with a small diameter. Thus, allowing activation spreading more than one hop from the given vertices results in activation of a large part of the network. The score for an item in top k is proportional to its rank. We define a base score to be proportional to the similarity rank of an entity document to a query:

$$S_B(v, q, k, G) = \begin{cases} 1 - rank(v, q, G)/k \Leftrightarrow rank(v, q, G) < k \\ 0 \Leftrightarrow rank(v, q, G) \geq k \end{cases}$$

Let $L' \subset L$ be the set of edge labels (e.g. equivalent to predicate types in RDF model) used for expansion (again, $L' := L$ would be the default setting). We define the candidate score S_C as:

$$S_C(v, q, G, k, L') = S_B(v, q, k, G) + \sum_{\substack{\forall (i,v) \in A: \\ \epsilon'((i,v)) \in L'}} S_B(i, q, k, G)$$

An example of candidate scores of a simple graph is depicted in Figure 1. We consider vertices with S_C greater than zero as the first approximation of the result and we will refer to this set as the candidate set C: $C(q, G, k, L') = \{v \in V : S_C(v, q, G, k, L') > 0\}$.

Example of intermediate results. To illustrate the process on our example query we provide intermediate results for this phase. The top five entities with the highest $vsim(v, q, G)$ for our example query about the Apollo astronauts are: 1. The Wonder of It All (2007 film) 2. List of spacewalkers, 3. Moon Landing (music drama), 4. List of Apollo astronauts, 5. Harrison Schmitt. Although the result set contains entities somewhat related to the query, the only high-ranked entity that should be part of the optimal result is 'Harrison Schmitt'. However, several of the entities retrieved are connected by edges to the entities that should be part of the optimal answer (e.g. List of Apollo astronauts).

After computing the S_C, using the spreading of activation (from the 10 top ranked vertices), the top ranked entities according to S_C are: 1. Astronaut, 2. NASA, 3. Moon, 4. Apollo 15, 5. Apollo 12, 6. Apollo 11, 7. List of Apollo astronauts, 8. Apollo program, 9. Buzz Aldrin, 10. Apollo 17, 11. Eugene Cernan. The computation of the C_S allowed us to bring more entities that match the query into the result set (i.e. 'Buzz Aldrin' and 'Eugene Cernan'); the result set is still far from optimal, as the optimal answer should comprise the twelve moon-walkers at the top ranks.

4.3 SemSets Score

We now describe the core of the proposed SemSet model that also gives it the name. Let us assume the existence of semantic sets that comprise entities from the knowledge base G, and let us assume that members of such a semantic set S (we will refer to a semantic sets as a SemSet through the rest of the paper) are semantically related. We will refer to a set of all SemSets as S'. We explain how to construct such semantic sets from the knowledge base in the Section 5. After the construction of a candidate set C (Section 4.2), we identify the SemSets that have at least a fraction p of its members in the candidate set $C(q, G, k)$:

$$CSS(q, G, k, p, L') = \left\{ S \in S' : \frac{|S \cap C(q, G, k, L')|}{|S|} \geq p \right\}$$

As we are searching for the entities that belong to a semantic set, the members of SemSets in CSS are good candidates for the query answer. However, in practice there often are multiple SemSets in CSS, in which case we have to distinguish how well a SemSets fit the input query. To measure similarity of a SemSet S and a query q, we construct a document for S by concatenation of entity documents of SemSets members:

$$sdoc(S, G) = \bigcup_{v \in S} doc(v, G)$$

and we measure the similarity of a SemSet S and a query q as:

$$ssim(S, q, G) = sim(sdoc(S, G), q)$$

where $sim(sdoc(S, G), q)$ is a document retrieval model. Let b be the SemSet boost parameter. We can then compute the *SemSet Score* of an entity in a knowledge base as:

$$S_S(v, q, G, k, p, b, L') = 1 + (b \times \sum_{\substack{v \in S; S \in \\ CSS(q,G,k,p,L')}} ssim(S, q, G))$$

Example of intermediate results. Let us assume the setting in which we use sets of entities belonging to Wikipedia categories as SemSets. We can compute the CSS, set of candidate SemSets, which would (in this setting and the example query) be: *CSS = { Category: People who have walked on the Moon, Category: Skylab program}*. As we can see, we have two candidate SemSets with different content, one of which comprises the correct answer entities. Let us assume the setting, where we use additional SemSets, formed by entities that include the same Wikipedia templates in their respective articles. In this setting, the situation would be even more complex, having: *CSS = { Category: People who have walked on the Moon, Category: Skylab program, Template: NASA Astronaut Group 3, Template: NASA Astronaut Group 5, Template: NASA Astronaut Group 2, Template: People who have walked on the Moon}*. To resolve the issue on which of the SemSets from *CSS* fits the query best, we can compute the textual similarity of the query and the SemSet document. The top three ranked items in CSS, in our example setting are the following: 1. Template: People who have walked on the Moon, 2. Category: People who have walked on the Moon , 3. Template: NASA Astronaut Group 3. In this way, we discover the best matching SemSets. The ranking of the vertices after the computation of the S_S score would comprise

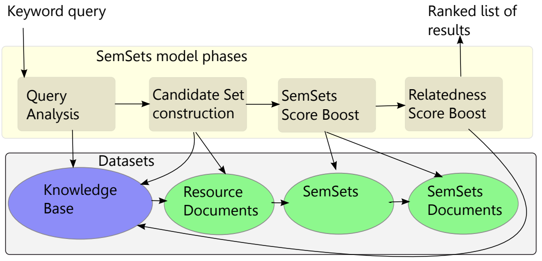

Figure 2: Block scheme of the SemSets model computation process. The upper block describes of model phases, while the lower block represents the data sets used in distinct phases.

entities 'Neil Armstrong', 'Buzz Aldrin' and the ten more astronauts who have walked the moon on top ranks, followed by other NASA astronauts, forming a good answer for the given query.

4.4 Principal Entity Relatedness Score

In the case that no SemSets are identified in the candidate set $C(q, G, k, L')$, we have only the basic S_C score relying on the document similarity and link evidence to rank the resources. When no SemSets are found, it often means that the information required by the query is either not covered by the knowledge base or is covered by the knowledge base but not by the used SemSets (e.g., in case the given query is not a semantic type query) Where no SemSets are identified but a principal entity of the query is identified given the property graph G, the information required by the query might be covered by the knowledge base. This might be an indication that another semantic retrieval model would be more suitable for the given query, e.g., in cases when the input query is not a semantic type query. However, we extend the SemSet model by an optional step, to improve the scores in such cases. We propose boosting the scores of items in $C(q, G, k, L')$ based on the similarity with principal entity *pent*(q, G) identified in the query analysis phase. In this step, we consider structural similarity of the nodes in the knowledge base. We can write:

$$S_P(v, q, G, k, c, L') = 1 + (c \times \mathit{struct\_sim}(v, \mathit{pent}(q, G), L'))$$

where c is the boost parameter for the principal entity relatedness and *struct_sim* is a graph structural similarity measure, computed taking into accounts edges with labels from L'. Namely, we utilize cosine similarity of two nodes, where the vector space is defined by the nodes' neighbours, as encoded in the underlying knowledge base. We consider this measure as it has proven efficiency for computing semantic relatedness [14].

Example of intermediate results. As the SemSets were identified during the SemSets processing for our example query, the computation of the principal entity relatedness score is just optional. However, for illustration, we continue with our example. The vertex corresponding to the Wikipedia article 'List of Apollo astronauts' has been identified as principal entity of the query. If we compute the structural similarity of this vertex and vertices in candidate set C, the top ranked vertices would be: NASA Astronaut Group 3, NASA Astronaut Group 5, Eugene Cernan, List of Apollo astronauts, Jack Swigert , John Young (astronaut). As shown in the evaluation section of the paper, using S_P score can improve slightly the overall performance of the proposed model.

4.5 SemSets Model Score

The final SemSet model score is simply the multiplication of the three scores introduced in previous subsections; the base score S_C and S_S and S_P. The block scheme of the phases and the data resources used are depicted in Figure 2. Let $v \in V$ be a vertex from the property graph G, let q be the keyword semantic type query. The parameters of the model are: k, b, c, p, L', where k is the parameter used in 4.2 , defining number of top k ranked results used for candidate set creation; b, c are boost parameters; p defines the fraction of SemSet members required to identify the SemSet from the candidate set and L' defines set of edge types used in computations related to the graph structure. Thus, the score is:

$$\mathit{SemSetModelScore}(v, q, G, k, b, c, p, L') =$$
$$S_C(v, q, G, k, L') \times S_S(v, q, G, k, p, b, L') \times S_P(v, q, G, k, c, L')$$

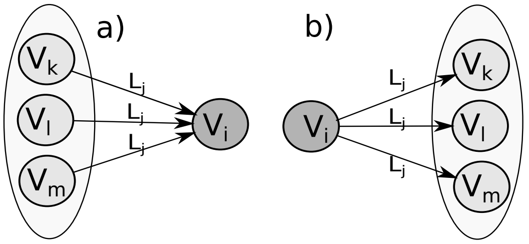

Figure 3: SemSets patterns in a property graph structure. A SemSet is a) vertices having outgoing edge of the same label to a common vertex, b) vertices having an incoming edge of the same type from a single vertex.

5. SEMSETS CONSTRUCTION

So far, we have assumed the existence of SemSets, without describing how to obtain them. We fill the gap in this subsection; we propose two methods for SemSets identification. The first one relies on the judgement of an expert user knowledgeable of the data set. As wez mentioned earlier, SemSets are sets of semantically related entities from the underlying knowledge base. As the knowledge base contains information on semantic relations between the entities, we should be able to construct such SemSets from the knowledge base. We define a SemSet in a property graph structure by following graph patterns: a) a set of vertices connected by an outgoing edge of the same label to a common vertex, or b) a set of vertices that have an incoming edge of the same label from a single vertex. We depict both cases in Figure 3.

Let us use again the DBpedia data set to provide a few examples. Members of Wikipedia categories are often a good example of a SemSet, e.g. the vertex in the property graph representing Wikipedia category 'Category: People who have walked on the Moon', comprising 12 astronauts is the SemSet that provides the answer for the example query we use in the paper. Members of a category are vertices connected by an outgoing edge labelled 'dcterms:subject' to the vertex representing the category. The example of a SemSet of the second type, where a set of semantically related nodes have the incoming edge with the same label from a single vertex can be e.g., members of a music band in DBpedia, where the vertices representing band members have incoming edge labelled 'dbpedia-owl:bandMember' from the vertex representing the band. The provided examples form sets of nodes that are usually perceived as semantically related. However, we can find examples of sets matching one of the two described patters which comprise members that do not match the human intuition of being semantically related; e.g. members of a category 'Category:1947 births' comprising people born in 1947, or the vertices representing persons connected by an incoming edge labelled 'dbpedia-owl:deathPlace' to a vertex representing a city. Although, the semantics of such sets is clear, a human judge would probably estimate the semantic relatedness of the set members to be quite low. Another problem is a practical one. The sheer number of possible SemSets can be impractical and difficult to handle. In theory, even if we constraint the minimal size of a SemSet to two members, there could be as much m SemSets for a graph G, where m is the number of its edges.

5.1 SemSets Based on Expert's Knowledge

A very large number of SemSets would be inconvenient in practice. The expert user with a good knowledge of the data set, its scheme and data itself can be helpful in identifying the SemSets that are useful. Although, it would not be reasonable to judge all possible SemSets one by one, the expert's knowledge can be exploited with just a small effort. We argue that not all the edge labels (types) (equivalents of RDF predicates) are equally helpful in construction of SemSets and the expert can identify the relations, labelled edges in the property graph, that are likely to define good SemSets. E.g. in our experiments with DBpedia data set, we have used SemSets defined by category membership edges and edges defining the inclusion of Wikipedia templates. As documented in the evaluation section, just by using those two edge labels for the SemSets definition, the improvement over the baseline has been significant. More formally, let T' be the subset of edge labels or types defined by an expert for construction of SemSets. In terms of the property graph, we can write that superset of SemSets S' is:

$S'(G) = \{S; S \subset V : \exists v \in V \exists t \in T' : ((\forall i \in S : (i,v) \in A \wedge \epsilon'((i,v)) = t) \vee (\forall i \in S : (v,i) \in A \wedge \epsilon'((v,i)) = t))\}$

5.2 Automatic SemSets Filtering

Although the SemSets identification by involving the expert user as described above is feasible, it requires the knowledge of the data set. An automatic method for identifying sets of semantically related entities would be suitable to give us the means of handling input data sets without prior knowledge of its schema or semantics. The semantic relatedness should reflect how a human user would judge the relatedness of the given concepts, documents or terms. It is, by its nature, a fuzzy concept, but the intuition is that the relatedness should be capturable or correlate well by some similarity measure. Thus, we have generated all possible sets matching the described patterns (Figure 3), comprising more than two vertices (entities) of the property graph knowledge base. We have than measured the similarity of SemSets members using similarity measures established in literature. The motivation is to study, whether we can use some of established similarity measures to automatically filter the SemSets and thus reduce their number. The goals of this exercise are twofold. The first goal is to measure whether such and automatically reduced set of all possible SemSets achieves competitive results when used in the proposed SemSets retrieval model. The second goal is to study the suitability of different measures for the task and the correlations of distinct measures.

The data we work with, the property graph, provides us with graph topology and we have also textual information, entity documents constructed for each vertex of the knowledge base, based on its properties. We have studied the textual similarity of the the SemSets as well as the structural similarities of its members.

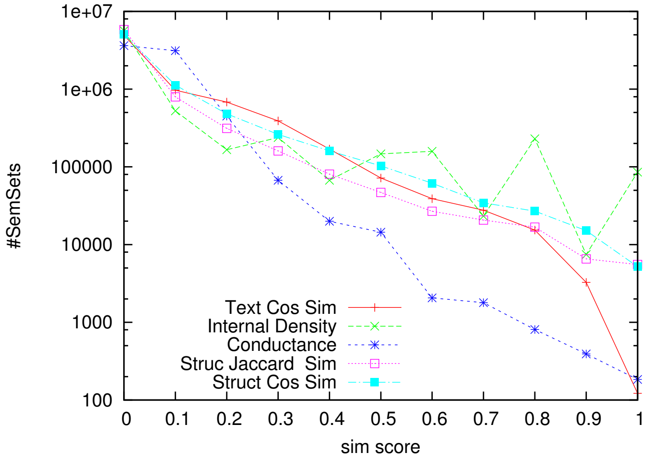

Figure 4: Similarity scores distribution. The y-axis is in log scale, depicting the number of SemSets having the defined similarity score.

For textual similarity, we have used the cosine similarity of the term vectors of the constructed entity documents. As we deal with a set of elements, not a pair, we have computed the average of the pairwise cosine similarity of the set members. To study structural similarity, we have used four measures, the first two measure how community like is a set of vertices, the rest were adopted from work studying semantic relatedness. The first structural measure is *conductance* that is defined as the ratio of edges outgoing from the given set of nodes to the rest of the graph and total number of edges outgoing from the given set of vertices. Formally, let $a_{i,j}$ be element of the adjacency matrix of G and $a(S) = \sum_{i \in S} \sum_{j \in V} a_{ij}$, the conductance can be defined as:

$$\varphi(S) = \frac{\sum_{i \in S, j \in \overline{S}} a_{i,j}}{min(a(S), a(\overline{S}))}$$

In the standard definition of conductance the lower the value is, the better the community structure is. As the other studied measures have image of their functions in $< 0, 1 >$ with the semantics that a higher value means higher similarity, we use value *1-conductance* in our experiments to facilitate the comparison.

The second studied structural measure is the *internal density*, expressing how clique-like is the subgraph generated by the given set of vertices, i.e., it is a ratio of edges within the given set and the number of all possible edges within the set. We can define the *internal density* as follows: let $a_{i,j}$ be element of the adjacency matrix for G, $sgn(x)$ is the signum function, the internal density is $\psi(S) = \sum_{i,j \in S; i \neq j} sgn(a_{ij}) / (|S| \times |S - 1|)$. Let us note that we are dealing with a multigraph structure which allows for multiple edges between two nodes. The provided definition counts multiple edges between two vertices as one and excludes loop edges (where edge's incoming and outgoing vertices are the same). Thus, the image of the function is the interval $< 0, 1 >$. The third and fourth structural similarity measures were averages of pairwise Cosine and Jaccard similarity. The vectors used for the computation are the vectors of the vertices neighbours, connected by outgoing edges. This similarity measures were inspired by the study of the semantic relatedness [14]. In Figure 4, we depict the distribution of the similarity measures scores. The x-axis is the interval $< 0, 1 >$, which is the image of all the similarity function, y-axis is the number of SemSets having the similarity score. The values of similarity scores were rounded to one decimal place for the sake of plot clarity. The plot shows that majority of the SemSets have low similarity scores for all the studied measures; thus, a simple thresholding can reduce the number of the SemSets significantly. We have used the sets with the average pairwise textual similarity higher than 0.1, with the observation that the achieved precision is very close to the precision of the method when the user defined

Table 1: The table of pairwise correlation of the similarity measures. TCS - textual cosine similarity; IDens - internal density; COND - conductance; SJC - structural Jaccard similarity, SCS - Structural Cosine Similarity

Sim. meassures	TCS	IDens	COND	SJC	SCS
TCS	1	0.497	0.221	0.676	0.726
IDens	0.497	1	0.216	0.246	0.312
COND	0.221	0.216	1	0.254	0.271
SJC	0.676	0.246	0.254	1	0.983
SCS	0.726	0.312	0.271	0.983	1

labelled edges were used to produce the SemSets. The evaluation setting and results are described in detail in Section 6.4. We have studied the correlation of distinct similarity measures. We have computed the Spearman's correlation coefficient (values ranging from -1 to 1, where 1 means a perfect linear dependency) for each pair of the studied measures. The results are depicted in Table 1. The correlation is quite significant for the textual cosine similarity of the term vectors of entity documents and the structural cosine similarity of the neighbourhood vectors. The implication is that the measures can be used interchangeably. One is based on the textual similarity, the other on the topological similarity. Thus, in case, when we the relations are defined poorly in the given data set, we can use the textual similarity. On the other hand, when the data set lacks in textual content, the structural similarity can be exploited.

6. EVALUATION

In this section, we present the evaluation of the SemSets retrieval model. We first discuss the SemSearch 2011 data set, and we describe our experimental setup. We then present the results achieved by SemSets model, using sets defined by an expert user and we compare them against the baseline, showing an important improvement in the precision. We discuss how the different steps of the SemSets model affect the final result. We compare the scores achieved by using distinct partial scoring functions of the SemSets model and their combinations. We compare the results achieved by using different retrieval models in step 4.2, we study results produced by using different edge labels to generate SemSets. Finally, we compare the SemSets defined by the expert user and the automatically constructed SemSets, when used in the SemSets retrieval model. The results shows that the SemSets acquired automatically have almost the same positive effect as the ones defined by the user.

6.1 SemSearch 2011 Data Set

Yahoo! SemSearch 2011[4] was a research challenge designed for the ad-hoc object retrieval from semantic data, triple collection (namely, Billion Triple Challenge 2009 data set (BTC)[5]). The list search track of SemSearch 2011 challenge has been especially designed for the task of answering keyword semantic type queries from a collection of triples and as such, was an obvious choice for the evaluation of the SemSets model. The data set consist of 50 keyword queries selected from the query log of a web search engine and the relevance judgements for the resources from BTC for each query. All of the selected queries were judged by human evaluators to be queries with the intent to retrieve list/set of entities. In addition, all of those queries were followed, in the query log, by a user click on a link leading to a Wikipedia page. The latter indicates that the results for the queries are covered by Wikipedia and we have thus decided to use DBpedia, a data set containing extracted structured information from Wikipedia, as our primary knowledge base. The relevance judgements for the queries were produced in course of SemSearch 2011 challenge, by using a crowd-sourcing solution for human intelligence computation, i.e., the Amazon's Mechanical Turk. The evaluators are human workers who gain a financial reward for completing a given task. Workers evaluated the results submitted by teams participating in the SemSearch 2011 challenge. The relevance judgements were performed on resources from the BTC collection. Each answer has been evaluated by five workers and the resources were assigned the values: 0, meaning non-relevant resource to the given query; 1, meaning that the resource is partially relevant to the query, and 2, denoting that resources is part of the correct answer for the given query. As the results were evaluated by humans and non-experts, the judgements are not perfect. Assignment of the partial relatedness (score 1) can be considered a bit random, as the concept of the partial relatedness is very subjective to a particular individual. Also several semantic errors are present, some resources that should belong to the correct answer for a query are judged with the score 0. In addition, for several queries there are no resources with the score of 2, simply because the correct answers were not present in the evaluated data set. Despite those imperfections, the SemSearch data set is the best available data set for the evaluation of the ad-hoc semantic type queries and we use it without any modifications or quality improvements. This allows us a head to head comparison of the SemSets model with other approaches to the ad-hoc list search.

6.2 Experimental Setup

In the following, we describe the setup used for the evaluation of the SemSets model.

Query analysis. The goal of the query analysis is to identify principal entity mentioned in the input query, which is a part of the used knowledge base. To map segments of the ad-hoc textual query to the resources in DBpedia data set, we use the Wikipedia miner toolkit [6]. It is an implementation of the method proposed by Milne et al. [15] and was primarily designed to annotate the text with the Wikipedia topics. Mapping from Wikipedia topics to DBpedia resources is a straightforward process. In case of multiple resources being identified within the query, we use the one with the highest relevance score as the principal entity. The method used is similar to the query fragmentation proposed in [7]. Results of query analysis are exploited in document retrieval for the query fragmentation and in the computation of the principal entity relatedness.

Construction of candidate entities scores. In order to generate set of candidate entities (cf. Section 4.2), we need to construct the entity documents for the vertices of the property graph knowledge base. To do that, we use concatenated entity properties with textual content, the main part of the entity documents being covered by the DBpedia English abstracts. We construct the index of the document collection created from the knowledge base using the Lucene framework[7]. To study the impact of the text similarity model used in this step on the SemSets model performance, we have performed experiments with multiple models: 1) the TF-IDF based, standard Lucene scoring function[8], 2) the Lucene standard scoring function exploiting the query fragmentation, provided by the query analysis phase (terms of the query fragment related to the principal entity of the query must occur in the document), 3) language modelling scoring, and 4) a combination of (2) and (3), where normalized

[4] http://semsearch.yahoo.com/

[5] http://vmlion25.deri.ie/

[6] http://wikipedia-miner.sourceforge.net/

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

[8] http://lucene.apache.org/java/3_0_1/api/.../search/Similarity.html

scores of the two retrieval models are combined. We use edges of the wikilink type, which represents the hyperlinks between Wikipedia articles, as the link types L'. The reason for using the wikilink edges is because the links between Wikipedia articles usually define some kind of semantic relationship, even though we are not able to extract the type of the relationship automatically; i.e., there is no relation of another type between the two resources in DBpedia.

SemSets score computation. For the initial set of experiments, we use SemSets defined by the expert users. In order to construct SemSets, we use the links connecting the resources of the knowledge base to the resources representing Wikipedia categories (i.e. 'subject' predicate in the DBpedia dataset). In addition, we use the Wikipedia templates and inclusion relations between templates and related DBpedia resources. The hypothesis is that the template inclusions often holds semantic information on the resources and could potentially improve the quality of the knowledge base and consequently the quality of the SemSets model results as well.

Computation of principal entity relatedness score. This step corresponds to Section 4.4. We use the principal entity identified in query analysis phase and we use wikilinks as the edges in the computation of the structural cosine similarity between the principal entity and the resources in the candidate set. The rest of the parameters were determined empirically.

Result postprocessing. In order to use the SemSearch 2011 data set for the evaluation, we have to perform a final postprocessing step.This is needed because the computed results from the described setup comprise only resources from DBPedia, whereas the SemSearch relevance judgements were produced for the BTC 2009 collection. The BTC collection comprise a large part of DBPedia snapshot from 2009 (but not the complete DBPedia data set), in addition it also comprises triples from other sources. In order to be able to use the SemSearch relevance judgements, we need to map resources produced as results by our experimental setup to the resources of the BTC. We do it by filtering the result set, removing all the DBpedia resources that are not part of the BTC collection. We than expand the filtered data set by the *sameAs* links that are part of the DBpedia collection and maps the DBpedia resources to the resources from other data sets. We filter the expanded result set, and again keep only the resources that are part of the BTC collection as well. After this postprocessing step, we can use directly the SemSearch 2011 relevance judgements to evaluate results produced by the SemSets model.

6.3 Evaluation Results

In the evaluation of the SemSets model, we used the experimental setup as described in Section 6.2. We have used the top 100 ranked resources from the results for all the queries and computed the mean average precision (MAP) against the relevance judgements from the SemSearch 2011 challenge (using TREC evaluation toolkit[9]). We performed the experiments with different configurations of the SemSets model setup to study the impact of distinct parts of the SemSets model on the final result. The results have been computed with and without the query analysis phase; for the candidate set construction (as described in Section 4.2), we have studied multiple text similarity scoring functions; for the SemSets score computation, we have studied different link types to define SemSets (category membership and templates inclusion) and we have studied the scores of distinct partial functions of the SemSets model.

The baseline. To our knowledge, there is no prior work focusing directly on answering semantic type queries from semantic data.

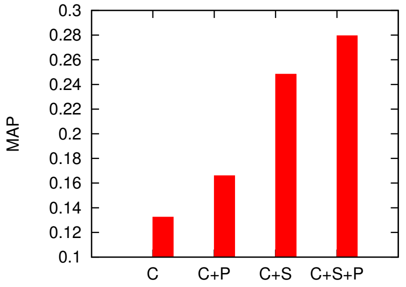

Figure 5: MAP scores for partial functions of SemSets model.

Table 2: SemSet model precision on SemSearch 2011 data set.

MAP	P@5	P@10	P@15	P@20	P@30
0.2795	0.3560	0.3660	0.3187	0.2890	0.2133

Thus, we have decided to use the sate-of-the-art retrieval method focusing on a similar task as a baseline for the comparison. Probably the most related retrieval task is the ad-hoc entity retrieval from semantic data, where the task is to rank the entities in the semantic knowledge base, based on an unstructured keyword query. According to the results of entity search track of SemSearch 2011 challenge, the best performing method for this task is the BM25MF [2] method, a modification of the popular BM25F information retrieval model. The BM25MF method achieved mean average precision of **0.1591** on the SemSearch 2011 - list search track data set.

SemSets model precision. We have tuned the parameters of the SemSet model on the first 15 queries of the data set. The configuration has been the following: a) Wikipedia miner tool for the query analysis (Section 4.1); b) the combination of the language modelling approach and Lucene scoring function with the query fragmentation based on the query analysis step for the text similarity computation was used for candidate entities score (Section 4.2); c) the category membership and template inclusion links were used to generate the SemSets for SemSets score computation (Section 4.3); d) the other SemSets model parameters were set as follows: $k = 12$; $p = 0.7$; $b = c = 100$.

The resulting MAP score of the experiment for this setup has been **0.2795**. Given the baseline of **0.1591**, this is a significant improvement in the precision of answering type queries from semantic data and fully justify specialized retrieval model for such queries. It is the most important result of the presented work. For completeness, the precision values at various ranks are summarized in Table 2 In the rest of the section we discuss the effect of different setup configurations on the MAP score.

Partial scores. In order to study the effect of the three scoring functions that are part of the SemSets model, we have performed runs with the use of different combinations of the scoring functions. The base S_C function can be interpreted as a naive baseline, where we just retrieve documents constructed for resources of a knowledge base, given a keyword query. The results are summarized in Figure 5. From the results, it is obvious that the main improvement of the final score is brought by the employment of the SemSets score S_S function, which boosts the scores of SemSets members, which are present in the candidate set. The principal entity score S_P brings also modest improvement when combined both with base $S_C \times S_P$ and also with $S_C \times S_S \times S_P$. From the aggregated results, it might intuitively seem that the improvements in score added by S_S and S_P are independent of each other. This

[9] http://trec.nist.gov/trec_eval

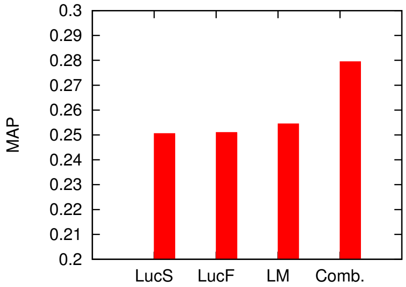

Figure 6: Impact of the text similarity scoring function. LucS - Lucene standard scoring, LucF - Lucene scoring function with query segmentation, LM is the language modeling approach and Comb. is the combination of the two latter functions.

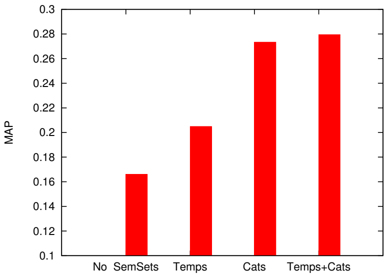

Figure 7: Impact of used SemSets. 'No SemSets' - no SemSets were used, 'Temps' - SemSets based on template inclusion relations, 'Cats' - SemSets based on the category membership.

intuition can be verified by inspecting scores of distinct queries. We compute the improvement of the $S_C \times S_S$ as $S_C - (S_C \times S_S)$. From the 50 input queries, there is 0 or negative improvement for 26 queries. Out of those 26, 8 queries have no positive judgments in the evaluation data set, which leaves us with 18 queries with 0 or negative improvement over the naive baseline; the introduction of the S_P lead to a positive improvement over the baseline in 11 cases (meaning that in 61.1% of the queries, where the S_S score does not brings any improvement to the baseline, the S_P does.) This indicates that S_P boost is complementary to the S_S boost.

As the main increase in MAP score is contributed by exploiting the SemSets and query similarity to the candidate SemSets (*CSS*, cf. Section 4.3), it is only appropriate to ask whether a simpler solution would not have similar results; we could just retrieve the SemSets documents with the highest similarity to the given query and then produce results by listing the resources belonging to appropriate SemSets. The short answer is, that such a solution does not work well. Our experiment with this solution resulted in the MAP score of 0.0998. Thus, constructing first the candidate set based on query similarity to individual entities of the knowledge base and their linked neighbours brings the important improvement in precisions when combined with the SemSets principle.

Impact of the query analysis. The query analysis provides information exploited in successive steps in the SemSets model. The results of the query analysis are used for 1) query segmentation in candidate set generation and 2) for determining the principal entity of a query. To measure how the information from the query analysis impacts the performance of the model, we executed runs with and without the query analysis part. All the other settings remained the same as in the original configuration, as described above. The MAP of the result set without query analysis was 0.2434, that is 0.0361 lower than the MAP score with the query analysis part.

Impact of the text similarity score. In order to measure the impact of the text similarity scoring on the SemSets model, we have executed runs with the four different retrieval models described in Section 6.2. The results are depicted in Figure 6. The important observation is that the improvements in the document retrieval model brings significant improvement to the overall performance of the SemSets model. In addition, combination of different retrieval models, with good individual performance, was very beneficial in terms of overall MAP score improvement.

Impact of the SemSets generation. To assess the importance of the SemSets quality used in the model, we have executed runs with different SemSets, constructed from the property graph. We study the SemSets defined by category membership and by the template inclusion. The SemSets defined by the category membership are constructed from entities that are connected by the link of 'subject' type to the same vertex. The template inclusion data set comprise links between DBpedia resources and the Wikipedia templates that are included by those articles. Figure 7 depicts the results.

6.4 User Defined SemSets vs. Automatically Constructed SemSets

So far, we have presented the results obtained using SemSets defined by an expert user and we have shown an important increase in the model precision when using information on entity membership in those SemSets. As the requirement of the user involvement lacks generality, we have proposed an approach how to identify semantically related sets from the knowledge base automatically. It is done by identifying candidate SemSets by matching two graph structural patters and computing similarity measures of their members. The important question that has been left unanswered is how well do the automatically identified SemSets substitute the SemSets defined by the expert, knowledgeable about the data set. To answer this question, we have performed experiment where the automatically identified SemSets with the average pairwise textual cosine similarity of the entity documents higher than 0.1 have been used (the threshold has been chosen empirically). The MAP value of the result set obtained by using this setting was 0.238 (the MAP value with user defined SemSets was 0.2795). Although this result is quite satisfactory, the loss of 0.0415 in MAP score did not correspond to our intuition after the inspection of several individual queries. Our interpretation of this loss is the quality of the relevance judgements, where because of three values for relevance were allowed (0-non relevant, 2-relevant, 1-partially relevant). As argued before, the assessments of the partially relevance vary highly depending on the human evaluator and the partial relevance judgements are a bit random. When we did the evaluation of the results against the relevance judgements stripped of the resources assessed as partially relevant, we achieved the MAP score of 0.2611 with automatically acquired SemSets and 0.2664 with a user defined SemSets. The conclusion is that we can use automatically identified SemSets and achieve results comparable to the setting with SemSets defined by experts.

7. DISCUSSION AND FUTURE WORK

There are several opportunities for the future work. As our experiments show, the SemSets model is quite efficient for answering semantic type ad-hoc queries. It is not, however, a perfect fit for all ad-hoc semantic queries, e.g., for entity queries or entity attribute queries. The important question is whether we can determine, by analysing the input keyword query, the suitability of the SemSets model for answering the given query. If we would be able to do so efficiently, we could combine SemSets model with other models or systems for ad-hoc semantic search which are suitable for other query types. We could thus construct a more robust system for semantic search, combining multiple approaches for different query types. Similar approach is used in IBM Watson [6], where multiple models are used to analyse the given question, each suggesting an answer and the confidence of the model. The final results are produced by synthesis of the results of distinct models and information on the confidence. E.g. the SemSets model can be used when the model's confidence on type of the query and the correctness of the answer is hight, otherwise other models can be used for the given question. The intuition is that when no SemSets are identified in the candidate set or the text similarity of the best candidate SemSet is low, it is highly probable that the SemSets model is not a good retrieval model for the given query and other models might be used to compute the answer. Although preliminary examination looks promising, this requires thorough examination and it will be the main focus of our future work on the topic.

For eight queries in total, out of fifty in SemSearch 2011 data set, our implementation of the SemSets model had MAP value of 0, meaning that no correct answer has been discovered by the model. The main reasons are: a) absence of the relevant SemSet in the used data set (the SemSet model yields precise results only when the required answer for the query is well covered by the data set in use), b) failure to retrieve relevant entities in the candidate generation phase (this leaves a room for the improvement in the used document retrieval model). The proposed method relies heavily on the traditional information retrieval and demonstrates the value of IR techniques for the processing of semantic data, especially for the ad-hoc querying. The lesson learned is that the rich textual information should be an integral part accompanying the semantic data, at least for the data sets where the ad-hoc querying is desirable.

8. CONCLUSION

In this paper, we have proposed the SemSets retrieval model for answering semantic type queries from a semantic knowledge base. The model combines and exploits document representation of knowledge base resources, relations between the resources and their membership in semantic sets to compute ranks of distinct resources give a user keyword query. The approach is complementary to the other research efforts on ad-hoc object retrieval, it also showcases the importance of the traditional information retrieval methods for ad-hoc querying of the semantic data. The proposed model has been evaluated using the SemSearch 2011 data set, especially designed for the semantic list search evaluation. To our knowledge, the proposed model has state-of-the-art performance on this data set and it brings important improvement in retrieval precision for the given task, compared to the baseline, the state-of-the-art retrieval model for entity search in semantic data. We have also proposed two approaches for the identification of the semantic sets from the knowledge base. The first one relying on the involvement of an expert user, the second one fully automatic. As shown by the experiments the automatic approach has almost the same positive effect on the SemSet model performance as the one guided by the expert user knowledge.

Acknowledgement. This work is supported by projects VEGA 2/0184/10, VEGA No. 2/0054/12, VENIS FP7-284984.

9. REFERENCES

[1] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. T. Duc. Entity search evaluation over structured web data. In *Proceedings of the 1st International Workshop on Entity-Oriented Search (EOS)*, 2011.

[2] S. Campinas, R. Delbru, N. A. Rakhmawati, D. Ceccarelli, and G. Tummarello. Sindice BM25MF at SemSearch 2011. In *System Descriptions from the Semantic Search Challenge 2011*, 2011.

[3] P. Cimiano, P. Haase, and J. Heizmann. Porting natural language interfaces between domains: an experimental user study with the ORAKEL system. In *Proceedings of IUI'07*, 2007.

[4] D. Damljanovic, M. Agatonovic, and H. Cunningham. Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In *Proceedings of ESWC'2010*, 2010.

[5] Ó. Ferrández, R. Izquierdo, S. Ferrández, and J. L. V. González. Addressing ontology-based question answering with collections of user queries. *Information Processing and Management*, 45(2):175–188, 2009.

[6] D. A. Ferrucci, E. W. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. M. Prager, N. Schlaefer, and C. A. Welty. Building Watson: an overview of the DeepQA project. *AI Magazine*, 31(3):59–79, 2010.

[7] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In *Proceedings of WWW'11*, 2011.

[8] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, and D. T. Tran. Evaluating ad-hoc object retrieval. In *Proceedings of IWEST'2010*, 2010.

[9] E. Kaufmann, A. Bernstein, and L. Fischer. NLP-Reduce: a "naïve" but domain-independent natural language interface for querying ontologies. In *Proceedings of ESWC'2007*, 2007.

[10] V. Lopez, A. Nikolov, M. Fernandez, M. Sabou, V. Uren, and E. Motta. Merging and ranking answers in the semantic web: The wisdom of crowds. In *Proceedings of ASWC'09*, 2009.

[11] V. Lopez, V. Uren, E. Motta, and M. Pasin. AquaLog: an ontology-driven question answering system for organizational semantic intranets. *Journal of Web Semantics*, 5(2), 2007.

[12] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. Rijke. Learning semantic query suggestions. In *Proceedings of ISWC'09*, 2009.

[13] P. Mika, E. Meij, and H. Zaragoza. Investigating the semantic gap through query log analysis. In *Proceedings of ISWC'09*, 2009.

[14] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of AAAI'2008*, 2008.

[15] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proceeding of CIKM'08*, 2008.

[16] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of WWW'10*, 2010.

[17] C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of WWW'04*, 2004.

[18] V. Tablan, D. Damljanovic, and K. Bontcheva. A natural language query interface to structured information. In *Proceedings of ESWC'08*, 2008.

[19] C. Unger and P. Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *Proceedings of NLDB'2011*, 2011.