Learning useful kick-off ontologies from Query Logs

Konstantinos Kotis  
Dept. of Information & Communications Systems Eng.  
University of the Aegean  
Karlovasi, Samos, Greece  
kotis@aegean.gr

Andreas Papasalouros  
Department of Mathematics  
University of the Aegean  
Karlovasi, Samos, Greece  
andpapas@aegean.gr

Abstract— Ontology engineering has been fully or partially practiced by knowledge engineers or knowledge workers, towards delivering either fully fledged conceptualizations of domains or providing lightweight ontology versions for less demanding but more frequent knowledge tasks. Domain-specific information can be shaped into ontologies either manually or (semi-)automatically using ontology learning techniques. The aim of the paper is to present a novice ontology learning approach that automatically constructs kick-off and useful ontologies from query logs. We place tasks related to the proposed learning approach in all phases of an ontology engineering life-cycle. By providing knowledge workers a useful kick-off ontology that is automatically built from “their needs (i.e. users’ search interests)” in order to address “their needs (i.e. use of the kick-off ontology to query data precisely)”, an approach that contributes as an incentive in the semantic content creation bottleneck is introduced.

Keywords- query logs; ontology learning; OE methodology, semantic content creation; SW incentives

I. INTRODUCTION

The original vision of the Semantic Web still remains to some extent unrealized. Web-scale automated computer interaction and intelligent information processing technology still have to prove their producing added value for humans. Nevertheless, the Semantic Web community was very active during the past decade and their efforts resulted in a wide range of maturing methodologies, methods, and tools for creating, processing, managing and using semantic content: ontologies or RDF data. However, a critical mass of useful semantic content is missing: Web users can only find few well-maintained and up-to-date domain ontologies on the Web and even though recently growing, the amount of RDF data publicly available is limited compared to the size of the traditional Web.

One reason for this problem is the lack of user involvement in semantic content creation tasks. Only a small number of Web users, typically members of the Semantic Web community, annotate their Web resources semantically or build and publish ontologies. This is in competition with several Web 2.0 applications, such as Wikipedia, Del.icio.us, Flickr, YouTube, Facebook or LinkedIn, which exhibit great popularity and user involvement and generate huge amounts of data at comparatively low costs and impressively high quality. To encourage large-scale user participation in the Semantic Web content creation, the research community has to look into incentives that will encourage and motivate humans to become part of the Semantic Web movement, to contribute their knowledge and time to create useful ontologies and use these in annotating documents, images, videos or even Web services.

By the term “useful” we describe domain ontologies that have been developed with the aim to represent specific domain knowledge that is or will be (possibly) most commonly searched by Semantic Web users. Their contribution in the Semantic Web Documents (SWDs’) retrieval process can be measured by their effectiveness in their retrieval and ranking of SWDs. For instance, concerning the retrieval of SWDs, the more a query-ontology (i.e. an ontology learned from the formalization of a NL query) is close to the semantics of the retrieved SWDs the more it can be considered a useful one (precision and recall in the retrieval and ranking of SWDs) [1]. Lightweight ontologies (e.g. ontologies including just simple taxonomic relations) can be useful if they can effectively retrieve SWDs in Semantic Web retrieval applications (e.g. in a Semantic Search Engine). However, in domains that are much ambiguous due to the nature of their discourse, an ontology with richer semantics is likely to be more effective for SWDs’ retrieval [1].

In the context of the proposed in this paper approach, human involvement in ontology creation within an organizational setting is distinguished in two types: a) direct involvement i.e. directly, participating (as knowledge workers) in the ontology development via a knowledge authoring task, and b) indirectly, contributing their knowledge as input to an ontology learning task i.e. mining organizational knowledge that is provided in different forms such as documents or queries used to retrieve them. In an ontology engineering life-cycle, both types of human involvement are possible, preferably not exclusively but in combination. Indirect involvement of workers can decrease the effort of the direct one since a learned ontology can be used as a kick-off ontology i.e. as an initial lightweight ontology version that is later refined and evolved). Such a way, knowledge workers are encouraged to participate in the development phase since they do not have to improvise, shaping their conceptualizations from scratch. On the other hand, direct involvement of knowledge workers is necessary
in order to correct and refine the learned kick-off ontology towards emerging richer semantics and vocabulary. Although both types of human involvement are important, the introduction of indirect human involvement to learn kick-off ontologies can be seen as the spur for knowledge workers that will motivate them to be extensively involved in an ontology engineering life-cycle, and consequently to contribute to overcome the Semantic Web content creation bottleneck.

Recent ontology engineering methodologies (HCOME [2], DILIGENT [3], DOGMA[4]) emphasize (a) the incorporation of ontology engineering tasks in knowledge-empowered organizations in ways that are seamless to the day-to-day activities of knowledge workers and (b) the active and decisive involvement of the knowledge workers in all stages of the ontology engineering processes. Particularly, the HCOME methodology accentuates the active and decisive participation of knowledge workers in the ontology life-cycle. Doing so, domain ontologies are developed and managed according to knowledge workers’ abilities, they are developed individually as well as conversationally, and they are put in the context of workers’ experiences and working settings, as an integrated part of workers’ “knowing” process. Besides the methodological issues, leveraging the role of knowledge workers in the ontology life-cycle entails the development of ontology engineering tools that provide greater opportunities for them to manage and interact with their conceptualizations in a direct and continuous way, not only by reusing and combining domain/development knowledge but also by communicating such knowledge between them effectively. In this paper, we aim to present an approach that tackles the bottleneck of semantic content creation by integrating an ontology learning task in the ontology engineering life-cycle of HCOME methodology. The main key issues of the approach are summarized as follows:

a. The proposed approach effectively supports the active and extensive involvement of humans in the development of domain ontologies by automatically learning kick-off ontologies. The approach extends the HCOME ontology engineering methodology (and any other collaborative methodology that misses such task) by adding an ontology learning task. The tight integration of an ontology learning task with OE methodologies has been recognized as a very important challenge for ontology engineering and evaluation and a crucial invest in the development of new ontology engineering methodologies which will be able to integrate the results of ontology learning systems in the OE process, keeping at the same time user involvement at a minimum level while concerned with the maximization of the produced ontologies’ quality (with respect to a particular domain) [5]. At the same time, the integration of an ontology learning task with OE methodologies serves as a mean for manually evaluating the ontology learning results (iteratively or stage by stage), avoiding such way the propagation of errors. We conjecture that in order to create ontologies of sufficient conceptual preciseness as well as of rich semantics, ontology learning results should be further engineered by other OE methodology phases i.e. evaluation, develop and maintenance. Learned ontologies may also be used within the development and maintenance phase, for extending existing domain ontologies (ontology merging).

b. The proposed approach effectively supports the learning of useful ontologies from query logs. Such ontologies can be used in Semantic Search applications (e.g. SAMOS [1]) or in the reformulation and enrichment of keyword-based traditional search (with semantics).

Ontology learning in general is an important task in OE methodologies. Learning ontologies from query logs in particular can be seen as an added value to this technological tie-up since query logs, comparing with other sources for ontology learning such as domain-specific text corpora or/and tags in folksonomies, they:

a. Directly reflect knowledge workers’ search interests [7]. Users (both Web users and knowledge workers in an Organizational K.M setting) uncover their needs for retrieving specific information by placing domain-specific queries. Such users are indirectly involved in ontology development by contributing their queries into an ontology learning task.

b. Are less noise and easier to process than text since queries are usually expressed in an extremely synoptic manner (a short sequence of keyword terms), focused on specific views in the domain of discourse [5]. Although terms in the bag are just a few (usually 2 or 3) it is possible, in some extend, to apply POS tagging and utilize such information for discovering semantic relations between the query terms. Additionally, using advanced techniques and external knowledge resources such as lexicons it is possible to obtain users’ intended meaning of terms with a quite precise manner [1].

c. Participate in the automatic accumulation of relational knowledge in a Knowledge Grid environment in a unique way. Queries constitute by nature an organizational resource that captures the interlinking between a) organizational content (queries are domain-specific terms that are searched for their inclusion in specific-domain text corpora), b) knowledge workers (queries are constructed by organizations’ knowledge workers thus they reflect their individual information search needs on the available organizational content), and c) content meta-information (queries are usually constructed guided by existing organizational resources’ structure e.g. a topics’ hierarchy). Knowledge acquisition in Knowledge Grid environments must be performed in a modern way, not only by acquiring knowledge from individual experts or knowledge engineers (Individual Knowledge) but also acquiring knowledge form massive contributions of indirect (organizational resources such as text, databases, query logs, tags, etc) involvement of knowledge workers (Relational Knowledge) [7].

In this paper, in contrast to the related approaches, we provide evidences that query logs, when combined with general lexicons or other external knowledge resources, can be used to automatically learn not only kick-off ontologies but also ontologies that are useful in SW applications.
The paper is structured as follows: Section 2 outlines related work, section 3 provides a description of the HCOME methodology revised with related to ontology learning tasks, section 4 presents the proposed approach of learning useful and kick-off ontologies from query logs, section 5 discusses evaluation issues and experimentation, and finally the paper concludes with discussion and future work.

II. RELATED WORK

A similar to the presented in this paper approach has been recently reported by Sekine and Suzuki [6]. A list of predefined name entities (NE) is matched against the query logs and frequencies are counted in order to identify typical contexts of NE categories. The approach, although proposed for the acquisition of ontological knowledge, do not report on issues related to the automatic learning of ontologies. Evaluation of the approach was extensive; however the usefulness of the learned ontologies in real SW applications as well as the comparison to gold-ontologies is not reported.

Another related work concerns the mining of query logs to assist ontology learning from relational databases [8]. The novelty of the approach is that it expands the ontology to the lower level by exploiting the data dimension. Formal Concept Analysis (FCA) method is used in order to build the concept hierarchy semi-automatically. The approach is heavily depended on the schema extracted from the database since it is used as input in the mining of the query log. Evaluation of the constructed hierarchies is done manually. More importantly, the usefulness of the learned ontologies is not measured in terms of evaluating them in real SW applications.

In Gulla et al [9], an unsupervised key-phrase extraction system has been used to speed up the construction of search ontologies. The extracted key-phrases serve as concept candidates in the ontology and give indications for how hierarchical relations should be defined. The learned ontologies are verified manually by domain experts and concepts are related to each other with various hierarchical and associative relationships appropriately (manual work is needed to complete the hierarchies and possibly add more abstract concepts that link everything together in complete ontologies). Evaluation of the usefulness of the learned ontologies in real applications is not reported.

In Park et al [10], a method for building ontologies on demand from scientific queries by applying text mining technologies is presented. The method induces ontological concepts and relationships relevant to the query by analyzing search result documents together with domain-specific knowledge sources available on the Web. The approach is heavily based on the analysis of the returned documents, even if they are in-correctly returned by the search engine. Furthermore, the constructed ontology does not utilize a set of queries and the interrelation of their terms, as the presented in this paper approach proposes, but rather it only formalizes a single query using information extracted only from the query itself.

In ORAKEL [11], a similar to the proposed in this paper approach is presented. However, a target corpus must be available to construct custom lexicons that will then assist the learning method of lightweight ontologies. Furthermore, the constructed ontology does not utilize a set of queries and the interrelation of their terms, as the presented in this paper approach proposes.

Finally, related work concerning the learning of ontologies directly from text corpora (e.g. [12], [13]) as well as semantically enriching tag clouds of Web 2.0 information resources (e.g. [14]) is acknowledged. Such efforts, although related to the ontology learning problem, do not report on the utilization (mining) of query logs. Furthermore, the learned ontologies are not enriched with additional semantics i.e. semantics that are extracted from external knowledge sources such as lexicons (e.g. synonyms, antonyms, meronyms related to the learned classes).

III. THE HCOME REVISED

Placing the proposed approach in the context of a specific ontology engineering (OE) methodology, it aims to advance the potential of reusing automatically learned formal conceptualizations of domain knowledge. Specifically, we aim to advance the HCOME methodology by incorporating an ontology learning task to support the collaborative ontology engineering process. HCOME provides a clear distinction between the different phases of an ontology life-cycle, the goals that should be achieved in each phase and the tasks that can be performed so as to achieve these goals. These tasks are performed iteratively until a consensus has been reached between knowledge workers. Tasks are performed either individually (in the personal space of workers) or conversationally (in a shared space). A worker can initiate any ontology engineering task in his personal space or participate in a task that has been initiated by other members of the community in a shared space. The following paragraphs briefly discuss the major tasks of HCOME phases and extend them with the ontology learning task.

During the HCOME specification phase, knowledge workers are joining groups that are concerned with the development of agreed ontologies. Having identified themselves within a group of collaborators, during this initial phase of ontology engineering, workers are discussing requirements, produce specification documents, and agree on the aim and the scope of the new ontology. The “Specification” phase of the ontology life-cycle is performed conversationally within the shared space and includes:

1. The specification of the scope and aim(s) of the ontology. This is essential in order workers to have an agreed initial reference of the way they understand the domain and the way they want initially to model it, according to their information needs.
2. An argumentation dialogue between the members of the group in order to obtain commonly agreed specification requirements.
3. Recording the agreed specifications in appropriate forms and/or documents.
4. The specification of the information sources which will be used to learn a kick-off ontology (e.g. a set of queries).
Having agreed on the scope and aim of the ontology to be developed, workers in their personal space can follow any approach or combination of approaches to the development of ontologies: They may improvise by integrating new concepts, learn a kick-off ontology from their queries, provide concepts with informal definitions, compare, merge and refine/generalize existing ontologies. Since the consultation of other well-known or widely acknowledged resources is critical to the ontology development process, the collaborators may perform this task before sharing their conceptualizations with others. Collaborators should be able to create, store, maintain, compare, merge, and manage different versions of ontologies or the learned kick-off ontology. The “conceptualization” phase includes the following tasks:

1. The import of existing ontologies, for the reuse of conceptualizations.
2. The learning of kick-off ontologies from information sources (e.g. queries)
3. The consultation of generic top ontologies, thesauruses supporting the decisions on conceptualizing the domain in development of personal ontologies as well as of the kick-off developed in personal spaces to the shared space. Shared ontology is achieved via structured conversation and refinement of terms using Latent Semantic Indexing method to map key terms to WordNet senses [15]. The pre-processing of a domain-specific query log is based on the assumption that all queries are related to a unique domain and thus, their terms should be somehow related between each other. We conjecture that such a relation is present not only between terms of an individual query but also between terms of every query of a particular domain-specific query log. Based on this, the vicinity of each query term is computed as follows:

   i) For a term \( t_i \) that occurs only in one query \( q_j = \{ t_1, t_2, ..., t_k \} \) of the query set \( Q = \{ q_1, q_2, ..., q_n \} \), the vicinity \( V_{t_i} \) of \( t_i \) comprises the rest terms of \( q_j \), i.e. \( V_{t_i} = q_j \setminus \{ t_i \} \).

   2. The browsing/exploitation of the learned ontology, bringing forward kick-off conceptualizations (for evaluation reasons).
3. The comparison of shared versions of an ontology, for identifying the differences between them, or the comparison of the kick-off with its revisions.
4. The posting of arguments upon versions of ontologies for supporting decisions for or against specifications.

Concluding the above, HCOME has been extended by new tasks related to the learning of kick-off ontologies from information sources (Note: learning tasks are presented in italics). Such tasks are integrated in the corresponded phases of the ontology engineering life-cycle, from the specification to the exploitation.

IV. THE ONTOLOGY LEARNING APPROACH

In this paper, an approach for mining domain-specific queries towards learning useful kick-off ontologies is presented. The approach meets the following specific requirements:

a) Learn kick-off domain ontologies to advance the SW content creation, integrating ontology learning tasks in HCOME O.E methodology
b) Learn ontologies from query logs (a subset of domain-specific queries)
c) Enrich learned ontologies with additional semantics that are extracted from external sources such as lexicons in order to improve their exploitation in SW search applications
d) Automate the learning process

In an organizational K.M setting, queries reflect domain-specific users’ search interests. Knowledge workers search their organization’s information spaces looking for domain-related information. Such queries are recorded in organizational query logs, usually, without relating them to information concerning the precision and recall of the returned information. The query log may contain queries that have been already placed in the past in different forms and also may contain queries from different knowledge workers.

The first step of the developed method is the pre-processing of the query log by analyzing the queries and identifying important terms, excluding terms that belong in stop words lists. In addition to this, the neighbor terms of each term in each query are identified. Such information is needed for the disambiguation of terms using Latent Semantic Indexing method to map key terms to WordNet senses [15]. The pre-processing of a domain-specific query log is based on the assumption that all queries are related to a unique domain and thus, their terms should be somehow related between each other. We conjecture that such a relation is present not only between terms of an individual query but also between terms of every query of a particular domain-specific query log. Based on this, the vicinity of each query term is computed as follows:
ii) For a term \( t_i \) that occurs in more than one queries (i.e. an important term), its vicinity \( V_{t_i} \) comprises the rest terms of all queries that the important term is identified i.e. 
\[
V_{t_i} = \bigcup_{q \in \text{eq}} q \setminus \{t_i\}.
\]

The second step is to assign part of speech tags to each query term of each query using a POS tagger (e.g. Stanford POS tagger)\(^1\). This step identifies mainly nouns, verbs, and adjectives in order to be able to apply simple heuristics e.g. for the identification of object properties.

1. Perform query set pre-processing
2. For each query \( q \)
3. Perform query normalization
4. For each key query term \( t \)
5. POS tag \( t \)
6. disambiguate \( t \) using neighbour terms of queries that have a \( t \) occurrence
7. return the mapped WordNet sense \( s \)
8. If \( t \) POS is Noun
9. If \( s \) is Instance
10. find its concept’s hyponym \( ch \) from WordNet
11. add \( ch \) in the ontology as an individual of class \( ch \)
12. add any synonyms of \( s \) as label of the class
13. add any hypernyms up to depth \( \text{UPPER\_DEPTH} \) (=0)
14. add any hyponyms up to depth \( \text{LOWER\_DEPTH} \) (=0)
15. else if \( t \) POS is Verb
16. add \( s \) in the ontology as a class
17. add any synonyms of \( s \) as label of the class
18. add any antonym of \( s \) as a disjoint class
19. add any hypernyms up to depth \( \text{UPPER\_DEPTH} \) (=0)
20. add any hyponyms up to depth \( \text{UPPER\_DEPTH} \) (=0)
21. add any synonyms of \( s \) as label of the class
22. add any antonym of \( s \) as a disjoint class
23. add any hypernyms up to depth \( \text{UPPER\_DEPTH} \) (=0)
24. Else
25. End if
26. End for

---

\( i \) represents the number of queries that the important term is identified i.e. \( \text{query-ontologies} \). The results obtained were encouraging to continue in this direction. More specific, the ontologies learned from free-formed queries (named “query-ontologies”) were used to retrieve Semantic Web Documents (SWD) (OWL ontologies) indexed by SWOOGLE semantic search engine. An OAEI-contest-evaluated ontology mapping tool, namely AUTOMS\(^2\), was used to match the query-ontologies with the SWDs. Such an evaluation is influenced by the precision performance of the ontology mapping tool and on the expressiveness of the SWDs. The evaluated approach was tested for single queries, one at a time, not for a subset of a query log data set. Further experimentation with SAMOS (or other similar semantic search systems) must be conducted. Finally, since latest mapping tools are able to discover

Figure 1. The algorithm for the main functionality of the proposed approach

The third step produces a set of semantically related query terms using the input of step 1 and step 2 and a mapping method to assign WordNet senses to query terms. The output of this step is used by an ontology construction module to transform such information into OWL axioms. In Figure 1 the algorithm for the main functionality of the proposed approach is outlined. The proposed algorithm currently discovers subsumption, synonym and disjoint relations between query terms (using WordNet Hypernym/Hyponym, Synonym and Antonym relative semantic relations between senses) and transforms these into OWL axioms (DL expressivity). Individual objects are also discovered using WordNet API\(^3\) provided functionality.

It must be pointed out that the learned ontology is not just a projection of a terminological subset of WordNet, since non-WordNet terms are also handled by the method. Such terms are a) single terms that have no entry in WordNet and b) compound terms. For instance, a query term lexicalized by the word “ecology-car” will be transformed to a class “ecology-car” and classified under class “car”. Furthermore, a class “Ecology” will be introduced that will be related with class “ecology-car” via a generic role “Related_to”. Different forms of compound terms are handled: “ecology_car”, “ecology-car”, “ecologyCar”.

Finally, it must be accentuated that, in different to other approaches as well as to the work presented in the semantic search system SAMOS [1], the constructed ontology utilizes a set of queries (a domain specific subset of a query log) and the interrelation of their terms in order to learn a single ontology (multiple queries to one ontology mapping, m:1).

V. Evaluation

Since it was rather difficult to collect evaluation data from an organizational setting, we have obtain a licensed Yahoo! query log data set of 1000 queries (most frequent web search queries), provided as part of the Yahoo! Research Alliance Webscope program, to be used for approved non-commercial research purposes [16]. The data, as received, was not classified in domains, and there were no links to users’ selected URL’s. In order to be able to cross-evaluate the presented in this paper approach, we have additionally collected queries from Google Search Engine – Google Toolbar (suggested queries). For 2 different domains, the total of approx. 200 queries was collected.

The proposed approach has been evaluated using queries from both Yahoo and Google datasets, with the following restriction: The terms’ disambiguation is performed using WordNet as external source of knowledge for discovering the semantics of query terms. Other external resources such as a) third party ontologies found in Semantic Web repositories such as SWOOGLE and b) Wikipedia and Wiktionary knowledge resources are left for future work.

The proposed approach has been evaluated in three different ways, as described in the next three sections respectively.

A. Evaluation in Semantic Web Applications

An early version of the presented in this paper approach has been evaluated in a semantic search application context [1]. The results obtained were encouraging to continue in this direction. More specific, the ontologies learned from free-formed queries (named “query-ontologies”) were used to retrieve Semantic Web Documents (SWD) (OWL ontologies) indexed by SWOOGLE semantic search engine. An OAEI-contest-evaluated ontology mapping tool, namely AUTOMS\(^3\), was used to match the query-ontologies with the SWDs. Such an evaluation is influenced by the precision performance of the ontology mapping tool and on the expressiveness of the SWDs. The evaluated approach was tested for single queries, one at a time, not for a subset of a query log data set. Further experimentation with SAMOS (or other similar semantic search systems) must be conducted. Finally, since latest mapping tools are able to discover

---

1 http://nlp.stanford.edu/software/tagger.shtml
2 http://lyle.smu.edu/cse/dbgroup/sw/jaws.htm
3 www.dit.unitn.it/~p2p/OM-2006/8-automs-OAEI06.pdf
mappings between subsumed or disjoint classes of two ontologies, and such semantic relations are learned in the kick-off ontologies of the presented approach, additional experiments must be conducted with alternative to AUTOMS tools.

The proposed in this paper ontology learning approach extends the one evaluated in SAMOS in terms of a) richer semantics that the learned ontology includes (e.g. disjoint classes), and b) better disambiguation of query terms due to the utilization of more than one query in the computation of the vicinity of query terms (bag of words used in the disambiguation method used i.e. mapping query terms to WordNet senses using Latent Semantic Indexing).

B. OE life-cycle evaluation

Manual evaluation of learned ontologies was also conducted. The quality of “Automobile” and “Movies” ontologies in terms of the domain knowledge captured and in terms of the consistency of the formal conceptualizations was examined. Learned ontologies were put in HCOME ontology engineering life-cycle as kick-off ontologies. Appropriate OE tools were used to browse and query the ontologies in knowledge workers personal space, during exploitation and evaluation tasks. Knowledge workers were satisfied with the exploitation of the learned ontologies. However, they have reported missing classes. Such classes were missed by the learning process since WordNet was unable to contribute them: they were not included in the mapped WordNet synsets (e.g. subclasses or instances of a mapped term) or they were specified as labels of the class that corresponded to the mapped synset (due to the way synonyms are handled by the proposed algorithm).

C. Gold ontology and automated tool evaluation

Learned ontologies were also compared with gold domain ontologies. Such evaluation was conducted using the Dellschaft and Staab’s approach [17], re-using the OntEval system. The approach takes two ontologies defined in OWL formats as input, one of which is assumed the Gold Standard (reference) ontology $O_r$ and the other is the machine computed ontology $O_c$; then it performs evaluation by computing measures such as Lexical Precision (LP), Lexical Recall (LR), Taxonomic Precision (TP), Taxonomic Recall (TR), F-Measure (TF). The lexical precision and recall reflect how good the learned lexical terms cover the target domain.

A more detailed description of the method used for the automated and tool-based evaluation of the learned ontologies is out of the scope of this paper. We conjecture that any other state-of-the-art automated evaluation method could be used, given that the input of the method is a gold ontology and a learned one. We also conjecture that any automated ontology mapping method could be used to discover alignments between the learned ontologies and the gold one, uncovering lexical and semantic similarities between the two ontologies.

Indicatively, Table 1 presents results obtained from the automated evaluation of the example learned ontology for the “car repair” domain-specific subset of queries. These results can be considered however “a best case” since the gold ontology provided for the evaluation was built in-house and had many concepts in common with the learned one (biased by the experiment set-up). An extended measurement of such figures must be performed for all the evaluation query logs that we have collected (both from Yahoo! and Google).

<table>
<thead>
<tr>
<th>TABLE I. GOLD-STANDARD EVALUATION RESULTS FOR THE EXAMPLE QUERY SUBSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Concepts in Reference Ontology: 93</td>
</tr>
<tr>
<td>Total Concepts in Computed Ontology: 100</td>
</tr>
<tr>
<td>Total Common Concepts in both ontologies: 92</td>
</tr>
<tr>
<td>LR($O_r$, $O_c$) = 0.989247311827957</td>
</tr>
<tr>
<td>TP($O_r$, $O_c$) = 1.0</td>
</tr>
<tr>
<td>TR($O_r$, $O_c$) = 0.9879227053140096</td>
</tr>
<tr>
<td>TF($O_r$, $O_c$) = 0.9939246658566221</td>
</tr>
</tbody>
</table>

VI. DISCUSSION

Experimenting with other types of input data, as an alternative to the proposed approach, could provide valuable feedback. Yahoo! Answers service provides a significant information space where concepts and semantic relations related to user queries can be more easily extracted. This is based on the fact that Yahoo! Answers provide an easy and highly popular way for users to post NL questions (queries), related them with specific categories (concepts in Yahoo! categories), and more importantly, to related them with specific user-intended answers (resulted documents), replied from Web users or from experts in the domain knowledge (Knowledge Partners’ Organizations).

Furthermore, it must be pointed out that the presented work is aligned with the latest proposal towards a “pragmatic Semantic Web” [18]. According to the pragmatic view of the SW, supported by the Web Science research initiative (http://webscience.org/) [19], and its impact on the real problems when applying SW technologies in organizational settings, the need of simple lightweight but useful (for querying RDF linked data) ontologies is more realistic than other related efforts.

In this paper the mining of query logs in order to build useful kick-off ontologies in an automatic fashion is presented. Such an approach extends the HCOME O.E methodology and contributes to the SW bottleneck of semantic content creation by supporting knowledge workers in early stages of ontology engineering, encouraging them to contribute their conceptualizations in order to learn, refine and exploit a machine-developed kick-off ontology. Although the presented approach performs well in terms of usefulness of the learned ontologies in the context of OE and SW applications (Search), further actions can be taken towards learning ontologies of richer semantics and vocabularies. Furthermore, the application of the approach in the open Web remains a hot issue that is under investigation. Experimentation with techniques for classifying Web queries into domain-specific subsets of query log data sets is in progress.
REFERENCES


