Mining Query-logs towards Learning Useful Kick-off Ontologies: an Incentive to Semantic Web Content Creation

Konstantinos Kotis , Andreas Papasalouros,
Manolis Maragoudakis

AI-Lab, Dept. of Information & Communications Systems
Engineering, University of the Aegean,
Karlovassi, Samos 83200, Greece
E-mail: {kotis, andpapas, mmarag}@aegean.gr

Abstract: The aim of the paper is to present an ontology learning method that automatically (unsupervised) builds useful kick-off ontologies from query logs. We introduce tasks related to the proposed method in all phases of an ontology engineering lifecycle, extending human-centered and collaborative engineering methodologies for devising continuously evolving ontologies such as HCOME (*H*uman *C*entered *O*ntology *E*ngineering *M*ethodology). Users, both knowledge workers of an organizational knowledge management setting and WWW users, are provided with a useful kick-off ontology that is automatically built from users' *search interests* in order to meet ontology engineering and ontology-based application needs. The paper provides evidences that this new approach plays a significant role as an *incentive* in the semantic content creation *bottleneck*.

Keywords: query logs, ontology learning, ontology engineering methodology, semantic content creation, Semantic Web incentives

1 Introduction

The Semantic Web still remains to some extent unrealized. Intelligent information processing technology has to prove its added value both for human and software agents. 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, both ontologies and RDF data. However, a critical mass of *useful* semantic content is missing; Web users can only find and use few well-maintained and up-to-date domain ontologies in their knowledge-based tasks and the amount of RDF data publicly available is limited compared to the size of the unstructured Web information.

One reason for this is the lack of motivation for users to be involved 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 contrast 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. To encourage and motivate large-scale user participation in the Semantic Web (SW) content creation, the research community has to look into what it is called SW incentives: to motivate humans to become part of the Semantic Web movement, to contribute their knowledge and time to create useful ontologies and to use them in annotating documents, images, videos or even Web services.

A *kick-off* (i.e. starting) ontology is similar to a lightweight ontology (an ontology with simple semantics such as a hierarch of classes) that has been developed automatically using an ontology learning method and is used as a starting (kick-off) point by knowledge workers/engineers in order to assist them in the ontology development lifecycle. Generally, a lightweight ontology may not be consistent and may contain only a small set of ontological axioms such as subsumption, and just a small number of classes (no individuals or properties). In this paper a kick-off ontology is considered a rich (more axioms than subsumption are added) and consistent version of a lightweight ontology.

A *useful* ontology is an ontology that plays a significant role mainly in the ontology development lifecycle of a collaborative and human-centred ontology engineering methodology. The importance of a useful ontology (usefulness) can be shaped in the following tasks:

a) consultation of a kick-off ontology during a process of improvising an ontology from scratch
b) reuse of a kick-off ontology in a process of developing an ontology (merge with another ontology)
c) comparing of a kick-off ontology with an improvised ontology and reusing (copy) parts of it

Furthermore, the paper considers learned kick-off ontology to be also a query-ontology i.e. a kick-off ontology that has been learned from query logs. Because of that, its usefulness must be also considered in terms of their exploitation in ontology-based applications. A kick-off query-ontology is a useful ontology since it plays also a significant role in ontology-based application environments. The importance of a useful ontology in terms of its use in applications can be shaped in the following tasks:

d) view/browse knowledge that users want to retrieve in a formal and structured form that the learned ontology provides
e) annotate documents/data that users want to query using the semantics of the learned ontology
f) use the learned ontology to re-formulate/enrich natural language queries towards retrieving unstructured information
g) use the learned ontology in formal queries in order to retrieve Semantic Web documents using ontology matching methods

In other line of this research (Spiliopoulos et al, 2008), a similar approach is presented with the aim to automatically discover and represent specific domain knowledge that will be most commonly used for searching Semantic Web Documents (SWDs) by Web users. The contribution of kick-off query-ontologies in the retrieval process of SWDs is measured by using and evaluating the ontology within an application context. For instance, concerning the retrieval of SWDs, the more a query-ontology, i.e. an ontology learned through the formalization of a single natural language query, is close to the semantics of the retrieved SWDs (an ontology or RDF data) the more it can be considered a useful one. In this paper, previous research concerning the SAMOS system (Spiliopoulos et al, 2008) is extended in the approach presented in this paper in the following ways: a) the approach deals with the mining of a set of queries that is automatically clustered from a query log in order to develop a single kick-off query-ontology, b) the learned ontology is richer in terms of semantics (e.g. disjoint classes, individuals, synonyms), and c) a different vector-based space model method is used in the disambiguation process 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 with the Latent Semantic Indexing method in order to map query terms to WordNet senses).

The aim of this paper is not to focus on the usefulness of kick-off query-ontologies in terms of ontology-based applications such as in (Spiliopoulos et al, 2008). This paper, in contrast to other related approaches, aims to provide evidences that query logs, when combined with general lexicons or other external knowledge resources, can be used to automatically learn lightweight as well as rich ontologies that are useful in the kick-off phase of an ontology development lifecycle of a collaborative and human-centered ontology engineering methodology for developing continuously evolving ontologies. Having said that, an empirical evaluation that engages end-users has been conducted (user interviews, questionnaire, usability testing), providing evidence regarding the usefulness of the proposed approach in ontology-based applications. Summing up, this paper aims to present a method for learning kick-off ontologies with richer semantics and vocabulary than the lightweight versions of related approaches (disjoint axioms, equivalent classes and individuals).

The paper is structured as follows: section 2 presents and compares related work, section 3 discusses relevant methodological issues, section 4 outlines the learning approach proposed in this paper, section 5 evaluates the approach both in terms of an ontological engineering evaluation approach and an empirical one, and section 6 and 7 provide discussion on limitations, future work and other important issues relate to the work presented in this paper.

2 Related Work

In Sekine and Suzuki (2007) 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. For instance, typical contexts identified for category “awards” are “winners”, “history”, “nominations” since the queries “academic + awards + winners”, “academic + awards + history” and “academic + awards + nominations” appeared in the query logs 86, 76 and 74 times respectively. A co-occurrence normalization formula is used to penalize frequent contexts for each category, appearing very often regardless of the category. The approach, although proposed for the acquisition of ontological knowledge, does not focus on issues related to the automatic learning of ontologies. Evaluation of the approach was extensive; however the usefulness of the learned ontologies in an ontology development lifecycle or in ontology-based application is not reported. An evaluation of the learned ontology against a gold one (an ontology manually developed by domain experts and knowledge/ontology engineers) is also not reported.

Another related work concerns the mining of query logs to assist ontology learning from relational databases (Zhang et al, 2006). The novelty of the approach lies in the expansion of the ontology to the lower level by exploiting the data dimension. Consider an example database, if a query to the ‘Person’ table with the WHERE clause equals to “age > 60 and gender = ‘male’” is frequently executed by a user, one may agree that a concept “elder man” should be elicited to be the sub-concept of ‘Person’. Additionally, the approach proposes a set of rules for schema extraction that provides initial input. Formal Concept Analysis (FCA) method is used to build the concept hierarchy semi-automatically. The approach depends 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 an ontology development lifecycle or in ontology-based application.

In Gulla et al (2007), 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 of how hierarchical relations should be defined. The candidate phrases were weighted using the *tf.idf* measure (*term frequencies*) used in information retrieval. This is a lightweight ontology learning approach, addressing the problem of searching in an adequate manner. 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 an ontology development lifecycle or in ontology-based application is not reported.

In Park et al (2003), 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 system processes documents returned by a search engine to find terms semantically related to the target query. It also identifies the relationships in which they participate. The ontology constructed is a lightweight ontology because it defines only the concepts represented by the query terms. The approach is heavily based on the analysis if the returned documents, even if they are incorrectly returned by the search engine. Furthermore, the constructed ontology does not utilize a set of queries and the interrelation of their terms, but rather it only formalizes a single query using information only from the query itself (not from the query set).

In ORAKEL (Cimiano et al, 2007) a similar 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.

Finally, related work concerning the learning of ontologies directly from text corpora that has been already conducted (e.g. Cimiano et al, 2004; Zavitsanos et al, 2007) as well as semantically enriching tag clouds of Web 2.0 information resources (e.g. Angeletou, 2008) 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 of the abovementioned related work do not include additional to the input data semantics i.e. semantics that are extracted from external knowledge sources such as lexicons (e.g. synonyms, antonyms) thus they cannot be considered as rich as the kick-off query-ontologies learned from the proposed approach.

3 Methodological Issues: HCOME revised

Human involvement in ontology creation can be distinguished in two types: a) direct involvement i.e. directly participating in the ontology development via a knowledge authoring task, and b) indirect involvement, i.e. contributing knowledge via an ontology learning task e.g. mining documents or queries to extract knowledge. In an ontology development lifecycle, both types of human involvement are possible, preferably not exclusively but in combination. Indirect involvement of humans can decrease the effort of the direct one since a learned ontology can be used as a kick-off ontology (as input to an ontology refinement and evolution process). In such a way, humans are encouraged to participate in the development phase since they do not need to specify their conceptualizations from scratch. On the other hand, direct involvement of humans is necessary in order to correct and refine the learned ontology towards a consistent ontology with richer semantics and vocabulary. Thus, 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 the ontology development lifecycle, thus to contribute in the Semantic Web content creation.

Table 1. The HCOME methodology phases to ontology engineering. *S* denotes the execution of a task in a *S*hared space and *P* in the *P*ersonal space. Tasks in bold letters extend the methodology in this revision by integrating "ontology learning" related tasks.

Lifecycle	Goals	Tasks
Specification	Define aim / scope/ requirements/ teams	▪ discuss requirements (S) ▪ produce documents (S) ▪ identify collaborators (S) ▪ specify the scope, aim of the ontology (S) ▪ **specify domain-related queries set (S)**
Conceptualisation	Acquire & learn knowledge	▪ import from ontology libraries (P) ▪ consult generic top ontology (P) ▪ consult domain experts by discussion (S) ▪ **learn kick-off ontology from queries (P)** ▪ **consult kick-off ontologies**
	Develop & Maintain Ontology	▪ improvise (P) ▪ manage conceptualizations (P) ▪ merge versions (P) ▪ compare own versions (P) ▪ generalize/specialize versions (P) ▪ add documentation (P) ▪ **re-use learned kick-off ontology (P)** ▪ **compare with kick-off ontology (P)**
Exploitation	Use ontology	▪ browse domain ontology (P) ▪ exploit in applications
	Evaluate ontology	▪ initiate arguments and criticism (S) ▪ compare others' versions (S) ▪ browse/exploit agreed ontologies (S) ▪ manage recorded discussions (S) ▪ propose new ontology versions (S) ▪ **browse/exploit kick-off ontology (S)**

Recent ontology engineering methodologies such as HCOME (Kotis and Vouros, 2006) and DILIGENT (Tempich et al, 2006) emphasize on (a) the incorporation of ontology engineering tasks in knowledge-empowered organizations in ways that are seamless to the day-to-day activities of the organization members and on (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 development lifecycle. 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 lifecycle entails the development of ontology engineering methods and 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.

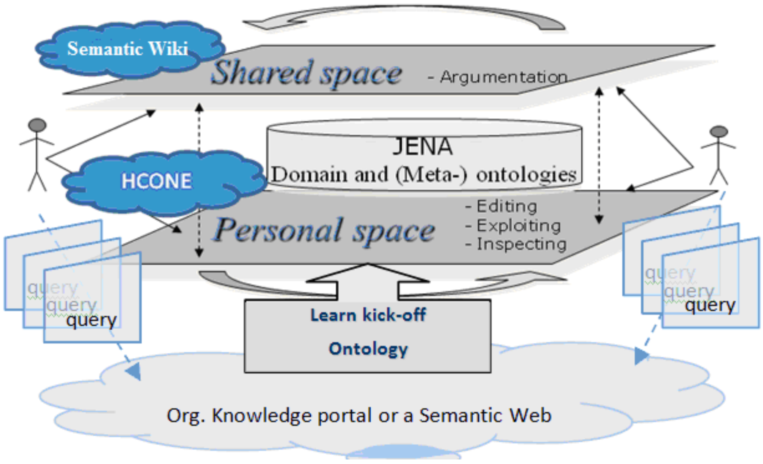

Fig. 1. The HCOME workflow extended with an "ontology learning-from-queries" task.

The paper presents an approach that tackles the bottleneck in semantic content creation by integrating tasks related to ontology learning in the ontology development lifecycle of HCOME methodology. Viewing the approach in the context of the specific Ontology Engineering (OE) methodology, it aims to advance the potential of consulting or reusing automatically learned formal conceptualizations of domain knowledge. Specifically, the aim is to advance the HCOME methodology and consequently to extend the HCOME workflow (Figure 1) by incorporating an ontology learning task to advance the human-centred collaborative ontology engineering process. The ontology-learning-related tasks that extend the HCOME methodology are presented in the "Tasks" column of Table 1 with bold letters.

HCOME methodology provides a clear distinction between the different phases of an ontology development lifecycle, 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/engineers.

Tasks are performed either individually (in the *personal space* using stand alone tools such as the HCONE[1] tool for the editing, exploitation and inspecting of personal ontologies) or conversationally (in a *shared space* using Web 2.0 and Semantic Wiki technology such as a SharedHCONE[2] tool that supports the recording of structured dialogues and argumentations). A knowledge worker/engineer can initiate any ontology engineering task in his personal or shared space, or participate in a task that has been initiated by other members of the community.

During the HCOME specification phase, knowledge workers/engineers 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/engineers are discussing requirements, produce specification documents, and agree on the aim and the scope of the new ontology. The “Specification” phase of the ontology lifecycle is performed conversationally within the shared space and includes:

a) The specification of the scope and aim(s) of the ontology. This is essential in order for workers/engineers 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.
b) An argumentation dialogue between the members of the group in order to obtain commonly agreed specification requirements.
c) The recording of the agreed specifications in appropriate forms and/or documents.
d) 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/engineers can follow any approach or combination of approaches to the development of ontologies in their personal space: 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:

a) The import of existing ontologies, for the reuse of conceptualizations.
b) The consultation of generic top ontologies, thesauruses and domain resources, for better understanding and clarification of the domain conceptualizations.
c) The from-scratch development of formal ontologies based on workers/engineers’ perception on the domain.
d) The mapping, merging and management of multiple versions of ontologies, supporting reuse and evolution.
e) The comparison of different versions of an ontology for inspecting ontologies’ evolution and for identifying ontologies that can possibly be merged.
f) Attaching to ontology classes/properties information items with further comments, examples and specification details.

[1] http://icsd-ai.aegean.gr/hcone
[2] http://icsd-ai.aegean.gr/sharedhcone

g) *The learning of kick-off ontologies from information sources (e.g. queries)*
h) *The consultation of a kick-off ontology during a process of improvising an ontology from scratch*
i) *The reuse of a kick-off ontology in a process of developing an ontology (merge with another ontology)*
j) *The comparison of a kick-off ontology with an improvised ontology, reusing (copy) parts of it*

According to HCOME, the need to achieve a common understanding of a domain inevitably pushes ontology developed in personal spaces to the shared space. Shared ontologies can be used within workers/engineers' settings, in the context of specific ontology-driven applications and tasks. The exploitation and assessment of an ontology version that has been developed by colleagues is seen as part of the ontology lifecycle, since it may provide feedback on the conceptualizations developed. The evaluation and further development of personal ontologies as well as of the kick-off ontology is achieved via structured conversation and criticism upon the ontology versions posted in the shared space. The recording of this conversation enables the tracking of changes and rationale behind ontology versions, supporting the decisions on conceptualizing the domain in the final ontology. The "Exploitation" phase includes:

a) The inspection of agreed or shared ontologies - either by individuals in their personal space or by collaborators in the shared space - for reviewing, evaluating and criticizing the specified conceptualizations.
b) The comparison of shared versions of an ontology, for identifying the differences between them, or the comparison of the kick-off ontology with its revisions.
c) The posting of arguments upon versions of ontologies for supporting decisions for or against specifications.
d) *The browsing/exploitation of the learned ontology, bringing forward kick-off conceptualizations (for evaluation reasons).*

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 the lists of tasks using italics). Such tasks are integrated in the corresponded phases of the ontology engineering lifecycle, from the specification to the exploitation.

The proposed approach effectively supports the active and extensive involvement of humans in the development of domain ontologies by automatically learning kick-off ontologies within an ontology development lifecycle. It extends the HCOME ontology engineering methodology (and any other OE methodology which lacks of such a task) by adding ontology-learning-related tasks. The proposed approach can be used in large-scale and open environments such as the Web, with an unsupervised and automated process of learning domain ontologies. The learned ontologies can represent rich conceptualizations of a domain, forming expressive and consistent OWL ontologies. Ontologies created from such an approach should be further evaluated and evolved, placing them in an ontology engineering methodology, before they can be used in real Web applications.

4 The ontology learning method

In this paper an approach for mining domain-specific queries towards learning useful kick-off query-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-related tasks in HCOME O.E methodology

b) Learn such ontologies from domain-specific query logs. The approach can take input of query logs from the open Web, e.g. Search Engines query logs, given that a pre-processing step is performed for the classification of queries in domain-specific query sets (clustering method)
c) Learn semantically rich ontologies using semantics that are extracted from external sources such as lexicons
d) Automate the learning process (unsupervised process)

4.1 The learning method

Query logs reflect knowledge workers' domain-specific search interests. Knowledge workers query information spaces, searching for domain-related information. Such queries are recorded in query logs, usually without linking them to meta-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.

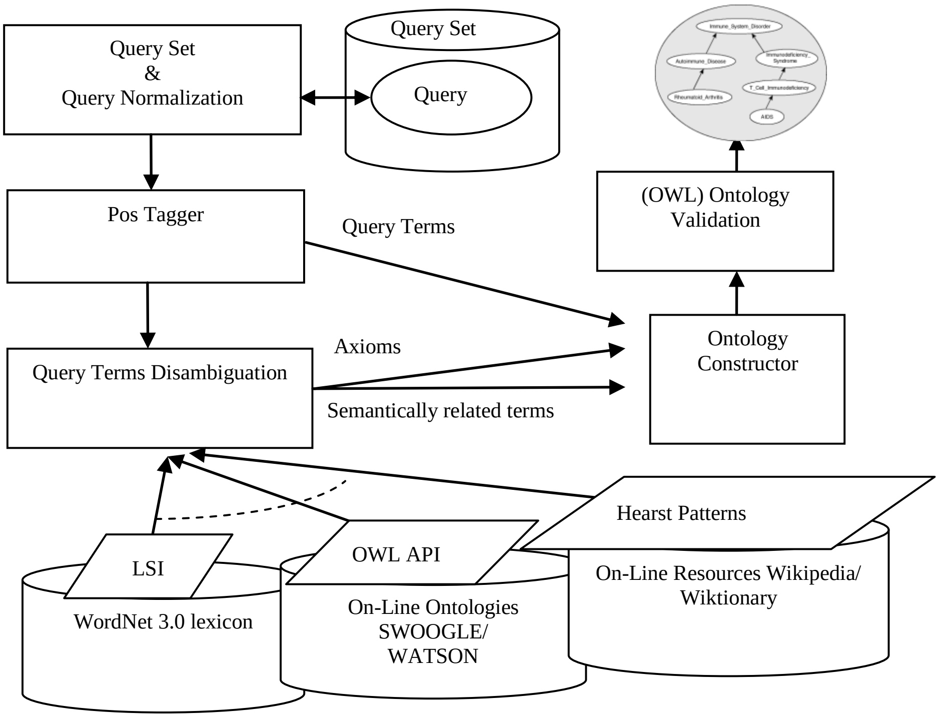

Fig. 2. Architecture of the proposed approach

The first step of the proposed method is to analyze the query set and identify the important terms i.e. terms that occur frequently (more than one time). In addition to this, the neighbour terms of each term (called the *vicinity* of a term) in every query are identified. Stop words are excluded from the vicinity of each term. Such information is needed for the disambiguation of terms using Latent Semantic Indexing (LSI) method to map key terms to WordNet senses (Kotis et al, 2006). The analysis 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, a domain-related one, is present not only between terms of an individual query but also between terms of every query of a particular domain-specific query log. Formally, the vicinity of each query term is computed as follows:

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

ii) For a term t that occurs in more than one queries (i.e. an *important* term), its vicinity V_t comprises the rest terms of all queries that the important term is contained i.e. $V_t = \bigcup_{t \in q_j} q_j \setminus \{t\}$.

```
(Perform query set pre-processing)
For each query q
  For each key query term t
    POS tag t,
      disambiguate t using neighbour terms of queries that have a t
        occurrence
    return the mapped WordNet sense s
    If t.POS is Noun
      If s is Instance
        find its concept's hypernym ch from WordNet
        add ch in the ontology
        add s in the ontology as an individual of class ch
      End if
      Else (* s is a class *)
        add s in the ontology as a class
        add any synomys of s as label of the class
        add any hypernyms up to depth UPPER_DEPTH (>=0)
        add any hyponyms up to depth LOWER_DEPTH (>=0)
    Else if t.POS is Verb
       add  s in the ontology as a class
       add any synomys of s as label of the class
       add any antonym of s as a disjoint class
       add any hypernyms up to depth UPPER_DEPTH (>=0)
    End if
  End for
End for
```

Fig. 3. The algorithm for the main (WordNet-based) functionality of the proposed approach

LSI method maps bags of query terms to WordNet synsets. Doing so, the method computes a mapping between each of these terms and a “hidden intermediate” ontology (Kotis et al, 2006). This mapping is computed by the LSI method which assumes that there is an underlying latent semantic space that it estimates by means of statistical techniques using an association matrix ($n \times m$) of terms-documents: Documents in our case correspond to WordNet senses. The use of a latent semantic space helps to deal with problems of imprecise and vague queries' descriptions, as well as with cases of polysemy and synonymy that other traditional techniques such as Vector Space Models (VSM) cannot handle. LSI is implemented via Latent Semantic Analysis (LSA) which computes the arrangement of a *k*-dimensional semantic space to reflect the major associative patterns in the data. This is in particular done by deriving a set of *k* uncorrelated indexing factors. Then, each term and document is represented by its vector of factor values, indicating its strength of association with each of these underlying latent "concepts". By virtue of dimension reduction from the *N* terms space to the *k* factors space, where $k<N$, terms that did not actually appear in a document may still end up close to the document, if this is consistent with the major patterns of association in the data.

The second step of the proposed ontology learning method is to identify the part of speech (POS) for each query term of each query, using a general POS tagging algorithm such as the Stanford tagger[1]. Future work on this step will focus on alternative techniques to POS identification to avoid incorrect tagging due to taggers inability to perform well with very small queries (information absence). 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.

Table 2. Learned ontology specifications in RDF triples form for a subset of "car repair" domain query log (query terms in bold and underlined form indicate correspondence with the presented triples).

Queries	**RDF triples form (<** Subject, **predicate,** Object **>)**
car engine **fix** in **Australia**	<Repair **rdf:subClassOf** Improvement>
	<Repair **rdfs:label** Fix>
replace her broken **car** **window**	…
	<Country **rdfs:subClassOf** Thing>
repair a busted **car** **window**	<Australia **rdf:type** Country>
	…
auto mobile **repair**, instructions for breaks	<Break **rdf:subClassOf** Destroy>
	<Break **owl:disjointWith** Repair>
ecology-car **repair** instructions	…
	<Window **rdf:subClassOf** Opening>
not-in-wordnetTerm **car seat** **repair**	<Car-window **rdf:subClassOf** Window>
	…
	<Car **rdf:subClassOf** Motor_vehicle>
	<Ecology-car **rdf:subClassOf** Car>
	…
	<Related_to **rdfs:domain** Ecology-car>
	<Related_to **rdfs:range** Ecology>
	…
	<Term **rdfs:subClassOf** Thing>
	<wordnetTerm **rdfs:subClassOf** Term>
	<not-in-wordnetTerm **rdfs:subClassOf** wordnetTerm>
	…
	<Seat **rdfs:subClassOf** Support>
	<Car-Seat **rdfs:subClassOf** Seat>

The third step is the core step of the proposed approach since it takes as input the first

[1] http://nlp.stanford.edu/software/tagger.shtml

and second step and a mapping method to assign WordNet senses to query terms, and returns in the output a set of semantically related query terms. The output of this step is used by an ontology construction module to transform such information into a W3C standard formalism i.e. OWL (Ontology Web Language). In Figure 3 the algorithm for the main (WordNet-based) functionality of the proposed approach is presented. 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). Individual objects are also discovered using WordNet API[1] provided functionality.

It must be pointed out that the learned ontology is not just a projection of a WordNet terminological subset, although currently it depends on it, since non-WordNet terms are also handled by the method (not depicted in the algorithm of Figure 3 for presentation reason). Such terms may be: a) single terms that have no entry in WordNet and b) compound terms. For instance, a query term lexicalized by the word "ecology-car" is transformed by the proposed algorithm to a class "ecology-car" classified under an introduced 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 also handled by the algorithm, using heuristic rules. For instance, terms such as "ecology_car", "ecology-car", 'ecologyCar" are also identified equally as compound terms.

Finally, it must be accentuated that, in extend to other approaches (e.g. Spiliopoulos et al, 2008, Cimiano et al, 2007), 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 (many queries to one ontology mapping, m:1). Another extension for future work is the learning of a kick-off ontology for a single query (one query to one ontology mapping, 1:1) using however terms from other "related" queries of the domain-specific subset of the query log that the query belongs to.

To demonstrate the learning algorithm, an example set of queries is used from the "car repair" domain (Table 2). A selected chunk of the related learned OWL specifications is presented in RDF triples form.

4.2 Clustering Web Query Logs

Query clustering is a significant pre-processing task for the application of the proposed approach to the open Web where query logs must be first organized (clustered) into domains, in order to reflect domain-specific users' search interest. A careful choice of a clustering algorithm is of outmost importance. The main characteristics (requirements) that guide our choice for the task at hand can be summed to the following:

a) As Web query log files tend to grow large, the selected algorithm should effectively cope with large data sets, in terms of time and computational cost.

b) No prior input as regards to the number of cluster should be needed, since Web corpora do not reveal the domain in advance.

c) The algorithm should be incremental, since new queries are constantly fed in to search engines.

In the presented approach, it is found that the incremental version of a density-based algorithm, namely Incremental DBSCAN (Ester et al., 1996), meets our requirements. It

[1] http://lyle.smu.edu/cse/dbgroup/sw/jaws.htm

does not consider the number of clusters as an input parameter and, as the name implies, it satisfies our third desired characteristic. A cluster is formed not by some centroid point, but rather by some minimum number of points (*MinPts*), thus eliminating very small clusters and for every point within a cluster, there exists another point of the same cluster whose distance is less than a given threshold *Eps*, thus giving dense clusters. From a technical point of view, the indexing of points is implemented as a spatial catalogue R-tree, a structure which is easy to locate any point from the core points of a cluster. All clusters containing less than the minimum number of points are considered as "noise", therefore they are discarded. Experimental evaluations have depicted that Incremental DBSCAN produces the same output as the original version, only that the former is by far faster than original DBSCAN (Ng and Han, 1994). Furthermore, the complexity is *O(nlogn)* which compared to the complexity of $O(n^3)$ that traditional clustering approaches such as *k-means* present is of major importance. Upon selection of a clustering methodology, the next step considers the selection of a similarity metric.

Experimenting with a Yahoo! and a Google query data set and analysing its terms, authors concluded that a combined similarity metric should be taken into consideration, since the set was consisted of queries, in which keywords, words in their order and phrases could potentially alter the whole content of each query. The following text explains the process of similarity function estimation.

The estimation of this metric is based on Information Retrieval theory. Therefore, each word, except from those belonging to a stop-word list (e.g. *for, in, the, etc*) is stemmed using Porter's algorithm (Porter, 1980) and is weighted according to the *tf*idf* approach. Afterwards, the cosine similarity function is applied:

$$S(q_1,q_2)=\frac{\sum_{i=1}^{k} cw_i(q_1)\,cw_i(q_2)}{\sqrt{\sum_{i=1}^{n} w_i^2(q_1)}\sqrt{\sum_{i=1}^{m} w_i^2(q_2)}}, \quad (1)$$

cwi(q1) and *cwi(q2)* are the weights of the *i-th* common keyword in queries *q1* and *q2* respectively. *wi(q1)* and *wi(q2)* are the weights of the *i-th* keyword of each query respectively, calculated based on the aforementioned *tf*idf* approach.

In case where phrases need to be considered instead of keywords, the same equation applies, but not on keywords but on phrases, which can easily be extracted using a simple noun phrase 'chunker' such the one used in Stamatatos et al (2000) or in De Lima and Pederson (1999).

As search engines tend to retrieve relative documents, given a user's query, this knowledge can be further exploited to identify the semantic similarity of two queries, even if they fail to share common keywords. In order to implement such an approach the following facts are taken into account: consider two sets of documents, *D(q1)* and *D(q2),* that a search system has presented to the user for a given query *q1* and *q2* respectively. Moreover, let *D_click* denote those that the user actually read from the above set (therefore they were clicked). The similarity of the two given queries is denoted by the intersection of the two *D_click(q1)* and *D_click(q2)* sets. The following equation expresses similarity as proportional to the number of commonly selected documents:

$$Sim(q_1, q_2) = \frac{|\, D\_click\,(q_1) \cap D\_click\,(q_2)\,|}{\max(|\; D\_click\,(q_1)\,|, |\, D\_click\,(q_2)\,|)}, \quad (2)$$

The nominator denotes the number of common documents clicked and the denominator expresses the maximum number of documents clicked for each query. Even though the consideration of solely the nominator could provide the similarity between two queries, the use of denominator as well results in a similarity which is proportional to the shared number of clicked (or selected) documents.

Despite its simplicity, the cross-reference similarity is able to relate queries that could not be related using any keyword-based approach. As an example consider the following user inputs: a) World Atlas, b) Google Maps, and c) Microsoft Earth. Using the aforementioned similarity metric, these queries can be related to a common cluster since they share common documents when posed to a search engine (e.g. documents on GPS software), while using only the lexicographic-based metric, these instances would appear separately.

Furthermore, cross-reference similarity is also useful in distinguishing between queries that happen to be worded similarly but stem from different information needs. As an example, consider the case that a user asked for “Aristarhos” and clicked on documents concerning the famous ancient Greek mathematician and astronomer while another user asked about the same term but clicked on documents that contain information about the Greek telescope on the mount of Helmos, Greece. These two cases could easily be separated by the similarity measure since the nominator of equation (2) which corresponds to the co-occurrence of documents for both the famous mathematician and the telescope would be significantly small. Furthermore, such an approach can provide essential information for sense disambiguation applications.

The functionality of this process has been used to cluster Web queries into domains as a requirement of the proposed ontology learning approach. However, it has been already reported in other lines of research that such an approach can also contribute in the actual process of the learning method since it can provide a mean for disambiguating queries: the selected documents returned from user queries provide a user-intended meaning for the query, thus its content can be used to extract important semantics (related concepts and relations between them) for each query term. Further work on this direction shall be conducted.

5 Evaluation

A Yahoo! query log data set (1000 queries from several domains) was obtained from Yahoo! Research Alliance Webscope program (Yahoo! Webscope, 2009). The data was not already classified in domains and there were no links to users’ selected URL’s (authors simulated the queries i.e. place them in real application environment of the search engine, asking in-house evaluators to click on (select) the returned Web page that they thought of as the most relevant to the query).

Additionally, a query log data set using Google Toolbar was also obtained, following specific procedure: users were asked to place queries related to different domains e.g. Automobile, Movies. For each query they placed, 10 additional (suggested by the Google Toolbar) queries were collected. The suggested queries are the most popular queries (from any user) stored in Google's repository for future reference. For each of the selected evaluation domains, a set of around 100 queries (submitted and suggested ones) was collected.

Table 3. Input and output (learned) data of the learning method

	Input	**Output Ontology**				
Domain	**Queries / terms**	**Classes**	**Properties**	**Individuals**	**Equivalent class axioms**	**Disjoint class axioms**
Google-Auto-Repair	6/23	181	1	2	12	1
Yahoo-Auto	9/17	148	0	0	10	0
Google-Auto	100/350	1804	3	684	103	0
Google-Movies	100/296	1703	4	252	94	0

The proposed approach has been evaluated with the following restriction: Terms' disambiguation is performed using semantic relations obtained only from WordNet (external source of knowledge for discovering the semantics of query terms). The rest external resources as depicted in Figure 2 i.e. a) third party ontologies found in Semantic Web repositories such as SWOOGLE or WATSON and b) Wikipedia and Wiktionary knowledge resources are not used in the presented experiments.

Table 3 presents the experiments' identity (input) and the results obtained (output) of the learning method, focusing on the 2 domains i.e. Automobile and Movies. As a general comment on the depicted data, the number of learned classes depends on the number of total terms that input queries contain. The large number of learned classes is justified by the fact that the learning algorithm introduces in the ontology a new (equivalent) class for each WordNet term that is synonym of the mapped (to WordNet) query term. Consequently, the number of individuals that are learned is larger for the ontologies with a larger set of learned concepts. The small or zero number of disjoint axioms is justified be the fact that antonyms in WordNet are quite rare.

5.1 Evaluation of learning method in ontology development lifecycle

The ontologies learned from the proposed ontology learning method were put in the context of an ontology development lifecycle of selected users with different experience/role (both knowledge workers and knowledge engineers). They were asked to assess the quality of the ontologies in terms of how good they reflect the domain of the query set and in terms of the consistency of the formal conceptualizations. More importantly, they were asked to assess the usability of the learned kick-off ontologies within an ontology development lifecycle. To do so, they were given both the learned ontology and the query set that the ontologies have learned from. From the domains of Automobile and Movies (with no particular reason for selecting those), four different ontologies/query-set pairs, with a variety of length (in terms of query number and number of learned ontology elements/axioms), were put in HCOME ontology development lifecycle as kick-off ontologies. The feedback from this evaluation process was taken in two ways: a) by a personal interview, b) by a questionnaire.

	A “kick-off ontology" is a lightweight ontology that has been developed automatically using an ontology learning method. It may be not a consistent ontology.
1	In an ontology development lifecycle, how important is the contribution of a kick-off ontology (High, Medium, Low)?
2	How you usually use an existing kick-off ontology? (Please choose one or more answers) a) as a consultation to your own improvised ontology b) import and re-use it in your own ontology c) compare it with your own ontology and copy parts of it only d) none of the above
	A kick-off **query**-ontology is a kick-off ontology that has been learned from query logs.
3	How useful (High, Medium, Low) is to consult a query-ontology : a) for using its vocabulary to annotate documents/data that you want to query b) for using it as a formal and structured view of the knowledge that you want to query c) for using its vocabulary as a re-formulated/enriched query to retrieve information d) for using it as a formal query to retrieve Semantic Web documents using ontology matching methods
	A kick-off query-ontology can be of different sizes (number of elements/axioms), depending on the query logs that are mined. A relative large kick-off query-ontology (learned from 100 queries) can have more than a 1000 classes and individuals
4	How useful (High, Medium, Low) is to consult a large kick-off query-ontology a) for using its vocabulary to annotate documents/data that you want to query b) for using it as a formal and structured view of the knowledge that you want to query c) for using its vocabulary as a re-formulated/enriched query to retrieve information d) for using it as a formal query to retrieve Semantic Web documents using ontology matching methods
	Considering the query-ontologies provided to you for the domains Auto (1 large, 2 small) and Movies (large), please answer the following questions:
5	In what way they were useful (High, Medium, Low) to you during an ontology development lifecycle a) as a consultation to your own improvised ontology b) import and re-use it in your own ontology c) compare it with your own ontology and copy parts of it only d) none of the above
6	What was your role in the ontology development lifecycle? a) Knowledge Worker b) Knowledge Engineer c) Both

Fig. 4. The questionnaire for evaluating learned ontologies in ontology development lifecycle

The evaluators (8 were selected) were Web users with an academic training in ontology engineering: familiar with the HCOME methodology and with ontology development tools such as Protégé and HCONE. The questionnaire (Figure 4) was given to the evaluators, together with the related material (learned ontologies and query-sets of Yahoo! and Google). Questions 1, 3, 4 and 5 are in three-level Likert scale form, with options High, Medium and Low. A qualitative and quantitative examination of the filled questionnaires is summarized in the following paragraph.

All evaluators considered the contribution of a kick-off ontology of high (4 out of 8 users) or medium (4 out of 8 users) importance. 6 out of 8 of the evaluators reported that they usually use a kick-off ontology mainly in order to compare it with their own ontology and copy parts of it in their own ontology and as a consultation for constructing their own ontology. That is, they find kick-off ontologies useful in the ontology development process. Questioned about the usefulness of query-ontologies in ontology-based applications, 5 out of 8 evaluators reported that using these ontologies to annotate documents they want to query is of high usefulness; 5 out of 8 reported that using ontologies in order to reformulate/enrich queries in order to retrieve information is also of high usefulness. In aggregate, most evaluators found query kick-off ontologies of high usefulness. However, when the question focuses on large kick-off ontologies, most evaluators found such ontologies of medium usefulness. This implies that a trade-off exist between extend and depth of kick-off ontology and the capability of the knowledge engineer to handle the volume of an extended ontology. Concerning the sample ontologies in the domains of Auto mobiles and Movies, most evaluators found it highly useful for comparison with the version of the ontology they were developing and for copying parts of it in their own ontology. However, they think that importing and reusing parts of kick-off domain ontologies is mostly of medium importance. This means that ontology engineers tend to use these pre-existing ontologies in an informal manner.

To conclude, evaluators, playing both the role of knowledge engineer and knowledge worker, found kick-off ontologies useful, in principle. They also found the particular ontologies created by query logs useful to the same extend. Large size of kick-off ontologies seems to obstruct ease of use.

5.2 Evaluation of learned ontologies with a Gold-ontology

Learned ontologies were also compared with gold domain ontologies. Such evaluation was conducted using the Dellschaft and Staab's approach (Dellschaft and Staab, 2006), re-using the OntoEval system[1]. The approach takes two ontologies defined in OWL format as input, one of which is assumed as the gold-standard (reference) ontology and the other as the machine computed ontology; then it performs evaluation by computing measures such as Lexical Precision (*LP*), Lexical Recall (*LR*), Taxonomic Precision (*TP*), Taxonomic Recall (TR), F-Measure (TF). Given a computed core ontology O_C and a reference ontology O_R, the lexical precision (*LP*) and lexical recall (*LR*) are defined as follows:

$$LP(\mathcal{O}_C,\mathcal{O}_R)=\frac{|\mathcal{C}_C\cap\mathcal{C}_R|}{|\mathcal{C}_C|}\qquad LR(\mathcal{O}_C,\mathcal{O}_R)=\frac{|\mathcal{C}_C\cap\mathcal{C}_R|}{|\mathcal{C}_R|}\qquad(1)$$

The lexical precision and recall reflect how good the learned lexical terms cover the target domain. For example (borrowed from Dellschaft and Staab, 2006), if one compares

[1] http://nlp.shef.ac.uk/abraxas/onteval.html

O_{C1} and *O_{R1}* (Figure 5) with each other, one gets $LP(O_{C1},O_{R1}) = 4/6 = 0.67$ and $LR(O_{C1},O_{R1}) = 4/5 = 0.8$.

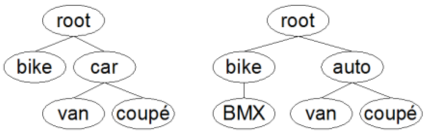

Fig. 5. Example reference ontology (O_{R1}, left) and computed ontology (O_{C1}, right)

For the local taxonomic precision the similarity of two concepts will be computed based on characteristic extracts from the concept hierarchy. Such an extract should characterize the position of a concept in the hierarchy, i.e. two extracts should contain many common objects if the characterized objects are at similar positions in the hierarchy. The proportion of common objects in the extracts should decrease with increasing dissimilarity of the characterized concepts. Given such a characteristic extract *ce*, the local taxonomic precision tp_{ce} of two concepts $c_1 \in O_C$ and $c_2 \in O_R$ is defined as:

$$tp_{ce}(c_1, c_2, O_C, O_R) := \frac{|ce(c_1, O_C) \cap ce(c_2, O_R)|}{|ce(c_1, O_C)|} \qquad (2)$$

Based on the local taxonomic precision (above), and the semantic cotopy sc of a concept (i.e. all its super and sub-concepts), the Global Taxonomic Precision and Recall can be computed. However, if one uses the semantic cotopy for defining the local taxonomic precision measure tp_{sc}, the results will be heavily influenced by the lexical precision of O_C because with decreasing lexical precision more and more concepts of $sc(c, O_C)$ are not contained in O_R and $sc(c, O_R)$. This influence of lexical precision and recall on the taxonomic measures can be avoided if one uses the common semantic cotopy *csc* as the characteristic extract (excludes all concepts which are not also available in the other ontology’s set of concepts). Taxonomic Precision and Recall using the semantic cotopy do not allow a separate evaluation of lexical term layer and concept hierarchy. Thus, the measures TP_{csc} and TR_{csc} have been used, where the building blocks are selected so that the influence of the lexical term layer is minimized. This is achieved by using the common semantic cotopy and by computing the taxonomic precision values only for the common concepts of both ontologies. The latter makes the estimation of local taxonomic precision values unnecessary.

$$\begin{aligned} TP_{csc}(O_C, O_R) &:= \frac{1}{|\mathcal{C}_C \cap \mathcal{C}_R|} \sum_{c \in \mathcal{C}_C \cap \mathcal{C}_R} tp_{csc}(c, c, O_C, O_R) \qquad (3) \\ TR_{csc}(O_C, O_R) &:= TP_{csc}(O_R, O_C) \end{aligned}$$

Finally, the taxonomic precision and recall have to be balanced, therefore the taxonomic F-measure is computed (*TF*), which is the harmonic mean of the global taxonomic precision and recall (The taxonomic F'-measure may additionally be computed, which is the harmonic mean of *LR* and *TF*).

It is assumed 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. It is also assumed 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.

Table 4 presents the results obtained from the automated evaluation of the example learned ontology for the “car repair” domain subset of queries (see Table 2, first column). These results are quite impressive but expected so since the gold ontology provided for the evaluation had many concepts in common with the learned one. The gold ontology was developed by an ontology engineer expert who consulted the related query set and WordNet lexicon as an input to her domain-specific knowledge extraction. An extended measurement of such figures must be performed for all the evaluation query logs that have been collected (both from Yahoo! and Google).

Table 4. Gold-standard evaluation results for the example query subset

```
Total Concepts in Reference Ontology: 93
Total Concepts in Computed Ontology: 100
Total Common Concepts in both ontologies: 92
LP(OcOr) = 0.92
LR(OcOr) = 0.989247311827957
TP(OcOr) = 1.0
TR(OcOr) = 0.9879227053140096
TF(OcOr) = 0.9939246658566221
TF’(OcOr) = 0.991580473
```

5.3 Evaluation of the query log clustering method

The query clustering approach has been evaluated upon Yahoo! and Google query sets. Due to the large number of queries and consequently the plethora of candidate returned web documents, two domains have been selected, namely *Movies* and *Automobile:* From the corpus of Yahoo! and Google queries, we chose the specific domains based on our own interests in such categories, without using any filtering approach to verify the correctness of our decision in terms of returned web pages. For demonstration and space reasons, a fragment of the *Automobile* (from the Yahoo! and Google query set respectively) and *Movies* domain queries clustering approach is depicted in Table 5.

Table 5 tabulates the clustering assignment expressed by queries as cluster members. Some terms were erroneously missed (e.g. ‘*Toyota’* and ‘*Nissan’* were unable to be clustered correctly under the domain), mostly due to the lexicographically reasons i.e. the distance of such terms between others that clearly belong to the domain (such as ‘*auto insurance’*) is significantly greater. Another possible cause is that some Web documents that were returned for queries belonging to company/industry names did not match with each other, even for similar notions (e.g. ‘*Honda’* and ‘*Ford’*). Subsequently, due to probably the same reasons as before, ‘*animation pictures*’ was miss-clustered while ‘*animation movies’* was correctly assigned.

Table 5. The clustering assignment expressed by queries as cluster members for the Automobile and Movies domain

Cluster	Queries found	Queries missed
Auto-Yahoo	auto insurance	Ford

	auto insurance quotes auto trader autozone car insurance quote car rental cars ebays motors nascar	honda nissan toyota
Auto-Google	auto leasing buy car buy vehicles four wheel nascar racing supercars suv used vehicles	city car club off-road wheel
Movies-Goole	action films action films actors action films best animation movies filmmaking film festivals review movies	animation pictures film ticket movie awards

Furthermore, an additional experiment was conducted where the impact that the number of web documents poses upon the average similarity (considered as a measure of density) of each cluster was evaluated. More specifically, 5 query terms from each domain (as regards to the Automobile domain, we have conducted two separate evaluations for queries from Yahoo! and Google respectively) and for each term were used, measuring the similarity with the rest of them by increasingly introducing more Web documents, ranging from 5 to 50. Results demonstrate a similar increase of the cluster density of each domain, as Web documents approach number 30, whereas the introduction of more documents affects the average similarity by a radical decrease, resulting in sparse clusters containing a plethora of outliers (Figure 6).

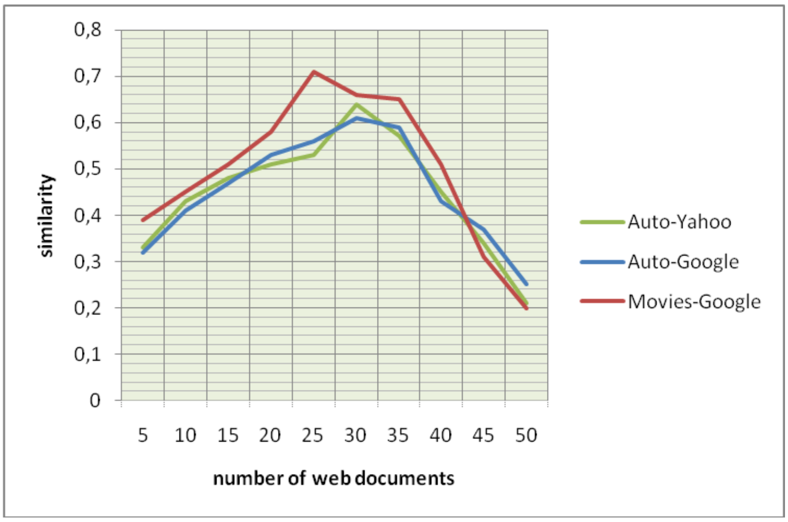

Fig. 6. Evolution of the average similarity measure per domain with respect to the number of Web

documents considered.

Going back to the evaluation of a learned ontology using a gold-standard, the subset of Yahoo! query log (Auto-Yahoo) when compared against a gold ontology for Automobiles (developed by an ontology engineer who was consulting the query set and WordNet), produces acceptable similarity measurements (Table 6), considering the 0.70 score of the *TF'* measure (harmonic mean of lexical recall and taxonomic F-measure).

Table 6. Gold-standard evaluation results for Yahoo! clustered "Auto" query subset

```
Total Concepts in Reference Ontology: 65
Total Concepts in Computed Ontology: 80
Total Common Concepts in both ontologies: 35
LP(OcOr) = 0.4375
LR(OcOr) = 0.5384615384615384
TP(OcOr) = 1.0
TR(OcOr) = 1.0
TF(OcOr) = 1.0
TF'(OcOr) = 0.7000000000000001
```

6 Discussion, limitations and future work

The tight integration of an ontology learning task to OE methodologies has been recognized by the research community 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) (Cimiano et al, 2006). 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 in such way the propagation of errors. It is assumed 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, and should be produced by considering others i.e. requirements specification. Learned ontologies may also be used within the development and maintenance phase for extending existing domain ontologies (consulting, comparing, merging tasks).

The abovementioned conjecture has motivated the work presented in this paper and is inline with the feedback obtained from an empirical evaluation that summarizes human evaluators' feedback on using learned ontologies in ontology development lifecycle. The usefulness of an automatically learned kick-off ontology in ontology development lifecycle has been accentuated by the evaluators: 6 out of 8 evaluators stated that they would use a kick-off ontology in order to compare it with their own ontology and copy parts of it in their own ontology. In addition, 6 out 8 evaluators stated that they would use a kick-off ontology also as a consultation for constructing their own ontology. Consequently, it can be generally stated that users find kick-off ontologies useful in the ontology development process.

Ontology learning in general is an important task in OE methodologies. Learning ontologies from query logs in particular is 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 (Zhuge, 2008). Users (both Web and users of an Organizational Knowledge Management setting) uncover their needs for retrieving specific information by placing domain-specific queries. Such users, acting as knowledge workers during the process of shaping such information (into knowledge), are indirectly involved in the ontology development by contributing their queries into the ontology learning phase of an OE methodology. The resulted ontology reflects their search interests, thus it is expected to contribute in a more effective retrieval process (thus, it can be considered a "useful" ontology).

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 (Cimiano et al, 2006). 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 (Kotis et al, 2006). Furthermore, to improve the process of terms' disambiguation in a query, the utilization of "important" terms extracted from "related" queries (taken from a domain-specific subset of a query log) can be considered.

c) Participate in the automatic accumulation of relational knowledge in a unique way, since it constitutes 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 organization's 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. an ontology). Knowledge acquisition in Web 2.0 or Web 3.0 environments must be performed in a modern way, not only by acquiring knowledge from individual experts or knowledge engineers (Individual Knowledge). Instead, knowledge must be also acquired by massive contributions of direct (human-centred collaborative ontology engineering environments) or indirect (learned ontologies from organizational resources' mining such as text, databases, query logs, tags, etc) involvement of knowledge workers (Relational Knowledge) (Zhuge, 2008).

Again, the abovementioned conjecture has motivated the work presented in this paper and is in line with the results obtained from the questionnaires that summarize human evaluators' feedback on using learned ontologies in ontology-based applications. The usefulness of an automatically learned kick-off ontology in ontology-based applications such as semantic annotation, knowledge viewing, and semantic search has been accentuated by the evaluators: Questioned about the usefulness of query-ontologies in ontology-based applications, 5 out of 8 evaluators reported that using these ontologies to annotate documents that they want to query is of high usefulness; 5 out of 8 evaluators reported that using ontologies in order to reformulate/enrich queries in order to retrieve information is also of high usefulness.

As a last but not least point related to the motivation of the presented work, it must be added that the presented work is aligned with the latest proposal towards a "pragmatic Semantic Web" (Alani et al, 2008). According to the pragmatic view of the SW, supported by the Web Science research initiative (http://webscience.org/) (Hendler et al, 2008), and its impact on the real problems when applying SW technologies in organizational settings, the need of useful kick-off ontologies for querying RDF linked data is more realistic than other related efforts.

6.1 Limitations

It is assumed that the effectiveness of an ontology learning method, and consequently the quality of learned ontologies is influenced by the nature of the queries. *Querying* is not only about following syntactic rules to form a query but also following logical rules related to the way humans express the intended meaning of a query or to the disambiguation of vague meanings. For instance, the query "*breaks auto repair instructions*" although follows some syntactic sugar allowing automated taggers to identify POS, the use of term "*auto*" is ambiguous (especially when used just before the term "*repair*") since it may not be automatically related to the intended meaning of "*car's breaks repair instructions*". Another logical interpretation could be the one of "*instructions for automatic repair of breaks*" where "*breaks*" can be intentionally related not to "*car breaks*" but to, for example, "*airplane breaks*".

Although the proposed method is aware of the abovementioned problems, the following limitations still remain to overcome in future work:

a. There are cases where incorrect POS tagging is frequent due to taggers inability to perform well with very small queries (information absence). Such problem not only depends on tagger performance but on the tagging strategy in general. Future work should involve the experimentation with alternative algorithms that utilize information externally i.e. not in the query but in corpuses related to them.

b. WordNet-based disambiguation of terms using LSI is a promising method that has been used recently in many lines of related research. However, its dependence to a specific lexicon may lead to incorrect results due to information absence. Although non-Wordnet terms are handled by the proposed approach in terms of their specification in the learned ontology, the validation of the intended meaning of these terms is left to humans.

Other limitations of the proposed approach are related to the general strategy of the learning method. Learning conceptualizations from queries must not depend heavily on just one external source (currently, this is WordNet). A combination of sources must be utilized and the union of the learned conceptualizations must be eventually specified in the ontology. Such sources are existing ontology repositories or Web thesauruses/lexicons documents.

6.2 Future Work

In future work, the proposed method will integrate new or extended techniques towards overcoming the abovementioned limitations. In addition, the implementation of a method that uses theory from Hearst Patterns will be applied on Wikipedia/Wiktionary information resources in order to extract semantics that will eventually enrich the kick-off query-ontologies. The union or other kind of operators (e.g. geometric mean) of the Wordnet-extracted semantics and of the semantics returned from this future implementation will be also examined.

Furthermore, the experimentation with other types of input data, as an alternative to the proposed approach that could provide valuable feedback will be conducted. 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 natural language questions (queries), relating them with specific categories (concepts in Yahoo! categories), and more importantly, relating them with specific user-intended answers (resulted documents), replied from Web users or/and from experts in the domain knowledge (Knowledge Partners' Organizations).

Finally, as the functionality of clustering Web queries into domains can also contribute in the actual process of the learning method, it will be examined as a future alternative implementation. Such approach may provide a mean for disambiguating the queries: the selected documents returned from user queries provide a user-intended meaning for the query, thus its content can be used to extract important semantics (related concepts and relations between them) for each query term.

7 Conclusion

An approach for mining query logs to build useful kick-off ontologies in an automatic (unsupervised) fashion is presented in this paper. Such an approach extends the HCOME ontology engineering methodology and contributes to the SW content creation bottleneck by supporting knowledge workers in early stages of ontology development lifecycle, encouraging them to contribute their conceptualizations in order to reuse, refine and exploit automatically learned kick-off ontologies. The ontologies learned with the proposed approach contain richer axioms and vocabulary than the lightweight versions of related approaches (disjoint axioms, equivalent classes, and individuals). Although the presented approach is promising in terms of its usefulness in the context of ontology development lifecycle, further actions can be taken towards improving its accuracy and testing its impact in ontology-based applications such as in semantic search engines.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. Also, we thank the evaluators that contributed with their encouraging feedback.

References

Alani, H., Chandler, P., Hall, W., O'Hara K., Shadbolt N., and Szomszor M. (2008) 'Building a Pragmatic Semantic Web', *IEEE Intelligent Systems*, 23 (3). pp. 61-68

Angeletou, S. (2008) 'Semantic Enrichment of Folksonomy Tagspaces', International Semantic Web Conference, Doctoral Consortium, Karlsruhe, Germany, Proceedings of the ISWC'08 conference

Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S. (2004) ‘Learning taxonomic relations from heterogeneous evidence’, In: Proceeding of ECAI 2004 Workshop on Ontology Learning and Evaluation

Cimiano, P., Haase, P., Heizmann, J. (2007) ‘Porting natural language interfaces between domains – a case study with the ORAKEL system’, In: Proceedings of the International Conference on Intelligent User Interfaces (IUI), pp. 180–189

Cimiano, P., Völker, J., Studer, R. (2006) ‘Ontologies on Demand? - A Description of the State-of-the-Art’, Applications, Challenges and Trends for Ontology Learning from Text, Information, Wissenschaft und Praxis, 57(6-7): 315-320

Dellschaft, K. & Staab, S. (2006) ‘On How to Perform a Gold Standard Based Evaluation of Ontology Learning’, International Semantic Web Conference, 228-241

De Lima, E. F. and Pedersen, J. O. (1999) ‘Phrase Recognition and Expansion for Short, Precision-Biased Queries Based on a Query Log’, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 145-152, ACM Press, 1999.*

Ester, M., Kriegel, H. P., Sander, J., Xiaowei, Xu (1996) ‘A density-based algorithm for discovering clusters in large spatial databases with noise’, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*: 226-231, AAAI Press.

Gulla, J., Borch, H., and Ingvaldsen, J. (2007) ‘Ontology Learning for Search Applications’, R. Meersman and Z. Tari et al. (Eds.): OTM 2007, Part I, LNCS 4803, pp. 1050–1062

Hendler, J., Shadboldt, N., Hall, W., Tim Berners-Lee, and Weitzner, D. (2008) ‘Web Science: An interdisciplinary approach to understanding the World Wide Web’, Communications of the ACM (cover story)

Kotis, K., Vouros, G., Stergiou, K. (2006) ‘Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach’, Elsevier's Journal of Web Semantics (JWS), 4(1): 60-79

Kotis, K. and Vouros, G. A. (2006) ‘Human-Centered Ontology Engineering: the HCOME Methodology’, International Journal of Knowledge and Information Systems (KAIS), 10(1): 109-131

Ng, R.T. and Han, J. (1994) ‘Efficient and effective clustering methods for spatial data mining’, In J. Bocca, M. Jarke, and C. Zaniolo, editors, Proceedings of the 20th Conference on Very Large Data Bases (VLDB), pages 144--155, San Francisco, CA, 1994. Santiago, Chile, Morgan Kauffman Publishers

Park, Y., Byrd, R., Boguraev, B. (2003) ‘Towards Ontologies on Demand’, Proceedings of Workshop on Semantic Web Technologies for Scientific Search and Information Retrieval, In Conjunction with the 2nd International Semantic Web Conference

Sekine, and Suzuki, H. (2007) ‘Acquiring Ontological Knowledge from Query Logs’, WWW 2007, May 8-12, 2007, Banf, Canada

Spiliopoulos, V., Kotis, K., Vouros, G. A. (2008) ‘Semantic retrieval and ranking of SW documents using free-form queries’, Int. J. Metadata, Semantics and Ontologies, 3(2): 95-108

Stamatatos, E., Fakotakis, N., Kokkinakis, G. (2000) ‘Automatic Text Categorization in Terms of Genre and Author', Computational Linguistics, 26(4), pp. 461-485, MIT Press, 2000

Tempich, C., Pinto, H. S., Staab, S. (2006) ’Ontology Engineering Revisited: An Iterative Case Study’, ESWC 2006: 110-124

Yahoo! Webscope (2009) Yahoo! Webscope dataset ydata-search-queries-multiple-langs-v1_0 [http://research.yahoo.com/Academic_Relations]

Zavitsanos, E., Vouros, G.A., Paliouras, G., and Petridis, S. (2007) ‘Discovering Ontology Concepts and Computing Subsumption Relations in Text Corpora Using the LDA Model and Performing Conditional Independence Tests’, In Proc. of the IEEE/WIC/ACM International

Conference on Web Intelligence

Zhang, J., Xiong M., and Yu, Y. (2006) ‘Mining Query Log to Assist Ontology Learning from Relational Database’, Springer – Verlag, LNCS 3841, pp. 437 – 448

Zhuge, H. (2008) ‘The Knowledge Grid Environment’, IEEE Intelligent Systems, November/December, 23(6): 63-71