Combining Semantic Web Search with the Power of Inductive Reasoning

Claudia d'Amato[1], Nicola Fanizzi[1], Bettina Fazzinga[2], Georg Gottlob[3,4], and Thomas Lukasiewicz[3,5]

[1] Dipartimento di Informatica, Università degli Studi di Bari, Italy
{claudia.damato,fanizzi}@di.uniba.it
[2] Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, Italy
bfazzinga@deis.unical.it
[3] Computing Laboratory, University of Oxford, UK
{georg.gottlob,thomas.lukasiewicz}@comlab.ox.ac.uk
[4] Oxford-Man Institute of Quantitative Finance, University of Oxford, UK
[5] Institut für Informationssysteme, TU Wien, Austria

Abstract. Extensive research activities are recently directed towards the Semantic Web as a future form of the Web. Consequently, Web search as the key technology of the Web is evolving towards some novel form of Semantic Web search. A very promising recent approach to such Semantic Web search is based on combining standard Web search with ontological background knowledge and using standard Web search engines as the main inference motor of Semantic Web search. In this paper, we propose to further enhance this approach to Semantic Web search by the use of inductive reasoning techniques. This adds especially the important ability to handle inconsistencies, noise, and incompleteness, which are very likely to occur in distributed and heterogeneous environments, such as the Web. We report on a prototype implementation of the new approach and experimental results.

1 Introduction

Web search [3] as the key technology of the Web is about to change radically with the development of the *Semantic Web* [2]. As a consequence, the elaboration of a new search technology for the Semantic Web, called *Semantic Web search* [6], is currently an extremely hot topic, both in Web-related companies and in academic research. In particular, there is a fast growing number of commercial and academic Semantic Web search engines. The research can be roughly divided into two main directions. The first (and most common) one is to develop a new form of search for searching the pieces of data and knowledge that are encoded in the new representation formalisms of the Semantic Web (e.g., [6]), while the second (and less explored) direction is to use the data and knowledge of the Semantic Web in order to add some semantics to Web search (e.g., [9]).

A very promising recent representative of the second direction to Semantic Web search has been presented in [8]. The approach is based on (i) using ontological (unions of) conjunctive queries (which may contain negated subqueries) as Semantic Web search queries, (ii) combining standard Web search with ontological background knowledge, (iii) using the power of Semantic Web formalisms and technologies, and (iv) using standard Web search engines as the main inference motor of Semantic Web search. It consists of an offline ontology compilation step, based on deductive reasoning techniques, and an online query processing step. In this paper, we propose to further enhance this approach to Semantic Web search by the use of inductive reasoning techniques for the offline ontology compilation step. To our knowledge, this is the first combination of Semantic Web search with inductive reasoning. The paper's main contributions can be summarized as follows:

- We develop a combination of Semantic Web search as presented in [8] with an inductive reasoning technique (based on similarity search [11] for retrieving the resources that likely belong to a query concept [5]). The latter serves in an offline ontology compilation step to compute completed semantic annotations.
- Importantly, the new approach to Semantic Web search can handle inconsistencies, noise, and incompleteness in Semantic Web knowledge bases, which are all very likely to occur in distributed and heterogeneous environments, such as the Web. We provide several examples illustrating this important advantage of the new approach.
- We report on a prototype implementation of the new approach in the context of desktop search. We also provide very positive experimental results for the precision and the recall of the new approach, comparing it to the deductive approach in [8].

2 System Overview

The overall architecture of our Semantic Web search system is shown in Fig. 1. It consists of the *Interface*, the *Query Evaluator*, and the *Inference Engine* (Fig. 1, dark parts), where the Query Evaluator is implemented on top of standard Web *Search Engines*. Standard *Web* pages and their objects are enriched by *Annotation* pages, based on an *Ontology*.

We thus assume that there are semantic annotations to standard Web pages and to objects on standard Web pages. Note that such annotations are starting to be widely available for a large class of Web resources, especially with the Web 2.0. Semantic annotations about Web pages and objects may also be automatically learned from the Web pages and the objects to be annotated (see, e.g., [4]), and/or they may be extracted from existing ontological knowledge bases on the Semantic Web. Another important standard assumption that we make is that Web pages and their objects have unique identifiers.

For example, in a very simple scenario, a Web page i_1 may contain information about a Ph.D. student i_2, called Mary, and two of her papers, namely, a conference paper i_3 entitled "*Semantic Web search*" and a journal paper i_4 entitled "*Semantic Web search engines*" and published in 2008. A simple HTML page representing this scenario is shown in Fig. 2, left side. There may now exist one semantic annotation each for the Web page, the Ph.D. student Mary, the journal paper, and the conference paper. The annotation for the Web page may simply encode that it mentions Mary and the two papers, while the one for Mary may encode that she is a Ph.D. student with the name Mary and the author of the papers i_3 and i_4. The annotation for the paper i_3 may encode that i_3 is a conference paper and has the title "*Semantic Web search*", while the one for the paper i_4 may encode that i_4 is a journal paper, authored by Mary, has the title "*Semantic Web search engines*", was published in 2008, and has the keyword "RDF". The semantic annotations of i_1, i_2, i_3, and i_4 are formally expressed as the sets of axioms $\mathcal{A}_{i_1}$, $\mathcal{A}_{i_2}$, $\mathcal{A}_{i_3}$, and $\mathcal{A}_{i_4}$, respectively:

$$
\begin{aligned}
\mathcal{A}_{i_1} &= \{\textit{contains}(i_1, i_2), \textit{contains}(i_1, i_3), \textit{contains}(i_1, i_4)\},\\
\mathcal{A}_{i_2} &= \{\textit{PhDStudent}(i_2),\ \textit{name}(i_2, \text{“}\textit{mary}\text{”}),\ \textit{isAuthorOf}(i_2, i_3),\ \textit{isAuthorOf}(i_2, i_4)\},\\
\mathcal{A}_{i_3} &= \{\textit{ConferencePaper}(i_3), \textit{title}(i_3, \text{“}\textit{Semantic Web search}\text{”})\},\\
\mathcal{A}_{i_4} &= \{\textit{JournalPaper}(i_4),\ \textit{hasAuthor}(i_4, i_2),\ \textit{title}(i_4, \text{“}\textit{Semantic Web search engines}\text{”}),\\
&\qquad \textit{yearOfPublication}(i_4, 2008), \textit{keyword}(i_4, \text{“RDF”})\}.
\end{aligned} \tag{1}
$$

Inference Engine. Using an ontology containing some background knowledge, these semantic annotations are then further enhanced in an offline ontology compilation step, where the *Inference Engine* adds all properties that can be deduced from the semantic

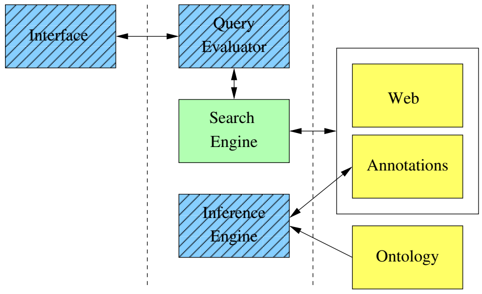

Fig. 1. System architecture.

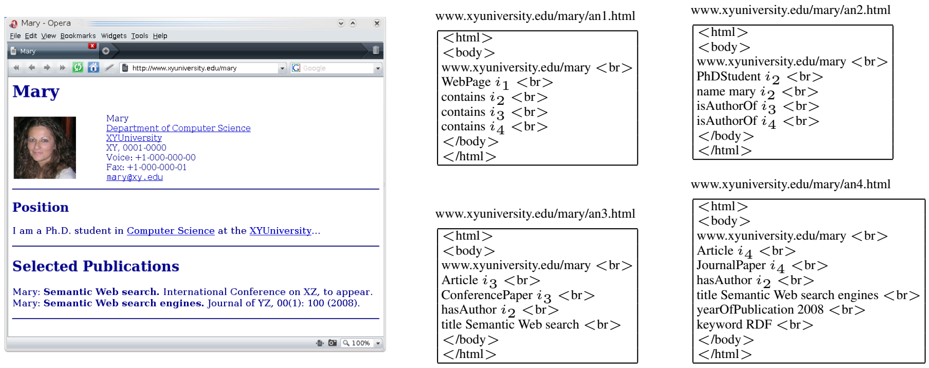

Fig. 2. Left side: HTML page p; right side: four HTML pages p_1, p_2, p_3, and p_4, which encode (completed) semantic annotations for p and the objects on p.

annotations and the ontology. In [8], we assume a deductive such step, while here we propose and explore an inductive one. The resulting (*completed*) semantic annotations are then published as Web pages, so that they can be searched by standard Web search engines. For example, an ontology may contain the knowledge that (i) conference and journal papers are articles, (ii) conference papers are not journal papers, (iii) *isAuthorOf* relates scientists and articles, (iv) *isAuthorOf* is the inverse of *hasAuthor*, and (v) *hasFirstAuthor* is a functional binary relationship, which is formally expressed by:

$$\begin{array}{l} \mathit{ConferencePaper} \sqsubseteq \mathit{Article},\ \mathit{JournalPaper} \sqsubseteq \mathit{Article},\ \mathit{ConferencePaper} \sqsubseteq \neg \mathit{JournalPaper}, \\ \exists \mathit{isAuthorOf} \sqsubseteq \mathit{Scientist},\ \exists \mathit{isAuthorOf}^- \sqsubseteq \mathit{Article},\ \mathit{isAuthorOf}^- \sqsubseteq \mathit{hasAuthor}, \\ \mathit{hasAuthor}^- \sqsubseteq \mathit{isAuthorOf},\ (\mathsf{funct}\ \mathit{hasFirstAuthor}). \end{array} \tag{2}$$

Using this ontological knowledge, we can derive from the above annotations that the two papers i_3 and i_4 are also articles, and both authored by John. These resulting searchable (completed) semantic annotations of (objects on) standard Web pages are published as HTML Web pages with pointers to the respective object pages, so that they (in addition to the standard Web pages) can be searched by standard search engines. For example, the HTML pages for the completed semantic annotations of the above $\mathcal{A}_{i_1}$, $\mathcal{A}_{i_2}$, $\mathcal{A}_{i_3}$, and $\mathcal{A}_{i_4}$ are shown in Fig. 2, right side. Note that on the HTML page of each individual, its identifier is located beside the atomic concept below the row specifying the URIs. Practically, such an identifier may simply be the HTML address of the Web page/object's annotation page. For example, considering the HTML pages of Fig. 2, the individual described by p_4 is i_4, and the one described by p_2 is i_2. Observe that we use a plain textual representation of the completed semantic annotations in order to allow their processing by existing standard search engines for the Web. It is important to point out that this textual representation is simply a list of properties, each eventually along with an identifier or a data value as attribute value, and it can thus immediately be encoded as a list of RDF triples.

Query Evaluator. The *Query Evaluator* (see Fig. 1) reduces each Semantic Web search query of the user in an online query processing step to a sequence of standard Web search queries on standard Web and annotation pages, which are then processed by a standard Web *Search Engine*. The Query Evaluator also collects the results and re-transforms them into a single answer which is returned to the user. As an example of a Semantic Web search query, one may ask for all Ph.D. students who have published an article in 2008 with RDF as a keyword, which is formally expressed as follows:

$$Q(x) = \exists y\, (\mathit{PhDStudent}(x) \wedge \mathit{isAuthorOf}(x, y) \wedge \mathit{Article}(y) \wedge \\ \mathit{yearOfPublication}(y, 2008) \wedge \mathit{keyword}(y, \text{“}RDF\text{”}))\,.$$

This query is transformed into the two queries $Q_1 = \mathit{PhDStudent}$ AND *isAuthorOf* and $Q_2 = \mathit{Article}$ AND *“yearOfPublication* 2008” AND *“keyword* RDF”, which can both be submitted to a standard Web search engine, such as Google. The result of the original query Q is then built from the results of the two queries Q_1 and Q_2. Note that a graphical user interface, such as the one of Google's advanced search, or even a natural language interface can help to hide the conceptual complexity of ontological queries to the user.

3 Semantic Web Search

We now introduce Semantic Web knowledge bases and the syntax and semantics of Semantic Web search queries to such knowledge bases. We then generalize the PageRank technique to our approach. We assume the reader is familiar with the syntax and the semantics of Description Logics (DLs) [1], which we use as underlying ontology languages.

Semantic Web Knowledge Bases. Intuitively, a Semantic Web knowledge base consists of a background TBox and a collection of ABoxes, one for every concrete Web page and for every object on a Web page. For example, the homepage of a scientist may be such a concrete Web page and be associated with an ABox, while the publications on the homepage may be such objects, which are also associated with one ABox each.

We assume pairwise disjoint sets $\mathbf{D}$, $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, and $\mathbf{V}$ of atomic datatypes, atomic concepts, atomic roles, atomic attributes, individuals, and data values, respectively. Let $\mathbf{I}$ be the disjoint union of two sets $\mathbf{P}$ and $\mathbf{O}$ of *Web pages* and *Web objects*, respectively. Informally, every $p \in \mathbf{P}$ is an identifier for a concrete Web page, while every $o \in \mathbf{O}$ is an identifier for a concrete object on a concrete Web page. We assume the atomic roles *links_to* between Web pages and *contains* between Web pages and Web objects. The former represents the link structure between concrete Web pages, while the latter encodes the occurrences of concrete Web objects on concrete Web pages.

Definition 1. A *semantic annotation* $\mathcal{A}_a$ for a Web page or object $a \in \mathbf{P} \cup \mathbf{O}$ is a finite set of concept membership axioms $A(a)$, role membership axioms $P(a, b)$, and attribute membership axioms $U(a, v)$, where $A \in \mathbf{A}$, $P \in \mathbf{R}_A$, $U \in \mathbf{R}_D$, $b \in \mathbf{I}$, and $v \in \mathbf{V}$. A *Semantic Web knowledge base* $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$ consists of a TBox $\mathcal{T}$ and one semantic annotation $\mathcal{A}_a$ for every Web page and object $a \in \mathbf{P} \cup \mathbf{O}$.

Informally, a Semantic Web knowledge base consists of some background terminological knowledge and some assertional knowledge for every concrete Web page and for every concrete object on a Web page. The background terminological knowledge may be an ontology from some global Semantic Web repository or an ontology defined locally by the user site. In contrast to the background terminological knowledge, the assertional knowledge will be directly stored on the Web (on annotation pages like the described standard Web pages) and is thus accessible via Web search engines.

Example 1. (Scientific Database). We use a DL knowledge base $KB = (\mathcal{T}, \mathcal{A})$ to specify some simple information about scientists and their publications. The sets of atomic concepts, atomic roles, atomic attributes, and data values are:

$$\mathbf{A} = \{\mathit{Scientist}, \mathit{Article}, \mathit{ConferencePaper}, \mathit{JournalPaper}\},$$
$$\mathbf{R}_A = \{\mathit{hasAuthor}, \mathit{isAuthorOf}, \mathit{contains}\},\ \mathbf{R}_D = \{\mathit{name}, \mathit{title}, \mathit{yearOfPublication}\},$$
$$\mathbf{V} = \{\text{“}\mathit{mary}\text{”}, \text{“}\mathit{Semantic\ Web\ search}\text{”}, 2008, \text{“}\mathit{Semantic\ Web\ search\ engines}\text{”}\}.$$

Let $\mathbf{I} = \mathbf{P} \cup \mathbf{O}$ be the set of individuals, where $\mathbf{P} = \{i_1\}$ is the set of Web pages, and $\mathbf{O} = \{i_2, i_3, i_4\}$ is the set of Web objects on the Web page i_1. The TBox $\mathcal{T}$ contains the axioms in Eq. 2. Then, a Semantic Web knowledge base is given by $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$, where the semantic annotations of the individuals in $\mathbf{P} \cup \mathbf{O}$ are the ones in Eq. 1.

Semantic Web Search Queries. We use unions of conjunctive queries with negated conjunctive subqueries as Semantic Web search queries to Semantic Web knowledge bases. We now first define the syntax of Semantic Web search queries and then the semantics of positive and general such queries.

Syntax. Let $\mathbf{X}$ be a finite set of variables. A *term* is either a Web page $p \in \mathbf{P}$, a Web object $o \in \mathbf{O}$, a data value $v \in \mathbf{V}$, or a variable $x \in \mathbf{X}$. An *atomic formula* (or *atom*) α is of one of the following forms: (i) $d(t)$, where d is an atomic datatype, and t is a term; (ii) $A(t)$, where A is an atomic concept, and t is a term; (iii) $P(t, t')$, where P is an atomic role, and t, t' are terms; and (iv) $U(t, t')$, where U is an atomic attribute, and t, t' are terms. An *equality* has the form $=(t, t')$, where t and t' are terms. A *conjunctive formula* $\exists \mathbf{y}\, \phi(\mathbf{x}, \mathbf{y})$ is an existentially quantified conjunction of atoms α and equalities $=(t, t')$, which have free variables among $\mathbf{x}$ and $\mathbf{y}$.

Definition 2. A *Semantic Web search query* $Q(\mathbf{x})$ is an expression $\bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_i(\mathbf{x}, \mathbf{y}_i)$, where each ϕ_i with $i \in \{1, \ldots, n\}$ is a conjunction of atoms α (also called *positive atoms*), negated conjunctive formulas $\mathit{not}\, \psi$, and equalities $=(t, t')$, which have free variables among $\mathbf{x}$ and $\mathbf{y}_i$, and the $\mathbf{x}$'s are exactly the free variables of $\bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_i(\mathbf{x}, \mathbf{y}_i)$.

Intuitively, Semantic Web search queries are unions of conjunctive queries, which may contain negated conjunctive queries in addition to atoms and equalities as conjuncts.

Example 2. (Scientific Database cont'd). Two Semantic Web search queries are:

$$\begin{aligned} Q_1(x) = {} & (\mathit{Scientist}(x) \wedge \mathit{not}\ \mathit{doctoralDegree}(x, \text{“}\mathit{oxford\ university}\text{”}) \wedge \mathit{worksFor}(x, \\ & \text{“}\mathit{oxford\ university}\text{”})) \vee (\mathit{Scientist}(x) \wedge \mathit{doctoralDegree}(x, \text{“}\mathit{oxford\ university}\text{”}) \wedge \\ & \mathit{not}\ \mathit{worksFor}(x, \text{“}\mathit{oxford\ university}\text{”})); \\ Q_2(x) = {} & \exists y\, (\mathit{Scientist}(x) \wedge \mathit{worksFor}(x, \text{“}\mathit{oxford\ university}\text{”}) \wedge \mathit{isAuthorOf}(x, y) \wedge \\ & \mathit{not}\ \mathit{ConferencePaper}(y) \wedge \mathit{not}\ \exists z\, \mathit{yearOfPublication}(y, z)). \end{aligned}$$

Informally, $Q_1(x)$ asks for scientists who are either working for *oxford university* and did not receive their Ph.D. from that university, or who received their Ph.D. from *oxford university* but do not work for it. Whereas query $Q_2(x)$ asks for scientists of *oxford university* who are authors of at least one unpublished non-conference paper. Note that when searching for scientists, the system automatically searches for all subconcepts (known according to the background ontology), such as e.g. Ph.D. students or computer scientists.

Semantics of Positive Search Queries. We now define the semantics of positive Semantic Web search queries, which are free of negations, in terms of ground substitutions via the notion of logical consequence.

A search query $Q(\mathbf{x})$ is *positive* iff it contains no negated conjunctive subqueries. A *(variable) substitution* θ maps variables from $\mathbf{X}$ to terms. A substitution θ is *ground* iff it maps to Web pages $p \in \mathbf{P}$, Web objects $o \in \mathbf{O}$, and data values $v \in \mathbf{V}$. A closed first-order formula ϕ is a *logical consequence* of a knowledge base $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$, denoted $KB \models \phi$, iff every first-order model $\mathcal{I}$ of $\mathcal{T} \cup \bigcup_{a \in \mathbf{P} \cup \mathbf{O}} \mathcal{A}_a$ also satisfies ϕ.

Definition 3. Given a Semantic Web knowledge base KB and a positive Semantic Web search query $Q(\mathbf{x})$, an *answer* for $Q(\mathbf{x})$ to KB is a ground substitution θ for the variables $\mathbf{x}$ (which are exactly the free variables of $Q(\mathbf{x})$) with $KB \models Q(\mathbf{x}\theta)$.

Example 3. (Scientific Database cont'd). Consider the Semantic Web knowledge base KB of Example 1 and the following positive Semantic Web search query, asking for all scientists who author at least one published journal paper:

$$Q(x) = \exists y\, (\mathit{Scientist}(x) \wedge \mathit{isAuthorOf}(x, y) \wedge \mathit{JournalPaper}(y) \wedge \exists z\, \mathit{yearOfPublication}(y, z)).$$

An answer for $Q(x)$ to KB is $\theta = \{x/i_2\}$. Recall that i_2 represents the scientist Mary.

Semantics of General Search Queries. We next define the semantics of general Semantic Web search queries by reduction to the semantics of positive ones, interpreting negated conjunctive subqueries $\mathit{not}\, \psi$ as the lack of evidence about the truth of ψ. That is, negations are interpreted by a closed-world semantics on top of the open-world semantics of DLs (we refer to [8] for more motivation and background).

Definition 4. Given a Semantic Web knowledge base KB and search query

$$Q(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_{i,1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \phi_{i,l_i}(\mathbf{x}, \mathbf{y}_i) \wedge \mathit{not}\, \phi_{i,l_i+1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \mathit{not}\, \phi_{i,m_i}(\mathbf{x}, \mathbf{y}_i)\,,$$

an *answer* for $Q(\mathbf{x})$ to KB is a ground substitution θ for the variables $\mathbf{x}$ such that $KB \models Q^+(\mathbf{x}\theta)$ and $KB \not\models Q^-(\mathbf{x}\theta)$, where $Q^+(\mathbf{x})$ and $Q^-(\mathbf{x})$ are defined as follows:

$$Q^+(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_{i,1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \phi_{i,l_i}(\mathbf{x}, \mathbf{y}_i) \text{ and}$$
$$Q^-(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y}_i\, \phi_{i,1}(\mathbf{x}, \mathbf{y}_i) \wedge \cdots \wedge \phi_{i,l_i}(\mathbf{x}, \mathbf{y}_i) \wedge (\phi_{i,l_i+1}(\mathbf{x}, \mathbf{y}_i) \vee \cdots \vee \phi_{i,m_i}(\mathbf{x}, \mathbf{y}_i))\,.$$

Roughly, a ground substitution θ is an answer for $Q(\mathbf{x})$ to KB iff (i) θ is an answer for $Q^+(\mathbf{x})$ to KB, and (ii) θ is not an answer for $Q^-(\mathbf{x})$ to KB, where $Q^+(\mathbf{x})$ is the positive part of $Q(\mathbf{x})$, while $Q^-(\mathbf{x})$ is the positive part of $Q(\mathbf{x})$ combined with the complement of the negative one. Observe that both $Q^+(\mathbf{x})$ and $Q^-(\mathbf{x})$ are positive queries.

Example 4. (Scientific Database cont'd). Consider the Semantic Web knowledge base $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$ of Example 1 and the following general Semantic Web search query, asking for Mary's unpublished non-journal papers:

$$Q(x) = \exists y\, (\mathit{Article}(x) \wedge \mathit{hasAuthor}(x, y) \wedge \mathit{name}(y, \text{“}\mathit{mary}\text{”}) \wedge \mathit{not}\, \mathit{JournalPaper}(x) \wedge \mathit{not}\, \exists z\, \mathit{yearOfPublication}(x, z)).$$

An answer for $Q(x)$ to KB is given by $\theta = \{\mathbf{x}/i_3\}$. Recall that i_3 represents an unpublished conference paper entitled "*Semantic Web search*". Observe that the membership axioms *Article*(i_3) and *hasAuthor*(i_2, i_3) do not appear in the semantic annotations $\mathcal{A}_a$ with $a \in \mathbf{P} \cup \mathbf{O}$, but they can be inferred from them using the background ontology $\mathcal{T}$.

Ranking Answers. As for the ranking of all answers for a Semantic Web search query Q to a Semantic Web knowledge base KB (i.e., ground substitutions for all free variables in Q, which correspond to tuples of Web pages, Web objects, and data values), we use a generalization of the PageRank technique: rather than considering only Web pages and the link structure between Web pages (expressed through the role *links_to* here), we also consider Web objects, which may occur on Web pages (expressed through the role *contains*), and which may also be related to other Web objects via other roles. More concretely, we define the *ObjectRank* of a Web page or an object a as follows:

$$R(a) = d \cdot \sum_{b \in B_a} R(b) \,/\, N_b + (1-d) \cdot E(a)\,,$$

where (i) B_a is the set of all Web pages and Web objects that relate to a, (ii) N_b is the number of Web pages and Web objects that relate from b, (iii) d is a damping factor, and (iv) E associates with every Web page and every Web object a source of rank.

4 Deductive Offline Ontology Compilation

In this section, we describe the (deductive) offline ontology reasoning step, which compiles the implicit terminological knowledge in the TBox of a Semantic Web knowledge base into explicit membership axioms in the ABox, i.e., in the semantic annotations of Web pages / objects, so that it (in addition to the standard Web pages) can be searched by standard Web search engines. For the online query processing step, see [8].

The compilation of TBox knowledge into ABox knowledge is formalized as follows. Given a satisfiable Semantic Web knowledge base $KB = (\mathcal{T}, (\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}})$, the *simple completion* of KB is the Semantic Web knowledge base $KB' = (\emptyset, ({\mathcal{A}_a}')_{a \in \mathbf{P} \cup \mathbf{O}})$ such that every ${\mathcal{A}_a}'$ is the set of all concept memberships $A(a)$, role memberships $P(a,b)$, and attribute memberships $U(a,v)$ that logically follow from $\mathcal{T} \cup \bigcup_{a \in \mathbf{P} \cup \mathbf{O}} \mathcal{A}_a$, where $A \in \mathbf{A}$, $P \in \mathbf{R}_A$, $U \in \mathbf{R}_D$, $b \in \mathbf{I}$, and $v \in \mathbf{V}$. Informally, for every Web page and object, the simple completion collects all available and deducible facts (whose predicate symbols shall be usable in search queries) in a completed semantic annotation.

Example 5. Consider the TBox $\mathcal{T}$ of Example 1 and the semantic annotations $(\mathcal{A}_a)_{a \in \mathbf{P} \cup \mathbf{O}}$ of Example 1. The simple completion contains in particular the new axioms *Article*(i_3), *hasAuthor*(i_3, i_2), and *Article*(i_4). The first two are added to A_{i_3} and the last one to A_{i_4}.

As shown in [8], general quantifier-free search queries to a Semantic Web knowledge base KB over *DL-Lite*$_{\mathcal{A}}$ [10] as underlying DL can be evaluated on the simple completion of KB (which contains only compiled but no explicit TBox knowledge anymore). Similar results hold when the TBox of KB is equivalent to a Datalog program, and the query is fully general. Hence, the simple completion assures (i) always a sound query processing and (ii) a complete query processing in many cases. For this reason, and since completeness of query processing is actually not that much an issue in the inherently incomplete Web, we propose to use the simple completion as the basis of our Semantic Web search.

Once the completed semantic annotations are computed, we encode them as HTML pages, so that they are searchable via standard keyword search. Specifically, we build one HTML page for the semantic annotation $\mathcal{A}_a$ of each individual $a \in \mathbf{P} \cup \mathbf{O}$. That is, for each individual a, we build a page p containing all the atomic concepts whose argument is a and all the atomic roles/attributes where the first argument is a (see Section 2).

5 Inductive Offline Ontology Compilation

We now describe an inductive inference based on similarity search, which we propose to use instead of deductive inference for offline ontology compilation in our approach to Semantic Web search. Section 6 then summarizes the central advantages of this proposal.

Inductive Inference Based on Similarity Search. In *similarity search* [11], the basic idea is to find the most similar object(s) to a query object (i.e., the one to be classified) with respect to a similarity (or dissimilarity) measure. We review the basics of the k-nearest-neighbor (k-NN) method applied to the Semantic Web context [5]. The objective is to induce an approximation for a discrete-valued target hypothesis function $h\colon \mathit{IS} \rightarrow V$ from a space of instances *IS* to a set of values $V = \{v_1, \ldots, v_s\}$ standing for the classes (concepts) that have to be predicted. Let x_q be the query instance whose class-membership is to be determined. Using a dissimilarity measure, the set of the k-nearest (pre-classified) training instances relative to x_q is selected: $NN(x_q) = \{x_1, \ldots, x_k\}$. Hence, the k-NN algorithm approximates h for classifying x_q on the grounds of the value that h is known to assume for the training instances in $NN(x_q)$. Precisely, the value is decided by means of a weighted majority voting procedure: it is the most *voted* value by the instances in $NN(x_q)$ weighted by the similarity of the neighbor individual. The estimate of the hypothesis function for the query individual is:

$$\hat{h}(x_q) := \operatorname*{argmax}_{v \in V} \sum_{i=1}^{k} w_i \delta(v, h(x_i)) \,, \tag{3}$$

where δ returns 1 in case of matching arguments and 0 otherwise, and, given a dissimilarity measure d, the weights w_i are determined by $w_i = 1/d(x_i, x_q)$.

Observe that this setting assigns to the query instance x_q a value, which stands for one in a set of pairwise disjoint concepts (corresponding to the value set V). In a multi-relational setting, as those of the Semantic Web (SW) context, this assumption cannot be made in general, since it is well known that an individual may be an instance of more than one concept. The problem is also related to the closed-world assumption (CWA) usually made in the knowledge discovery context. To deal with the open-world assumption (OWA), generally adopted for the SW representations, the absence of information on whether a training instance x belongs to the extension of a query concept Q should not be interpreted negatively, as in the standard settings which adopt the CWA, rather, it should count as neutral (uncertain) information. Assuming this alternate viewpoint, the multi-class classification problem is transformed into a ternary one and the $V = \{+1, -1, 0\}$ value set is adopted for the classification of an individual with respect to a query concept Q and where the three values denote, respectively, membership, non-membership, and uncertainty. Hence, the task is cast as follows: given a query concept Q, determine the membership of an instance x_q through the NN procedure (see Eq. 3) where $V = \{-1, 0, +1\}$ and the hypothesis function values for the training instances are determined as:

$$h_Q(x) = \begin{cases} +1 & \mathcal{K} \models Q(x) \\ -1 & \mathcal{K} \models \neg Q(x) \\ 0 & \textit{otherwise.} \end{cases}$$

That is, the value of h_Q for the training instances is determined by logical entailment (denoted $\models$) of the corresponding assertion from the knowledge base. Alternatively, a look-up in the ABox of the knowledge base could be considered, thus obtaining a classification process less complex but also possibly less accurate.

For measuring the similarity between individuals, a totally semantic and language independent family of dissimilarity measures has been used [5]. It is based on the idea of comparing the semantics of the input individuals along a number of dimensions represented by a committee of concept descriptions, say $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, which stands as a group of discriminating *features* expressed in the OWL-DL sub-language taken into account. It is formally defined as follows [5]:

Definition 5 (family of measures). Let $KB = (\mathcal{T}, \mathcal{A})$ be a knowledge base. Given a set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, corresponding weights $w_1, \ldots, w_m$, and $p > 0$, a family of dissimilarity functions $d_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0, 1]$ is defined by:

$$\forall a, b \in \mathsf{Ind}(\mathcal{A}): \quad d_p^{\mathsf{F}}(a, b) := \frac{1}{|\mathsf{F}|} \left[\sum_{i=1}^{|\mathsf{F}|} w_i \mid \delta_i(a, b) \mid^p \right]^{1/p} ,$$

where the dissimilarity function δ_i $(i \in \{1, \ldots, m\})$ is defined as follows:

$$\forall a, b \in \mathsf{Ind}(\mathcal{A}): \quad \delta_i(a, b) = \begin{cases} 0 & F_i(a) \in \mathcal{A} \wedge F_i(b) \in \mathcal{A} \\ 1 & F_i(a) \in \mathcal{A} \wedge \neg F_i(b) \in \mathcal{A} \text{ or} \\ & \neg F_i(a) \in \mathcal{A} \wedge F_i(b) \in \mathcal{A} \\ 1/2 & otherwise. \end{cases}$$

An alternative definition for the projections requires the entailment of an assertion (instance-checking) rather than the simple ABox look-up; this can make the measure more accurate yet more complex to compute. Moreover, using instance checking, induction is performed on top of deduction, thus making it a kind of completion of deductive reasoning.

As for the weights w_i employed in the family of measures, they should reflect the impact of the single feature concept F_i relative to the overall dissimilarity. This is determined by the quantity of information conveyed by a feature, which is measured as its entropy. Namely, the extension of a feature F_i relative to the whole domain of objects may be probabilistically quantified as $P_{F_i} = |F_i^{\mathcal{I}}|/|\Delta^{\mathcal{I}}|$ (relative to the canonical interpretation $\mathcal{I}$). This can be roughly approximated by $|\mathsf{retrieval}(F_i)|/|\mathsf{Ind}(\mathcal{A})|$. Hence, considering also the probability $P_{\neg F_i}$ related to its negation and the one related to the unclassified individuals (relative to F_i), denoted P_U, we may give an entropic measure for the feature:

$$H(F_i) = -\left(P_{F_i} \log(P_{F_i}) + P_{\neg F_i} \log(P_{\neg F_i}) + P_U \log(P_U)\right) .$$

The measures strongly depend on F. Here, we make the assumption that the feature-set F represents a sufficient number of (possibly redundant) features that are able to discriminate really different individuals. However, an optimal discriminating feature set could be learned [7]. Experimentally, we obtained good results by using the very set of both primitive and defined concepts found in the knowledge base [5].

Measuring the Likelihood of an Answer. The inductive inference made by the procedure shown above is not guaranteed to be deductively valid. Indeed, inductive inference naturally yields a certain degree of uncertainty. So, from a more general perspective, the main idea behind the above inductive inference for Semantic Web search is closely related to the idea of using probabilistic ontologies to increase the precision and the recall of querying databases and of information retrieval in general. But, rather than learning probabilistic ontologies from data, representing probabilistic ontologies, and reasoning with probabilistic ontologies, we directly use the data in the inductive inference step.

In order to measure the likelihood of the decision made by the inductive procedure (individual x_q belongs to the query concept denoted by value v maximizing the argmax argument in Eq. 3), given the k-nearest training individuals in $NN(x_q) = \{x_1, \ldots, x_k\}$, the quantity that determined the decision should be normalized by dividing it by the sum of such arguments over the (three) possible values:

$$l(class(x_q) = v | NN(x_q)) = \frac{\sum_{i=1}^{k} w_i \cdot \delta(v, h_Q(x_i))}{\sum_{v' \in V} \sum_{i=1}^{k} w_i \cdot \delta(v', h_Q(x_i))} . \tag{4}$$

Hence, the likelihood of the assertion $Q(x_q)$ corresponds to the case when $v = +1$. The computed likelihood can be used for building a probabilistic ABox (which is a collection of pairs, each consisting of a classical ABox axiom and a probability value).

6 Inconsistencies, Noise, and Incompleteness

In this section, we illustrate the main advantages of using inductive reasoning in Semantic Web search, namely, that inductive reasoning (differently from deductive reasoning) can handle inconsistencies, noise, and incompleteness in Semantic Web knowledge bases, which are all very likely to occur when knowledge bases are stored in a distributed and heterogeneous fashion, like on the Web.

Inconsistencies. Since our inductive method is based on the majority vote of the individuals in the neighborhood, it may be able to give a correct classification even in the case of inconsistent knowledge bases. This aspect is illustrated by the following example.

Example 6. Consider the description logic knowledge base $KB = (\mathcal{T}, \mathcal{A})$ that consists of the following TBox $\mathcal{T}$ and ABox $\mathcal{A}$:

$$\begin{aligned}\mathcal{T} = \{&\mathit{Man} \equiv \mathit{Male} \sqcap \mathit{Human};\ \mathit{Professor} \equiv \mathit{Person} \sqcap \exists \mathit{abilitatedTo.Teaching} \sqcap \\ &\exists \mathit{isSupervisorOf.PhDThesis} \sqcap \mathit{Researcher};\ \mathit{Researcher} \equiv \mathit{GraduatePerson} \sqcap \\ &\exists \mathit{worksFor.ResearchInstitute} \sqcap \neg\exists \mathit{isSupervisorOf.PhDThesis}; \ldots\}\,; \\ \mathcal{A} = \{&\mathit{Professor}(\mathit{Franz});\ \mathit{isSupervisorOf}(\mathit{Franz}, \mathit{DLThesis});\ \mathit{Professor}(\mathit{John}); \\ &\mathit{isSupervisorOf}(\mathit{John}, \mathit{RoboticsThesis});\ \mathit{Professor}(\mathit{Flo});\ \mathit{isSupervisorOf}(\mathit{Flo}, \mathit{MLThesis}); \\ &\mathit{Researcher}(\mathit{Nick});\ \mathit{Researcher}(\mathit{Ann});\ \mathit{isSupervisorOf}(\mathit{Nick}, \mathit{SWThesis}); \ldots\}\,.\end{aligned}$$

Actually, *Nick* is a *Professor*, indeed, he is the supervisor of a PhD thesis in $\mathcal{A}$. However, by human mistake, he is asserted to be a *Researcher* in $\mathcal{A}$, and by the axiom for *Researcher* in $\mathcal{T}$, he cannot be the supervisor of any PhD thesis. Hence, KB is inconsistent, and thus a deductive reasoner cannot answer whether *Nick* is a *Professor* or not (since everything can be deduced from an inconsistent knowledge base). On the contrary, by inductive reasoning, it is highly probable that the returned classification result is that *Nick* is an instance of *Professor*. This is because the most similar individuals are *Franz*, *John*, and *Flo*, and all of them vote for the concept *Professor*.

Noise. Inductive reasoning may also be able to give a correct classification in the presence of noise in a knowledge base (containing, e.g., incorrect concept and/or role membership assertions), which is illustrated by the following example.

Example 7. Consider the description logic knowledge base $KB = (\mathcal{T}', \mathcal{A})$, where the ABox $\mathcal{A}$ is as in Example 6 and the TBox $\mathcal{T}'$ is obtained from the TBox $\mathcal{T}$ of Example 6 by replacing the axiom for *Researcher* by the following axiom:

$$\mathit{Researcher} \equiv \mathit{GraduatePerson} \sqcap \exists \mathit{worksFor.ResearchInstitute}\,.$$

Again, *Nick* is actually a *Professor*, but by human mistake asserted to be a *Researcher* in KB. But due to the slightly modified axiom for *Researcher*, there is no inconsistency in KB anymore. By deductive reasoning, however, *Nick* turns out to be a *Researcher*, whereas by inductive reasoning, it is highly probable that the returned classification result is that *Nick* is an instance of *Professor*, as above, because the most similar individuals are *Franz*, *John*, and *Flo*, and all of them vote for the concept *Professor*.

Incompleteness. Clearly, inductive reasoning may also be able to give a correct classification in the presence of incompleteness in a knowledge base. That is, inductive reasoning is not necessarily deductively valid, and may produce new knowledge.

Example 8. Consider the description logic knowledge base $KB = (\mathcal{T}', \mathcal{A}')$, where the TBox $\mathcal{T}'$ is as in Example 7 and the ABox $\mathcal{A}'$ is obtained from the ABox $\mathcal{A}$ of Example 6 by removing the axiom *Researcher*(*Nick*). Then, the resulting knowledge base is neither inconsistent nor noisy, but it is now incomplete. Nonetheless, by the same line of argumentation as in Examples 6 and 7, it is highly probable that the classification result by inductive reasoning is that *Nick* is an instance of *Professor*.

7 Implementation and Experiments

In this section, we describe our prototype implementation for a semantic desktop search engine. Furthermore, we report on very positive experimental results on the precision and the recall under inductively vs. deductively completed semantic annotations.

Implementation. We have implemented a prototype for a semantic desktop search engine. We have realized both a deductive and an inductive version of the offline inference step for generating the completed semantic annotation for every considered resource. The deductive version uses PELLET[1], while the inductive one is based on the k-NN technique, integrated with an entropic measure, as proposed in Section 5. Specifically, each individual i of a Semantic Web knowledge base is classified relative to all atomic concepts and all restrictions $\exists R^{-}.\{i\}$ with roles R. The parameter k was set to $\log(|\mathsf{Ind}(\mathcal{A})|)$, where $\mathsf{Ind}(\mathcal{A})$ stands for all individuals in the knowledge base. The simpler distances d_1^{F} were employed, using all the atomic concepts in the knowledge base for determining the set F.

Precision and Recall of Inductive Semantic Web Search. We next give an experimental comparison between Semantic Web search under inductive and under deductive reasoning. We do this by providing the precision and the recall of the latter vs. the former. Our experimental results with queries relative to the FINITE-STATE-MACHINE (FSM) and the SURFACE-WATER-MODEL (SWM) ontology from the Protégé Ontology Library[2] are summarized in Table 1. For example, Query (8) asks for all transitions having no target state, while Query (16) asks for all numerical models having either the domain "lake" and public availability, or the domain "coastalArea" and commercial availability. The experimental results in Table 1 essentially show that the answer sets under inductive reasoning are very close to the ones under deductive reasoning.

8 Summary and Outlook

We have presented a combination of Semantic Web search as presented in [8] with an inductive reasoning technique, based on similarity search [11] for retrieving the resources that likely belong to a query concept [5]. As a crucial advantage, the new approach to Semantic Web search allows for handling inconsistencies, noise, and incompleteness, which are very likely in distributed and heterogeneous environments, such as the Web. We have also reported on a prototype implementation and very positive experimental results on the precision and the recall of the new inductive approach to Semantic Web search.

[1] http://www.mindswap.org
[2] http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library

Table 1. Precision and recall of inductive vs. deductive Semantic Web search.

	Onto-logy	**Query**	No. Results Deduction	No. Results Induction	No. Correct Results Induction	**Precision Induction**	**Recall Induction**
1	FSM	$\mathit{State}(x)$	11	11	11	1	1
2	FSM	$\mathit{StateMachineElement}(x)$	37	37	37	1	1
3	FSM	$\mathit{Composite}(x) \wedge \mathit{hasStateMachineElement}(x, \mathit{accountDetails})$	1	1	1	1	1
4	FSM	$\mathit{State}(y) \wedge \mathit{StateMachineElement}(x) \wedge \mathit{hasStateMachineElement}(x, y)$	3	3	3	1	1
5	FSM	$\mathit{Action}(x) \vee \mathit{Guard}(x)$	12	12	12	1	1
6	FSM	$\exists y, z\,(\mathit{State}(y) \wedge \mathit{State}(z) \wedge \mathit{Transition}(x) \wedge \mathit{source}(x, y) \wedge \mathit{target}(x, z))$	11	2	2	1	0.18
7	FSM	$\mathit{StateMachineElement}(x) \wedge \mathit{not}\ \exists y\,(\mathit{StateMachineElement}(y) \wedge \mathit{hasStateMachineElement}(x, y))$	34	34	34	1	1
8	FSM	$\mathit{Transition}(x) \wedge \mathit{not}\ \exists y\,(\mathit{State}(y) \wedge target(x, y))$	0	5	0	0	1
9	FSM	$\exists y\,(\mathit{StateMachineElement}(x) \wedge \mathit{not}\ \mathit{hasStateMachineElement}(x, \mathit{accountDetails}) \wedge \mathit{hasStateMachineElement}(x, y) \wedge \mathit{State}(y))$	2	2	2	1	1
10	SWM	$\mathit{Model}(x)$	56	56	56	1	1
11	SWM	$\mathit{Mathematical}(x)$	64	64	64	1	1
12	SWM	$\mathit{Model}(x) \wedge \mathit{hasDomain}(x, \mathit{lake}) \wedge \mathit{hasDomain}(x, \mathit{river})$	9	9	9	1	1
13	SWM	$\mathit{Model}(x) \wedge \mathit{not}\ \exists y\,(\mathit{Availability}(y) \wedge \mathit{hasAvailability}(x, y))$	11	11	11	1	1
14	SWM	$\mathit{Model}(x) \wedge \mathit{hasDomain}(x, \mathit{river}) \wedge \mathit{not}\ \mathit{hasAvailability}(x, \mathit{public})$	2	8	0	0	0
15	SWM	$\exists y\,(\mathit{Model}(x) \wedge \mathit{hasDeveloper}(x, y) \wedge \mathit{University}(y))$	1	1	1	1	1
16	SWM	$\mathit{Numerical}(x) \wedge \mathit{hasDomain}(x, \mathit{lake}) \wedge \mathit{hasAvailability}(x, \mathit{public}) \vee \mathit{Numerical}(x) \wedge \mathit{hasDomain}(x, \mathit{coastalArea}) \wedge \mathit{hasAvailability}(x, \mathit{commercial})$	12	9	9	1	0.75

In the future, we aim especially at extending the desktop implementation to a real Web implementation, using existing search engines, such as Google. Another interesting topic is to explore how search expressions that are formulated as plain natural language sentences can be translated into the ontological conjunctive queries of our approach. It would also be interesting to investigate the use of probabilistic ontologies rather than classical ones.

Acknowledgments. Georg Gottlob's work was supported by the EPSRC grant Number EP/E010865/1 "Schema Mappings and Automated Services for Data Integration." Georg Gottlob, whose work was partially carried out at the Oxford-Man Institute of Quantitative Finance, gratefully acknowledges support from the Royal Society as the holder of a Royal Society-Wolfson Research Merit Award. Thomas Lukasiewicz's work was supported by the German Research Foundation (DFG) under the Heisenberg Programme.

References

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Sci. Am.*, 284:34–43, 2001.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1–7):107–117, 1998.

[4] P.-A. Chirita, S. Costache, W. Nejdl, and S. Handschuh. P-TAG: Large scale automatic generation of personalized annotation TAGs for the Web. In *Proc. WWW-2007*.

[5] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: An inductive approach. In *Proc. ESWC-2008*.

[6] L. Ding, T. W. Finin, A. Joshi, Y. Peng, R. Pan, and P. Reddivari. Search on the Semantic Web. *IEEE Computer*, 38(10):62–69, 2005.

[7] N. Fanizzi, C. d'Amato, and F. Esposito. Induction of classifiers through non-parametric methods for approximate classification and retrieval with ontologies. *International Journal of Semantic Computing*, 2(3):403–423, 2008.

[8] B. Fazzinga, G. Gianforme, G. Gottlob, and T. Lukasiewicz. From Web search to Semantic Web search. Technical Report INFSYS RR-1843-08-11, Institut für Informationssysteme, TU Wien, November 2008. http://www.kr.tuwien.ac.at/research/reports/rr0811.pdf.

[9] R. V. Guha, R. McCool, and E. Miller. Semantic search. In *Proc. WWW-2003*.

[10] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.

[11] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search — The Metric Space Approach*, *Advances in Database Systems* 32. Springer, 2006.