Abduction for Extending Incomplete Information Sources

Carlo Meghini[1], Yannis Tzitzikas[2], and Nicolas Spyratos[3]

[1] Consiglio Nazionale delle Ricerche, Istituto della Scienza e delle Tecnologie della Informazione, Pisa, Italy
meghini@isti.cnr.it
[2] Department of Computer Science, University of Crete, Heraklion, Crete, Greece
tzitzik@csi.forth.gr
[3] Université Paris-Sud, Laboratoire de Recherche en Informatique, Orsay Cedex, France
spyratos@lri.fr

Abstract. The extraction of information from a source containing term-classified objects is plagued with uncertainty, due, among other things, to the possible incompleteness of the source index. To overcome this incompleteness, the study proposes to expand the index of the source, in a way that is as reasonable as possible with respect to the original classification of objects. By equating reasonableness with logical implication, the sought expansion turns out to be an explanation of the index, captured by abduction. We study the general problem of query evaluation on the extended information source, providing a polynomial time algorithm which tackles the general case, in which no hypothesis is made on the structure of the taxonomy. We then specialize the algorithm for two well-know structures: DAGs and trees, showing that each specialization results in a more efficient query evaluation.

1 Introduction

The extraction of information from an information source (hereafter, IS) containing term-classified objects is plagued with uncertainty. From the one hand, the indexing of objects, that is the assignment of a set of terms to each object, presents many difficulties, whether it is performed manually by some expert or automatically by a computer programme. In the former case, subjectivity may play a negative role (e.g. see [4]); in the latter case, automatic classification methods may at best produce approximations. On the other hand, the query formulation process, being linguistic in nature, would require perfect attuning of the system and the user language, an assumption that simply does not hold in open settings such as the Web.

A collection of textual documents accessed by users via natural language queries is clearly a kind of IS, where documents play the role of objects and words play the role of terms. In this context, the above mentioned uncertainty is typically dealt with in a quantitative way, i.e. by means of numerical methods:

G. Antoniou et al. (Eds.): SETN 2006, LNAI 3955, pp. 267–278, 2006.

in a document index, each term is assigned a *weight,* expressing the extent to which the document is deemed to be about the term. The same treatment is applied to each user query, producing an index of the query which is a formal representation of the user information need of the same kind as that of each document. Document and query term indexes are then matched against each other in order to estimate the relevance of the document to a query (e.g. see [1]).

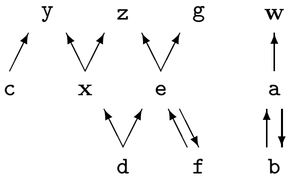
In the present study, we take a different approach, and deal with uncertainty in a *qualitative* way. We view an IS as an agent, operating according to an open world philosophy. The agent knows some facts, but it does not interpret these facts as the only ones that hold; the agent is somewhat aware that there could be other facts, compatible with the known ones, that might hold as well, although they are not captured for lack of knowledge. These facts are, indeed, *possibilities.* One way of defining precisely in logical terms the notion of possibility, is to equate it with the notion of *explanation.* That is, the set of terms associated to an object is viewed as a *manifestation* of a phenomenon, the indexing process, for which we wish to find an explanation, justifying why the index itself has come to be the way it is. In logic, the reasoning required to infer explanations from given theory and observations, is known as *abduction.* We will therefore resort to abduction in order to define precisely the possibilities that we want our system to be able to handle. In particular, we will define an operation that extends an IS by adding to it a set (term, object) pairs capturing the sought possibilities, and then study the property of this operation from a mathematical point of view. The introduced operation can be used also for ordering query answers using a *possibility*-based measure of relevance.

2 Information Sources

Definition 1. An *information source* (IS) S is a pair $S = (O, U)$ where (a) O, the *taxonomy,* is a pair $O = (T, K)$ where T is a finite set of symbols, called the *terms* of the taxonomy, and K is a finite set of conditionals on T, *i.e.* formulas of the form $p \rightarrow q$ where p and q are different terms of the taxonomy; (b) U is a *structure on O,* that is a pair $U = (Obj, I)$ where Obj is a countable set of objects, called the *domain* of the structure, and I is a finite relation from T to Obj, that is $I \subseteq T \times Obj$. K is called the *knowledge base* of the taxonomy, while I is called the *interpretation* of the structure.

As customary, we will sometimes write $I(t)$ to denote the set $I(t) = \{o \in Obj \mid (t, o) \in I\}$, which we call the *extension* of term t. Dually, given an object $o \in Obj$, the *index of o in S, $ind_S(o)$*, is given by the set of terms which have o in their extension: $ind_S(o) = \{t \in T \mid (t, o) \in I\}$. Finally, the *context of o in S, $C_S(o)$*, is defined as: $C_S(o) = ind_S(o) \cup K$. For any object o, $C_S(o)$ consists of terms and simple conditionals that collectively form all the knowledge about o that S has.

Example 1. Throughout the paper, we will use as an example the IS $S = ((T, K), (Obj, I))$ given in the righthand side of Figure 1. The lefthand side of the Figure graphically illustrates the taxonomy of S.



$$\begin{aligned} T &= \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}, \mathtt{w}, \mathtt{x}, \mathtt{y}, \mathtt{z}\} \\ K &= \{\mathtt{c} \to \mathtt{y},\ \mathtt{x} \to \mathtt{y},\ \mathtt{x} \to \mathtt{z},\ \mathtt{d} \to \mathtt{x},\ \mathtt{d} \to \mathtt{e},\ \mathtt{e} \to \mathtt{z}, \\ &\qquad \mathtt{e} \to \mathtt{g},\ \mathtt{e} \to \mathtt{f},\ \mathtt{f} \to \mathtt{e},\ \mathtt{a} \to \mathtt{w},\ \mathtt{a} \to \mathtt{b},\ \mathtt{b} \to \mathtt{a}\} \\ Obj &= \{1\} \\ I &= \{(\mathtt{x}, 1),\ (\mathtt{w}, 1)\} \end{aligned}$$

Fig. 1. An information source

We focus on ISs which satisfy an intuitive minimality criterion, to introduce which a few basic notions from propositional logic are now recalled [3]. Given a set of propositional variables P, a *truth assignment for* P is a function mapping P to the true and false truth values, respectively denoted by $\mathbf{T}$ and $\mathbf{F}$. A truth assignment V *satisfies* a sentence σ of the propositional calculus (PC), $V \models \sigma$, if σ is true in V, according to the classical truth valuation rules of PC. A set of sentences Σ *logically implies* the sentence α, $\Sigma \models \alpha$, iff every truth assignment which satisfies every sentence in Σ also satisfies α.

The *instance set* of an object o in an IS S, denoted as $N_S(o)$, is the set of terms that are logically implied by the context of o in S : $N_S(o) = \{t \in T \mid C_S(o) \models t\}$. For each term t in $N_S(o)$, we will say that o is an *instance* of t. Clearly, $ind_S(o) \subseteq N_S(o)$, therefore o is an instance of each term in $ind_S(o)$.

Definition 2. The index of object o in IS S, $ind_S(o)$, is *non-redundant* iff

$$A \subset ind_S(o) \text{ implies } \{v \in T \mid A \cup K \models v\} \subset N_S(o).$$

An IS is *non-redundant* if all its indices are non-redundant.

In practice, the index of an object is non-redundant if no term in it can be removed without loss of information. It can be easily verified that the IS introduced in the previous example is non-redundant. From now on, we will consider “IS” as a synonym of “non-redundant IS”.

Definition 3. Given a taxonomy $O = (T, K)$, the *query language for* O, $\mathcal{L}_O$, is defined by the following grammar, where t is a term in T:

$$q ::= t \mid q \wedge q' \mid q \vee q' \mid \neg q \mid (q)$$

Any expression in $\mathcal{L}_O$ is termed a *query.* Given an IS $S = (O, U)$, for every object $o \in Obj$, the *truth model of* o *in* S, $V_{o,S}$, is the truth assignment for T defined as follows, for each term $t \in T$:

$$V_{o,S}(t) = \begin{cases} \mathbf{T} \text{ if } C_S(o) \models t \\ \mathbf{F} \text{ otherwise} \end{cases}$$

Given a query φ in $\mathcal{L}_O$, the *answer of* φ *in* S is the set of objects whose truth model satisfies the query:

$$ans(\varphi, S) = \{o \in Obj \mid V_{o,S} \models \varphi\}.$$

In the Boolean model of information retrieval, a document is returned in response to a query if the index of the document satisfies the query. Thus, the above definition extends Boolean retrieval by considering also the knowledge base in the retrieval process.

Query evaluation requires the computation of the truth model of each object o, which in turn requires deciding whether each query term is logically implied by the object context $C_S(o)$. Computing propositional logical implication is in general a difficult task. However, the specific form of the propositional theories considered in this study, makes this computation much simpler, as the remainder of this Section shows. In order to devise an efficient query evaluation procedure, we will resort to graph theoretic concepts.

The *term graph* of a taxonomy O is the directed graph $G_O = (T, E)$, such that $(t, t') \in E$ iff $t \rightarrow t'$ is in K. Figure 1 shows indeed the term graph of the example IS. For simplicity, we will use "term" also to refer to a vertex of the term graph. The *tail of a term* t in G_O, *tail(t),* is the set of terms that can be reached from t by walking the graph edges backward, that is:

$$tail(t) = \{u \in T \mid \text{ there exists a path from } u \text{ to } t \text{ in } G_O\}$$

Proposition 1. For all ISs S and queries $\varphi \in \mathcal{L}_O$, $ans(\varphi, S) = \alpha_S(\varphi)$, where α_S is the *solver* of the IS S, defined as follows:

$$\begin{aligned}
\alpha_S(t) &= \bigcup \{I(u) \mid u \in tail(t)\} \\
\alpha_S(q \wedge q') &= \alpha_S(q) \cap \alpha_S(q') \\
\alpha_S(q \vee q') &= \alpha_S(q) \cup \alpha_S(q') \\
\alpha_S(\neg q) &= Obj \setminus \alpha_S(q)
\end{aligned}$$

The proof of the Proposition relies on structural induction on the query language and on the following Lemma: Given an IS S, a set of terms $A \subseteq T$ and a term $t \in T$, $A \cup K \models t$ iff there is a path in G_O from a letter in A to t.

Example 2. In the IS previously introduced, the term $\mathtt{z}$ can be reached in the term graph by each of the following terms: $\mathtt{z}, \mathtt{x}, \mathtt{d}, \mathtt{e}, \mathtt{f}$. Hence, $tail(\mathtt{z}) = \{\mathtt{z}, \mathtt{x}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. According to the last Proposition, then: $ans(\mathtt{z}, S) = \alpha_S(\mathtt{z}) = I(\mathtt{z}) \cup I(\mathtt{x}) \cup I(\mathtt{d}) \cup I(\mathtt{e}) \cup I(\mathtt{f}) = \{1\}$.

As a consequence of the last Proposition, we have that $\alpha_S(t)$ can be computed in $O(|T| \cdot |Obj| \cdot log\,|Obj|)$ time. Indeed, computing $\alpha_S(t)$ requires the following steps: (a) to derive *tail(t)* by searching the term graph in order to identify every vertex that is backward reachable from t; (b) to access the extension of each term in *tail(t)*; and (c) to compute the union of the involved extensions. The time complexity of step (a) is $|T|^2$, corresponding to the case in which every term is backward reachable from t in the term graph. We assume that step (b) can be performed in constant time, which is negligible with respect to the other values at stake. Let us now consider step (c). By adopting a merge-sort strategy, the union between two extensions can be performed in $n\, log\, n$ time in the size of the input. Since in the worst case the union of $|T|$ extensions must be computed, and each extension is the whole set Obj, we have a $O(|T| \cdot |Obj| \cdot log\,|Obj|)$ time complexity for step (c). Overall, the upper bound for evaluating single-term queries is therefore $O(|T|^2 + |T| \cdot |Obj| \cdot log\,|Obj|)$. Since the size of the domain is expected to be significantly larger than the size of the terminology, the sought upper bound for singles-term queries evaluation is $O(|T| \cdot |Obj| \cdot log\,|Obj|)$.

3 Extended Information Sources

Let us suppose that a user has issued a query against an IS and is not satisfied with the answer, as the answer does not contain objects that are relevant to the user information need. Further, let us assume that the user is not willing to replace the current query with another one, for instance because of lack of knowledge on the available language or taxonomy. In this type of situation, database systems offer practically no support, as they are based on the assumption that users can always articulate their information need in the form of a query. In an information retrieval setting a user in the above described situation could use relevance feedback to pinpoint interesting (relevant) or uninteresting objects among those returned by the system, and ask the system to re-evaluate the query taking into account this information; but what if all the displayed objects are not relevant? In all these, and probably other, cases, the index of the IS suffers from *incompleteness:* it contains correct information, but at least in some cases not *all* the correct information. In other words, there are other facts, compatible with the known ones, which hold as well, although they are not captured for *lack of knowledge.*

To overcome this lack of knowledge, the idea is to relax the index, by expanding it, in a way that is as *reasonable* as possible with respect to the original classification of objects. But what could be a reasonable expansion? By reasonable expansion we mean a *logically grounded* expansion, that is an expansion that *logically implies* the index as it has been created in the first place. Then, the expansion we are talking about is in fact a logical *explanation* of the index. The most general form of explanation in logic is *abduction,* seen as the generation of causes to explain the observed effects of known laws. In our case, the known laws are the sentences of the IS knowledge base, the observed effects are contents of the index, while the cause is the sought index expansion. We will therefore resort to abduction in order to define precisely the expansion that would address the incompleteness of the index.

3.1 Propositional Abduction Problems

The model of abduction that we adopt is the one presented in [2]. Let $\mathcal{L}_V$ be the language of propositional logic over a finite alphabet V of propositional variables. A *propositional abduction problem* is a tuple $\mathcal{A} = \langle V, H, M, Th \rangle$, where $H \subseteq V$ is the set of hypotheses, $M \subseteq V$ is the manifestation, and $Th \subseteq \mathcal{L}_V$ is a consistent theory. $S \subseteq H$ is a solution (or explanation) for $\mathcal{A}$ iff $Th \cup S$ is consistent and $Th \cup S \models M$. $Sol(\mathcal{A})$ denotes the set of the solutions to $\mathcal{A}$. In the context of an IS S, we will consider each object separately. Thus,

- the terms in T play both the role of the propositional variables V and of the hypotheses H, as there is no reason to exclude *apriori* any term from an explanation;
- the knowledge base K plays the role of the theory Th;
- the role of manifestation is played by the index of the object.

Definition 4. Given an IS $S = (O, U)$ and object $o \in Obj$, the *propositional abduction problem for o in S*, $\mathcal{A}_S(o)$, is the propositional abduction problem $\mathcal{A}_S(o) = \langle T, T, ind_S(o), K \rangle$.

The solutions to $\mathcal{A}_S(o)$ are given by:

$$Sol(\mathcal{A}_S(o)) = \{A \subseteq T \mid K \cup A \models ind_S(o)\}$$

where the consistency requirement on $K \cup A$ has been omitted since for no knowledge base K and set of terms A, $K \cup A$ can be inconsistent. Usually, certain explanations are preferable to others, a fact that is formalized in [2] by defining a preference relation $\preceq$ over $Sol(\mathcal{A})$. Letting $a \prec b$ stand for $a \preceq b$ and $b \not\preceq a$, the set of preferred solutions is given by:

$$Sol_{\preceq}(\mathcal{A}) = \{S \in Sol(\mathcal{A}) \mid \nexists S' \in Sol(\mathcal{A}) : S' \prec S\}.$$

Also in the present context a preference relation is desirable, satisfying criteria that reflect the goals of our framework. Here are these criteria, in order of decreasing importance:

1. explanations including only terms in the manifestation are less preferable, as they do not provide any additional information;
2. explanations altering the behavior of the IS to a minimal extent are preferable; this requirement acts in the opposite direction of the previous one, by making preferable solutions that, if incorporated in the IS, minimize the differences in behavior between the so extended IS and the original one;
3. between two explanations that alter the behavior of the IS equally, the simpler one is to be preferred. As explanations are sets, it is natural to equate simplicity with smallness in size.

All the above criteria can be expressed in terms of the effects produced by the extension of an IS, which we term "perturbation".

Definition 5. Given an IS $S = (O, U)$, an object $o \in Obj$ and a set of terms $A \subseteq T$, the *perturbation of A on S with respect to o*, $pert_o(A)$ is given by:

$$pert_o(A) = \{t \in T \mid (C_S(o) \cup A) \models t \text{ and } C_S(o) \not\models t\}$$

that is the set of additional terms in the instance set of o once the index of o is extended with the terms in A.

We can now define the preference relation over solutions of the above stated abduction problem.

Definition 6. Given an IS $S = (O, U)$, an object $o \in Obj$ and two solutions A and A' to the problem $\mathcal{A}_S(o)$, $A \preceq A'$ if either of the following holds:

1. $pert_o(A') = \emptyset$
2. $0 < |pert_o(A)| < |pert_o(A')|$
3. $0 < |pert_o(A)| = |pert_o(A')|$ and $A \subseteq A'$.

The strict correspondence between the clauses in the last Definition and the criteria previously set for the preference relation should be evident. Solutions having an empty perturbation are obviously subsets of the instance set of the object, therefore the first condition of the last Definition captures the first of the three criteria. The second condition establishes preference for solutions that minimize the number of terms that change their truth value from **F** to **T** in the truth model of the object, and thus alter the behavior of the IS *with respect to query answering* to a minimal extent. Between two solutions producing the same alteration, the third condition makes preferable the smaller in size, and so simplicity, criterion number three, is implemented.

We now introduce the notion of *extension* of an IS. The idea is that an extended IS (EIS for short) includes, for each object, the terms of the original index plus those added through the abduction process illustrated above. However, in so doing, non-redundancy may be compromised, since no constraint is posed on the solutions of the abduction problem in order to avoid it. There can be two sources of redundancy when extending a non-redundant index with the solutions to an abduction problem: (1) a solution to an abduction problem may contain taxonomical cycles, and including a whole cycle in the index of an object clearly violates non-redundancy (all terms in the cycle but one can be removed without losing information); and (2) a term in a solution may be a direct descendant of a term in the index. The coexistence of these two terms in the new index violates redundancy, thus the latter can be added only if the former is removed. In order to cope with the former problem, we introduce the operator ρ, which takes as input a set of terms and replaces each cycle occurring in it by any term in the cycle. In order to cope with the latter problem, we introduce a special union operator $\sqcup$ which takes as input two interpretations and adds each pair (t, o) of the second interpretation to the first interpretation after removing any pair (u, o) in the first interpretation such that $t \rightarrow u \in K$. Formally,

$$I_1 \sqcup I_2 = I_2 \cup \{(t, o) \in I_1 \mid \text{for no pair } (v, o) \in I_2,\ v \rightarrow t \in K\}.$$

Definition 7. Given an IS $S = (O, U)$ and an object $o \in Obj$, the *abduced index* of o, $abind_S(o)$, is given by:

$$abind_S(o) = \bigcup Sol_{\preceq}(\mathcal{A}_S(o)) \setminus ind_S(o).$$

The *abduced interpretation of* S, I^+, is given by

$$I^+ = I \sqcup \{\langle t, o\rangle \mid o \in Obj \text{ and } t \in \rho(abind_S(o))\}.$$

Finally, the *extended* IS, S^e, is given by $S^e = (O, U^e)$ where $U^e = (Obj, I^+)$.

3.2 Querying Extended Information Sources

A key role in solving propositional abduction problems is played by single-letter solutions (SLSs). Given an IS S, an object o and a term $t \in T \setminus N_S(o)$, the *single-letter solution of* t is the set $\mu(t)$ given by:

$$\mu(t) = \{t\} \cup (ind_S(o) \setminus \sigma(t))$$

where $\sigma(t) = \{u \in T \mid \text{there is a path in } G_O \text{ from } t \text{ to } u\}$. It can be proven that, for any IS S, object $o \in Obj$ and term $t \in T$, $\mu(t)$ is a solution to $\mathcal{A}_S(o)$ whose perturbation is given by $pert_o(\mu(t)) = \sigma(t) \setminus N_S(o)$. Moreover, if $t \notin N_S(o)$, $\mu(t)$ has the smallest perturbation among the solutions to $\mathcal{A}_S(o)$ including t.

Example 3. Let us consider again the IS S introduced in Example 1, and the problem $\mathcal{A}_S(1)$. The manifestation is given by $ind_S(1) = \{\mathtt{w}, \mathtt{x}\}$ while $N_S(1) = \{\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{w}\}$. Table 1 gives, for each term in $T \setminus N_S(1)$, the σ value, the single-letter solution and its perturbation.

We can now state the main result on query answering in EIS.

Table 1. The single-letter solutions of $\mathcal{A}_S(1)$ and their perturbations

t	$\sigma(t)$	$\mu(t)$	$pert_o(\mu(t))$
a	$\{\mathtt{w}, \mathtt{a}, \mathtt{b}\}$	$\{\mathtt{a}, \mathtt{x}\}$	$\{\mathtt{a}, \mathtt{b}\}$
b	$\{\mathtt{w}, \mathtt{a}, \mathtt{b}\}$	$\{\mathtt{b}, \mathtt{x}\}$	$\{\mathtt{a}, \mathtt{b}\}$
c	$\{\mathtt{c}, \mathtt{y}\}$	$\{\mathtt{c}, \mathtt{x}, \mathtt{w}\}$	$\{\mathtt{c}\}$
d	$\{\mathtt{d}, \mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$	$\{\mathtt{d}, \mathtt{w}\}$	$\{\mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}\}$
e	$\{\mathtt{e}, \mathtt{f}, \mathtt{z}, \mathtt{g}\}$	$\{\mathtt{e}, \mathtt{x}, \mathtt{w}\}$	$\{\mathtt{e}, \mathtt{f}, \mathtt{g}\}$
f	$\{\mathtt{f}, \mathtt{e}, \mathtt{z}, \mathtt{g}\}$	$\{\mathtt{f}, \mathtt{x}, \mathtt{w}\}$	$\{\mathtt{e}, \mathtt{f}, \mathtt{g}\}$
g	$\{\mathtt{g}\}$	$\{\mathtt{g}, \mathtt{x}, \mathtt{w}\}$	$\{\mathtt{g}\}$

Proposition 2. For all ISs S and terms $t \in T$, $ans(t, S^e) = \alpha_{S^e}(t) = \alpha_S(t) \cup \beta(t)$, where

$$\beta(t) = \{o \in Obj \mid t \in abind_S(o)\}.$$

By unfolding the relevant definitions, we have that $ans(t, S^e) = \{o \in Obj \mid ind_S(o) \cup K \models t\} \cup \{o \in Obj \mid abind_S(o) \cup K \models t\} = \alpha_S(t) \cup \{o \in Obj \mid abind_S(o) \cup K \models t\}$. As it can be proven, the second term of the last union operation is given by:

$$\alpha_S(t) \cup \{o \in Obj \mid t \in abind_S(o)\},$$

and the Proposition obtains. In turn, the set $abind_S(o)$ is derived, based on the above stated properties of SLSs, as follows:

Proposition 3. Given an IS S and an object $o \in Obj$, let d_o be the least positive perturbation of the solutions to $\mathcal{A}_S(o)$, that is:

$$d_o = min\{|pert_o(A)| \mid A \in Sol(\mathcal{A}_S(o)) \text{ and } pert_o(A) > 0\}.$$

Then, $abind_s(o) = \{t \in T \setminus N_S(o) \mid |\sigma(t) \setminus N_S(o)| = d_o\}$.

Example 4. Let us consider again the problem $\mathcal{A}_S(1)$ of the last Example. From Table 1 and the last Proposition, it follows that $abind_S(1) = \{\mathtt{c}, \mathtt{g}\}$ and that $I^+ = I \cup \{(\mathtt{c}, 1), (\mathtt{g}, 1)\}$.

Proposition 3 suggests the following algorithm for computing $\beta(t)$: for each object $o \in Obj$, select the letters u in $T \setminus N_S(o)$ that minimize the size of $\sigma(u) \setminus N_S(o)$. If t is amongst these letters, $o \in \beta(t)$, otherwise $o \notin \beta(t)$. This allows us to establish that $\alpha_{S^e}(t)$ can be computed in $O(|T| \cdot |Obj|^2 \cdot log\,|Obj|)$ time (the proof is omitted for reasons of space). It follows that query evaluation on EISs worsens complexity by a factor equals to the size of the domain, and is consistent with the complexity results derived from propositional abduction problems [2].

4 Special Information Sources

We conclude this study by applying the ideas developed so far to two special classes of ISs, corresponding to two special structures of the taxonomy. The first class consists of the ISs whose term graphs are directed acyclic graphs (DAGs) with a maximal element $\top$. We call these ISs "hierarchical". Indeed, hierarchical taxonomies are common in object-oriented models, where subsumption is a partial ordering relation amongst classes and maximal elements are introduced in order to tie classes up, thus making each class reachable from the top. The second class includes a special case of the first, that is term graphs that are rooted trees. This kind of taxonomies are common in catalogs, directories and information retrieval systems on the Web.

Proposition 4. Let a *hierarchical* information source (HIS) be an IS whose term graph is a DAG with a greatest element $\top$. Then, for all HISs S and terms $t \in T$, $t \neq \top$, $ans(t, S^e) = \cap\ \{\alpha_S(u) \mid t \to u \in K\}$.

Accordingly, an object o is in the result of query t against the EIS S^e just in case it is an instance of all the immediate generalizations of t in S. Indeed, if o were an instance of t, it would, as a consequence, be an instance of all t's generalizations, thus the explanation of the current index offered by the system is the most reasonable one can conceive. As a result, the behavior of the query mechanism turns out to be compliant with intuition. Notice that asking the query $\top$ on the extended IS, a case not dealt with by the last Proposition, does not make much sense, since already $ans(\top, S) = Obj$.

Example 5. Let us consider the HIS S having as taxonomy the one shown in Figure 2, where the index of the only object 1 consists of the terms `Miniatures` and `MovingPictureCams`. The problem $\mathcal{A}_S(1)$ has two minimal solutions, given by $\mu(\mathtt{Reflex})$ and $\mu(\mathtt{UnderwaterDevices})$. Thus, object 1 should be returned in response to the query `Reflex` on S^e, and in fact $ans(\mathtt{Reflex}, S^e)$ is given by:

$$\begin{aligned} \cap\ \{\alpha_S(u) \mid \mathtt{Reflex} \to u \in K\} &= \alpha_S(\mathtt{StillCams}) \\ &= I(\mathtt{StillCams}) \cup I(\mathtt{Miniatures}) \cup I(\mathtt{Reflex}) = \{1\} \end{aligned}$$

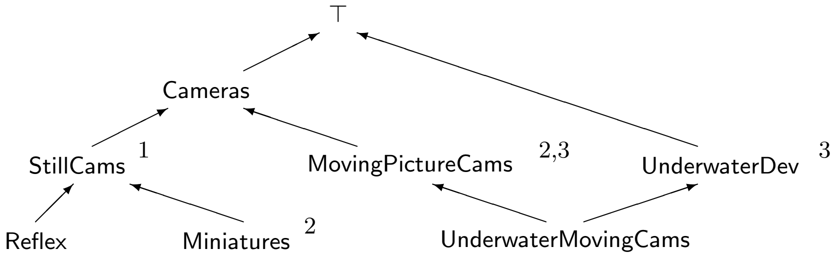


Fig. 2. A hierarchical information source

Letting P stand for S^e and considering $\mathcal{A}_P(1)$, 1 should be included in the answer to the query `UnderwaterMovingCams` on P^e. Indeed,

$$ans(\texttt{UnderwaterMovingCams}, P^e) = \cap \{\alpha_P(u) \mid \texttt{UnderwaterMovingCams} \rightarrow u \in K\}$$
$$= \alpha_P(\texttt{MovingPictureCams}) \cap \alpha_P(\texttt{UnderwaterDevices}) = \{1\}$$

since 1 is an instance of `MovingPictureCams` in S and has become an instance of `UnderwaterDevices` in P.

From a complexity point of view, the Proposition permits to compute an upper bound on the evaluation of queries on an extended HIS. Specifically, it can be proved that $\alpha_{S^e}(t)$, where S is a HIS, can be computed in $O(|T|^2 \cdot |Obj| \cdot log\,|Obj|)$ time. Table 2 summarizes the complexity results obtained for query evaluation on the classes of IS examined in this study. The last result indicates that the evaluation of queries on extended HISs is worse than that on ISs by a factor proportional to the size of the terminology. This is a significant improvement over the general case, where the factor is proportional to the size of the domain (Proposition 0). This difference reflects the fact that in the general case, $\alpha_{S^e}(t)$ must be computed in an object-based (or class-based) fashion, *i.e.* by considering one object (class) at a time, while the evaluation of $\alpha_{S^e}(t)$ on a HIS proceeds in a term-based fashion, *i.e.* by considering the terms that are immediate successors of t in the term graph. This also simplifies the implementation, as it avoids to compute and keep track of classes.

Table 2. Summary of complexity results for query evaluation

IS type	Complexity
Simple	$O(\|T\| \cdot \|Obj\| \cdot log\,\|Obj\|)$
Extended	$O(\|T\| \cdot \|Obj\|^2 \cdot log\,\|Obj\|)$
Extended Hierarchical	$O(\|T\|^2 \cdot \|Obj\| \cdot log\,\|Obj\|)$
Extended Tree	$O(\|T\| \cdot \|Obj\| \cdot log\,\|Obj\|)$

Proposition 5. Let a *tree* information source (TIS) be an IS whose term graph is a rooted tree with root $\top$. Then, for all TISs S and terms $t \in T$, $t \neq \top$,

$$ans(t, S^e) = \{\alpha_S(u) \mid t \rightarrow u \in K\}.$$

Proof: A TIS is a HIS in which every term different from $\top$ has exactly one immediate successor in the term graph.

The complexity of query evaluation on extended TISs is clearly the same as that on ISs.

5 Ranked Answers

The abduction framework described can be also exploited for obtaining ordered answers. In order to illustrate how, let us use a superscript to indicate the iteration at which an IS is generated, that is, $S = S^0$, $S^e = S^1$, $(S^e)^e = S^2$ and so on. Moreover, let N be the iteration at which the fixed point is reached, *i.e.* $S^{N-1} \subset S^N = S^{N+1} = S^{N+2} = \dots$. The set of objects that the user will get in response to a query φ on the extensions of the IS S, is given by:

$$answers_S(\varphi) = \bigcup_{i=0}^{N} \alpha_{S^i}(\varphi)$$

We can give all of these objects to the user as a response to the query φ on S, ordered according to the iteration at which each object would start appearing in the answer. In particular, we can define the *rank* of an object $o \in answers_S(\varphi)$, denoted by $rank_S(o, \varphi)$, as follows:

$$rank_S(o, \varphi) = min\{\ k \mid o \in \alpha_{S^k}(\varphi)\}$$

The answer that will be returned by φ on S, the *ranked answer*, is an ordering of sets, i.e. the ordering:

$$rans(\varphi, S) = \langle \{o \mid rank_S(o, \varphi) = 1\}, \dots, \{o \mid rank_S(o, \varphi) = N\} \rangle$$

For example, consider the hierarchical IS presented in Figure 2, where the extension of each term is shown on the right of the term (*i.e.*, $I(\texttt{MovingPictureCams}) = \{2, 3\}$). Let suppose that $\varphi = \texttt{UMC}$. In this case we have: $\alpha_{S^0}(\texttt{UMC}) = \emptyset$, $\alpha_{S^1}(\texttt{UMC}) = \{3\}$, $\alpha_{S^2}(\texttt{UMC})\{1, 2, 3\}$. So the ranked answer to UMC is $rans(\texttt{UMC}, S) = \langle \{3\}, \{1, 2\} \rangle$.

6 Conclusions

Indexing accuracy and consistency are difficult to maintain. To alleviate this problem we have proposed a mechanism which allows liberating the index of a source in a gradual manner. This mechanism is governed by the notion of explanation, logically captured by abduction.

The proposed method can be implemented as an answer enlargement[1] process where the user is not required to give additional input, but from expressing his/her desire for more objects. The introduced framework can be also applied for ranking the objects of an answer according to an explanation-based measure of relevance.

[1] If the query contains negation then the answer can be reduced.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *"Modern Information Retrieval"*. ACM Press, Addison-Wesley, 1999.
2. T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, January 1995.
3. H. Enderton. *A mathematical introduction to logic.* Academic Press, N. Y., 1972.
4. P. Zunde and M. Dexter. "Indexing Consistency and Quality". *American Documentation*, 20(3):259–267, July 1969.