Multimedia Interpretation as Abduction

S. Espinosa Peraldi, A. Kaya, S. Melzer, R. Möller, M. Wessel

Hamburg University of Technology, Germany

Abstract. In this work we present an approach to interpret information extracted from multimedia documents through Abox abduction, which we consider as a new type of non-standard retrieval inference service in Description Logics (DLs). We discuss how abduction can be adopted to interpret multimedia content through explanations. In particular, we present a framework to generate explanations, and introduce a preference measure for selecting 'preferred' explanations.[1]

1 Introduction

Automated extraction of information from different types of multimedia documents such as image, text, video, and audio becomes more and more relevant for intelligent retrieval systems. An intelligent retrieval system is a system with a knowledge base and capabilities that can be used to establish connections between a request and a set of data based on the high-level semantics of the data (which can also be documents). Typically, nowadays, automated semantics extraction from multimedia occurs by using low-level features and is often limited to the recognition of isolated items if even. Examples are single objects in an image, or single words (or maybe phrases) in a text. However, multimedia documents such as images usually present more than objects detectable in a bottom-up fashion. For instance, an image may illustrate an abstract concept such as an event. An event in a still image can hardly be perceived without additional high-level knowledge.

We see multimedia interpretation as abduction (reasoning from effects to causes) in that we reason from observations (effects) to explanations (causes). The aim of this work is to present a novel approach for multimedia interpretation through Abox abduction, which we consider as a new type of non-standard retrieval inference service in DLs. In particular, we focus on the use of DL-safe-like rules for finding explanations and introduce a preference measure for selecting 'preferred' explanations.

2 Related Work in Media Interpretation and Abduction

The idea of formalizing interpretation as abduction is investigated in [4] in the context of text interpretation. In [8], Shanahan presents a formal theory of robot

[1] This work is partially supported by the EU-funded projects BOEMIE (Bootstrapping Ontology Evolution with Multimedia Information Extraction, IST-FP6-027538) and TONES (Thinking ONtologiES, FET-FP6-7603).

perception as a form of abduction. In this work, low-level sensor data is transformed into a symbolic representation of the world in first-order logic and abduction is used to derive explanations. In the context of scene interpretation, recently, in [7] the use of DLs for scene interpretation processes is described.

In this paper we present a novel approach based on the combination of the works in [4, 8] and [7], and indicate how formal representation and reasoning techniques can be used for interpretation of information extracted from multimedia documents. The approach used description logics and rules, with abduction implemented with backward-chaining applied to the rules. In contrast to approaches such as [5], which use abduction in the context of rules in logic programming, we use description-logic reasoning for proving subgoals of (non-recursive) rules. Other approaches for abduction in description logics (e.g., [1]) have dealt with concept abduction only. In [3] among other abductive reasoning tasks in DLs also Abox abduction is discussed. A solution to the Abox abduction problem is formally presented, but it is not shown how to derive solutions.

Abduction is investigated for supporting information retrieval based on high-level descriptions on media content. The approach builds on [6] and, in contrast to later related work such as [2], the approach is integrated into a mainstream description logic system and is based on high-level descriptions of media content.

3 Retrieval Inference Services

Before introducing abduction as a new inference service, we start with an overview of retrieval inference services that are supported by state-of-the-art DL reasoners.

The *retrieval* inference problem w.r.t. a Tbox $\mathcal{T}$ is to find all individuals mentioned in an Abox $\mathcal{A}$ that are instances of a certain concept C: $\{x$ mentioned in $\mathcal{A} \mid (\mathcal{T}, \mathcal{A}) \models x : C\}$. In addition to the basic retrieval inference service, expressive query languages are required in practical applications. Well-established is the class of conjunctive queries. A *conjunctive query* consists of a *head* and a *body*. The head lists variables for which the user would like to compute bindings. The body consists of query atoms (see below) in which all variables from the head must be mentioned. If the body contains additional variables, they are seen as existentially quantified. A query answer is a set of tuples representing bindings for variables mentioned in the head. A query is a structure of the form $\{(X_1, \ldots, X_n) \mid atom_1, \ldots, atom_m\}$.

Query atoms can be *concept* query atoms ($C(X)$), *role* query atoms ($R(X, Y)$), *same-as* query atoms ($X = Y$) as well as so-called *concrete domain* query atoms. The latter are introduced to provide support for querying the concrete domain part of a knowledge base and will not be covered in detail here. Complex queries are built from query atoms using boolean constructs for conjunction (indicated with comma) or union ($\vee$).

In *standard* conjunctive queries, variables (in the head and in query atoms in the body) are bound to (possibly anonymous) domain objects. A system supporting (unions of) standard conjunctive queries is QuOnto. In so-called *grounded*

conjunctive queries, $C(X)$, $R(X,Y)$ or $X = Y$ are true if, given some bindings α for mapping from variables to *individuals mentioned in the Abox* $\mathcal{A}$, it holds that $(\mathcal{T},\mathcal{A}) \models \alpha(X) : C$, $(\mathcal{T},\mathcal{A}) \models (\alpha(X),\alpha(Y)) : R$, or $(\mathcal{T},\mathcal{A}) \models \alpha(X) = \alpha(Y)$, respectively. In grounded conjunctive queries the standard semantics can be obtained for so-called tree-shaped queries by using corresponding existential restrictions in query atoms. Due to space restrictions, we cannot discuss the details here. In the following, we consider only grounded conjunctive queries, which are supported by KAON2, Pellet, and RacerPro.

In practical applications it is advantageous to name subqueries for later reuse, and practical systems, such as for instance RacerPro, support this for grounded conjunctive queries with non-recursive rules of the following form

$$P(X_1,\ldots,X_{n_1}) \leftarrow A_1(Y_1),\ldots,A_l(Y_l),R_1(Z_1,Z_2),\ldots,R_h(Z_{2h-1},Z_{2h}). \quad (1)$$

The predicate term to the left of $\leftarrow$ is called the head and the rest is called the body (a set of atoms), which, informally speaking, is seen as a conjunction of predicate terms. All variables in the head have to occur in the body, and rules have to be non-recursive (with the obvious definition of non-recursivity). Since rules have to be non-recursive, the replacement of query atoms matching a rule head is possible (unfolding, with the obvious definition of matching). The rule body is inserted (with well-known variable substitutions and variable renamings). If there are multiple rules (definitions) for the same predicate P, corresponding disjunctions are generated. The unfolding process starts with the set of atoms of a query. Thus, we start with a set of atom sets.

$$\{\{atom_1, atom_2, \ldots atom_k\}\}$$

Each element of the outer set represents a disjunct. Now, wlog we assume that there are n rules matching $atom_2$. Then, the set $\{atom_1, atom_2, \ldots atom_k\}$ is eliminated and replaced with the sequence of sets $\{atom_1\} \cup$ replace_vars(body$(rule_1), head(rule_1), atom_2) \cup \{\ldots atom_k\}, \ldots, \{atom_1\} \cup$replace_vars(body $(rule_n), head(rule_n), atom_2) \cup \{\ldots atom_k\}$. The unfolding process proceeds until no replacement is possible any more (no rules match). The unfold operator is used in the abduction process, which is described in the next section.

4 Abduction as a Non-Standard Inference Service

In this paper, we argue that abduction can be considered as a new type of non-standard retrieval inference service. In this view, observations (or part of them) are utilized to constitute queries that have to be answered. Contrary to existing retrieval inference services, answers to a given query cannot be found by simply exploiting the knowledge base. In fact, the abductive retrieval inference service has the task of acquiring what should be added to the knowledge base in order to positively answer a query.

More formally, for a given set of Abox assertions Γ (in form of a query) and a knowledge base $\Sigma = (\mathcal{T},\mathcal{A})$, the abductive retrieval inference service aims to

derive all sets of Abox assertions Δ (explanations) such that $\Sigma \cup \Delta \models \Gamma$ and the following conditions are satisfied:

- $\Sigma \cup \Delta$ is satisfiable, and
- Δ is a minimal explanation for Γ, i.e., there exists no other explanation Δ' in the solution set that is not equivalent to Δ and it holds that $\Sigma \cup \Delta' \models \Delta$.

In addition to minimality (simplicity), in [4] another dimension called consilience is mentioned. An explanation should explain as many elements of Γ as possible. Both measures are contradictory.

In the next section, we will focus on the use of abductive retrieval inference services for multimedia interpretation and address two important issues, namely finding explanations that meet the conditions listed above and selecting 'preferred' ones.

5 Interpretation of Multimedia Documents

For intelligent retrieval of multimedia documents such as images, videos, audio, and texts, information extracted by media analysis techniques has to be enriched by applying high-level interpretation techniques. The interpretation of multimedia content can be defined as the recognition of abstract knowledge, in terms of concepts and relations, which are not directly extractable by low-level analysis processes, but rather require additional high-level knowledge. Furthermore, such abstract concepts are represented in the background knowledge as aggregate concepts with constraints among its parts.

In this section, we start by specifying the requirements for the abduction approach by defining its input and output. Then, we proceed with describing the framework for generating explanations, and finally introduce a scenario with a particular example involving for image interpretation where various explanations are generated and the usefulness of a preference score is demonstrated.

5.1 Requirements for Abduction

The abduction approach requires as input a knowledge base Σ consisting of a Tbox $\mathcal{T}$ and an Abox $\mathcal{A}$. We assume that the information extracted from a multimedia document through low-level analysis (e.g., image analysis) is formally encoded as a set of Abox assertions (Γ). For example, in the context of images for every object recognized in an image, a corresponding concept assertion is found in Γ. Usually, the relations that can be extracted from an image are spatial relations holding among the objects in the image. These relations are also represented as role assertions in Γ. In order to construct a high-level interpretation of the content in Γ, the abduction process will extend the Abox with new concept and role assertions describing the content of the multimedia document at a higher level.

The output of the abduction process is formally defined as a set of assertions Δ such that $\Sigma \cup \Delta \models \Gamma$, where $\Sigma = (\mathcal{T}, \mathcal{A})$ is the knowledge base (usually the

Abox A is assumed to be empty), Γ is a given set of low-level assertions, and Δ is an explanation, which should be computed. The solution Δ must satisfy certain side conditions (see Section 4). To compute the explanation Δ in our context we modify this equation into

$$\Sigma \cup \Gamma_1 \cup \Delta \models \Gamma_2, \tag{2}$$

where the assertions in Γ will be split into bona fide assertions (Γ_1) and assertions requiring fiats (Γ_2).[2] Bona fide assertions are assumed to be true by default, whereas fiat assertions are aimed to be explained. The abduction process tries to find explanations (Δ) such that Γ_2 is entailed. This entailment decision is implemented as (boolean) query answering. The output Δ of the abduction process is represented as an Abox. Multiple solutions are possible.

5.2 The Abduction Framework

The abduction framework exploits the non-recursive rules of Σ to answer a given query in a backward-chaining way (see Framework 1). The function `compute_explanations` gets Σ, Γ_1, and Γ_2 as input. We assume a function `transform_into_query` that is applied to a set of Abox assertions Γ_2 and returns a set of corresponding query atoms. The definition is obvious and left out for brevity. Since the rules in Σ are non-recursive, the unfolding step (see Line 2 in Framework 1) in which each atom in the transformed Γ_2 is replaced by the body of a corresponding rule is well-defined. The function `unfold` returns a set of atom sets (each representing a disjunct introduced by multiple matching rules, see above).

The function `explain` computes an explanation Δ for each $\gamma \in \Gamma_2'$. The function `vars` (or `inds`) returns the set of all vars (or inds) mentioned in the argument structures. For each variable in γ a new individual is generated (see the set *new_inds* in Line 7). Besides old individuals, these new individuals are used in a non-deterministic variable substitution. The variable substitution σ_{γ,new_inds} (line 8) is inductively extended as follows:

- $\sigma_{\gamma,new\_inds}(\{a_1,\ldots,a_n\}) =_{def} \{\sigma_{\gamma,new\_inds}(a_1),\ldots,\sigma_{\gamma,new\_inds}(a_n)\}$
- $\sigma_{\gamma,new\_inds}(C(x)) =_{def} C(\sigma_{\gamma,new\_inds}(x))$
- $\sigma_{\gamma,new\_inds}(R(x,y)) =_{def} R(\sigma_{\gamma,new\_inds}(x), \sigma_{\gamma,new\_inds}(y))$
- $\sigma_{\gamma,new\_inds}(x) =_{def} x$ if x is an individual

The function `transform` maps $C(i)$ into $i : C$ and $R(i,j)$ into $(i,j) : R$, respectively. All satisfiable explanations Δ derived by `explain` are added to the set of explanations Δs. The function `compute-preferred-explanations` transforms the Δs into a poset according to a preference measure and returns the maxima as a set of Aboxes. The preference score of a Δ used for the poset order relation is: $S_{pref}(\Delta) := S_i(\Delta) - S_h(\Delta)$ where S_i and S_h are defined as follows.

[2] With the obvious semantics we slightly abuse notation and allow a tuple of sets of assertions Σ to be unioned with a set of assertions $\Gamma_1 \cup \Delta$.

– $S_i(\Delta) := |\{i | i \in inds(\Delta) \text{ and } i \in inds(\Sigma \cup \Gamma_1)\}|$
– $S_h(\Delta) := |\{i | i \in inds(\Delta) \text{ and } i \in new\_inds\}|$

Algorithm 1 The Abduction Framework

1: **function** compute_explanations($\Sigma, \Gamma_1, \Gamma_2, S$) : set of Aboxes
2: $\Gamma_2' :=$ unfold(transform_into_query(Γ_2), Σ) // $\Gamma_2' = \{\{atom_1, \ldots, atom_m\}, \ldots\}$
3: $\Delta s := \{\Delta \mid \exists \gamma \in \Gamma_2'.(\Delta =$ explain(Σ, Γ_1, γ), $\Sigma \cup \Gamma_1 \cup \Delta \not\models \bot)\}$
4: **return** compute_preferred_explanations($\Sigma, \Gamma_1, \Delta s, S$)

5: **function** explain(Σ, Γ_1, γ) : Abox
6: $n :=$ |vars(γ)|
7: $new\_inds := \{new\_ind_i \mid i \in \{1 \ldots n\}\}$, where $new\_inds \cap (inds(\Sigma) \cup inds(\Gamma_1)) = \emptyset$
8: $\Delta :=$ {transform(a) | $\exists \sigma_{\gamma, new\_inds} : vars(\gamma) \mapsto (inds(\Sigma) \cup inds(\Gamma_1) \cup new\_inds)$.
9: $(a \in \sigma_{\gamma, new\_inds}(\gamma), (\Sigma \cup \Gamma_1) \not\models a)\}$
10: **return** Δ

11: **function** compute_preferred_explanations($\Sigma, \Gamma_1, \Delta s, S$) : set of Aboxes
12: **return** maxima(poset($\Delta s, \lambda(x, y) \bullet S(x) < S(y)$))

Depending on the preference function given as the actual parameter for the argument S, the procedure `compute_explanations` can be considered as an approximation w.r.t. the minimality and consilience condition defined in Section 4. It adds to the explanation those query atoms that cannot be proven to hold.

For the abduction framework, only the rules are considered. The GCIs should be used for abduction as well, however. We might accomplish this by approximating the Tbox with the DLP fragment and, thereby, see the Tbox axioms from a rules perspective in order to better reflect the Tbox in the abduction process. The procedure does not add irrelevant atoms (spurious elements of an explanation), in case the rules are well-engineered and do not contain irrelevant ways to derive assertions. The procedure could be slightly modified to check for those redundancies.

5.3 An Example for Image Interpretation as Abduction

For the image shown in Figure 1, we suppose the Abox in Figure 2 is provided by low-level image analysis. Furthermore, a sample Tbox of the athletics domain and a small set of rules are assumed to be provided as background knowledge Σ (see Figure 3).

In order to find a 'good' high-level interpretation of this image, we divide the Abox Γ into Γ_1 and Γ_2 following Equation 2. In this example Γ_1 contains $\{pole_1 : Pole, human_1 : Human, bar_1 : Bar\}$ and Γ_2 contains $\{(bar_1, human_1) : near\}$. Consequently, the abductive retrieval inference service computes the following boolean query in line 2: Q_1:={() | $near(bar_1, human_1)$}. In this paper we do not elaborate on the strategy to determine which Γ_2 to actually choose.
Obviously, both rules in Σ match with the 'near' atom in query Q_1. Therefore, the abduction framework first generates explanations by non-deterministically

Fig. 1. A pole vault event.

$$
\begin{aligned}
pole_1 &: Pole \\
human_1 &: Human \\
bar_1 &: Bar \\
(bar_1,\, human_1) &: near
\end{aligned}
$$

Fig. 2. An Abox Γ representing the results of low-level image analysis.

$$
\begin{aligned}
Jumper &\sqsubseteq Human \\
Pole &\sqsubseteq Sports\_Equipment \\
Bar &\sqsubseteq Sports\_Equipment \\
Pole \sqcap Bar &\sqsubseteq \bot \\
Pole \sqcap Jumper &\sqsubseteq \bot \\
Jumper \sqcap Bar &\sqsubseteq \bot \\
Jumping\_Event &\sqsubseteq \exists_{\leq 1} hasParticipant.Jumper \\
Pole\_Vault &\sqsubseteq Jumping\_Event \sqcap \exists hasPart.Pole \sqcap \exists hasPart.Bar \\
High\_Jump &\sqsubseteq Jumping\_Event \sqcap \exists hasPart.Bar \\
near(Y,Z) &\leftarrow Pole\_Vault(X), hasPart(X,Y), Bar(Y), \\
&\quad\; hasPart(X,W), Pole(W), hasParticipant(X,Z), Jumper(Z) \\
near(Y,Z) &\leftarrow High\_Jump(X), hasPart(X,Y), Bar(Y), \\
&\quad\; hasParticipant(X,Z), Jumper(Z)
\end{aligned}
$$

Fig. 3. A tiny example Σ consisting of a Tbox and DL-safe rules.

substituting variables in the query body with different instances from Γ_1 or with new individuals. Some intermediate Δ results turn out to be unsatisfiable (e.g., if the bar is made into a pole by the variable subsitution process). However, several explanations still remain as possible interpretations of the image. The preference score is used to identify the 'preferred' explanations. For example, considering the following explanations of the image

- $\Delta_1 = \{new\_ind_1 : Pole\_Vault, (new\_ind_1, bar_1) : hasPart, (new\_ind_1, new\_ind_2) : hasPart,\ new\_ind_2 : Pole,\ (new\_ind_1, human_1) : hasParticipant,\ human_1 : Jumper\}$
- $\Delta_2 = \{new\_ind_1 : Pole\_Vault,\ (new\_ind_1, bar_1) : hasPart,\ (new\_ind_1, pole_1) : hasPart,$
  $(new\_ind_1, human_1) : hasParticipant,\ human_1 : Jumper\}$
- $\Delta_3 = \{new\_ind_1 : High\_Jump, (new\_ind_1, bar_1) : hasPart, (new\_ind_1, human_1) : hasParticipant,$
  $human_1 : Jumper\}$

the preference measure of Δ_1 is calculated as follows: Δ_1 incorporates the individuals $human_1$ and bar_1 from Γ_1 and therefore $S_i(\Delta_1)$=2. Furthermore, it hypothesizes two new individuals, namely new_ind_1 and new_ind_2, such that

$S_h(\Delta_1)$=2. The preference score of Δ_1 is $S(\Delta_1)$= $S_i(\Delta_1)$- $S_h(\Delta_1)$=0. Similarly, the preference scores of the second and third explanations are $S(\Delta_2)$=2 and $S(\Delta_3)$=1. After transforming the Δs into a poset, the algorithm computes the maxima. In our case, the resulting set of Aboxes contains only one element, Δ_2, which represents the 'preferred' explanation. Indeed, the result is plausible, since this image should better be interpreted as showing a pole vault and not a high jump, due to the fact that low-level image analysis could detect a pole, which should not be ignored as in the high-jump explanation.

6 Summary

In this paper we presented a novel approach to interpret multimedia data using abduction with description logics that makes use of a new type of non-standard retrieval service in DLs. We showed that results from low-level media analysis can be enriched with high-level descriptions using our Abox abduction approach. In this approach, backward-chained DL-safe-like rules are exploited for generating explanations. For each explanation, a preference score is calculated in order to implement the selection of 'preferred' explanations. Details of the approach have been discussed with a particular example for image interpretation. An implementation of the abduction process described in this paper is available as a non-standard retrieval service integrated in RacerPro.

References

1. S. Colucci, T. Di Noia, E. Di Sciascio, M. Mongiello, and F. M. Donini. Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. In *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*, pages 41–50, 2004.
2. E. Di Sciascio, F.M. Donini, and M. Mongiello. A description logic for image retrieval. In *Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence*, number 1792 in Lecture Notes in Computer Science, pages 13–24. Springer, 1999.
3. C. Elsenbroich, O. Kutz, and U. Sattler. A Case for Abductive Reasoning over Ontologies. In *OWL: Experiences and Directions*, 2006.
4. J. R. Hobbs, M. Stickel, D. Appelt, and P. Martin. Interpretation as abduction. *Artificial Intelligence*, 63:69–142, 1993.
5. A. Kakas and M. Denecker. Abduction in logic programming. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond. Part I*, number 2407 in LNAI, pages 402–436. Springer, 2002.
6. R. Möller, V. Haarslev, and B. Neumann. Semantics-based information retrieval. In *Proc. IT&KNOWS-98: International Conference on Information Technology and Knowledge Systems, 31. August- 4. September, Vienna, Budapest*, pages 49–6, 1998.
7. B. Neumann and R. Möller. On Scene Interpretation with Description Logics. In H.I. Christensen and H.-H. Nagel, editors, *Cognitive Vision Systems: Sampling the Spectrum of Approaches*, number 3948 in LNCS, pages 247–278. Springer, 2006.
8. Murray Shanahan. Perception as Abduction: Turning Sensor Data Into Meaningful Representation. *Cognitive Science*, 29(1):103–134, 2005.