Data mining techniques for Multimedia content management

by Turlov Andrey

Supervisor: Prof. Dr. Ralf Möller
Advisor: Michael Wessel.

Technische Universität Hamburg-Harburg

Submitted in partial fulfilment of the requirements for the degree
Master of Science in Information and Media Technologies

Hamburg, January 2009

Declaration

I declare that:
this work has been prepared by me,
all literal or content based quotations are clearly pointed out,
and no other sources or aids than the declared ones have been used.

Hamburg, January 2009
Turlov Andrey

Acknowledgements

I would like to thank Prof. Dr. Ralf Möller for providing me with a very interesting and challenging topic of research and for giving me the opportunity to perform this work at the Technical University Hamburg-Harburg.

I would also like to thank Michael Wessel for helping me to find a better solution to the problem and technical support.

Further I want to thank Atila Kaya for providing me with the test material and advices.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Motivation

The growing amount of electronic documents is a problem found in proprietary as well as in public repositories. In this context, the web is a representative example where the need of logic-based information retrieval to enhance precision and recall is evident. The project *BOEMIE (Bootstrapping Ontology Evolution with Multimedia Information Extraction)*, funded by the European Commission, is currently involved in developing a system to extract automatically information from multimedia content. The BOEMIE project proposes a specific approach to knowledge acquisition, which uses multimedia ontology. An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts and can be used to reason about the properties of that domain. The evaluation of the BOEMIE project concerns the collection of information about the domain of athletic events that includes concepts tournaments, meetings, training, athletes, persons, faces, etc.

The extracted modality-specific concepts are said to be mid-level, high-level or both. Mid-level concepts are those that can be directly instantiated by the relevant analysis module, using some modality-specific analysis technique. High-level concepts are those that are instantiated by the reasoning services, by means of instantiated mid-level concepts and rules within the ontology. Mid-level objects (such as face, body, etc.) are extracted from several modalities such as still images, text or video. Figure 1 depicts an example of a multimedia document that contains of two modalities: image that has a caption and text.

Figure 1. An example of multimedia document

The result of the extraction is represented in an OWL file (for more information see Section 3.3) which consists of three parts corresponding to the segmentation of the document, the classification of segments as modality-specific mid-level concepts and their relations. In terms of description logic this information forms a knowledge base, which is represented in two parts: T-BOXes and A-BOXes (see Section 3.2 for more details). T-BOXes and A-BOXes deliver an appropriate set of concept and role assertions (w.r.t. a domain ontology) as input for the higher-level multimedia interpretation where more abstract (high-level) knowledge (a person for example) can be discovered with the help of reasoning about multimedia ontologies.

The reasoning procedure uses predefined rules that determine a set of mid-level concepts and the relationship between them needed for the extraction of high-level concepts. Unfortunately, in some cases it is possible that multimedia analysis is not able to deliver enough assertions to satisfy the conditions of the rules and therefore the interpretation of multimedia objects at a high level fails. Similarly, the set of interpretation rules that is defined by a human is sometimes not able to cover some structures of instances of mid-level concepts. The performance of high-level interpretation procedure can be increased on two different levels. On one hand, the methods used for multimedia analysis of different modalities (still images, video, text, etc) can be improved. On the other hand, a set of interpretation rules can be enriched with the help of different rule-learning techniques or with additional information from other modalities.

1.2 Scope of this thesis

The task of this thesis is first to make a research about possible approaches for rule learning and then to define an appropriate method to increase the performance of high-level interpretation procedure on the level of interpretation rules.

The significant advantage of multimedia ontology used in the BOEMIE project is that it allows defining concepts and relations of a difficult nature (for example sport events with certain type of athletes). This fact forces the definition of new approaches in rule learning, instead of using standard ones. The problem of missing assertions could be solved by presenting a rule learning technique that will automatically define new rules needed for the high-level interpretation. This master thesis represents the idea of creation of a tool that will use the functionality of RacerPro (see Section 3.4 for more details) to construct interpretation rules based on the information derived from image and caption A-BOXes. Image A-BOXes that contain enough assertions for the interpretation of a multimedia object at a high level are considered to be positive examples, which are used as a pattern for negative examples, while caption A-BOXes define a high-level concept needed to be extracted. This information is combined to define a suggestion about a new interpretation rule that will cover a negative example.

The material described in this thesis assumes that a reader has some background knowledge in first order logic.

1.3 Outline

Chapter 2 introduces the basics of the BOEMIE project, its goals and design principles. The chapter explains how the BOEMIE project works and highlights the problem that will be solved in this thesis. Chapter 3 represents the background techniques, like Semantic Web, OWL and Description Logic that are used in the project. Chapter 4 depicts possible techniques that can be used for rule learning and represents the founded solution that takes into account the characteristics of the BOEMIE project. Chapter 5 describes the tool that was written to test a method chosen to increase the performance of high-level interpretation procedure. The report ends in Chapter 6 by drawing conclusions and presenting the future work that can further improve semantic extraction.

Chapter 2. The BOEMIE project

2.1 The goal of the BOEMIE project

The aim of the BOEMIE project is to propose a specific approach to knowledge acquisition, which uses semantics extraction on the basis of the ontology-driven analysis of complex structured documents with multimedia content (video, image, audio and text). The project involves the following activities (see [1] for more details):

- *semantics extraction from still images*, concerning the detection and classification of image areas with domain-pertinent information, using both region-based and holistic approaches and based on the extraction of low-level image descriptors (e.g., scalable colour descriptor);
- *semantics extraction from video sequences*, concerning the detection and classification of spatiotemporal segments through analysis of global and local motion patterns or through model-based analysis of object trajectories;
- *video OCR*, concerning the detection, segmentation and recognition of text found in video sequences;
- *semantics extraction from audio/speech*, concerning the extraction of information about the existence of known audio events, events extracted using name recognition from speech data and non-speech audio events;
- *semantics extraction from text*, concerning the extraction from the textual part of documents information about the existence of names of persons, dates, etc., the relations that may occur between them as well as about the occurrence of terms for various domain-specific events;
- *coordination and fusion of multimedia content analysis*, concerning the combination of information stemming from the specific analysis of each modality, in order to enable semantics extraction and ontology evolution to a degree that cannot be achieved using the individual modalities ;
- *reasoning based multimedia interpretation*, concerning the extraction of high-level knowledge in the domain ontology based on multimedia content analysis.

For the BOEMIE project the purpose of defining a methodology for semantics extraction from multimedia content is described as follows (see [1]):

"Through the proposed methodology we will specify how information from the multimedia semantic model can be used to achieve semantic extraction from various modalities (text, image, video and audio). The outcome of the proposed methodology will be an open architecture, which will communicate with the ontology evolution modules, accessing existing semantic information and providing back newly extracted information . . . The architecture will also . . . specify the interface for the extraction and fusion tools. Thus, it will be completely open to the replacement of the tools with new ones in the future. Additionally, the methodology will cover the evaluation of the whole extraction process based on the separate evaluations of the approaches for single-media analysis."

Although algorithms for automatic extraction of mid-level concepts from visual content were significantly improved, little progress has been achieved in the area of high-level interpretation. The BOEMIE project brings a new paradigm in the process of knowledge acquisition from multimedia content by introducing and implementing the concept of evolving multimedia ontologies, which allows interpreting of high-level concepts in image, video, audio and text and fusing these features for optimal extraction. The significant advantage of ontologies is that concepts and relations are defined in a way to allow a specific formal reasoning to be applied. The ontologies will be continuously populated and enriched using the extracted semantic content. When a significant amount of content is available, in the way that it can lead to the evolution of the semantic model, the ontology enrichment will add one or more mid-level concepts to the modality-specific part of the ontology, together with rules that associate them with the mid-level concepts that subsumes them and with the specific high-level concept. In parallel those ontologies are deployed to enhance the robustness of the multimedia extraction system. To achieve this, the BOEMIE project represents multimedia ontologies and related knowledge with the help of the multimedia semantic model (see Figure 2) which allows expressing knowledge in the form of domain-specific and mid-level concept terms. The information from the multimedia semantic model can be used to achieve extraction from various media (text, image, video and audio) and to combine this information, using data fusion techniques, in order to improve the extraction performance. Moreover, the architecture supports the evolution of the system, by using both supervised and unsupervised machine learning techniques in processing algorithms for each separate media. The content for the background knowledge is collected from official and personal Web resources.

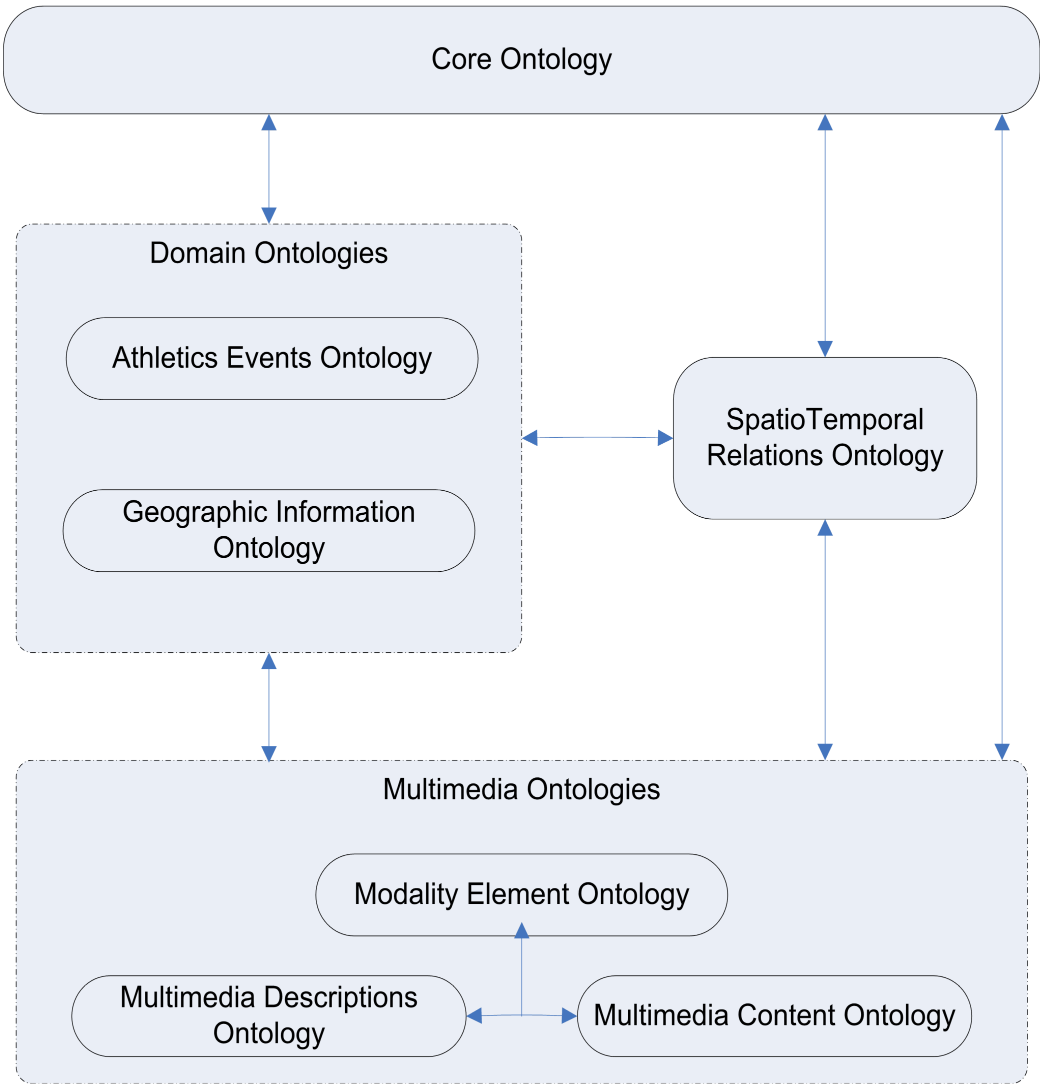


Figure 2. The BOEMIE multimedia semantic model

2.2 Design principles of the BOEMIE project

Nowadays multimedia documents possess a complex structure of different types of information resources. To process them, a large number of specific per-media techniques that enables the interpretation of a domain application and its adaptation to the context are needed. Consequently, the architecture for semantics extraction from multimedia content must be designed to meet the following criteria:

- Independent development of processing and learning techniques per medium;
- Transparent coordination of per-medium analysis modules;
- Enable reasoning-based feedback on analysis results;

To meet these requirements the BOEMIE project is designed to include separate channels for analysing different sources of information as well as tools for improving and expanding the multimedia analysis. The design principles of the project are depicted in the Figure 3.

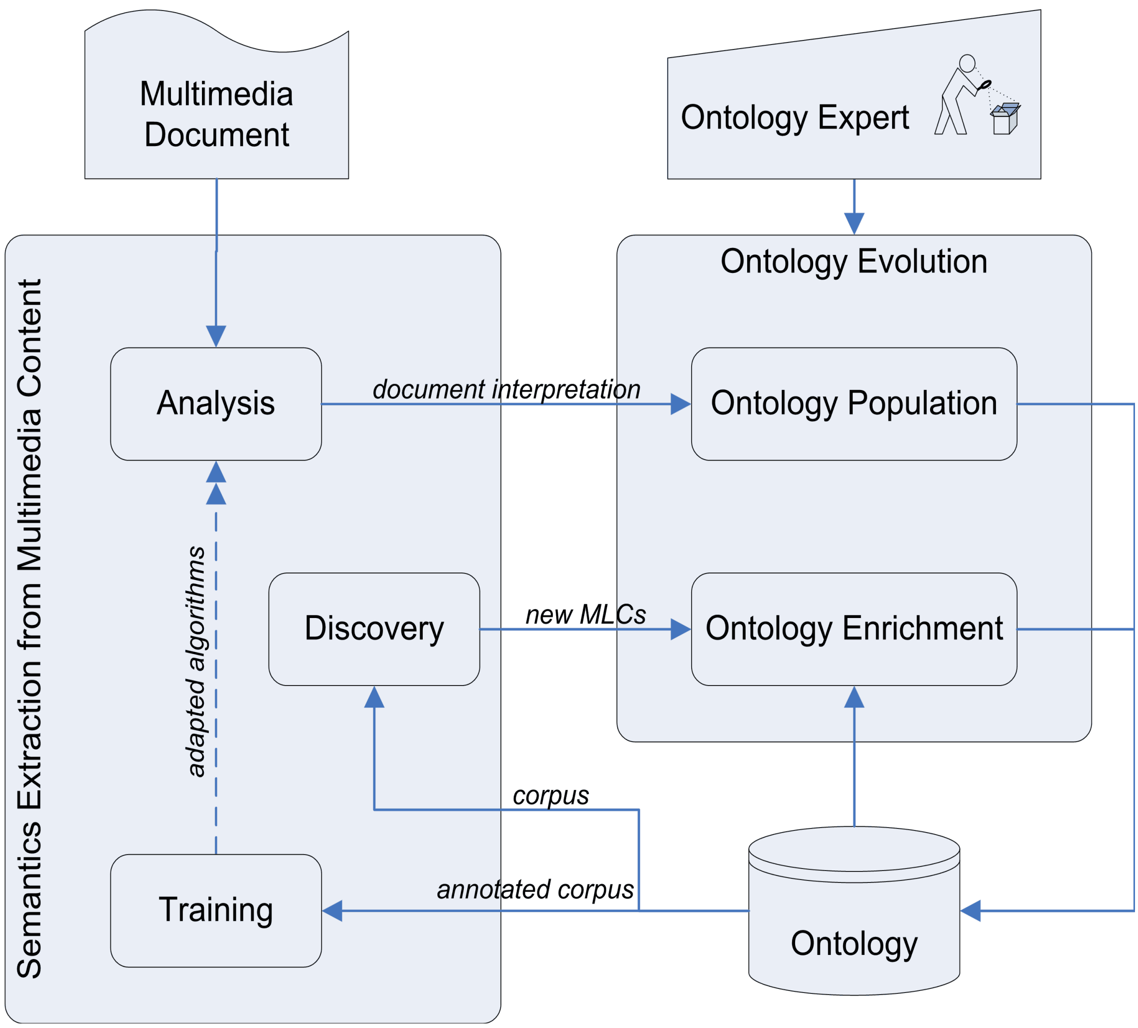


Figure 3. Design principles of the BOEMIE project

The methodology, used in BOEMIE, comprises three distinct modes of operation that are responsible for the ontology population, the adaptability to new content and the enrichment:

1. **Analysis.** This mode is applied each time a new multimedia document becomes available. During this step each document is separated according to the type of its information resource. Figure 4 depicts the methodology for multimedia analysis.

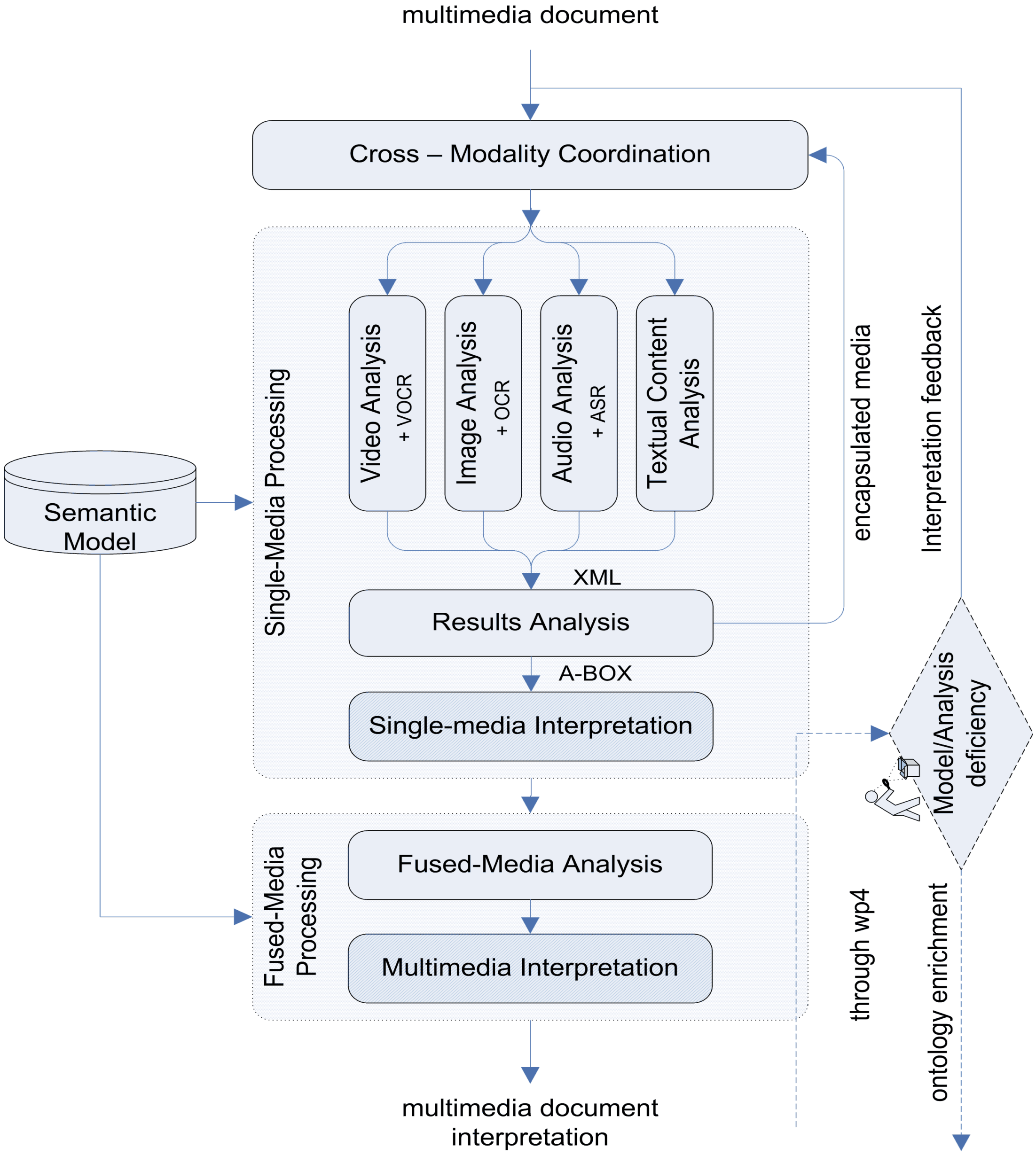


Figure 4. Analysis

Semantics extraction from a multimedia document breaks down to the semantics extraction from each modality-specific sub-document (see [1] for more information):

- **Still Images.** The aim of semantics extraction from still images is, for any input image, to provide information about the existence of image dependent mid-level concepts (such as face, body, etc.), their maps (unique region numbers that identify the image area that is covered by a particular mid-level concept), their low-level descriptors (e.g., scalable colour descriptor, etc) and complementary information about unknown image regions (i.e., MPEG-7 colour, texture and shape descriptors).

- **Video.** The aim of semantics extraction from video documents is to provide information about the existence of mid-level concept instances in video data.
- **Audio.** The aim of semantics extraction from audio/speech is, for any input audio from the considered domain, to provide the information about the existence of known audio events, events extracted using name recognition from speech data and non-speech audio events recognised from speech and non-speech data, their position with respect to other events, their intensity.
- **Text.** The aim of extraction from the textual part of documents is to provide information about the existence of mid-level concepts (e.g. names of persons, names of events, dates, ages, performance, etc.), the relations between them (e.g. that a person with a name N1 has performance P1), as well as about the occurrence of terms for the various sporting events.

The output of the analysis modules is a set of *xml* files containing the list of the extracted elements and element relations together with sufficient properties to describe the related mid-level concepts, the position of the elements in the subdocument, the extracted features used to conduct the analysis, as well as the confidence estimation of the classification. This information is represented in Web Ontology Language (see 3.3 for more details) and is used later on for the high-level multimedia interpretation where more abstract (high-level) knowledge is discovered with the help of reasoning about multimedia ontologies. An example of a *caption owl* file is given in Figure 5.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:gio="http://repository.boemie.org/ontology_repository_tbox/gio.owl#"
  xmlns:mco="http://repository.boemie.org/ontology_repository_tbox/mco.owl#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:aeo="http://repository.boemie.org/ontology_repository_tbox/aeo.owl#"
  xmlns="http://repository.boemie.org/ontology_repository_tbox/aeo.owl#"
  xml:base="http://repository.boemie.org/ontology_repository_tbox/aeo.owl">
  <owl:Ontology rdf:about="http://repository.boemie.org/ontology_repository_tbox/aeo.owl">
  <owl:imports rdf:resource="http://repository.boemie.org/ontology_repository_tbox/aeo-1.owl"/>
  <owl:imports rdf:resource="http://repository.boemie.org/ontology_repository_tbox/mco-1.owl"/>
  <owl:imports rdf:resource="http://repository.boemie.org/ontology_repository_tbox/gio-1.owl"/>
  </owl:Ontology>
  <mco:Caption rdf:about="boemie_html_ncsr-skel_seg6-mco_Caption">
    <mco:depicts rdf:resourcel=”#IND-9657"/>
  </mco:Caption>
  <Person rdf:about="#IND-9657">
    <hasPersonName rdf:resource="l#boemie_text_ncsr-skel_name_2014"/>
  </Person>
  <PersonName rdf:about=" #boemie_text_ncsr-skel_name_2014">
      <hasPersonNameValue rdf:datatype="#string">Yamile Aldama
      </hasPersonNameValue>
```

```
  </PersonName>
  <gio:City rdf:about="#boemie_text_ncsr-skel_city_2015">
  </gio:City>
  <gio:Stadium rdf:about="#boemie_text_ncsr-skel_stadium_name_2016">
  </gio:Stadium>
</rdf:RDF>
```

Figure 5. A typical owl file of a single-media semantics extraction process.

Once the result of one of modalities is transformed in a form of an A-BOX it is forwarded to reasoning, in the context of the given ontology. The interpretation of the results helps to identify high-level concepts and to escape inadequacies (for example missing or redundant instances, according to the current ontology).

It is possible that in some cases multimedia analysis is not able to deliver an appropriate set of concept and role assertions (w.r.t. a domain ontology) to satisfy conditions of interpretation rules (see Figure 14 for the example) and interpretation of a multimedia object at a high level fails. In such situations an abductive approach is required for high-level interpretation. Abduction is usually defined as a form of reasoning from effects to causes and aims at finding explanations (causes) for observations (effects). The input for abduction process is a knowledge base ∑ that consists of a T-BOX T and an A-BOX A (for more information about A-BOX and T-BOX see chapter 3). As it was discussed above, a low-level analysis extracts from multimedia document information that is represented in the form of a set of A-BOXes (Γ). For every object that was recognized in an image a concept and relations, holding among these objects, will be created in Γ. In order to construct a high-level interpretation of the content in Γ, the abduction process will extend the A-BOX with a new instance of a concept and role assertions describing the content of the multimedia document at a higher level. In general, abduction is formalized as

$$\Sigma \cup \Delta \cup \Gamma_1 \models \Gamma_2$$

where ∑ is the ontology, Δ is the sought-after explanation and finally Γ_1 and Γ_2 represent different kind of assertions. Γ_1 contains bona fide assertions, which are believed to be true by default, and Γ_2 contains assertions, which are to be entailed by the abduction process (see [3] for more information). Given some training data, low-level multimedia analysis recognises objects and constrains of spatial or temporal nature among them to generate assertions Γ_1 and Γ_2. Differently from the standard retrieval inference services, the abductive retrieval inference service tries to understand what should be added to the knowledge base in order to positively answer a query. As an example of abduction consider the example described in [4]:

For the image shown in Figure 6 the A-BOX in Figure 7 is provided by low-level image analysis. Furthermore, a sample T-BOX of the athletics domain and a small set of rules are assumed to be provided as background knowledge ∑.

Figure 7. A pole vault event

pole1: Pole
human1: Human
bar1: Bar
(bar1, human1): near

Figure 6. The A-BOX

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

Figure 8. The T-BOX and rules

In order to find a good high-level interpretation of this image, the A-BOX Γ is divided into Γ_1 and Γ_2 following equation, described above. In this example Γ_1 contains {`pole1: Pole, human1: Human, bar1: Bar`} and Γ_2 contains {(`bar1, human1`): `near`}.

Consequently, the abductive retrieval inference service computes the following Boolean query: Q1 := { ( ) | `near(bar1, human1)` }.

Obviously, both rules in ∑ match with the `near` atom in query Q1. Therefore, the abduction framework first generates explanations by 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 a `bar` is made into a `pole` by the variable substitution 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 `human1` and `bar1` from Γ_1 and therefore $S_i(\Delta_1) = 2$. Furthermore, it hypothesizes two new individuals, namely `new_ind1` and `new_ind2`, 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 that the algorithm computes the maxima. In our case, the resulting set of A-BOXes 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.

For more examples of text and image abduction see [1] and [3].
The result of the analysis from different modalities is fused to enrich the informational content of a separate modality (see Section 2.3.3 for more details about fusion). Finally, reasoning on the fused media instances is used to identify fused high-level concept instances as well as missing or redundant fused mid-level concept instances.

2. **Training.** The second mode is used when newly analysed content becomes available. With the help of the received content and learning algorithms, this mode tries to improve the procedure of analysis. Figure 9 depicts the methodology for improving multimedia analysis.
The input is an A-BOX describing the structure of the media item, in terms of mid-level concepts of the background ontology, the output consists of improved versions of the single-media and fused-media analysis modules, expected to lead to improved analysis accuracy in the future. Mid-level concept instances from the previous mode are used as a training set for each separate media analysis module. Given the information of detection success for each instance, analysis modes are checked on accuracy.
Once single-media analysis modules are enhanced, fused-media analysis learning takes place. The learning takes into account new results to adjust parameters for fusion of modalities, such as the confidence levels attributed to each single-media analysis module. Both single and fused media analysis modules are then stored to be used for subsequent analysis requests.

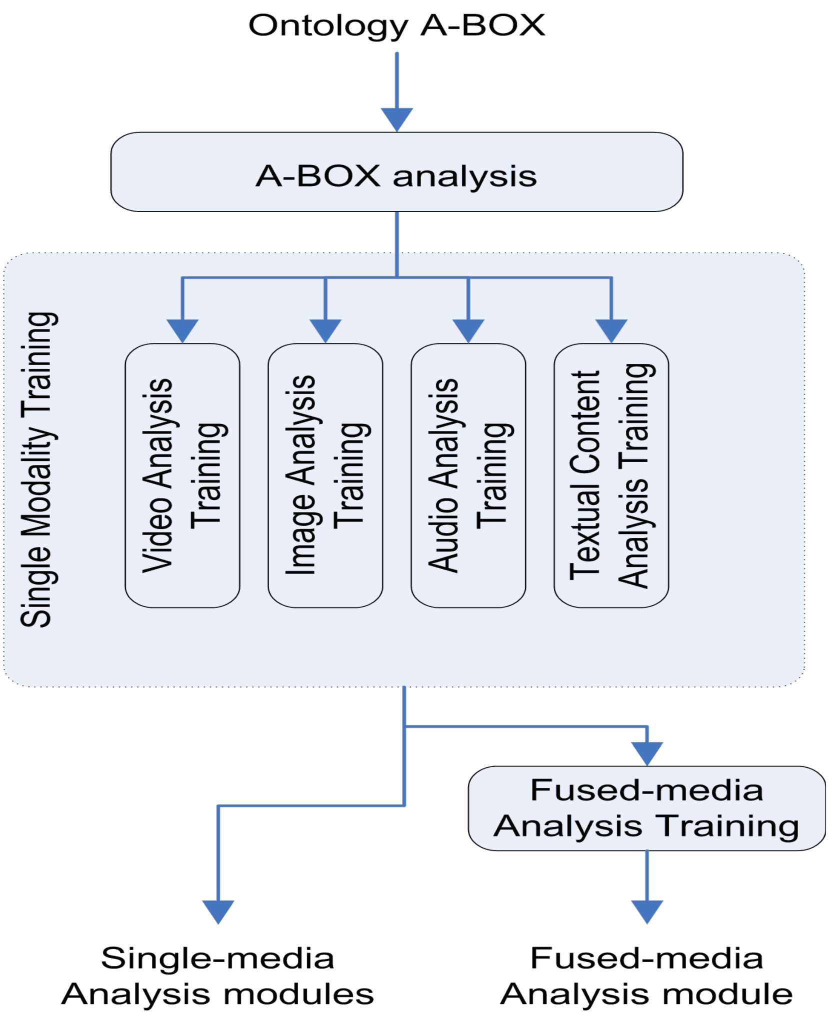


Figure 9. Training

3. **Discovery.** This mode is applied to extend the set of modality-specific concepts, as soon as an appropriate amount of content, needed for the evolution of the semantic model is available. Figure 10 depicts the methodology for expanding multimedia analysis.
   The input is an A-BOX describing the structure of the media item, in terms of mid-level concepts of the background ontology, comprising all the instances of multimedia elements so far analysed by the BOEMIE system. The output is a suggestion for new modality specific mid-level concepts, which can ultimately lead to improved analysis of new multimedia content.

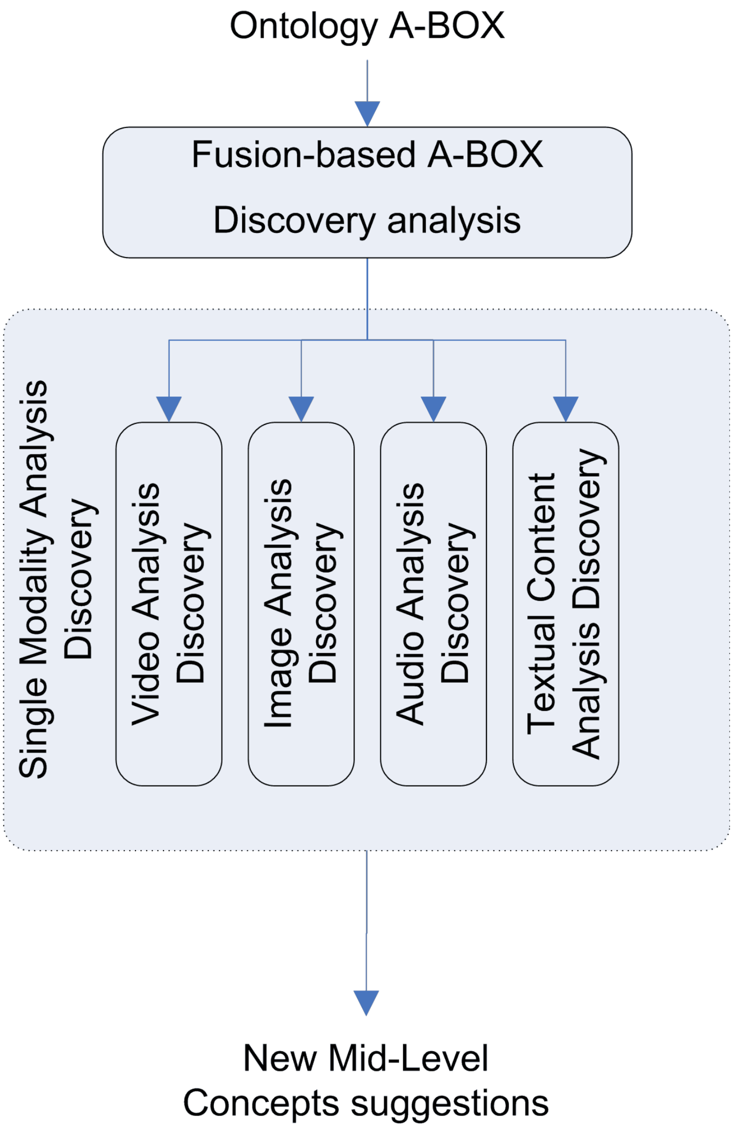


Figure 10. Discovery

2.3 Application of the BOEMIE project

2.3.1 Introduction

The evaluation of the BOEMIE project concerns the collection of information about athletic events. The domain of *athletic events* includes tournaments, meetings, training, etc. The results of the annotation process, i.e., the identified entities and their properties, will be linked to geographical locations and stored in a content server. The application can be considered as an intelligent information service for athletic events. The user will be provided with immediate access to the annotated multimedia content base, through the user-friendly interface of digital maps, which will also provide immediate navigation guidance to the place of interest. An important advantage of this application scenario is that it is associated with a wealth of complementary multimedia content that is evolving over time.

In the BOEMIE project, a collection of images from the athletics domain will serve as the training data for image analysis. In order to gain the necessary training data, certain regions of these images will be manually annotated with mid-level-concepts. Given a new image, image analysis first segments it into regions and then analyses each region using certain low-level features such as colour, shape or texture.

2.3.2 Example of the application scenario

As a concrete example of the application scenario, consider concepts in an athletics domain of *pole-vault and high-jump events*. All events in the athletics domain represent a relational structure that cannot be observed directly in images and thus will not be determined by the low-level analysis modules. In the BOEMIE project events are hypothesised by a higher-level multimedia interpretation where high-level knowledge is discovered with the help of reasoning about multimedia ontology.

Given an athletics domain, the BOEMIE prototype will provide the end user with uniform, user-friendly Web access to the collected information about pole-vault and high-jump events in the three cities covered by the prototype. Furthermore, related material from other locations will be retrievable, due to its semantic association with the material about the three main cities. (For concrete examples see [2]).

As an input the BOEMIE project receives different Web-pages with various media (text, image, video and audio). For each type of media the architecture creates an A-BOX that contains the extracted information. In this master thesis we will concentrate on two types of media: text (a caption of an image) and image. Examples of preinterpreted A-BOXes for image and caption are depicted in the Figures 11 and 12.

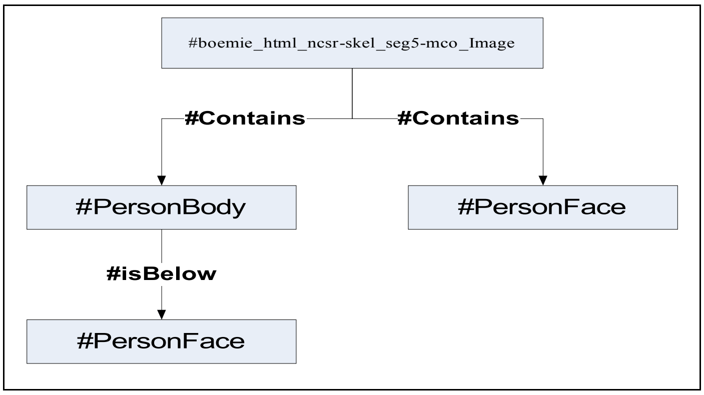


Figure 11. Preinterpreted image A-BOX

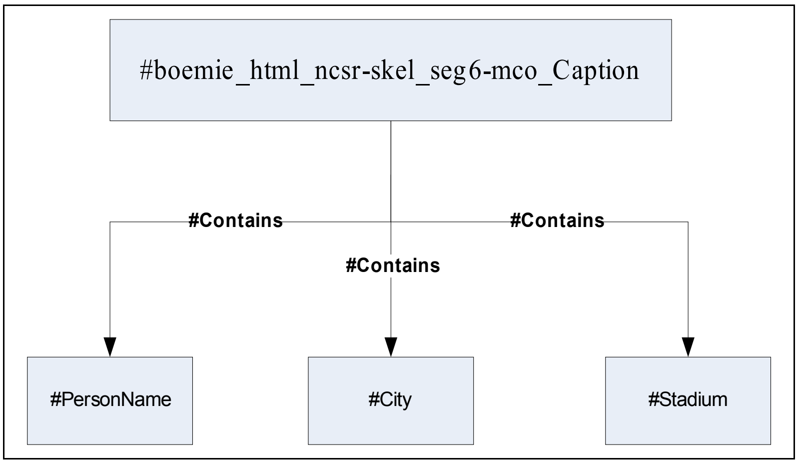


Figure 12. Preinterpreted caption A-BOX

To create a preinterpreted image A-BOX, the extraction algorithm, according to some strategy, applies low-level feature extractors on certain regions of an image and obtains some characteristic values as a result for each region. On the next step, the high-level multimedia interpretation takes place. In a best case high-level concept instances of a multimedia object are extracted through the assertions delivered by multimedia analysis and reasoning deduction using background knowledge. For the interpretation to be successful a preinterpreted A-BOX must contain a structure that satisfies one of the interpretation rules that determine a set of mid-level concepts and relationship between them needed to extract a high-level concept. The example of rules is presented in the Figure 13.

```
(define-rule (?y ?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBelow|)
  (and (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|)
       (?x ?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPart|)
       (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|)
            (?x ?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPart|)
       (?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|))
   :forward-rule-p nil)

(define-rule (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonName|)
          (and (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|)
                                                                                  (?y                ?x                |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPersonName|))
           :forward-rule-p nil)
```

Figure 13. Interpretation rules.

The first rule creates an instance of a high-level concept `Person` that has a body and a face as parts if a preinterpreted A-BOX contains instances of mid-level concepts `PersonBody` and `PersonFace` with a relationship `isBelow`. A second rule extracts an instance of a high-level concept `Person` with a name if a preinterpreted A-BOX contains an instance of a mid-level concept `PersonName`.

The Preinterpreted image and caption A-BOXes depicted in the Figures 11 and 12 contain mid-level instances required in the interpretation rules described above. Such A-BOXes are considered to be positive examples. The result of the high-level multimedia interpretation for these caption and image is depicted in the Figures 14 and 15.

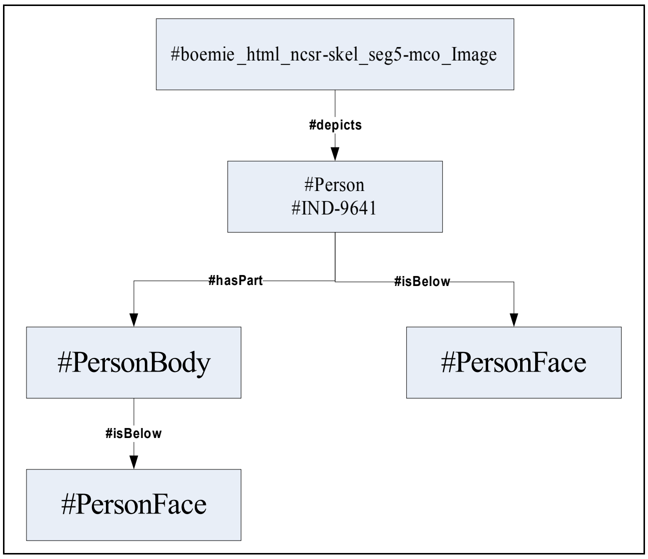


Figure 15. Interpreted Image A-BOX

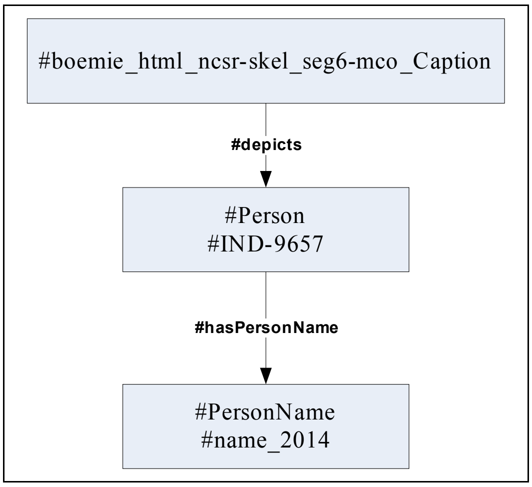


Figure 14. Interpreted Caption A-BOX

The result of the analysis from different modalities is fused to enrich the informational content of the separate modality. In the case of the successful extraction of high-level concept instances the rule of fusion is executed and instances of concepts from a caption A-BOX are added to the image A-BOX (see Figure 17). Following the Figure 16, for the fusion process to take place each modality must depict an instance of a high-level concept. The execution of the rule creates a linkage (same-as) between instances from different modalities.

```
(define-rule (?x |http://repository.boemie.org/ontology_repository_tbox/mco.owl#CaptionedImage|)
   (and (?x ?y |http://repository.boemie.org/ontology_repository_tbox/mco.owl#contains|)
        (?x ?z |http://repository.boemie.org/ontology_repository_tbox/mco.owl#contains|)
        (?y |http://repository.boemie.org/ontology_repository_tbox/mco.owl#Image|)
        (?y ?a |http://repository.boemie.org/ontology_repository_tbox/mco.owl#depicts|)
        (?z |http://repository.boemie.org/ontology_repository_tbox/mco.owl#Caption|)
        (?z ?b |http://repository.boemie.org/ontology_repository_tbox/mco.owl#depicts|)
        (?a ?b same-as))
         :forward-rule-p nil)
```

Figure 16. Fusion rule

A `same-as` linkage depicts that an instance of high-level concept in the caption is the same as in the image and is used later on for the fusion between a captioned image and a text of a web-page.

The result of fusion for an image A-BOX is depicted in the Figure 17.

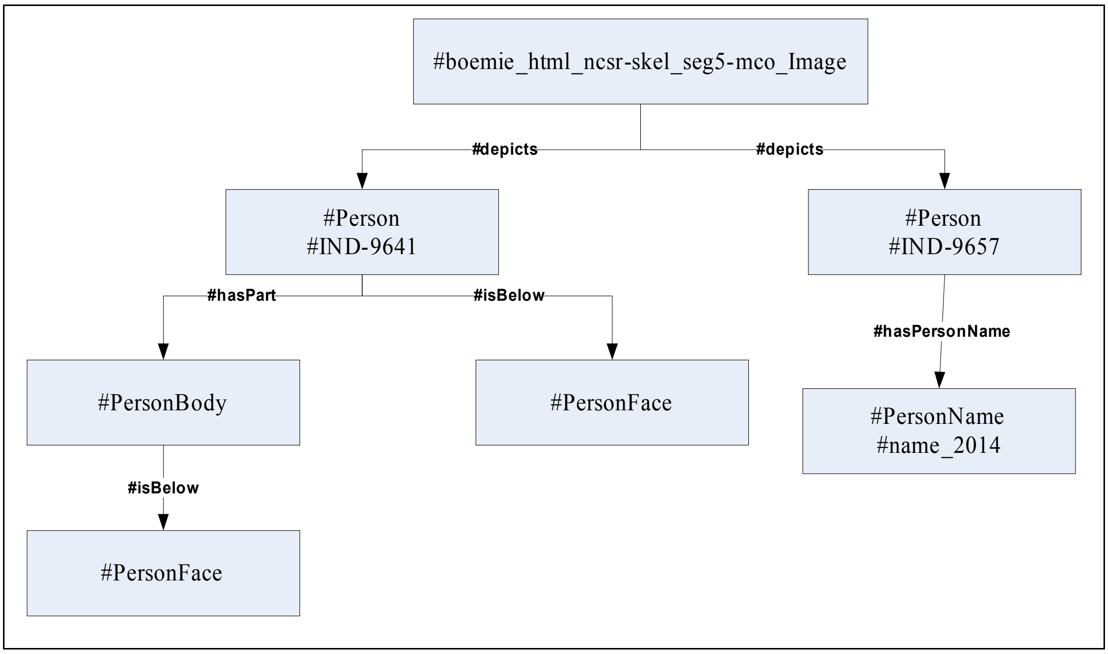


Figure 17. Fused Image

2.4 The problem of the high-level extraction

Unfortunately, often the extraction of high-level concepts from an image A-BOX fails and as a result the execution of fusion process also doesn't enrich the modality. Such A-BOX is considered to be a negative example. A reason for this can be a lack of instances of mid-level concepts, needed for the interpretation rules or failure of abduction, due to the lack of knowledge in the ontology. The example of a falsly interpreted A-BOX is depicted in the Figure 18.

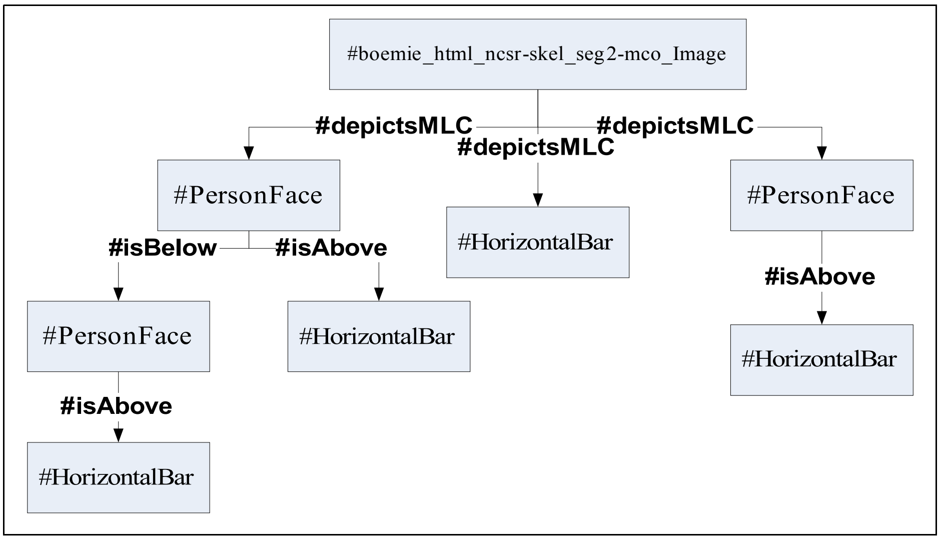


Figure 18. False interpreted Image A-BOX

Although, an A-BOX contains an instance of a mid-level concept `#PersonFace` that could be interpreted as a high-level concept `#Person` it lacks an instance of a mid-level concept `#PersonBody` (see the first rule in the Figure 13). To solve this problem some additional rules can be created to cover this false negatives or the information about mid-level concepts in the image A-BOX can be extended with the help of high-level concepts in the caption, as in the BOEMIE project it is hypothesized that a high-level concept from the caption automatically describes the image.

Chapter 3. Background technologies used in the BOEMIE project

3.1 Semantic Web

The **Semantic Web** provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. (see [1] of web-sites). The Semantic Web can be seen as a mesh of information linked up in such a way as to be easily to process by machines, on a global scale.

The Semantic Web is generally built on syntaxes which use URIs to represent data and usually is represented in triples based structures. A triple can simply be described as three URIs. A language which utilises three URIs in such a way is called RDF (The Resource Description Framework), which is a general-purpose language for representing information in the Web.

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
            xmlns:gio="http://repository.boemie.org/ontology_repository_tbox/gio.owl#"
            xmlns:mco=http://repository.boemie.org/ontology_repository_tbox/mco.owl#>

<PersonName rdf:about="http://#boemie_text_ncsr-skel_name_2014">
    <hasPersonNameValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Yamile Aldama
    </hasPersonNameValue>
</PersonName>
</rdf:RDF>
```

This RDF produces the following triples:

```
<> <http://#boemie_text_ncsr-skel_name_2014> _:x0

_:x0 <http://www.w3.org/2001/XMLSchema#string> Yamile Aldama
```

The main benefit of RDF is that the information maps *directly* and *unambiguously* to a model, a model which is decentralized, and for which there are many generic parsers already available. In the BOEMIE project an extension of RDF OWL (Web Ontology Language) is used to represent information extracted from multimedia documents.

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full (see [1] of web-sites):

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies.

- *OWL DL* supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics.
- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

One of the driving principles of the Semantic Web is inference that enables to derive new data from data that you already know. For the Semantic Web to become expressive enough it is necessary to construct a powerful logical language for making inferences. In the BOEMIE project description logic is used for this purpose.

3.2 Description Logics

Description Logics is a family of knowledge representation formalisms that represent the knowledge of an application domain by first defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain. As the name Description *Logics* indicates, one of the characteristics of these languages is that they are equipped with a formal, logic-based semantics. Another distinguished feature is the emphasis on reasoning as a central service: reasoning allows one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base [5].

3.2.1 The *ALC* description logic

The *ALC* is a family member of description logic languages. Syntax and semantic of *ALC* concept constructors is shown in Table 1. As usual in logics, interpretations are used to assign a meaning to syntactic constructs. Let N_I denote the set of objects, N_C denote the set of atomic concepts, and N_R denotes the set of roles. An interpretation I consists of a non-empty interpretation domain Δ^I and an interpretation function $\cdot^I$, which assigns to each object a from N_I an element of Δ^I, to each concept A from N_C a set A^I from Δ^I, and to each role r from N_R a binary relation r^I from $\Delta^I \times \Delta^I$. Interpretations are extended to concepts as shown in Table 1, and to other elements of a knowledge base in a straightforward way. An interpretation, which satisfies an axiom (set of axioms), is called a model of this axiom (set of axioms) (for more information see [6]).

construct	syntax	semantics
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
existential	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{a \mid \exists b.(a, b) \in r^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\}$
universal	$\forall r.C$	$(\forall r.C)^{\mathcal{I}} = \{a \mid \forall b.(a, b) \in r^{\mathcal{I}} \text{ implies } b \in C^{\mathcal{I}}\}$

Table 1. ALC syntax and semantics.

3.2.2 Knowledge representation in DL

In description logic systems information is stored in a knowledge base, which is a set of axioms. It is divided in two parts T-BOXes and A-BOXes. Examples of A-BOX and T-BOX are depicted in the Figures 20 and 21. The A-BOX contains assertions about objects and relates objects to concepts and roles. The T-BOX contains intensional knowledge in the form of a terminology (hence the term "T-BOX," but "taxonomy" could be used as well) and is built through declarations that describe general properties of concepts. The A-BOX contains extensional knowledge—also called assertional knowledge (hence the term "A_BOX")—knowledge that is specific to the individuals of the domain. Intensional knowledge is usually thought not to change—to be "timeless," in a way—and extensional knowledge is usually thought to be dependent on a single set of circumstances, and therefore subject to occasional or even constant change (see [5] for more details).

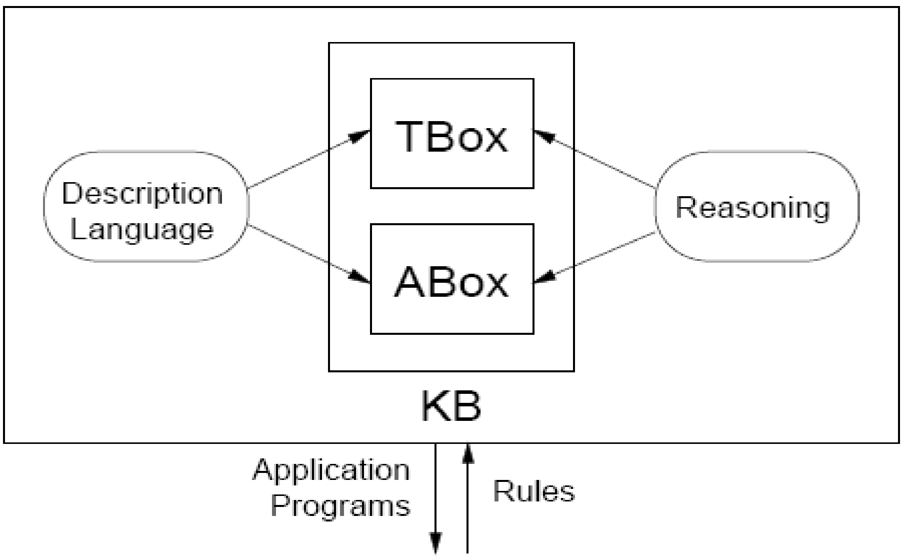


Figure 19. A knowledge representation system based on Description Logics.

The T-BOX can be used to assign names to complex descriptions. The description language has a model-theoretic semantics. Thus, statements in the T-BOX and in the A-BOX can be identified with formulae in first-order logic or, in some cases, a slight extension of it. A knowledge representation system based on Description Logic gives also the opportunity to reason about terminologies and assertions. Typical reasoning tasks for a terminology are to determine whether a description is satisfiable, or whether one description is more general than another one.

Important problems for an A-BOX are to find out whether the assertions in the A-BOX entail that a particular individual is an instance of a given concept description. Satisfiability checks of descriptions and consistency checks of sets of assertions are useful to determine whether a knowledge base is meaningful at all (see [5] for more details).

$$
\begin{array}{rl}
date_1 & : Date \\
(date_1,\ \text{`13 August 2002'}) & : hasValue \\
country_1 & : Country \\
(country_1,\ \text{`Russia'}) & : hasValue \\
hjName_1 & : HighJumpName \\
(hjName_1,\ \text{`high jump'}) & : hasValue \\
perf_2 & : Performance \\
(perf_2,\ \text{`2.26'}) & : hasValue \\
country_2 & : Country \\
(country_2,\ \text{`Finland'}) & : hasValue \\
(hjName_1,\ date_1) & : sportsNameToDate \\
(hjName_1,\ city_1) & : sportsNameToCity \\
(pName_1,\ country_1) & : personNameToCountry \\
(pName_2,\ country_2) & : personNameToCountry \\
city_1 & : City \\
(city_1,\ \text{`Helsinki'}) & : hasValue
\end{array}
$$

Figure 20. An A-BOX example.

$$
\begin{array}{rl}
Person & \sqsubseteq \exists hasName.PersonName \\
 & \sqcap \exists hasNationality.Country \\
Athlete & \sqsubseteq Person \\
HighJumper & \sqsubseteq Athlete \\
PoleVaulter & \sqsubseteq Athlete \\
HighJumpName & \sqsubseteq SportsName \sqcap \neg PoleVaultName \\
PoleVaultName & \sqsubseteq SportsName \\
SportsTrial & \sqsubseteq \exists hasParticipant.Athlete \\
 & \sqcap \exists hasPerformance.Performance \\
 & \sqcap \exists hasRanking.Ranking \\
HighJump & \sqsubseteq SportsTrial \sqcap \forall hasParticipant.HighJumper \\
 & \sqcap \neg PoleVault \\
PoleVault & \sqsubseteq SportsTrial \sqcap \forall hasParticipant.PoleVaulter \\
SportsRound & \sqsubseteq \exists hasName.RoundName \sqcap \exists hasDate.Date \\
 & \sqcap \exists hasPart.SportsTrial \\
HighJumpRound & \sqsubseteq SportsRound \sqcap \forall hasPart.HighJump \\
 & \sqcap \neg PoleVaultRound
\end{array}
$$

Figure 21. A T-BOX example.

3.4 RacerPro

RacerPro stands for **R**enamed **A**-BOX and **C**oncept **E**xpression **R**easoner **P**rofessional. RacerPro can be seen as a system for managing semantic web ontologies based on OWL and as a semantic web information repository with optimized retrieval engine because it can handle large sets of data descriptions.

RacerPro combines description logics reasoning with, for instance, reasoning about spatial (or temporal) relations within the A-BOX query language nRQL. RacerPro is a knowledge representation system that implements a highly optimized tableau calculus for very expressive description logic. It offers reasoning services for multiple T-BOXes and A-BOXes and supports the specification of general terminological axioms. RacerPro allows a T-BOX to contain general concept inclusions, which state the subsumption relation between two concept terms, and multiple definitions or even cyclic definitions of concepts.

Given a T-BOX, various kinds of queries can be answered:

- Concept consistency;
- Concept subsumption;
- Find all inconsistent concept names mentioned in a T-box;
- Determine the parents and children of a concept;

Given an A-BOX, following queries can be answered:

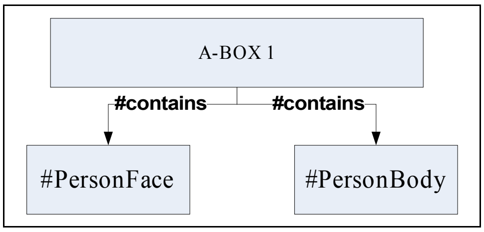
- Check the consistency of an A-box;
- Instance testing;
- Instance retrieval;
- Retrieval of instances that satisfy certain conditions;
- Computation of the direct types of an individual;
- Computation of the fillers of a role with reference to an individual;
- Check if certain concrete domain constraints are entailed by an A-box and a T-box;

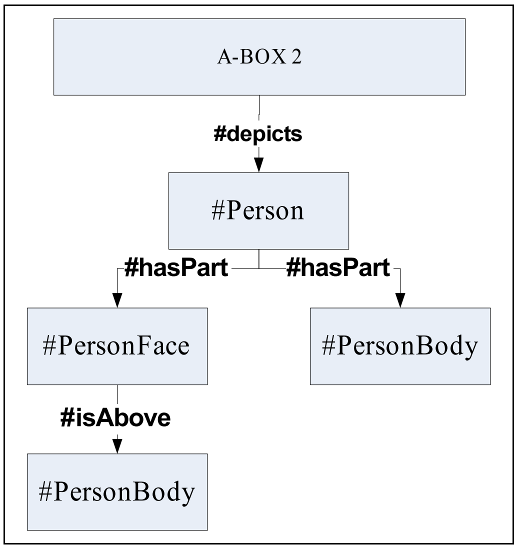
RacerPro can read RDF, RDFS, and OWL files. Information in an RDF file is represented using an A-BOX in such a way that usually triples are represented as related statements: the subject of a triple is represented as an individual, the property as a role, and the object is also represented as an individual.

The triples in RDFS files are processed in a special way. They are represented as T-box axioms. If the property is **rdf:type**, the object must be **rdfs:Class** or **rdfs:Property**. These statements are interpreted as declarations for concept and role names, respectively. Three types of axioms are supported with the following properties: **rdfs:subClassOf**, **rdfs:range**, and **rdfs:domain**.

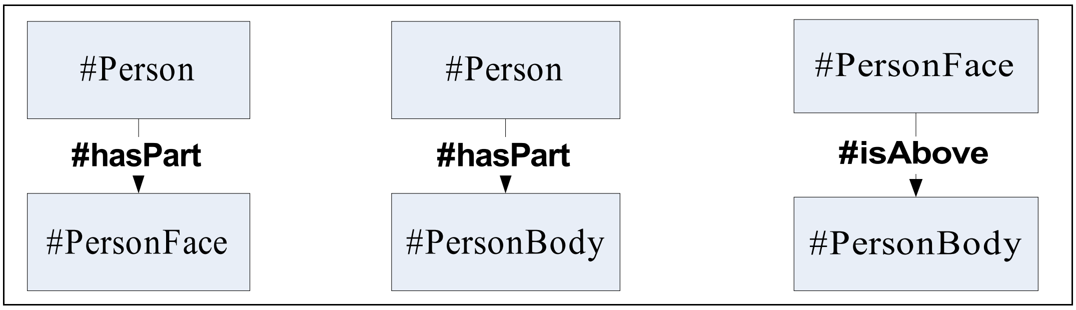
RacerPro can be used in many areas, some of them are: Semantic Web, Electronic Business, Medicine/Bioinformatics, Natural Language Processing and Knowledge-Based Vision, Process Engineering, Knowledge Engineering, Software Engineering.

RacerPro was used in this thesis as a retrieval engine for managing and extracting information from OWL files delivered by the BOEMIE project. One of the functions of RacerPro that was used for the tool is compute-abox-difference function (for the information about the tool see chapter 5). To demonstrate the performance of this function consider the following two A-BOXes:





The result of the execution
(compute-abox-difference abox2 abox1)
are an instance of high-level concept `#Person` and three relations:



(compute-abox-difference)

This function was specially written to solve the problem described above.
Description: Computes the difference between two A-BOXes.

Syntax: (compute-abox-difference (a b &rest args))
Explanation:

- a and b are required arguments (A-BOX names).
- (optimizer-max-plans 30) is a default argument for the argument optimizer-max-plans: 30.

Default optional arguments can be overwritten as follows:
(compute-abox-difference a b :known-correspondances ((i c) (j d)))
This means that "i" in A-BOX “a” is called "c" in A-BOX “b”, and "j" is called "d".
Otherwise, if auto-corresponcance-p = t (true), the function tries to compute the intersection of the individuals of "a" and "b" and assumes they are the same.

The cutoff function determines when to reject a path in the search tree:
(compute-abox-difference a b :cutoff-fn (:hypothesized-assertion < 4))
This means that as soon as a difference needs more than 4 hypothesized assertions, the search path is rejected. Default is at most 5 assertions are hypothesized.

It is also to limit the search space by supplying the ":how-many" argument. In general, if this argument is used, completeness can no longer be guaranteed.
(compute-abox-difference a b :how-many 100)
Terminates after the first 100 differences have been computed, and the best of them is returned. But this should only be used if termination cannot be achieved otherwise.

Chapter 4. Rule learning methods

4.1 Introduction

As stated in the Section 2.4 a possible solution to the problem of missing instances of high-level concepts in an interpreted A-BOX could be a rule learning approach. Based on the knowledge of positive and negative examples of interpretation, the set of rules can be extended to include false negatives. To realize this approach several methods were taken in the consideration.

4.2 Association Rule Learning

Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases, based on the concept of strong rules discovered in databases using different measures of interestingness. Association rules were introduced for discovering regularities between products in large scale transaction data recorded by point-of-sales systems in supermarkets. This method was used to construct following rules: “if a customer purchases three-way calling, then that customer will also purchase call waiting”. To illustrate the concepts, consider a small example from the supermarket domain. The set of items is I = {milk, bread, butter, beer} and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table 2 (This example was taken from [2] of web-pages). An example rule for the supermarket could be {milk, bread} =>{butter}meaning that if milk and bread is bought, customers also buy butter.

transaction ID	milk	bread	butter	beer
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Table 2. Transaction database

The basic process for finding association rules includes:
- Choosing the right set of items;
- Generating rules by deciphering the counts in the table;

In the case of a transaction database, the data used for finding association rules is typically the detailed transaction data delivered by the point of sales. In the case of A-BOXes, the table is constructed from the mid-level concepts, extracted from the modalities.

To generate a rule from the set of all possible rules, various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence. The support *supp*(*X*) of an itemset *X* is defined as the proportion of transactions in the data set which contain the itemset. In the example described above, the itemset {milk, bread} has a support of 2 / 5 = 0.4 since it occurs in 40% of all transactions (2 out of 5 transactions).
The *confidence* of a rule is defined as follows:

$$\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X)$$

For example, the rule {milk, bread} => {butter} has a confidence of 0.2 / 0.4 = 0.5 in the database, which means that for 50% of the transactions containing milk and bread the rule is correct. Confidence can be interpreted as an estimate of the probability *P*(*Y* | *X*), the probability of finding the right-hand-side of the rule in transactions under the condition that these transactions also contain the left-hand-side. The *lift* of a rule is defined as

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(Y) * \mathrm{supp}(X)}$$

or the ratio of the observed confidence to that expected by chance. The rule {milk, bread} => {butter} has a lift of

$$\frac{0.2}{0.4 * 0.4} = 1.25.$$

The *conviction* of a rule is defined as

$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}.$$

and is interpreted as the ratio of the expected frequency that X occurs without Y if they were independent to the observed frequency. The rule {milk, bread} => {butter} has a conviction of

$$\frac{1 - 0.4}{1 - .5} = 1.2,$$

Association rules are required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. To achieve this, association rule generation is a two-step process. First, minimum support is applied to find all frequent itemsets in a database. In a second step, these frequent itemsets and the minimum confidence constraint are used to form rules. While the second step is straight forward, the first step needs more attention. For more information see [8].

After presentation of the preinterpreted A-BOXes in the form of the transaction table, this learning mechanism could be used to construct the rules in the following form: {mid-level concept1, relation, mid-level concept2} => high-level concept.

The main problem, that does not allow applying this method, is the level of complexity of knowledge that is used in the BOEMIE project. The taxonomy used in the BOEMIE project describes a great amount of concepts and complex relations between them that makes it impossible to present it in the form of a table. A part of a taxonomy is depicted in the Figure 24.

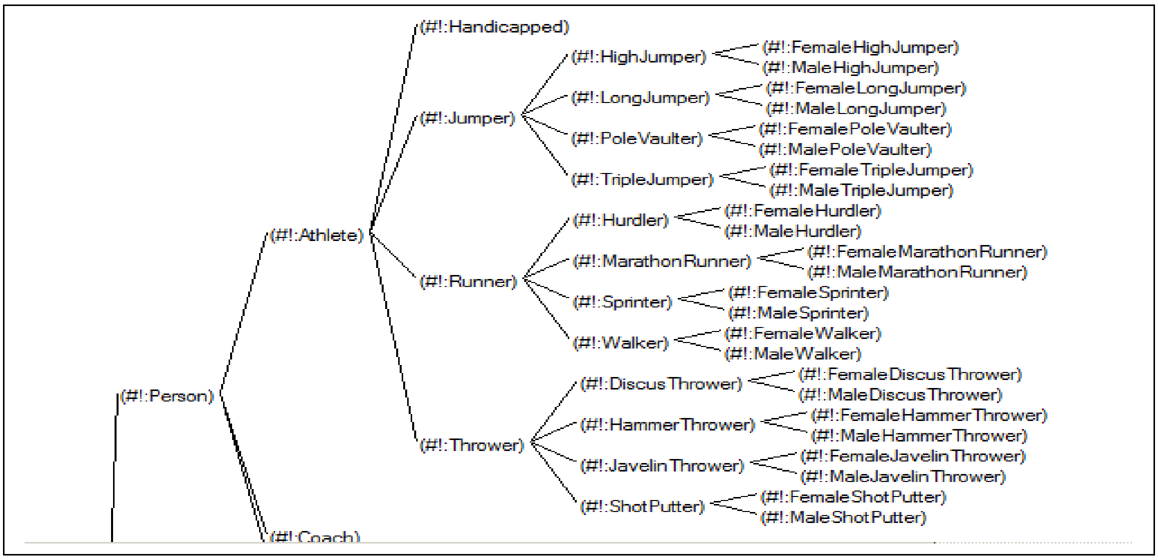


Figure 24. A part of a taxonomy.

4.3 Neural Networks

Neural networks are a class of powerful, general-purpose tools readily applied for prediction, classification and clustering. Neural networks consist of basic units that mimic, in a more simple fashion, behaviour of biological neurons that can be seen in nature. The basic idea is that each neural unit has many inputs that the unit combines and transforms to produce a single output value. These together are called the activation function. The most common activation functions are based in the biological model where the output remains very low until the combined inputs reach a threshold value. When the combined inputs reach the threshold, the unit is activated and the output is high.

The activation function has two parts (see Figure 25). The first part is the combination function that merges all the inputs into a single value. The most common combination function is the weighted sum, where each input is multiplied by its weight and these products are added together. Other possible combination functions include the maximum of the weighted inputs, the minimum and the logical AND or OR of the values. Although there is a lot of flexibility in the choice of combination functions, the standard weighted sum is one of the most used. The second part of the activation function is the transfer function, which transfers the value of the combination function to the output of the unit.

Training a neural network is the process of setting the best weights on the edges connecting all the units in the network. The goal is to use the training set to calculate weights where the output of the network is as close to the desired output as possible for as many of the examples in the training set as possible.

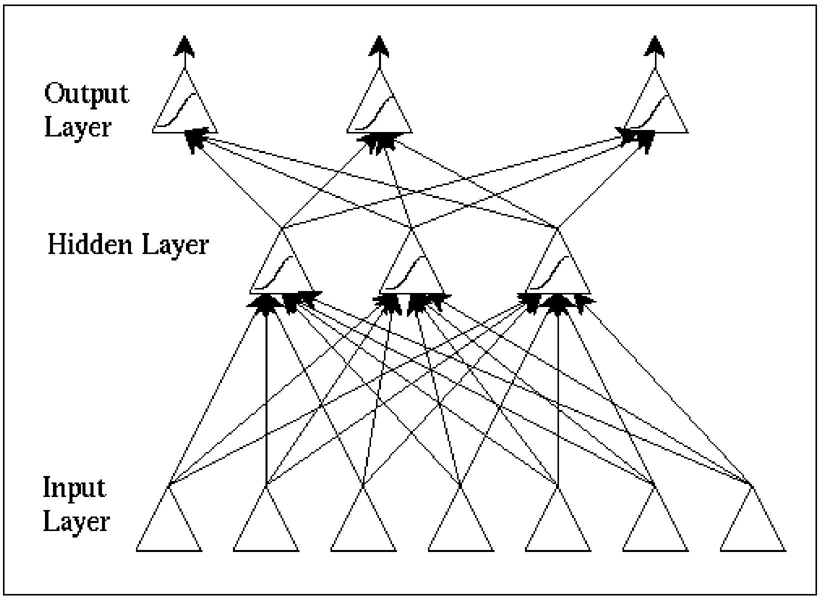


Figure 25. Neural Network

Each interpretation rule has a form of LHS (left-hand side) => RHS (right-hand side) or backward. An example of a rule: if a preinterpreted A-BOX contains a face and a body of a person and between them there is a relation `isBelow` or `isLeft`, then this A-BOX contains an instance of a high-level concept `Person`. The structure of an image A-BOX can be represented in the form of the LHS of a rule. For example the A-BOX depicted in the Figure 11 will be represented as a set of:

- `Image – contains -> PersonBody`
- `PersonFace – isBelow -> PersonBody`
- `Image – contains -> PersonFace`

If one of the triples is similar with a rule, then this rule will be fired and an instance of a high-level concept will be extracted. In many cases though, a preinterpreted A-BOX does not contain a sufficient amount of instances of mid-level concepts to fire a rule. For example the A-BOX in the Figure 18 contains an instance of a mid-level concept `PersonFace`, but misses an instance of a mid-level concept `PersonBody`. Additionally, the caption of this image contains an instance of a high-level concept `Person`. This aggregate information can be used to create a flexible rule that will be fired if a certain amount of information is presented. For this purpose each of the triple and the information extracted from the caption can be represented as one of inputs to the "unit" of a neural network. If a weighted sum of this input exceeds a certain level, then a rule will be fired and an instance of a high-level concept will be extracted. The weight of triples can be calculated in two different ways:

- The concept distance between the triple and a rule. Each concept as well as each relation has its position in the taxonomical tree. The distance can be the number of hops needed to make to move from one concept to another. A triple with the least distance to one of the rules will have a higher weight.
- The number of assertions that must be done. For example the image A-BOX in the Figure 18 has an instance of a mid-level concept `PersonFace` and a relation `isBelow`. Consequently, only one instance of a concept must be hypothesized: `PersonBody`. A triple with the least number of assertions will have a higher weight.

The problem of this approach is the weight of the caption information. Instances of concepts from the caption represent intermodality information and can not be weighted easily. The possible solution can be a heuristic number that can be estimated with the help of statistical approach or observations. This approach will use the idea of fuzzy numbers and can not be realized on the basis of the BOEMIE project.

4.4 Forward chaining rule learning from manually annotated content (rule extraction)

One of the methods that are based on the functionality of RacerPro is forward chaining rule learning from manually annotated context. The idea of this method is to construct new rules based on the evolution of A-BOXes. After the analysis mode of the BOEMIE project interpreted A-BOXes are enriched with new instances of concepts, consequently the difference between preinterpreted and interpreted A-BOXes can be used for the rule extraction.

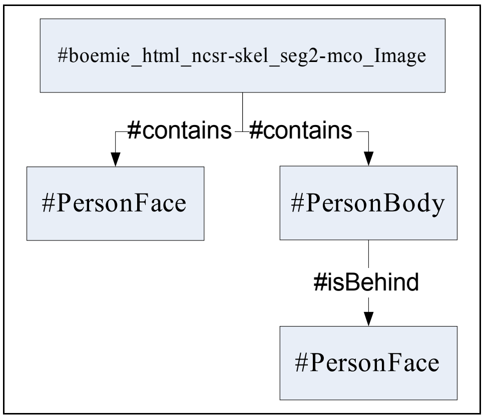


Figure 26. Preinterpreted image A-BOX

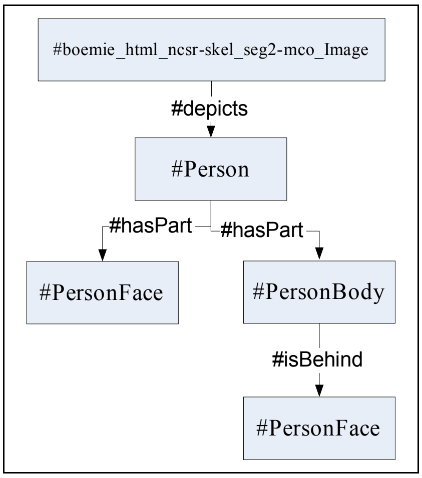


Figure 27. Interpreted image A-BOX

The A-BOX difference between preinterpreted and interpreted A-BOXes (see Figure 26 and 27) is an instance of a high-level concept `#Person` and two relations:

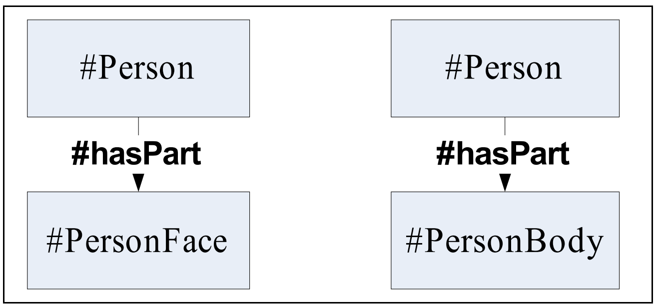


Figure 28. A result of A-BOX difference

From this information the following rule can be extracted:

```
(define-rule (?y ?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBehind|)
  (and (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|)
      (?x ?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPart|)
      (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|)
          (?x ?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPart|)
      (?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|))
   :forward-rule-p nil)
```

Consequently, each positive example of an A-BOX can be used to extract a rule. A set of all extracted rules can be generalized by aggregation in one general rule that will cover all possible cases.

The problem of this method is that a general rule can cover also false positives and thus will bring to false high-level extraction. Hence, more sophisticated methods for generalization are needed.

Another problem that occurs in this method is that rules created in such a meaner are forward chaining and non-horn rules. The abduction process described in the Section 2.3 uses backward chaining rules and rules that have only one atom in head. To create horn rules from non-horn is a complex task that is out of scope of this master thesis.

4.5 Cross-modality interpretation rules

As it was described in the Section 2.4, the information about mid-level concepts in an image A-BOX can be extended with the help of instances of high-level concepts in the caption, as in the BOEMIE project it is hypothesized that an instance of a high-level concept from the caption automatically describes the image.

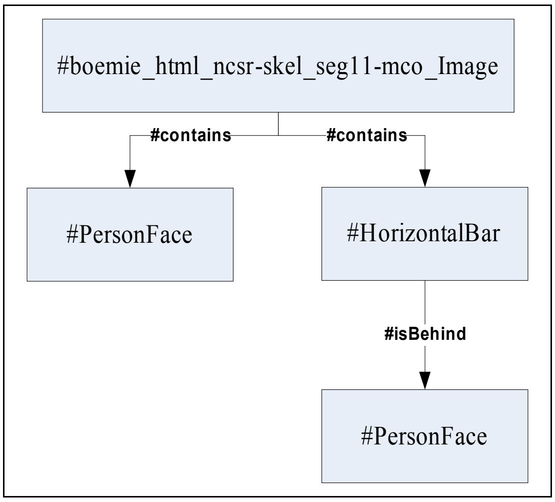


Figure 29. Interpreted image A-BOX

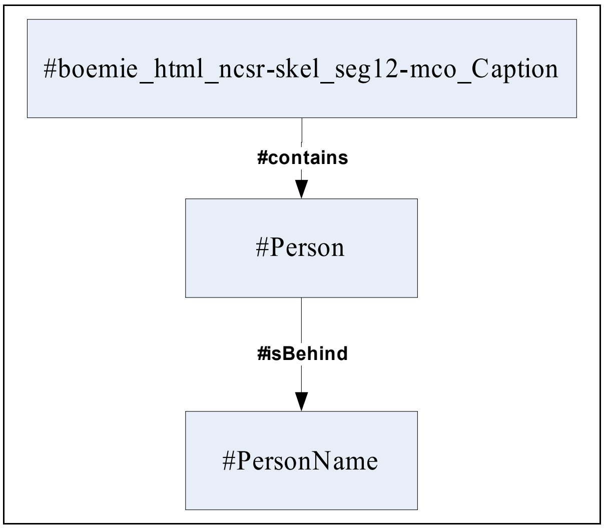


Figure 30. Interpreted caption A-BOX

Information about an instance of a high-level concept in the caption can be used to create a cross-modality interpretation rule that will help to make image interpretation rules more reach and less strict.
The problem of this method is the architecture of the BOEMIE project that was build to separate modalities before analysis. For each modality the BOEMIE project uses a certain set

of rules and information between modalities is exchanged only during fusion. Cross-modality rules will also contain more then one atom in the head and so will be non-horn rules. This idea is partly used in the method described below.

4.6 A-BOX difference rule learning

Because of the complex nature of information used in the BOEMIE project another approach to the solution of the problem was needed.

As described above, a possible solution to the problem of missing instances of high-level concepts in an interpreted image can be additional rules that will cover false negatives. To create these rules the following idea was suggested and implemented. The structure of each preinterpreted image A-BOX was represented in the form of the left-hand side of a rule (see below the example of this representation) that is used in the BOEMIE project. If one of the triples coincides with a left-hand side of a rule, then this rule will be fired and an instance of a high-level concept will be extracted. In this case, given that the process of high-level extraction from preinterpreted caption also successive, the fusion procedure will be successful. On this step captioned images are spitted on positive and negative examples.

To identify the closest to the rule triple, the "degree of applicability" of rules was measured, which is the number of assertions needed to be done to convert a negative example into a positive.

For each triple in all negative examples a concept distance between a triple and a rule is calculated. Each concept as well as each relation has its position in the taxonomical tree. The distance is calculated as a number of hops needed to move from one concept to another. The triple with the least distance is chosen. The tool will inform the user about a concept or a relation from this triple that need to be hypothesized to fit the closest rule. For example, a preinterpreted image has a structure:

- `#Image - contains ->  #UnknownMLC`
- `#Image - contains -> #PersonFace`
- `#Image - contains -> #PersonFace`
- `#UnknownMLC – isAboveRight -> #PersonFace`
- `#PersonFace – isOverlapping -> #PersonFace`

The last triple is the closes to the rule `#PersonBody – isOverlapping -> #PersonFace` => `#Person`. In this case, the tool will inform the user that a mid-level concept `#PersonBody` must be hypothesized.

For the extraction of a structure of a possible new rule an A-BOX difference was used. A difference between each negative example and positive examples gives a structure that is missing in a negative example to fit a rule.

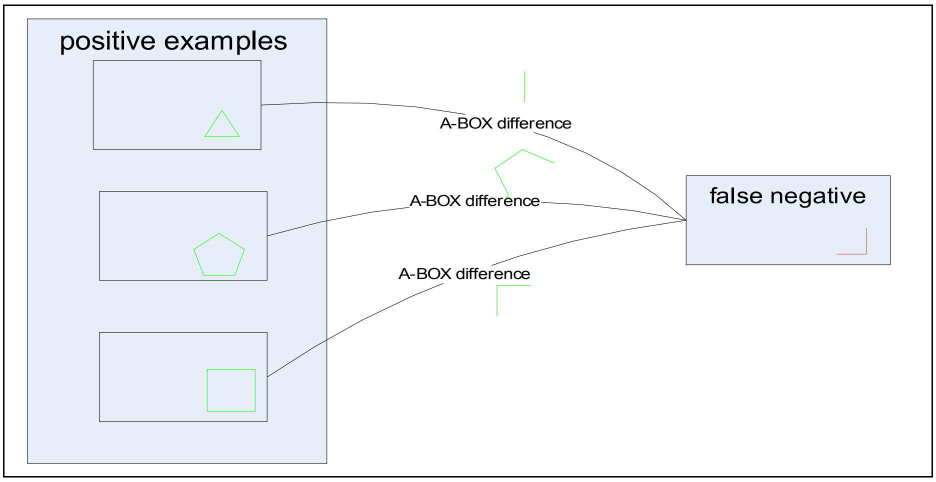


Figure 31. A-BOX difference

As it is depicted in the Figure 31, each a-box difference represents a structure that is missed by a negative example to become a positive. This information together with the hypothesized mid-level concept gives us a left-hand side of a new rule. An instance of high-level concept extracted from a preinterpreted caption is used to represent a right-hand side of a new rule. Consequently, each new negative and positive example give a possibility to create a new rule that will cover a negative example. A set of new rules can be generalized to create one rule.

A pitfall of the extraction of a structure for a new rule from an A-BOX difference, as it becomes obvious from the Figure 31, is the number of possible missed structures that can convert a negative example into a positive. To choose an appropriate one several approaches can be used: the most specific, the most general, the minimal structure. In the tool described below the structure with the minimal taxonomical difference was chosen.

4.7 Learning by Analyzing Differences

This method is based on the step by step learning procedure that uses positive and negative examples. To describe this method we will use an example of learning about arches.

From the first example on the Figure 22.1 the procedure understands the idea of what an arch is and constructs an initial description (Figure 23.1).

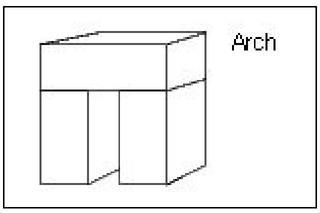


Figure 22.1. Example of an Arch

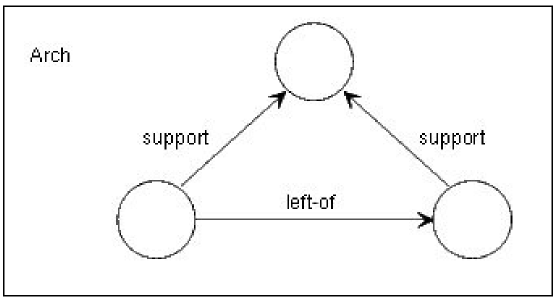


Figure 23.1. Initial description

The following two examples provide the procedure with negative examples, in this case a near miss (Figure 22.2 and 22.3). To respond to near miss example the procedure uses the initial description to find which links are important. The description on the Figure 23.2 is not a description of an arch, but as it is only slightly different from the arch it is interpreted as a near miss.

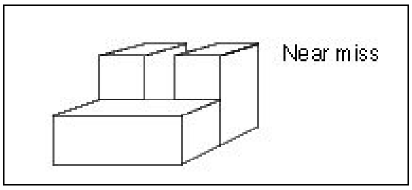


Figure 22.2. Example of Near miss

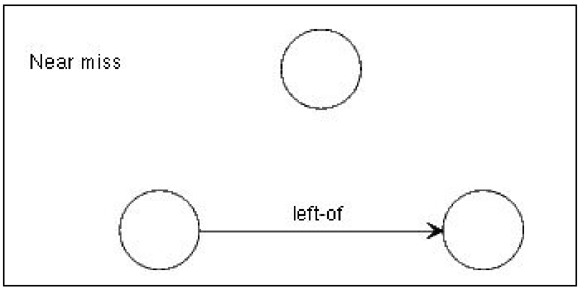


Figure 23.2. Without support links example

The absence of the supporting links and the comparison with the initial description allows the procedure to conclude that arches require support links. Thus, the procedure refines the initial description by replacing the Support links with the emphatic form, Must-support (Figure 23.3).

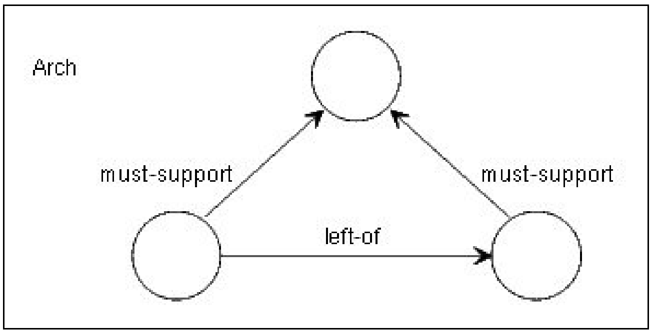


Figure 23.3. A new description of the arch

The next negative example (Figure 22.3) teaches about the importance of Touch links

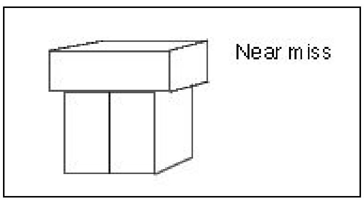


Figure 22.3. Example of Near miss

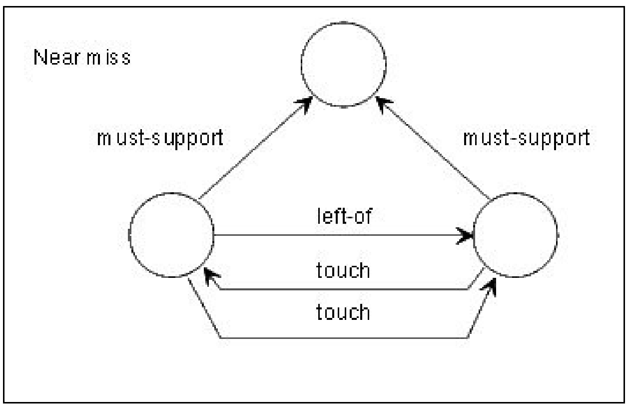


Figure 23.4. With touch links example

From the comparison between the evolving model in Figure 23.3 and the near miss in Figure 23.4 procedure concludes that the new links should be forbidden and converts each Touch link to the negative emphatic link, Must-not-touch (Figure 23.5).

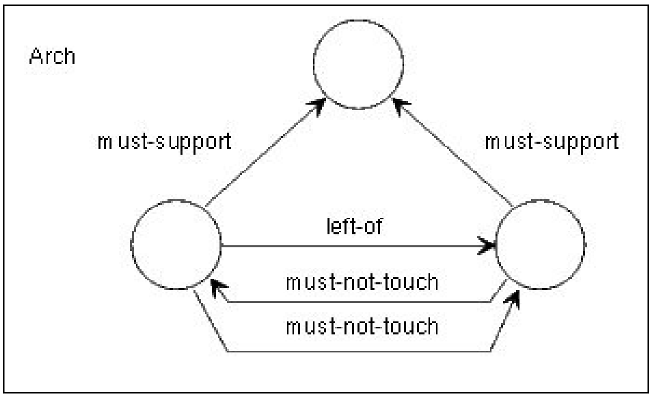


Figure 23.5. A new description of the arch

Consequently, during a learning procedure positive examples relax the model by expanding what can be an arch and negative examples (near misses) restrict the model by limiting what can be an arch.

The procedure described above can be formalized as follows:

To learn using procedure,

- Let the description of the first example, which must be an initial example.
- For all subsequent examples,
  - ➢ If the example is a near miss, use SPECIALIZE.
  - ➢ If the example is an example, use GENERALIZE.

Where SPECIALIZE procedure:

SPECIALIZE procedure makes a model more restrictive,

- Match the evolving model to the example to establish correspondences among parts.
- Determine whether there is a single, most important difference between the evolving model and the near miss.
  - ➢ If there is a single, most important difference,
    - ✓ If the evolving model has a link that is not in the near miss, use the require-link heuristic.
    - ✓ If the near miss has a link that is not in the model, use the forbid-link heuristic.
  - ➢ Otherwise, ignore the example.

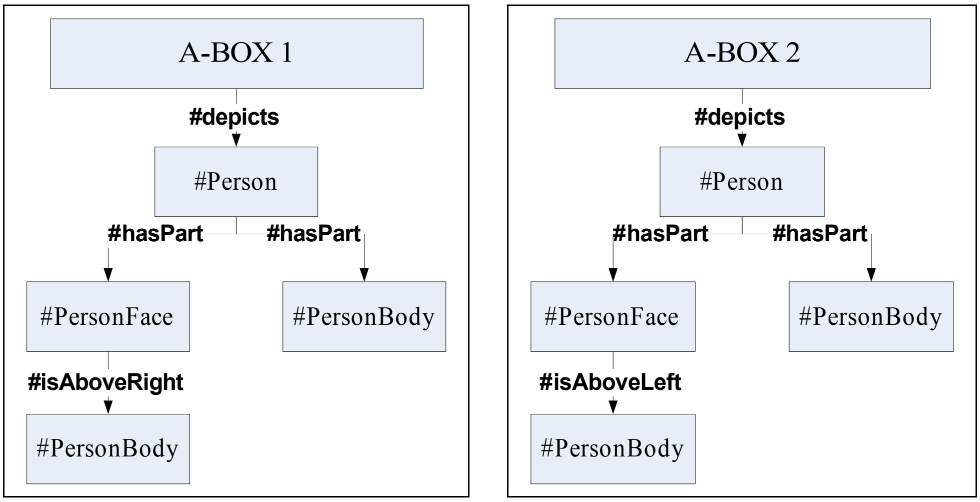
And GENERALIZE procedure:

GENERALIZE procedure makes a model more permissive,

- Match the evolving model to the example to establish correspondences among parts.
- For each difference, determine the difference type:
    - ➢ If a link points to a class in the evolving model different from the class to which the link points in the example,
        - ✓ If the classes are part of a classification tree, use the climb-tree heuristic.
        - ✓ If the classes form an exhaustive set, use drop-link heuristic.
        - ✓ Otherwise, use the enlarge-set heuristic.
    - ➢ If a link is missing in the example, use the drop-link heuristic.
    - ➢ If the difference is that different numbers, or an interval and a number outside the interval, are involved, use the close-interval heuristic.
    - ➢ Otherwise, ignore the difference.

In the scope of this master thesis this method will be used to generalize or specialize the rules delivered by the tool (see Section 5 for more details about the tool). Consider the following example:

Using the positive examples of A-BOXes:



the tool will deliver the following rules:

```
(define-rule (?y ?z | #isAboveRight |)
  (and (?x | #Person |)
       (?x ?y | #hasPart |)
       (?y | #PersonBody |)
       (?x ?z | #hasPart |)
       (?z | #PersonFace |))
  :forward-rule-p nil)
```

```
(define-rule (?y ?z | #isAboveLeft |)
  (and (?x | #Person |)
       (?x ?y | #hasPart |)
       (?y | #PersonBody |)
       (?x ?z | #hasPart |)
       (?z | #PersonFace |))
  :forward-rule-p nil)
```

The GENERALIZE procedure of the method described above will generalize these rules in a rule:

```
(define-rule (?y ?z | #isAbove |)
  (and (?x | #Person |)
       (?x ?y | #hasPart |)
       (?y | #PersonBody |)
       (?x ?z | #hasPart |)
       (?z | #PersonFace |))
   :forward-rule-p nil)
```

However, in the case of a negative example of an A-BOX the SPECIALIZE procedure will specialize the rule to exclude this example.

Chapter 5. The tool and results

5.1 The tool

An additional task to the solution of the problem described above was to write a tool that will also be able to organise and control the information received from the BOEMIE engine.

As an input the BOEMIE receives a set of web-sites that describe some sport event. An output is a set of A-BOXes organized in the following way:

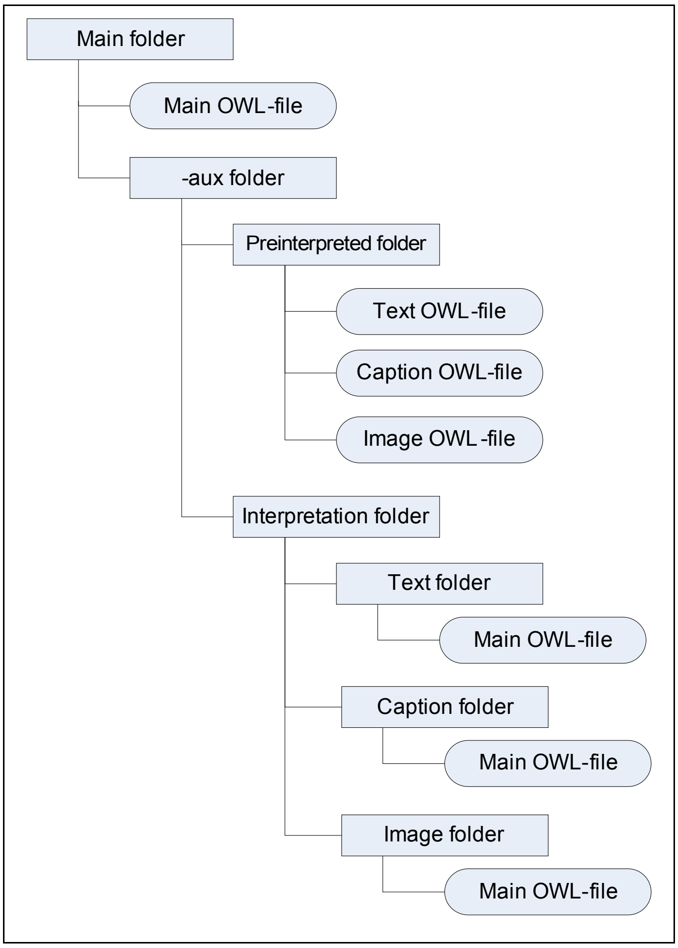


Figure 32. The output of BOEMIE project

Where, main OWL-file is the file that contains aggregated information about the whole web-page and the preintepreted folder contains A-BOXes for all modalities found on the web-page.

Besides the rule learning procedure, the tool delivers the information about the rate of empty captions and images, the percentage of successful fusion procedures and depicts a set of A-BOXes in a user-friendly meaner.

The tool was written in Java. Using the functionality of the RacerPro server the tool reconstructs a tree that contains individuals and relations between them from each preinterpreted, interpreted, and fused A-BOX (see Figure 18 for the example of a tree). This structure allows to analyze the evolution of each modality and to compare separate modalities of different web-sites. From the difference between preinterpreted and interpreted trees the tool receives individuals added by the analysis mode. This information is used to monitor structures that are missed by the interpretation procedure and will be used for rule construction. The difference between fused and interpreted tree helps to identify image A-BOXes that were successfully fused and to control individuals added from another modality.

On the following step each tree brunch is represented in the triple form to identify individuals that an A-BOX misses to fire one of the rules. For this purpose the interpretation rules are also represented in the triple form (see function `initRules` in Appendix A). From all triples of an A-BOX the tool chooses the one with the smallest number of assertions and a smallest taxonomical difference needed to fire the rule. To calculate this, the tool uses the taxonomy stored in T-BOXes and the functionality of the RacerPro. On this step individuals that are needed to be hypothesized are added to the suggestion. The triple representation of an A-BOX was chosen to match the rules that are used in the BOEMIE project. In general, representation can be adapted to the rules of different structures. This allows developing the tool together with the project.

To get the structure of a new rule, the tool calculates the A-BOX difference between each negative and positive example. For each negative example a structure with the minimal taxonomical difference is chosen. Although, the rules that are used now for the interpretation have a simple structure and contain only two atoms connected with one relation, the A-BOX difference will allow the tool to adopt in the future to more sophisticated structures.

To get a right-hand side of a new rule, an instance of a high-level concept is extracted from a corresponding caption.

A missing instance of concept, structure and a high-level concept from a caption form a suggestion about a possible new rule that will be able to cover a negative example. The tool produces the following output:

The name of the main owl-file:
Top:Data/newsId=26575_html/boemie_text_ncsr-skel_top-and-boemie_html_ncsr-skel_top_interpretation1.owl

The name of the caption:
Caption:|
http://repository.boemie.org/ontology_repository_abox/abox.owl#F1AF869F-6865-11DD-89ED-00137238A351-boemie_html_ncsr-skel_seg3-mco_Caption|

The name of the image:
Image:|
http://repository.boemie.org/ontology_repository_abox/abox.owl#F1AF869F-6865-11DD-89ED-00137238A351-boemie_html_ncsr-skel_seg2-mco_Image|

The result of the fusion procedure:
Fusion Success: false

Representation of the preinterpreted image A-BOX in the triple form:
|http://repository.boemie.org/ontology_repository_tbox/mco.owl#Image|
|http://repository.boemie.org/ontology_repository_tbox/mco.owl#contains|
|http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|

The minimal distance to one of the rules is calculated as described above:
10

|*http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Pillar*| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isAboveRight| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|
5

|*http://repository.boemie.org/ontology_repository_tbox/aeo.owl#HorizontalBar*| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBelow| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|
5

The triple with the least distance to one of the interpretation rules:
|*http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace*| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBehind| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|
2

The suggestion about a possible hypothesis:
to hypothesize
An instance of concept that is needed to be hypothesised:
|*http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody*|
To extract an instance of high-level concept:
for the high-level concept |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|

Instances of high-level concepts in caption:
|*http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person*|

A-BOX difference:
|*http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody*| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isAbove| |
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|

From this information the following rule can be constructed:

```
(define-rule (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|)
  (and (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|)
       (?x ?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBehind|)
   :forward-rule-p nil)
```

5.2 Results

During the study 180 sites 538 captioned images were analysed. 469 of them were empty that gives 87%. From 69 images 41 were successful fused, what gives 60% of success. From 28 not fused images 16 contain individuals extracted from background information of an image:

sports event name, country, ranking, person name and thus could be covered with the help of caption interpretation rules. 3 images can be covered with the additional rule:

```
(define-rule (?y ?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBelowRight|)
  (and (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|)
      (?x ?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPart|)
      (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|)
           (?x ?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#hasPart|)
      (?z |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|))
   :forward-rule-p nil)
```

For 9 images the tool gave the suggestion about the rules:

```
(define-rule (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|)
  (and (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|)
      (?x ?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBehind|)
   :forward-rule-p nil)
```

```
(define-rule (?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|)
  (and (?x |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|)
      (?x ?y |http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isBehind|)
   :forward-rule-p nil)
```

Chapter 6. Conclusion and future work

6.1 Conclusion

The task of this thesis was to commit a research of possible approaches for rule learning and to define an appropriate method to increase the performance of high-level interpretation procedure on the level of interpretation rules. Different rule learning methods were analyzed; between them were association rule learning, neural networks, cross-modality interpretation rules and others, but the choice of the appropriate method was bounded by the scope of the BOEMIE project. One of the main problems, that did not allow applying some methods, was the level of complexity of knowledge that is used in the BOEMIE project. The taxonomy used in the BOEMIE project describes a great amount of concepts and complex relations between them what made it impossible to present it in the form needed for these methods. The problem of another method was an approach to the weight of the caption information, as instances of concepts from the caption represented intermodality information and could not be weighted easily. The possible solution could be a heuristic number that could be estimated with the help of statistical approach or observations. This approach would have to use the idea of fuzzy numbers and could not be realized on the basis of the BOEMIE project.

The A-BOX difference rule learning method was able to satisfy all the necessary requirements and was used for the creation of the tool. The tool written for the thesis used the functionality of the RacerPro to construct the suggestion for the interpretation rules based on the information derived from image and caption A-BOXes. In addition, it was able to organise and control the information received from the BOEMIE engine.

The analysis of results delivered by the tool shows that nowadays the problem of the high-level interpretation is not as critical as the problem of the mid-level extraction. In almost 90% of the cases multimedia analysis fails. From the rest 10% of web-pages 60% is successfully fused, which is a remarkable result.

Nevertheless, the tool shows that the performance of the high-level extraction can be increased on the level of interpretation rules by addition of new rules and the usage of information from different modalities. The method and the tool introduced in this thesis can be used for the automatic control of the results deliver by the BOEMIE project and for the extraction of new rules to cover false negatives.

6.2 Future work

The main disadvantage of the research done in this thesis is the lack of the control of newly added rules. New rules must be added to the set of interpretation rules and the analysis of web-pages must be repeated. From one point of view, new rules will increase the percentage of successfully fused images. From another point of view, they can generate a number of false positives, when instances of high-level concepts will be falsly extracted. In this case a method of generalization and specialization described in the Section 4.7 should be used.

The functionality of the RacerPro is constantly extended. This gives the possibility to invent new and to improve existing methods of high-level extraction.

Appendix A

```
public void GetSuccessors(RacerClient racer){
            String ind, mynode, connection = "";
            String[] tuples;
            int count = 0, n;
            DopFunctions df = new DopFunctions();

            try {
                  ind = racer.send("(get-individual-successors
                                    this.getIndividual() +")");
                  if(ind == null)
                  return;

                ind = ind.substring(1, ind.length() - 1);
                tuples = df.mySplit(ind);
                for (String tuple : tuples){
                  mynode = df.cleanFrom(tuple);
                  n = 0;
                      while(tuple.substring(n, n +
                            1).equalsIgnoreCase("(")){
                        count++;
                            n++;
                      }

                      if(count == 1)
                        connection = mynode;

                      if(count > 1){
                        TreeNode node = new TreeNode(mynode, connection);
                        node.addDescription(racer);
                        this.addSubNode(node);
                        node.GetSuccessors(racer);
                  }
                      n = tuple.length() - 1;
                      while(tuple.substring(n, n+1).equalsIgnoreCase(")")){
                        count--;
                      n--;
                    }
                  }
            } catch (Exception e) {
                  e.printStackTrace();
            }
    }


public void addIntCaption(Abox abox, RacerClient racer, String Cind){
      String tuple, str;

      str = Cind.split("boemie")[2];
      tuple = str.substring(0, str.length() - 1);
      for (String file : abox.getCaptionAboxsInt()){
            if(file.contains(tuple)){
                  try {
                        racer.send("(owl-read-file \"" + file + "\")");
                        racer.send("(set-current-abox " + file + ")");
                        str = df.checkIndividual(file, tuple, racer);
```

```
                        if(str != null){
                              intCaption = new TreeNode(str, "main");
                              intCaption.setDescription("Caption");
                          intFile = file;
                          intCaption.GetSuccessors(racer);
                        }
                        else{
                              intCaption = new TreeNode(Cind, "main");
                          intFile = file;
                        }
                        racer.send("(forget-abox " + file + ")");
                  } catch (Exception e) {
                        e.printStackTrace();
                  }
            }
          }
    }


public void initRules(){
            Rule rf = new Rule();         rf.setIndividual("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|")
;           rf.setRelation("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isAdjacent|")
;           rf.setIndividual2("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|")
;           rf.setRHS("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|");
            this.rules.add(rf);

            Rule rf2 = new Rule();        rf2.setIndividual("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|")
;           rf2.setRelation("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isAdjacent|")
;           rf2.setIndividual2("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|")
;           rf2.setRHS("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|");
            this.rules.add(rf2);

            Rule rf3 = new Rule();        rf3.setIndividual("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|")
;           rf3.setRelation("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isNear|");
            rf3.setIndividual2("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|")
;           rf3.setRHS("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|");
            this.rules.add(rf3);

            Rule rf4 = new Rule();        rf4.setIndividual("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|")
;           rf4.setRelation("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isNear|");
            rf4.setIndividual2("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|")
;
      rf4.setRHS("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|");
            this.rules.add(rf4);
```

```
            Rule rf5 = new Rule();        rf5.setIndividual("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|")
;           rf5.setRelation("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isAbove|");
            rf5.setIndividual2("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|")
;           rf5.setRHS("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|");
            this.rules.add(rf5);

            Rule rf6 = new Rule();        rf6.setIndividual("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonFace|")
;           rf6.setRelation("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#isAbove|");
            rf6.setIndividual2("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#PersonBody|")
;           rf6.setRHS("|
http://repository.boemie.org/ontology_repository_tbox/aeo.owl#Person|");
            this.rules.add(rf6);

}
```

References

[1] Methodology for Semantics Extraction from Multimedia Content. BOEMIE project.

[2] BOEMIE. Project description.

[3] “On Ontology Based Abduction for Text Interpretation.” Irma Sofia Espinosa Peraldi, Atila Kaya, Sylvia Melzer, Ralf Möller.

[4] “Multimedia Interpretation as Abduction“. S. Espinosa Peraldi, A. Kaya, S. Melzer, R. Möller, M. Wessel

[5] „The Description logic handbook“. Edited by Franz Baader, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider

[6] „*ALC* Concept Learning with Refinement Operators“ Jens Lehmann, Pascal Hitzler.

[7] “RacerPro User's Guide“

[8] „Data minning techniques for marketing, sales, and customer relationship management“ Michael J.A. Berry, Gordon S. Linoff.

[9] “On the Scalability of Description Logic Instance Retrieval” Ralf Möller_, Volker Haarslev, Michael Wessel.

[10] “A Statistical Approach to Rule Learning” Ulrich Rückert, Stefan Kramer

[11] “Foundations of Refinement Operators for Description Logics” Jens Lehmann, Pascal Hitzler. In Proceedings of the 17th International Conference on Inductive Logic Programming (ILP),2007.

[12] “Learning context-free grammars to extract relations from text“ Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Constantine D. Spyropoulos.

[13] “Can OWL and Logic Programming Live Together Happily Ever After?” Boris Motik, Ian Horrocks, Riccardo Rosati, and Ulrike Sattler

[14] “A formal framework for theory learning using Description Logics.” Jordi Alvarez. In James Cussens and Alan M. Frisch, editors, ILP Work-in-progress reports, volume 35 of CEUR Workshop Proceedings. CEUR-WS.org, 2000.

[15] “A refinement operator for Description Logics.” Liviu Badea and Shan-Hwei Nienhuys-Cheng Lecture Notes in Computer Science, 1866:40–58, 2000.

[16] “Learning and inductive inference”, volume III of “The Handbook of Artificial Intelligence” Thomas G. Dietterich, Bob L. London, Kenneth Clarkson, and Geof Dromey, chapter XIV, pages 323–512. William Kaufmann, 1982.

[17] „A refinement operator based learning algorithm for the ALC Description Logic.” Jens Lehmann and Pascal Hitzler In Proceedings of the 17th International Conference on Inductive Logic Programming (ILP), 2007.

[18] “Artificial intelligence. The modern approach” Stuart Russell, Peter Norvig.

Web-sites

[1] www.w3.org
[2] http://de.wikipedia.org