Available online at www.sciencedirect.com
ScienceDirect
Int. J. Human-Computer Studies 66 (2008) 759–788
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Ontology-based information extraction and integration from heterogeneous data sources

Paul Buitelaar[a], Philipp Cimiano[b,*], Anette Frank[c], Matthias Hartung[c], Stefania Racioppa[a]

[a]*DFKI GmbH—Language Technology Lab, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany*
[b]*Institut AIFB, Universität Karlsruhe (TH), Englerstr. 11, D-76131 Karlsruhe, Germany*
[c]*Seminar für Computerlinguistik, Universität Heidelberg, Im Neuenheimer Feld 325, D-69120 Heidelberg, Germany*

Received 7 August 2007; received in revised form 11 July 2008; accepted 15 July 2008
Communicated by F. Ciravegna
Available online 19 July 2008

Abstract

In this paper we present the design, implementation and evaluation of SOBA, a system for ontology-based information extraction from heterogeneous data resources, including plain text, tables and image captions. SOBA is capable of processing structured information, text and image captions to extract information and integrate it into a coherent knowledge base. To establish coherence, SOBA interlinks the information extracted from different sources and detects duplicate information. The knowledge base produced by SOBA can then be used to query for information contained in the different sources in an integrated and seamless manner. Overall, this allows for advanced retrieval functionality by which questions can be answered precisely. A further distinguishing feature of the SOBA system is that it straightforwardly integrates deep and shallow natural language processing to increase robustness and accuracy. We discuss the implementation and application of the SOBA system within the SmartWeb multimodal dialog system. In addition, we present a thorough evaluation of the different components of the system. However, an end-to-end evaluation of the whole SmartWeb system is out of the scope of this paper and has been presented elsewhere by the SmartWeb consortium.

Keywords: Ontology-based natural language processing; Information extraction; Knowledge integration; Question answering

1. Introduction

One of the biggest current research challenges in human–computer interaction and information retrieval is to provide users intuitive access to the growing amount of information that is ubiquitously available in the form of text, tables, images, videos, etc.

For instance, let us assume a user interested in a specific domain, say football, who would like to get a precise and concise answer to questions/requests such as:

(1)
(a) How many goals did Ronaldo score in the world championship 2006?
(b) Show pictures in which Ronaldo commits a foul.
(c) Show pictures of fouls which led to a penalty kick.
(d) How many goals were scored as a result of a penalty kick by a substitute player?
(e) Show me pictures of saves from the World Cup 2006.

*Corresponding author. Tel.: +49 721 608 7363; fax: +49 721 608 6580.
E-mail address: cimiano@aifb.uni-karlsruhe.de (P. Cimiano).

1071-5819/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2008.07.007

Fig. 1. Pictures of saves of the World Cup 2006.

It is clear that an explicit answer to such questions cannot be found by using Google or open-domain question answering systems. The reason for this is simply that answering such questions requires either counting (e.g. the goals that Ronaldo scored in the world cup 2006), knowledge about cause–effect relations (e.g. knowing which fouls lead to a penalty kick or which penalty kicks lead to a score as a result) as well as knowledge about what pictures actually show (e.g. fouls, saves, etc.). For example, as an answer to the question (e), we would like to see all the pictures shown in Fig. 1.

1.1. Requirements

Given these examples, we can derive the following requirements for a knowledge-based system which is capable of answering such questions:

(1) The knowledge needs to be stored in a structured form, i.e. in a data or knowledge base in order to allow for answering questions that involve counting, aggregation, comparison, computing statistics, etc.
(2) The knowledge base needs to be up-to-date, at least in domains where information is highly dynamic and is affected by changes, updates, etc. which need to be processed quickly. Meeting this requirement is especially important for the domain of football, but also for other domains and applications where the users demand up-to-date information (e.g. financial information systems).
(3) As it is not feasible to populate (and maintain) the knowledge base manually, some automatic mechanism for knowledge extraction is required. Knowledge will need to be extracted from different (complementary) data sources. For example, the goals scored by each player in a certain world cup will probably be best extracted from tables, which are typically complete and offer the information in a regular and compact form amenable to automatic extraction. Other types of information, such as which scoring opportunities were missed, which fouls where committed, etc. are typically not encountered in tabular form, so that they have to be extracted from free text.
(4) Mechanisms for associating linguistic knowledge with domain knowledge are needed as the interpretation of textual data needs to be linked with the appropriate structures in the knowledge base.
(5) The content of images needs to be made explicit to allow for image retrieval. As fully automatic content extraction from images remains still a serious challenge, image captions (if available) provide a valuable resource for determining the content of a picture. Moreover, the information extracted from such captions needs to be integrated with the information extracted from other sources, e.g. free text or tables.

(6) Knowledge extracted from different sources (e.g. texts, tables and image captions) needs to be combined into a coherent knowledge structure, in order to detect and integrate mentions of one and the same entity in different sources.

In this paper we describe the SOBA system,[1] which fulfills the above requirements and automatically creates a knowledge base that can be used for question answering as well as for other applications in the football domain—see for instance Buitelaar et al. (2008). In this paper, we emphasize the following aspects:

- The fact that the system is able to automatically populate and maintain a domain-specific knowledge base, thus fulfilling requirement (1). Actually, the requirement of "up-to-dateness" (requirement (2)) is fulfilled by integrating a web crawler that constantly monitors relevant web sites and triggers an extraction process in the case of updates.
- Further, we show how different information extraction techniques are integrated into our approach. We apply wrapper-like techniques[2] to extract information from tabular data as well as techniques relying on a combination of deep and shallow natural language processing for processing textual data (requirements (3) and (4)). When processing image captions, the images can be annotated with the extracted knowledge, thus fulfilling requirement (5) as a byproduct.
- Additionally, we describe an *information consolidation* component which updates the knowledge base with the output of the different information extraction systems (requirement (6)). The main task of the consolidation component is to (i) identify if an entity extracted from text is already in the knowledge base and (ii) establish appropriate links to the information existing in the knowledge base in case of updates. For example, if an event of type *foul* is extracted from the text, this fact should not only be asserted in the knowledge base, but also be linked to the particular match and the player who committed the foul. Otherwise the knowledge base will consist only of small "islands" of information that are not interlinked and therefore less useful. In connection to this, we also present a discourse analysis component which is able to infer relations between events. These relations can be used to query for causal connections.
- Finally, we present a thorough and systematic evaluation of these different components.

1.2. Ontologies

A crucial question that should be addressed about the general design of the SOBA system is in which way it is in fact *ontology-based*. Typically, an ontology is defined as a *formal specification of a conceptualization* (Gruber, 1993). However, for the purposes of this paper, we simply assume that an ontology is a schema agreed upon by a group of interest in order to formalize the data relevant for the domain in question. Along these lines, the ontology specifies *what* is relevant for the domain in question as well as *how* it is expressed according to the vocabulary defined in the schema. Thus, while we build on RDF(S) (Brickley and Guha, 2004) and F-Logic (Kifer et al., 1995)[3] as languages to describe our domain ontology, for the purposes of this paper a database schema could also be seen as an ontology. The main difference between an ontology and a database schema is that the latter essentially constrains the possible states of the database, while the former has typically a model-theoretic semantics and thus allows to infer new knowledge (in a deductive fashion). In the approach presented in this paper, we do not rely on expressive reasoning other than rule-based reasoning, which could arguably also be performed with deductive databases. In general, our approach can integrate as much reasoning capabilities as the underlying inference engine is capable of. However, expressive reasoning is not a necessary requirement. Thus, the choice of F-Logic or RDF(S) as formalism is an engineering choice rather than a principled one. While our data are described in RDF(S) or F-Logic, they could be stored persistently in any relational database. The concrete formalism used is not essential for our approach. Nevertheless the choice of resorting to ontology languages such as RDF(S), F-Logic or OWL may still be a principled one. While we are not concerned with this issue in this paper, we assume that more complex inferences will be needed for applications, such that the choice of more expressive formalisms (compared to a plain RDBMS) seems reasonable. Moreover, ontologies support better the kind of dynamic domains that we have in mind. In fact, while database schemas are in general regarded as static, ontology schemas are typically assumed to be highly dynamic and evolving objects (see Noy and Klein, 2004).

1.3. Contributions

The contributions of our work described here may be of relevance to several communities, i.e. the information extraction community as we show how a variety of information extraction techniques on different kinds of data can be integrated into an end-to-end system which constantly monitors the web and automatically maintains a coherent knowledge base; the

[1]SOBA originally was an acronym for "SmartWeb Ontology-based Annotation". However, SOBA now covers ontology-based information extraction beyond semantic annotation proper.

[2]Wrappers are simple procedures, e.g. based on regular expressions, for extracting and structuring information from semi-structured data such as HTML tables.

[3]As inference engine for F-Logic we use OntoBroker (see Decker et al., 1999).

question answering community as we show how a domain-specific knowledge base can be created and maintained automatically such that it allows to answer questions requiring aggregation, counting and some level of inference; the knowledge acquisition community as we show which problems need to be dealt with when populating a knowledge base with facts extracted automatically from different sources and by presenting an elegant and domain-independent solution for the incremental integration of these extracted facts (albeit a slightly brittle one as it depends on exact string matches; we describe the limitations and future extensions in Section 8).

Finally, our work may be of interest also to the natural language processing community in general as we address the combination of shallow, finite-state-based linguistic analysis with deep linguistic parsing for a real-world application domain such as football match reports. Overall, we regard as our biggest contribution the fact that we show how techniques from different disciplines (deep parsing, information extraction (IE), ontologies) can be put together in a larger system to provide an added value for concrete applications, in our case question answering.

1.4. Structure of the paper

The structure of the paper is as follows. In Section 2 we provide the background of the SOBA system, which is part of the SmartWeb multimodal dialog system. In addition, we also provide more in-depth motivation for the development and design decisions of the SOBA system. In Section 3 we give an overview of the SOBA system and the data sources used. Section 4 discusses the consolidation component of SOBA and describes how the different components of the system are conceptually related. In Section 5 we discuss the linguistic analysis components used. In particular, we present the application of a new architecture for linguistic processing that integrates finite-state-based technologies for shallow text analysis with a deep linguistic parser. Section 6 presents an evaluation of different aspects of the system, while Section 7 addresses the application of SOBA in the context of the SmartWeb system. Finally, Section 8 discusses related work while in Section 9 we draw some conclusions of our research and provide an outlook on future work.

2. Background and motivation

SmartWeb is a multimodal dialogue system which aims at providing intuitive access to the semantic web. The system has been developed with a focus on the football domain in order to be demonstrated during the world cup 2006 in Germany. Users are able to access the system from different devices: a PDA, while riding a motor-bike and from inside a car (see also Reithinger et al., 2007). For the purpose of question answering, SmartWeb implements two different and complementary approaches.

One approach is built on an open-domain question answering system (see Neumann and Sacaleanu, 2005 for a description of this system), where "open-domain" implies the capacity to handle arbitrary questions about any domain of interest. However, open-domain QA systems typically rely on answers given explicitly in underlying text collections such as the web and do not rely on domain-specific background knowledge in the form of ontologies or knowledge bases (see Strzalkowski and Harabagiu, 2006 for a recent description of the state-of-the-art in QA). As a result, it is very difficult for such systems to answer complex questions that require counting, aggregation or inference. Further, most open-domain QA systems are not able to consider non-textual sources for answering questions although semi-structured data such as tables are in fact very important as a source of information. Tables typically contain a wealth of accurate and in many cases complete knowledge that can be easily extracted using wrapper-like techniques. Further, considering images or videos enhances the user experience for question answering systems, as users typically do not only want to see a textual answer, but also some audio-visual content which can provide additional information in a very convenient and efficient way.

The second approach is complementary to the first one in that it builds on a structured knowledge base for answering domain-specific questions. The SOBA system discussed here is used in the SmartWeb multimodal dialogue system to build up and maintain a knowledge base about football, in particular about all world cup tournaments since 1930. By processing relevant web pages, SOBA builds up a structured knowledge base that can be used for answering questions requiring counting, comparisons, aggregation, etc. In addition, as a byproduct of extracting information from text, it is also able to keep references between images and extracted content, such that pictures can also be delivered as answers. Capturing the content of an image or video automatically by means of image analysis techniques still remains a serious challenge, so that approximating the content by analyzing captions is a promising first solution.

According to our observations, different types of resources express different types of information. In the football domain, tables typically express very basic information about matches, the teams, their players, the match result, the number of scores as well as red and yellow cards assigned. Information that is typically not contained in tables comprises, for instance, the number of fouls, the number of goals produced as a result of a penalty kick, the number of corners as well as the causal relationship between different events. Image captions typically describe the content of a scene and can serve to capture the meaning of a picture for retrieval purposes. Thus, following our requirement (3), we need different IE systems in order to process tabular data but also free text. Further, as we do not want to produce "islands" of information which

are not connected, we need a consolidation component which introduces connections between the different bits and pieces of the information extracted. These connections can be established either across types (e.g. between information extracted from an image caption and the information extracted from a table) or between events extracted from one text, even from one and the same paragraph. For this purpose, the ontology can provide valuable background knowledge as it can tell us which types of events can have causal relationships (e.g. a foul can lead to a red card, a penalty kick or corner kick can lead to a goal, a cross can lead to a shot, etc.). Further, the ontology can specify which properties are functional, thus helping to decide whether two events can be merged or not. For example, the property `atMinute` of a goal is functional, such that we have to conclude that if two goals have different minutes, they cannot represent one and the same goal.

In order to keep interfaces clean, we assume that IE systems will not be in charge of consolidating information, but rather to produce target knowledge structures as output which are compliant with the ontology in question. The consolidation component then takes the output of the IE systems and is responsible for updating the knowledge base such that redundancies are eliminated and the knowledge is interconnected. In this way, the consolidation component is responsible for updating the knowledge base. In our use case, we have defined a set of operations which need to be performed during this task: (i) detect duplicates, (ii) merge non-functional properties of these duplicates and (iii) connect the entity to other entities within the knowledge base. Operations (i) and (iii) can be carried out by querying the knowledge base for entities with certain key attributes. In essence, this amounts to specifying appropriate queries to the knowledge base. Thus, it seems that the types of operations are universal and independent of the specific ontology used. Hence, our aim was to strictly separate the implementation of these generic procedures from the particularities of the domain in question. In fact, we have created a declarative formalism which allows to specify the operations to be carried out for each ontological type when updating the knowledge base. These operations can thus be specified separately from the code executing them and thus be defined by a domain expert without any knowledge of programming languages. We think this is a crucial step towards simplifying the customization of such systems as described in this paper.

3. System overview and data sources

The ontology-based information extraction and integration system SOBA consists of a web crawler, linguistic annotation components and a component for the transformation of linguistic annotations into a knowledge base according to the *SWIntO* ontology. SWIntO (SmartWeb Integrated Ontology) is the core knowledge resource used by SOBA. SWIntO has been developed in the context of the SmartWeb project and integrates a number of domain and task ontologies for representing knowledge about football, navigation, discourse and multimedia. It includes the SUMO top ontology and the foundational ontology DOLCE and consists of 2384 concepts in total with 631 football concepts (for complete details on the design and use of SWIntO see Oberle et al., 2007). The web crawler acts as a monitor on relevant web domains (i.e. the FIFA web site[4]), automatically downloads relevant web documents and sends these to a linguistic annotation web service. Linguistic annotation and information extraction is based on the Heart-of-Gold (HoG) architecture (Callmeier et al., 2004), which provides a uniform and flexible infrastructure for building multilingual applications with XML-based natural language processing components. The linguistically annotated documents are further processed by the semantic transformation component, which generates a knowledge base of football-related entities (players, teams, etc.) and events (matches, goals, etc.) by mapping annotated entities and events to instances of ontology classes and their properties (Fig. 2).

3.1. Crawler and data set

The crawler process enables the automatic creation of the data set that we use in our experiments discussed below. The data set consists of tables, texts, and images on World Cup football matches (1930–2006) that are derived from the original HTML documents. For each match, we extract from the FIFA web site: (i) a table with players, goals, referees, etc. (ii) one or more textual match reports that can be associated with the particular match described by the table, and (iii) images with their corresponding captions related to the textual match report.

To align these heterogeneous data sources, we link all files that are related to a particular match uniquely to a central `crossref` file that acts as a metafile for that match. Each crossref file in turn corresponds to exactly one tabular match report as derived from the FIFA web site.

An important step in the crawling process is to link only those textual match reports that are in fact reports about a particular match and not a more general news report on the World Cup. We therefore implemented a simple classification procedure that decides whether a FIFA news item is indeed a match report and which match it refers to on the basis of the mentioning of (i) a limited number of teams—ideally two—and (ii) mentioning of players belonging to these teams. Thus, if

[4]http://fifaworldcup.yahoo.com; last access on 06.07.2008.

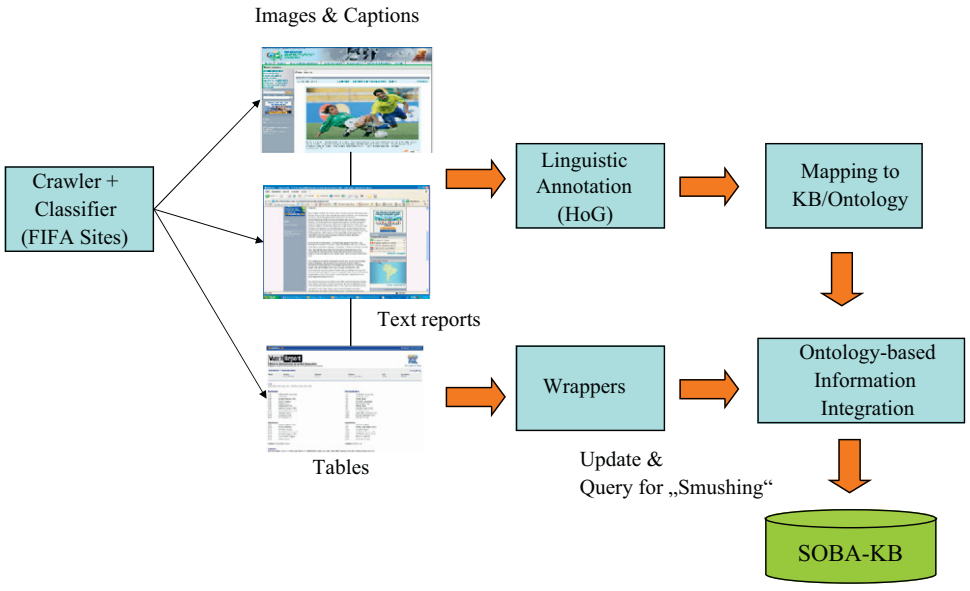

Fig. 2. SOBA overview.

only the teams of Bolivia and Ghana together with a number of players are mentioned in the news report, we assume that this is indeed a match report on the Bolivia–Ghana match.

The resulting data set consists of a tightly linked collection of semi-structured data (tables for each match), textual data (one or more match reports for each match) and multimedia data (images with captions that are directly linked to a textual match report and indirectly to a tabular match report). A limited version of this data set has been made publicly available.[5]

3.2. Data processing

The remainder of the paper will be concerned with a description of how this data set has been used for extracting relevant information on World Cup matches and turning this into a knowledge base comprising facts on teams, team players and the events in which they were engaged.

Information extraction in SOBA is based on a combination of wrapping techniques for the analysis of semi-structured data and shallow and deep linguistic analysis for the extraction of information from textual match reports and image captions. Details of the textual analysis are discussed in Section 5.

The wrapping technique we use for the analysis of tables is straightforward and based on a standard regular expression approach. These regular expressions were created by hand and iteratively refined until coverage and accuracy was close to 100%. Thus, the regular expressions are completely tailored to the purpose at hand and lack any generality. The main problem we encountered when defining these regular expressions is the wide variety in the use of abbreviations in the description of match events and results across different editions of the World Cup, e.g. 2006 vs. 2002.

The results of the wrapping and textual analysis are encoded in proprietary XML-formats that have been designed based on the SWIntO ontology, i.e. XML-tags used in this format correspond to labels of SWIntO ontology classes and properties. The XML-format for the wrapping results has been designed specifically for SOBA, whereas the XML-format for the textual analysis results is based on the SProUT output format as used by the HoG processing platform.

The resulting XML-encoded files are further processed by the knowledge consolidation component of SOBA, which transforms the extracted entities and events encoded in XML into ontology instances encoded in F-Logic or RDF. For this purpose, a mapping language has been designed and implemented in the form of an XML-based rule language that (i) maps XML structures onto ontology-conform frame-based structures, (ii) checks the existing knowledge base for duplicate facts, and (iii) integrates newly extracted information with existing facts in the knowledge base. Details of this process are discussed in Section 4.

[5]The SmartWeb data set (http://www2.dfki.de/sw-lt/olp2/dataset/) has been made publicly available in the context of the 2nd Workshop on Ontology learning and Population (http://olp.dfki.de/olp2/cfp.htm).

4. Information consolidation

In this section we describe how the results of the different IE components are integrated into one coherent knowledge base which can be used for question answering. SOBA relies on the following important assumptions:

- The wrapper procedures extracting information from tables produce reliable and complete knowledge that can be directly inserted into the knowledge base.
- The information extraction system extracts information from text by annotation, i.e. it introduces tags linked to text positions. In the particular system used (SProUT), these are stand-off annotations representing feature structures.
- A consolidation component is needed in order to integrate the information extracted from textual data into the knowledge base constructed on the basis of data extracted from tabular report.

The tasks that the consolidation component needs to accomplish are thus as follows:

- *mapping* of feature structures to appropriate structures compliant with the ontology in question, possibly creating more complex structures;
- *integrating* the information extracted from various sources into one *big picture*, linking extracted resources to each other as well as to entities already existing in the knowledge base;
- *detecting duplicates*, i.e. determining whether newly extracted information is already contained in the knowledge base as well as performing a merge, thus avoiding that information is duplicated. Note that merging is possible only for such values of properties which are not specified as functional in the underlying ontology;
- making *discourse relations* explicit in the knowledge base by relying on a repertoire of specific semantic relations created for this purpose.

Being part of the process of updating a knowledge base, the consolidation component thus takes the output of the IE systems, which is specified in the form of annotations, and transforms these annotations into appropriate ontological structures. In general, the structures in the knowledge base can be much more complex than the tags used by the IE system. For example, named entities corresponding to football players are stored in the knowledge base as three entities, i.e. a *football player* (entity 1) which is *impersonated by* a *natural person* (entity 2) which *has a denomination* (entity 3), which has properties *firstname*, *second name*, *alias*, etc. In addition, in order to avoid the insertion of duplicates, a number of queries needs to be sent to the knowledge base to check whether an entity with the same key attributes is already available. Obviously, the consolidation component needs to be instantiated for each application again. The operations applied in the consolidation component are the same across domains but need to be instantiated with respect to the ontological structures created for each type as well as the queries sent to the inference engine in order to detect duplicates. Thus, we have designed a declarative formalism which allows to define the behavior of the consolidation component independently of the code executing the operations. This eases the task of customizing the system to different domains and allows a knowledge engineer without any programming background to instantiate the consolidation component. In the following sections we describe in which way the information extracted from tables is used as stable background knowledge (see Section 4.1) as well as how this stable core is enriched with information extracted from the text (see Section 4.2). Finally, Section 4.3 discusses how similar procedures are applied to the annotation and semantic indexing of images.

4.1. Tabular match reports as stable background knowledge

Tabular match reports (semi-structured data) are processed using wrapper-like techniques to transform HTML tables into XML files which are translated into knowledge structures (F-Logic, Kifer et al., 1995, RDF Brickley and Guha, 2004) and used to update the knowledge base. The knowledge structures generated from the tabular reports include knowledge about the date and time of the match, the stadium it took place in, the number of attendees, the referee, the teams and their players, but also goals scored as well as yellow and red cards assigned in the match. Fig. 3 gives an example of the knowledge structures (in F-Logic syntax[6]) automatically generated for the match between Italy and France on the 9th of July during the World Championship of 2006.

As mentioned in Section 3.1, the wrappers used to transform HTML tables to an XML representation have been created by hand and are thus completely tailored to the purpose at hand. As a result, a satisfactory level of accuracy allows us to regard the facts extracted in this manner as stable and reliable background knowledge with respect to which the textual match reports can be interpreted. In this sense, the role of the facts extracted from the semi-structured data is to constrain

[6]Special converters allow us to transform back and forth between F-Logic and RDF depending on the purpose.

```
semistruct#'IT_vs_FR_9_Juli_2006_20:00':sportevent#PlayOffFootballMatch
    [
    externalRepresentation@(de) ->> "Italien vs. Frankreich (9. Juli 2006 20:00 Uhr)";
    dolce#'HAPPENS-AT' -> semistruct#'9._Juli_2006_2000_interval';
    sportevent#heldIn -> semistruct#'Olympiastadion_Berlin';
    sportevent#matchNumber -> 64;
    sportevent#team1Result -> 5;
    sportevent#team2Result -> 3;
    sportevent#attendance ->69000;
    sportevent#team1 -> semistruct#'IT_vs_FR_9_Juli_2006_20:00IT_MatchTeam';
    sportevent#team2 -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_FR_MatchTeam';
    sportevent#inTournament -> sportevent#FIFAWorldCup_2006;
    sportevent#inRound -> semistruct#'_WM_2006_Finale';
    sportevent#team1 -> semistruct#'_IT_vs_FR_9_Juli_2006_20_00_Italien_MatchTeam';
    sportevent#team2 -> semistruct#'_IT_vs_FR_9_Juli_2006_20_00_Frankreich_MatchTeam';
    sportevent#matchEvents -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Gianluca_Zambrotta_5_YellowCard';
    [...]
    sportevent#matchEvents -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard';
    sportevent#matchEvents -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane7_Score';
    ...
    ].

semistruct#'IT_vs_FR_9_Juli_2006_20:00_FR_MatchTeam':sportevent#FootballMatchTeam
    [
externalRepresentation@(de) ->> "Frankreich";
    sportevent#name -> "Frankreich";
    sportevent#partOf -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_FR_Squad';
    sportevent#lineup -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Fabien_Barthez_Lineup_PFP';
    (...)
    sportevent#bench -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_JeanAlain_Boumsong_Bench_PFP';
    sportevent#bench -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Vikash_Dhorasoo_Bench_PFP';
    (...)
    sportevent#bench -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Sidney_Govou_Bench_PFP';
    ].

semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_Lineup_PFP':sportevent#FieldMatchFootballPlayer
    [
    externalRepresentation@(de) ->> "Zinedine Zidane (10)";
    sportevent#number -> 10;
    sportevent#hasUpperRole -> semistruct#'_Zinedine_Zidane_Role'
    ].

semistruct#'Zinedine_Zidane_Role':sportevent#FootballPlayer
    [
    sportevent#impersonatedBy -> sportevent#SportEventOntology_Instance_451119
    ].
```

Fig. 3. Result of processing semi-structured data (tables) in F-Logic notation.

the text interpretation process, as will be described in the next section. Overall, the results we present in Section 6.1 show that the accuracy achieved by our hand-crafted wrappers is indeed very satisfactory.

4.2. Text-based enrichment

In addition to processing tabular reports about each match, SOBA also processes text linked to the match in order to extract additional information, specifically additional events that are not represented in the semi-structured data. For example, the semi-structured data do not contain any information about passes, special types of passes (e.g. *crosses*), special types of shots (e.g. *corner*, *freekick*, *penaltykick*) as well as illegal actions (e.g. *fouls*, *headbutts*, etc.), all of which can be represented in the ontology.

This information can indeed be extracted from textual data. In this sense, the information extracted from semi-structured and textual data will complement each other. While the first leads to stable background knowledge, the second links new information to the already existing entities, thus enriching the knowledge base.

For the processing of the texts, the ontology-based integration component relies on text annotated with feature structures as produced by the SProUT system as described in Section 5. The semantic transformation and consolidation component maps extracted events to ontology class instances and links these to the knowledge structures created from the tabular reports. The linking is achieved by querying the knowledge base for players involved in the extracted event, thus

```
soba#id1770:sportevent#Headbutt [
    sportevent#committedBy->semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_Lineup_PFP'
    smartsumo#consequence->'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard'
    dolce#HAPPENS-AT->soba#'TimePointRelative_9_Juli_2006_20:00+110'
].

soba#'TimepointRelative_9_Juli_2006_20:00+110' [
    dolce#ABSOLUTE -> semistruct#'TimePoint_9_Juli_2006_20:00'
    dolce#OFFSET -> "110"
]

semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard':sportevent#ShowingRedCard
[
    sportevent#committedOn-> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_Lineup_PFP'
].

semistruct#'IT_vs_FR_9_Juli_2006_20:00':sportevent#LeagueFootballMatch
[
    sportevent#matchEvents -> soba#id1770;
    sportevent#matchEvents -> semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard'
].

bodyeleminst#'http://smartweb/media/(...)/de_060617_14wg9_10_struct.xml':media#BodyElement
[
    media#talksAbout->soba#id1770;
    media#talksAbout->semistruct'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard';
    media#talksAbout->semistruct#'IT_vs_FR_9_Juli_2006_20:00'
].
```

Fig. 4. Result of processing textual match reports.

linking the newly extracted information to the ID of a player already stored in the knowledge base. All events that can be extracted from the text are linked to a match instance that has been created from the tabular match reports.

For instance, from a text report on the Italy vs. France match on July 9th, 2006, we could extract the event that the player Zinedine Zidane attacked an opponent with a headbutt at minute 110. We can then generate an instance for this event and link it to already available information on this match by pointing to the correct ID for Zinedine Zidane as shown in Fig. 4. The figure shows also that a red card assignment to Zinedine Zidane has been extracted from the text. Instead of creating a new ID, the red card assignment is identified with the red card assignment already available in the knowledge base, i.e. the one with ID `'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard'`. Finally, the discourse analysis component has established a consequence relation between the headbutt and the red card assignment. Furthermore, both extracted events are linked to the text fragment they were extracted from, thus allowing the SmartWeb dialogue component to show this text fragment as answer context on demand.

In summary, what the consolidation component has achieved here is:

- mapping the feature structure that represents the headbutt to an appropriate ontological instance of type `sportevent#Headbutt`,
- finding the appropriate player in the knowledge base to be inserted into the `sportevent#committedBy` slot,
- setting the slot `DOLCE:HAPPENS-AT` to a time point 110 min after the time point at which the game started,
- identifying the extracted red card event with the red card event already available in the knowledge base,
- establishing a *consequence* relation between the headbutt and the redcard event in the knowledge base, and
- linking additional textual information to the red card event available in the knowledge base by making explicit the text fragment from where the event was extracted. In the specific example shown in Fig. 4, this is accomplished via an instance of `BodyElement` which represents the text occurrence where the information was found and points to the Headbutt with ID `soba#1770` via the `media#talksAbout` property.

This shows how in general our text processing approach has indeed the potential to enrich a knowledge base by providing new events and additional links as well as additional textual material.

The mapping from SProUT feature structures to knowledge structures in F-Logic/RDF is specified in a declarative form (XML) and is thus extensible in a flexible manner by adding or modifying the existing rules. In essence, these rules specify the mapping from feature structures to ontological structures in a declarative form. The rule that maps the feature structure for an extracted "headbutt" event to the appropriate ontological structures is shown in Fig. 5.

```
<type orig="s_playeraction" target=sportevent#Headbutt">
  <condition attribute="SPORTACTIONTYPE" value="headbutt">
    <link type=sportevent#LeagueFootballMatch" method=sportevent#matchEvents"
          id="sportevent#"$MATCH""/>
      <map>
        <case>
          <subcase>
            <input>
              <arg orig="COMMITTEDBY:IMPERSONATEDBY:GIVEN_NAME" target="VAR1"/>
              <arg orig="COMMITTEDBY:IMPERSONATEDBY:SURNAME" target="VAR2"/>
              <opt orig="COMMITTEDON:IMPERSONATEDBY:GIVEN_NAME" target = "VAR3">
              <opt orig="COMMITTEDON:IMPERSONATEDBY:SURNAME" target ="VAR4">
              <opt orig="SPORTACTIONPOINT" target ="VAR5">
            </input>
            <output method=sportevent#committedBy" bind = "VAR6" value="q(FORALL Z <- EXISTS Y,R,W,V
            ($MATCH[sportevent#team1 -> Y] OR $MATCH[sportevent#team2 -> Y]) AND
             Y[sportevent#lineup -> Z] AND Z[sportevent#hasUpperRole -> W] AND
             W[sportevent#impersonatedBy -> R] AND R[smartdolce#"HAS-DENOMINATION" -> V] AND
             V[smartdolce#FIRSTNAME -> "VAR1"] AND V[smartdolce#LASTNAME -> "VAR2"]. orderedby Z)"/>
             <output method=sportevent#committedOn" value="q(FORALL Z <- EXISTS Y,R,W,V
            ($MATCH[sportevent#team1 -> Y] OR $MATCH[sportevent#team2 -> Y]) AND
             Y[sportevent#lineup -> Z] AND Z[sportevent#hasUpperRole -> W] AND
             W[sportevent#impersonatedBy -> R] AND R[smartdolce#"HAS-DENOMINATION" -> V] AND
             V[smartdolce#FIRSTNAME -> "VAR3"] AND V[smartdolce#LASTNAME -> "VAR4"]. orderedby Z)"/>
             <output link="dolce#HAPPENS-AT" new="dolce#time-point-relative"
                     method=sportevent#OFFSET" value ="VAR5">
             <key query="FORALL Y <- EXISTS Z $MATCH[sportevent#matchEvents -> Y] AND
             Y:sportevent#Headbutt AND Y[sportevent#committedBy -> VAR6] AND
             Y[smartdolce#'HAPPENS-AT' -> Z] AND Z[smartdolce#OFFSET -> "VAR5]."">
          </subcase>
          <subcase>
            <input>
              <arg orig="COMMITTEDBY:IMPERSONATEDBY:SURNAME" target="VAR1"/>
              <opt orig="COMMITTEDON:IMPERSONATEDBY:SURNAME" target ="VAR2">
              <opt orig="SPORTACTIONPOINT" target ="VAR3">
            </input>
            <output method=sportevent#committedBy" bind="VAR4" value="q(FORALL Z <- EXISTS Y,W,R,V
            ($MATCH[sportevent#team1 -> Y] OR $MATCH[sportevent#team2 -> Y]) AND
            Y[sportevent#lineup -> Z] AND Z[sportevent#hasUpperRole -> W] AND
            W[sportevent#impersonatedBy -> R] AND R[smartdolce#"HAS-DENOMINATION" -> V] AND
            V[smartdolce#LASTNAME -> "uc(VAR1)"]. orderedby Z)"/>
            <output method=sportevent#committedOn" value="q(FORALL Z <- EXISTS Y,W,R,V
            ($MATCH[sportevent#team1 -> Y] OR $MATCH[sportevent#team2 -> Y]) AND
            Y[sportevent#lineup -> Z] AND Z[sportevent#hasUpperRole -> W] AND
            W[sportevent#impersonatedBy -> R] AND R[smartdolce#"HAS-DENOMINATION" -> V] AND
            V[smartdolce#LASTNAME -> "uc(VAR2)"]. orderedby Z)"/>
            <output link="dolce#HAPPENS-AT" new="dolce#time-point-relative"
                    method=sportevent#OFFSET" value ="VAR3">
            <key query="$MATCH[sportevent#matchEvents -> Y] AND
             Y:sportevent#Headbutt AND Y[sportevent#committedBy -> VAR4] AND
             Y[smartdolce#'HAPPENS-AT' -> Z] AND Z[smartdolce#OFFSET -> "VAR3]."">">
          </subcase>
        </case>
      </map>
    </condition>
    </type>
```

Fig. 5. Example illustrating how rules can be declaratively specified in our XML-based formalism (in order to map feature structures of a certain type to appropriate knowledge structures). This rule in particular deals with headbutts.

In what follows, we explain these rules step-by-step. The *type* tag with the attributes "*orig*" and "*target*" indicates the type of the source feature structure (i.e. `s_playeraction` in the example) and the type of the target KB entity, `sportevent#Headbutt` in this case. The *condition* tag poses constraints on the feature structures which cause a rule to fire. For instance, in the above example, the value of the `SPORTACTIONTYPE` attribute needs to be "*headbutt*". The *link* tag specifies any other entities which should point to the entity denoted by the feature structure via some relation. In our example rule, it is specified that the ID of the extracted headbutt event needs to be linked to the ID of the match in question (bound during runtime to the variable $MATCH) through the `matchEvents` relation. This shows how a relation to the existing match extracted from the semi-structured reports can be established. The *map* section then specifies how values

from the feature structure should be mapped to values in the resulting knowledge structures. First of all, different cases can be distinguished (this is represented by the different *subcases*). The first case describes the situation in which the first name and the surname of the player are represented in the feature structure, while the second case corresponds to a feature structure in which only the surname is specified. In both cases, the values of paths in the feature structure are bound to variables `VAR1` or `VAR2`. For example, the value of the path `COMMITTEDBY:IMPERSONATEDBY:SURNAME` is bound to a variable, i.e. `VAR1` in the first case and `VAR2` in the second. These variables are then used in a query in the *output* section to find a player in the knowledge base taking part in the match (as member of the lineup of team1 or the opponent team2), having `VAR1` as `FIRSTNAME` and `uc(VAR2)`, i.e. `VAR2` converted to uppercase, as `SURNAME`.

The output part starting with the attribute "*link = dolce#HAPPENS-AT*" shows that more complex cases for the output can be specified. In fact, what this output rule specifies is that a new entity of type *dolce#time-point-relative* is to be instantiated where the value of the slot `OFFSET` is set to the value of the `SPORTACTIONPOINT` path of the corresponding feature structure. The resulting structure is then linked to the headbutt entity in question via the slot `dolce#HAPPENS-AT`.

Finally, the *key* tag specifies a *query* which is instantiated and sent to the inference engine to find out if the entity is already present in the knowledge base. In this particular case, the query asks for a headbutt event at the same minute and committed by the same player. In case such a headbutt is found, the headbutt extracted from the text would be assigned the same ID as the one already existing in the knowledge base. This is exactly how the red card event extracted from text in the example given in Fig. 4 is identified with an appropriate `sportevent#ShowingRedCard` event already existing in the knowledge base.

The rule described in Fig. 5, for example, would translate the feature structure

$$\begin{bmatrix} \text{SPORTACTIONTYPE} & \text{headbutt} \\ \text{COMMITTEDBY} & \begin{bmatrix} \text{IMPERSONATEDBY} & \begin{bmatrix} \text{GIVEN\_NAME} & \text{Zinedine} \\ \text{SURNAME} & \text{Zidane} \end{bmatrix} \end{bmatrix} \\ \text{SPORTACTIONPOINT} & 110 \end{bmatrix}$$

into the following F-Logic structure (partially depicted also in Fig. 4):

```
soba#id1770:sportevent#Headbutt [
  sportevent#committedBy-
>semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_Lineup_PFP'
  smartsumo#consequence->'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_110_RedCard'
  dolce#HAPPENS-AT->soba#'TimePointRelative_9_Juli_2006_20:00+110'
].
soba#'TimepointRelative_9_Juli_2006_20:00+110' [
  dolce#ABSOLUTE -> semistruct#'TimePoint_9_Juli_2006_20:00'
  dolce#OFFSET -> "110"
]
```

In order to create this structure, first the corresponding player is found in the knowledge base via the following query, an instantiation of the query template of the output method shown in Fig. 5:

```
FORALL Z <- EXISTS Y,R,W,V (
'IT_vs_FR_9_Juli_2006_20:00'[sportevent#team1 -> Y] OR
'IT_vs_FR_9_Juli_2006_20:00'[sportevent#team2 -> Y] ) AND
Y[sportevent#lineup -> Z] AND Z[sportevent#hasUpperRole -> W]
AND W[sportevent#impersonatedBy -> R] AND
R[smartdolce#"HAS-DENOMINATION" -> V] AND
V[smartdolce#FIRSTNAME->"Zinedine"] AND
V[smartdolce#LASTNAME->"ZIDANE"]. orderedby Z)
```

which returns the entity representing "Zinedine Zidane" in the game in question: `semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_Lineup_PFP'`. Once the appropriate player has been identified in the knowledge base, a further query is sent to make sure that the headbutt does not already exist in the knowledge base:

```
FORALL Y <- EXISTS Z
'IT_vs_FR_9_Juli_2006_20:00'[sportevent#matchEvents -> Y] AND Y:sportevent#Headbutt AND
Y[sportevent#committedBy ->
semistruct#'IT_vs_FR_9_Juli_2006_20:00_Zinedine_Zidane_Lineup_PFP']
AND Y[smartdolce#'HAPPENS-AT' -> Z] AND Z[smartdolce#OFFSET ->"110"].
```

In case a headbutt performed by Zinedine Zidane at minute 110 does already exist in the knowledge base, the headbutt is not added to it. In our case the headbutt is not already contained in the knowledge base, such that the net result is that the headbutt committed by Zinedine Zidane is linked to the correct entity representing Zinedine Zidane in the knowledge base and further the headbutt is explicitly encoded as an event of the match between Italy and France on the 9th of July 2006.

4.3. Processing image captions

SOBA also integrates images into the automatically generated knowledge base, which allows for semantic-level image retrieval in the SmartWeb system. For this task, we exploit entities and events that can be extracted from the image captions to annotate and integrate the corresponding image into the knowledge base. To process the image captions, SOBA follows the same process as with text reports, but additionally creates a knowledge base entity for the image. For instance, let us assume that SOBA has extracted a foul-event committed by Gianluca Zambrotta from an image caption. This would then result in the creation of the following knowledge structures:

```
soba#id1785:sportevent#Foul [
  sportevent#committedBy-
>semistruct#['IT_vs_FR_9_Juli_2006_20:00_Gianluca_Zambrotta_Lineup_PFP'
].
semistruct# 'IT_vs_USA_17_Juni_2006_21_00':sportevent#LeagueFootballMatch
[
  sportevent#matchEvents -> soba#id1785;
].
mediainst#'http://localhost:8080/smartweb/media/(...)/Images/
3550564448.jpg':media#Picture
[
  media#shows->soba#id1785;
  media#shows->semistruct#'IT_vs_FR_9_Juli_2006_20:00'
].
```

It is important to note that we do not only record that the image shows the foul extracted from the image caption, but also that this event occurred in the match with ID `'IT_vs_USA_17_Juni_2006_21_00'`. As a result, this allows to ask, for example, for all the images about a certain match but also for pictures showing fouls and even more specifically for all pictures showing a foul by a specific player (i.e. Gianluca Zambrotta in our example).

5. Linguistic analysis and information extraction

In this section, we describe the linguistic analysis components used for extracting information from football texts, in particular facts about football events that are not contained in the structured knowledge sources. Many of the relevant entities (such as players, game results, etc.) can be easily captured by shallow named entity recognition techniques. However, more interesting facts about football events, especially those not captured in match tables, are rather difficult to handle with shallow IE technologies. We therefore designed an extension of a shallow information extraction component that incorporates deeper linguistic analysis in a seamless fashion. In what follows, we first describe the shallow information extraction system SProUT (Shallow Processing with Unification and Typed feature structures) and its interfaces to the SmartWeb ontology. After discussing the limitations of this shallow processing approach, we present an extension of the system that integrates deep syntactic analysis to improve the system's capacity of recognizing complex events. Finally, we discuss a shallow approach to discourse analysis which allows to extract information distributed over several sentences.

5.1. Shallow NLP processing for information extraction

In SOBA, knowledge extraction from textual data is based on a cascade of natural language analysis tools that are available in the HoG architecture (Callmeier et al., 2004), in particular the information extraction system SProUT (Drozdzynski et al., 2004).

5.1.1. SProUT: a shallow IE system using typed feature structures

The SProUT IE system combines finite-state techniques with unification-based processing using typed feature structures (TFSs). It allows the definition of finite-state transduction rules that apply to (sequences of) TFSs, as opposed to atomic symbols. The left-hand side of a transduction rule specifies a regular expression over TFSs as a recognition pattern; the

```
np :> morph & [CAT det, CASE #1, NUM #2, GEND #3] ?
      morph & [CAT adj, CASE #1, NUM #2, GEND #3] *
      morph & [CAT noun & #4, CASE #1, NUM #2, GEND #3] {1,2}
   -> phrase & [CAT #4, CASE #1, NUM #2, GEND #3]
```

Fig. 6. A SProUT example rule: recognizing an NP structure.

right-hand side specifies the output, again in terms of a TFS. Co-references across the feature structures of a rule express unification constraints, and are used to define attribute values in the output feature structure of a rule.

This unique combination of *TFS unification* with *finite-state technology* permits the encoding of highly generalized and compact IE recognition rules. The system includes a *gazetteer component* that associates names of persons, countries, companies, etc. with a corresponding named entity type defined in the recognition grammars. In addition, SProUT allows the user to specify so-called *functional operators* that can define additional constraints for the application of a rule. Recent extensions of the underlying TFS formalism include simple forms of *sets*, *negation* with weak forms of *coreferences*, as well as several output merging techniques (cf. Krieger et al., 2004, for more detail).

The SProUT system incorporates tokenization and morphological analysis tools for many languages, ranging from English, German, French, Spanish, Italian and Dutch over Polish, Czech and Greek to Chinese and Japanese (cf. Drozdzynski et al., 2004; Schäfer and Beck, 2006). Basic IE extraction grammars for MUC-type entities are provided for some of the major languages.

Recently, SProUT has been extended to cascaded processing, such that the output of a set of rule applications can provide the input for another set of rules. This permits the design of *modular*, *cascaded IE grammars*, separating, e.g. the recognition of classical *named entities* such as persons, locations, etc., from more complex information objects, such as *events and their participants*. This novel feature has been exploited in the SmartWeb IE component for the design of a new linguistic analysis architecture, enabling the extraction of complex information types, in particular events and their participants (see Section 5.2). In the following we first describe the way grammars are encoded in SProUT (Section 5.1.2), as well as the rules which are used to extract information using SProUT (Section 5.1.3).

5.1.2. Grammar encoding in SProUT

The example rule in Fig. 6 illustrates how rules can combine *regular expression-based encoding*, using the classical operators `?`, `*` and `{n,m}` for optionality, Kleene star and restricted repetition, respectively, and *TFSs*. The example rule is named `np`, by way of the name tag left to the separator `:>`. It specifies a sequence of three objects of type *morph*: the first one is marked optional (`?`), the second may occur in an infinite (or null) sequence (`*`), and the third is constrained to occur 1–2 times in sequence `{1,2}`. The objects are feature structures of type *morph* that are further specified and distinguished using categorial (`CAT`) attribute values `det`, `adj` and `noun`. The intersection of the type *morph* (defined elsewhere) and the feature structures (stated in the rule in square brackets) is defined using the ampersand sign (`&`). The structures are further constrained by unification of their agreement features `CASE`, `NUM` and `GEND`, which is indicated by way of co-reference tags (#). The three TFS objects defined in a regular expression in the left-hand side of the rule (i.e. left to the arrow (`->`)) constitute constraints on the input structures that need to be satisfied for the rule to be applied. The output of the rule is defined, again as a TFS on the right-hand side of the rule. Here, it is defined as a feature structure of type *phrase*, whose attribute values are further constrained by specifying co-reference with values of the structures on the left-hand side of the rule.[7]

Both for input and output structures, SProUT assumes fully typed TFSs. For the recognition part, the available grammars come with pre-defined type hierarchies for the linguistic modeling aspects, covering mainly tokenization and morphological information (inflection, lemmatization, etc.). For the recognition of event structures on the basis of deep syntactic analysis results (see Section 5.2), this hierarchy was extended with special types (e.g. *syn_args*) for the encoding of linguistic predicate argument structures. Fig. 7 displays a sketch of the basic underlying linguistic type hierarchy.

The following rule illustrates the usage of these formal devices for the task of named entity recognition:

```
goalscore :>  morph & [STEM football & #1_football]
              token{0,2}
              morph & [STEM "in" & #2_prep]
              ( morph & ¬ [STEM "eigen"] & [POS adj, SURFACE #3_attribute]
              | gazetteer & [GTYPE gaz_nationality, FIFA3LCODE #3_attribute] ) ?
              ( morph & [STEM "tor" & #4_goal]
              | morph & [STEM net & #4_goal] )
        ->    s_playeraction & [SPORTACTIONTYPE scoregoal, SPORTACTIONDESCR #5_desc]
where #5_desc = ConcWithBlanks(#1_football, #2_prep, #3_attribute, #4_goal).
```

[7]For more details on the rule syntax, see Drozdzynski et al. (2004).

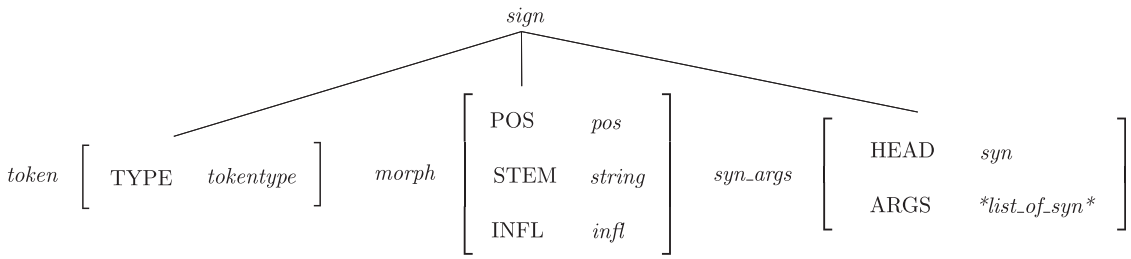

Fig. 7. A snapshot of the SProUT type hierarchy for linguistic objects.

The rule is intended to recognize and define *scoregoal* events in linguistic contexts such as *den Ball* (*nur noch*)? *ins* (*leere*|*italienische*|¬*eigene*)? (*Tor*|*Netz*) (zu schieben/...),[8] and employs regular expression operators to encode the required vs. optional elements of the target expressions. The regular expression encodes constraints on a sequence of TFS objects, defined in terms of their basic type (*morph, token*) and using regular expression operators (such as restricted iteration {0, 2} and simple optionality (?)), as well as disjunction (|). Linguistic terms are either specified by reference to the lemma (STEM "*tor*"), or else by reference to a lexical type (STEM *net*), which subsumes alternative terms for a given concept (see below for more detail). Co-reference variables (#) are used to refer to the values of specific features for the definition of the output object, here a description of the recognized event, yielding e.g. "*ball in italienisch tor*". (The functional operator in the *where* field defines string concatenation.) Note also the combined use of disjunction, negation and optionality to encode an open set of default values and exceptions. In particular, the rule excludes adjectives like *eigen* (*own*) to prevent recognition of own-goals, but allows any other adjectives, e.g. *gegnerisch* (*opponent*). In the second disjunct, the rule allows reference to nationality or FIFA codes, as provided by the gazetteer.

5.1.3. Ontology-based information extraction with SProUT

SProUT comes with basic grammars for the annotation of typical named entity types, such as persons, locations, numerals and date and time expressions. As domain-specific extensions, we implemented rules for the extraction of football-specific entities, such as actors in soccer (trainer, player, referee,...), teams and tournaments. On top of these entity types, we also implemented rules for the extraction of football-specific events, such as player activities (shots, headers,...), match events (goal, card,...) and match results.

As the SOBA IE approach relies on a tight integration of linguistic (terms) and conceptual information (domain semantics), we developed an innovative lexicon model for ontologies, called LingInfo (Buitelaar et al., 2006a, b). LingInfo allows for the representation of linguistic information for each term, in particular a representation of its morphosyntactic structure (gender, number, part of speech, case, etc.). LingInfo objects (i.e. terms) have a representation of their semantics through a back link into the ontology, i.e. the SWIntO domain ontology on football.

Based on the information encoded by the LingInfo objects, we automatically extract a *type hierarchy* as used by SProUT. The following example illustrates this; it displays an excerpt of the SWIntO ontology that has been compiled into a type hierarchy defined in TDL,[9] the representation language used by SProUT:

```
PlayerAction :< SportMatchAction.
SingleFootballPlayerAction :< PlayerAction.
FootballTeamAction :< PlayerAction.
GoalKeeperAction :< SingleFootballPlayerAction.
AnyPlayerAction :< SingleFootballPlayerAction.
```

Properties associated with these concepts are translated into *TDL attributes* of the corresponding types, e.g. the property `inMatch` of the SWIntO class `SportMatchAction` translates to the TDL attribute `INMATCH` that is inherited by all subtypes of the TDL type `SportMatchAction`. The SWIntO property `CommittedBy` that is defined for the SWIntO class `SingleFootballPlayerAction` translates to a corresponding TDL attribute `COMMITTEDBY` of the TDL type `SingleFootballPlayerAction`, and is again inherited by all its subtypes:

```
SportMatchAction
     := swinto_out & [INMATCH Football].
SingleFootballPlayerAction
     := swinto_out & [COMMITTEDBY FootballPlayer].
```

[8] *the ball* (*only*)? *in the* (*empty*|*italian*|¬*own*)? (*goal*|*net*) to (push)

[9] Type Description Language, see Krieger and Schäfer (1994) for details.

As explained above, terms in different languages that express SWIntO concepts are encoded as LingInfo objects and are compiled into *TDL lexical types* thus supporting information extraction. Below, we see the encoding of German terms for corresponding SWIntO concepts:

```
"erzielen" :< GoalScore.
"treffen" :< GoalScore.
"verwandeln" :< GoalScore.
"treffer" :< GoalScore.
"auswärtstor" :< AwayGoal.
"eigentor" : < OwnGoal.
"führungstor" :< LeadingGoal.
"sperren" :< Banned.
```

Ambiguous terms, such as *Tor* (*goal*) in the `Object` vs. `GoalScore` readings are represented by use of multiple inheritance. Other types of ambiguities involve terms that express an event type such as *Abseits* (*offside*) or a player role such as *Abwehr* (*defense*) vs. the corresponding position in the field.

```
"tor" := GoalObject & GoalScore.
"abseits" := Offside & OffsidePosition.
"abwehr" := Defender & DefenceLine.
```

SProUT extraction patterns can thus be triggered by lexical types and define output structures that correspond directly to the classes and properties in the SWIntO ontology. For instance, a "*banned_player*" rule defines an extraction pattern for the SWIntO class `BanEvent` with attributes `CommittedBy` and `InMatch`. This rule is defined to be triggered, for instance, by the German term (LingInfo object) "sperren" (*to ban*). Example sentences from the SmartWeb development corpus to which this rule applies are as follows:

(2) "... ist Petrow für die Partie gegen Schweden gesperrt."
("... has Petrow been banned for the match against Sweden")
(3) "... ist David Trezeguet von der FIFA für zwei Spiele gesperrt worden."
("... has David Tezeguet been banned by the FIFA for two matches")

5.2. Event recognition: limitations of shallow IE systems

Shallow IE techniques based on finite-state processing are highly efficient and appropriate for recognizing entities that can be identified with high confidence using local contextual constraints. Prime examples are classical entity types (persons, times, goal results, teams, etc.) as well as event mentions that are realized in local syntactic configurations, for instance simple nominal phrase structures. Configurations as in (4) can be easily captured by patterns based on regular expressions that specify sequences of nouns and prepositions that satisfy relevant terminological constraints for domain concepts (*Führungstor*—`LeadingGoal`, *Ecuador*—`Team`, *Lara*—`Player`) and an appropriate class of semantically indicative prepositions (*für* (*for*), as opposed to *gegen* (*against*) and *durch* (*by*)). Applied to (4), a simple rule as depicted in Fig. 8 can easily recognize that the team Ecuador fills the `Team` role in the concept `ScoreGoal`, and that the player *Lara* fills the role `CommittedBy`.[10]

(4) Das Führungstor$_{Scoregoal}$ für Ecuador$_{Team}$ durch Lara$_{Player}$
The leading-goal for Ecuador by Lara
"The goal by Lara giving Ecuador the lead"

However, events are typically realized using more complex verbal constructions involving free word order, coordination, long distance constructions, etc. which make it difficult to identify the arguments of event concepts. This is illustrated in (5), the full context of example (4).

[10]The rule in Fig. 8 makes use of SProUT's *seek* operator (@seek), which refers to (the results of) independently defined recognition rules, here rules for `team` and `player`. The rule `player_context` recognizes player names not via gazetteer entries, but local context information, such as *Verteidiger* (defender) or *Mittelfeldspieler* (midfield player).

```
leading_goal :> morph & [STEM leadinggoal & #event]
                morph & [STEM "für"]
                @seek(team) & [NAME #team]
                ( morph & [STEM goalscore] | morph & [STEM "durch"] )
                ( @seek(player) & [IMPERSONATEDBY #player, HASROLE #role]
                | @seek(player_context) & [IMPERSONATEDBY #player] )
-> s_playeraction & [SPORTACTIONTYPE #type, SPORTACTIONDESCR #event,
                     COMMITTEDBY s_footballplayer &
                                 [IMPERSONATEDBY #player,
                                 HASROLE #role, INMATCHTEAM #team]],
where #type = GetParentType(#event).
```

Fig. 8. Extraction rules based on local context window.

(5) Das Führungstor$_{e4}$ für Ecuador$_{Team}$ durch Lara$_{Player}$ fiel nach
The leading-goal for Ecuador by Lara was scored after
einer Vorlage$_{e3}$ des technisch ausgezeichneten Nicer Reasco$_{Player}$,
a delivery of the technically excellent Nicer Reasco,
der einen langen und zu ungenauen Pass$_{e1}$ des Argentiniers Carlos
who a long and too inaccurate cross by the Argentine Carlos
Tévez$_{Player}$ in den gegnerischen Strafraum abfangen$_{e2}$ konnte.
Tévez into the penalty area intercept could.

"*The goal by Lara giving Ecuador the lead was scored after a delivery from the skilled Nicer Reasco, who intercepted a long and inaccurate cross by the Argentine player Carlos Tévez into the penalty area.*"

There are four events to be recognized in (5), which occur in the temporal order e1<e2<e3<e4.[11]

e1: Pass: [CommittedBy Tévez]
e2: Intercept: [CommittedBy Reasco, CommittedOn Tévez]
e3: Assist: [CommittedBy Reasco]
e4: ScoreGoal: [CommittedBy Lara, Team: Ecuador]

While recognizing *Carlos Tévez* as the agent of Pass (*Pass*) in its local NP construction (*Pass des Argentiners Carlos Tévez*) is straightforward, the agent of `Intercept` (*abfangen*), *Nicer Reasco*, cannot be identified with sufficient confidence without taking syntactic structure into account—here a complex object argument (*einen langen und zu ungenauen Pass des Argentiniers Carlos Tévez in den gegnerischen Strafraum*) that separates the verb from its syntactic subject. Typical heuristics applied in finite-state-based processing, such as choosing the nearest constituent of type Player would yield the wrong player, namely *Carlos Tévez*.

5.3. Integrating shallow IE with deep syntactic analysis

Non-local configurations of this type represent a challenge for finite-state-based extraction techniques. A number of methods have been proposed for the integration of "deep" and "shallow" grammar processing models in so-called *hybrid NLP architectures*, which try to combine the robustness of shallow processing tools with the higher precision and fine-grainedness of deep linguistic analysis (cf. Crysmann et al., 2002; Frank et al., 2003, 2004).

For the recognition of complex event structures in the football domain, we have designed a novel integration architecture that builds on the core machinery for shallow processing, offering a seamless extension of the IE system architecture to incorporate deeper linguistic knowledge in a focused way. We make use of existing interface modules of SProUT to import selected information about syntactic dependencies from an external grammar component, tailoring this additional level of information to the specific formalism and processing methods of the shallow IE system.

An overview of the integration architecture is displayed in Fig. 9.[12] Concurrently with the main processing thread using the SProUT engine, we run a robust statistical PCFG parser for German, the Sleepy parser (Dubey, 2005).[13] From the syntactic analysis results delivered by the parser, we extract *local dependency structures* of verbal syntactic heads. These

[11]Currently, we do not try to extract temporal relations at the level of event recognition. Discourse relations together with their temporal implications are inferred in the discourse processing step (see Section 5.4).

[12]This integrated processing architecture has been realized as a web service, and was enhanced with interfaces to support efficient grammar development.

[13]The Sleepy parser has been trained on the syntactically annotated TIGER corpus (Brants et al., 2002).

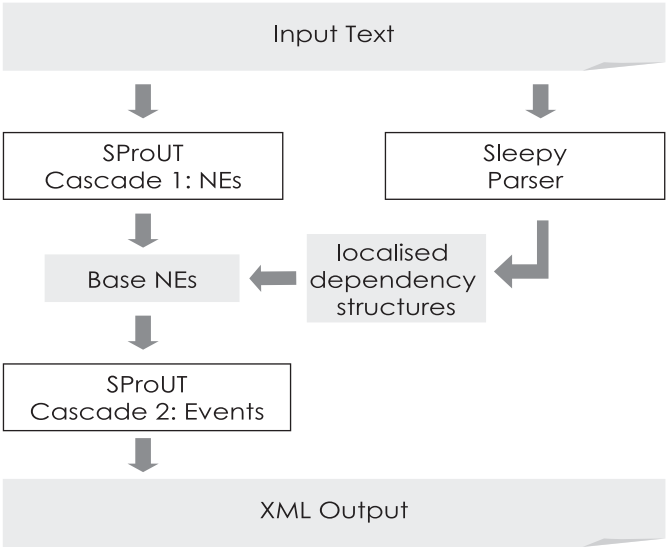

Fig. 9. Processing architecture: importing syntactic analysis into SProUT.

local argument structures are imported into the SProUT system as TFSs that are predefined as extended linguistic data structures in the SProUT linguistic hierarchy.

Example (6) illustrates the basic principle. For each syntactic head, we extract syntactic information about its dependents, as delivered by the parser.

(6) a.

[Dort] kam [Herrera] [an den Ball] und

There came Herrara to the ball and

drückte [ihn] [per Kopf] [zu seinem Mitspieler mit der Nr 10] [herunter].

pitched it by head to his team-mate with the no. 10 down

"There Herrera came to the ball and pitched it down with his head to his team-mate with the number 10"

b. Localized dependency information for lexical heads

head verb *kam*:
[CAT adv, LB mod, STEM dort, SURFACE 'dort']
[CAT np, LB act_subj, STEM Herrara, SURFACE 'Herrera']
[CAT pp, LB mod, STEM an, SURFACE 'an den Ball']
head verb *drückte*:
[CAT np, LB act_subj, STEM Herrara, SURFACE 'Herrera']
[CAT pper, LB obj, STEM pro, SURFACE 'ihn']
[CAT pp, LB mod, STEM per, SURFACE 'per Kopf']
[CAT pp, LB mod, STEM zu, SURFACE 'zu …Nr 10']
[CAT adv, LB mod, STEM herunter, SURFACE 'herunter']

The extracted data structures specify lexical and syntactic properties of the head, and the list of its dependents, each of them again defined in terms of syntactic category (CAT), grammatical function (LB), lemma (STEM) and surface information (SURFACE), given in terms of the constituent's character span. Where appropriate, the actual syntactic categories defined by the parser output can be further normalized. For example, in our small hierarchy of syntactic types, displayed in Fig. 10, we distinguish between active and passive subjects, to ease correct reference to event participants, and

```
syn_args := sign & [ HEAD syn, ARGS *list_of_syn* ].
syn := *avm* & [ CAT cat,
                 LB fnct,
                 STEM string,
                 TYPE sportaction,
                 PASSIVE boolean,
                 CSTART string,
                 CEND string ].
```

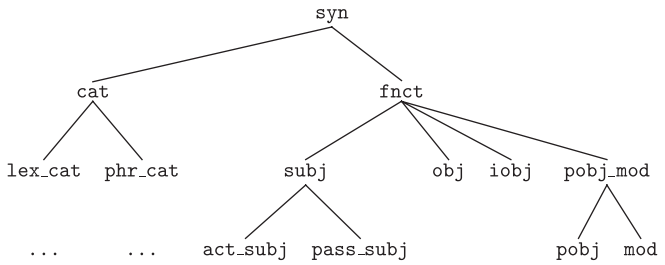

Fig. 10. Subhierarchies for the encoding of local dependency structures.

defined an underspecified type for prepositional modifier (mod) and argument (pobj) functions, which are often difficult to distinguish in parsing.

By representing the syntactic dependency structure of heads as a local feature structure of the projecting lexical head, here the verb, we can greatly simplify the reference to non-local dependents: The syntactic dependents that are recorded in the verb's argument feature structure (cf. (6.b) and Fig. 10) are characterized using their surface position in the input sentence (`CSTART, CEND`). That is, we can refer to the arguments identified by the concurrent parsing process without any reference to the complex syntactic structures constructed by the parser. Instead, we encode the constituents identified by the parser by reference to their surface position, together with categorial and functional information, and can thus define SProUT recognition rules that can access local or non-local dependents of the verb by simple reference to grammatical features and the surface position of the dependents encoded in its local argument structure, i.e. without traversing complicated syntactic structures. As a consequence, the imported and locally represented syntactic information permits access to verbal dependents in non-local configurations that are far beyond the scope of standard finite-state-based methods.

For the realization of this integration architecture, the SProUT named entity recognition grammar has been redesigned as a *cascaded grammar architecture* that separates the recognition of basic named entity types, treated in the bottom cascade level, and events, which are defined in the second cascade. The second cascade takes as input the basic NE output structures recognized by the first cascade, combined with the external syntactic knowledge sources provided by the concurrent syntactic parser.[14] For the definition of syntax-based recognition rules, the SProUT system was extended with a number of functional operators to manipulate list-valued feature structures.

A very general rule for event recognition is illustrated in Fig. 11. The rule recognizes a variety of different event types, referred to by way of the lexicalized action types in the head's `STEM` attribute. The argument structure information imported from syntactic parsing is accessible via the *syn_args* `ARGS` attribute; the functional operator `InListFS` allows reference to individual syntactic functions in the dependents list (e.g. the *act*(*ive*)*_subj* in Fig. 11) and their attributes, via the *inFSFeature* operator, to refer, for example, to the named entity information (NE_FS) of the selected constituent that derives from the first grammar cascade. The values of the attributes referred to in this way are again used to define the semantic output structure.

The rule in Fig. 11 applies rather generically to verbs of different semantic classes (e.g. *stürmen* (to strike)—*FootballTeamAction*, *blockieren* (to block)—*SingleFootballPlayerAction*, or *abfälschen* (deflect)—*BallEvent*) in a variety of syntactic configurations, as illustrated in the example text passages in (7). It identifies and outputs the respective action type information, together with the information about the agent of the action in the event's `COMMITTED_BY` attribute.

(7)

a. *Der Stürmer Ballack* köpfte den Ball ins Netz ('The striker Ballack made a header into the net')
b. *Guevara* verwandelte den folgenden Strafstoß zum 3:2 ('Guevara transformed the next penalty into 3:2')
c. ... scheiterte *Gabriel Batistuta* mit einem Kopfball ('failed Gabriel Batistuta with a header')

[14]The named entities recognized in the first grammar cascade are integrated into the local syntactic dependency structures using the character span information of the dependents as an index for assembly.

```
action :> ( syn_args & [HEAD verb & [STEM FootBallTeamAction & #descr],
                        ARGS #args]
          | syn_args & [HEAD verb & [STEM SingleFootBallPlayerAction & #descr]
                        ARGS #args]
          | syn_args & [HEAD verb & [STEM BallEvent & #descr],
                        ARGS #args]
          | syn_args & [HEAD verb & [STEM Sportsituation & #descr],
                        ARGS #args]
-> s_playeraction &
                [SPORTACTIONDESCR #descr, SPORTACTIONTYPE #type,
                 COMMITTEDBY s_footballplayer &
                    [HASROLE #role1, INMATCHTEAM s_team & [NAME #team1],
                     IMPERSONATEDBY ne-person & [GIVEN_NAME <#gname1>,
                                                 SURNAME #sname1,
                                                 NATIONALITY #nat1]]],
where
#subj    =InListFS(act_subj, #args),
#ne_subj=InFsFeature("NE_FS", #subj),
#role1   =InFsFeature("HASROLE", #ne_subj),
#person1=InFsFeature("IMPERSONATEDBY", #ne_subj),
#inteam1=InFsFeature("INMATCHTEAM", #ne_subj),
#team1  =InFsFeature("NAME", #inteam1),
#gname1=InFsFeature("GIVEN_NAME", #person1),
#sname1=InFsFeature("SURNAME", #person1),
#nat1   =InFsFeature("NATIONALITY", #person1),
#type   =GetParentType(#descr),
IsAtLeastOneDefined(#person1).
```

Fig. 11. Event recognition: access to syntactic dependency information.

d. ... erzielte *Luiz Fabiano* ein Tor ('attained Luiz Fabiano a goal')
e. ... traf schließlich vier Minuten vor dem Schluss *Herrera* ('hit finally four minutes before the end Herrera')
f. Das 5:0 erzielte schließlich vier Minuten vor dem Schluss *Herrera* ('The 5:0 attained finally four minutes before the end Herrera')
g. Und es war dann auch *Pacheco*, der in der 17. Spielminute folgerichtig das erste Tor erzielte ('And it was therefore Pacheco who attained in the 17th minute the first goal')

Other rules impose finer constraints on the linguistic structure and/or semantic types of the arguments. For example, to capture contexts like ⟨*player*⟩ goes for ⟨*action*⟩, as in *Ronald Gomez entscheidet sich für einen direkten Torschuss* (Gomez goes for a direct goal shot), the rule checks for the presence of a prepositional object with preposition *für* and of semantic type *sportactiontype*, of which "*direkter Torschuss*" is just one possible instance.

The syntactic recognition patterns can specify alternative syntactic contexts, for example prepositional objects with a number of different prepositions, such as *über* (*over*/*above*) and *neben* (*next to*) to recognize ball events of type *miss* in realizations like (8). Example (8.b) clearly illustrates the benefits of our hybrid approach, which incorporates syntactic information about dependencies: reference to the subject argument correctly identifies *Camoranesi* as the agent of the action, as opposed to *Pirlo*—deeply embedded within the intervening object argument.

(8)

a. der Stürmer schoss eine Freistossvorlage knapp [neben den Pfosten]
   the striker shot a freekick tightly next to the post

b. setzte Camoranesi [eine Freistossvorlage von Andrea Pirlo]
   performed Camoranesi a freekick by Andrea Pirlo
   mit dem Kopf [knapp über das deutsche Tor]
   with the head narrowly over the German goal

In sum, with SProUT recognition rules being able to make use of externally provided deep syntactic information, it is possible to reliably identify concepts in linguistic constructions that are usually beyond the scope of shallow IE recognition

systems. The integration architecture is designed as to permit integration of different parsers, and can be carried over to different languages.

5.4. Discourse analysis

As is widely acknowledged (cf. Asher and Lascarides, 2003, among others), information about events in texts is often distributed over several sentences. Since the football domain is no exception with regard to this observation, our goal was to build a module which is capable to discover such anaphoric information and can be seamlessly integrated into SOBA's processing pipeline in order to provide answers to complex questions asking for relationships between events.

In these lines, we have implemented a shallow component for discourse analysis which is based on the computation of discourse relations. Discourse relations imply semantic effects, i.e. given that a specific discourse relation R holds between two events e_1 and e_2, we can conclude a certain semantic relatedness of e_1 and e_2. Within the SOBA system, we focus on three types of discourse relations: *prepares*, *result* and *elaboration*. Being a football-specific adaptation of SDRT's *Narration* relation (Asher and Lascarides, 2003), the *prepares* relation denotes the immediate temporal precedence between an event e_1 (e.g. a *pass* or a *cross*) and an event e_2 (e.g. a *scoregoal*) occurring afterwards without any intervening event e_3.

The relations *result* and *elaboration* are also taken from the inventory presented by Asher and Lascarides (2003), where the presence of a relation *result* (e_1, e_2) is defined to imply a causative connection between those events and *elaboration* (e_1, e_2) leads to the interpretation that e_2 is temporally included within e_1.

As opposed to approaches which make use of deeper knowledge sources and more elaborate reasoning techniques for the computation of these discourse relations (e.g. Hartung, 2006), our method is restricted to prototypical knowledge about the connections between events in a specific real-world scenario. One of the reasons which prevents us from adopting deeper discourse processing within SOBA stems from the shallow event extraction procedure which yields rather low performance with regard to argument structure in some cases. The rules we apply to compute discourse relations between events have the following form:

(9) $sportevent\#Pass(e_1) \wedge sportevent\#ScoreGoal(e_2) \wedge e_1 \prec e_2 \Rightarrow smartsumo\#prepares(e_1, e_2)$
$sportevent\#Shot(e_1) \wedge sportevent\#GoalScore(e_2) \wedge e_1 \prec e_2 \Rightarrow smartsumo\#result(e_1, e_2)$
$sportevent\#ScoreGoal(e_1) \wedge sportevent\#Header(e_2) \wedge e_1 \prec e_2 \Rightarrow smartsumo\#elaboration(e_1, e_2)$

In the above rules, variables are assumed to be universally quantified and $\prec$ is the relation denoting the surface order between events. The first rule covers cases where a *Pass* is mentioned in a text before a *ScoreGoal* event, such that our discourse processing component assumes that the *ScoreGoal* event has been *prepared* by the *Pass*. The second rule states that a *GoalScore* and a preceding *Shot* should be interpreted in such a way that the former *results* from the latter. The third rule indicates that in the context of a preceding *ScoreGoal* mention, a *Header* should be interpreted as a sub-part of the complex event of scoring a goal and thus as an *elaboration* on the mentioned *ScoreGoal* event.

As an additional heuristic, we assume that discourse relations can only occur among events contained in one and the same paragraph, i.e. we exclude discourse relations across paragraphs. Thus, our algorithm basically loops over the set of events extracted from the current paragraph of a text and successively matches pairs of events against one of the discourse rules if appropriate.

Generally, our approach to discourse interpretation is based on the premise that we can conclude from the textual order of events on the surface to some of their temporal and semantic features. The backbone which licenses this step is world knowledge about prototypical chains of events as modeled in the discourse rules above. Some of the typical errors produced by such a rule-based approach are displayed in the following examples:

(10) Pavel Pardos [Freistoß]$_{Freekick}$ von der rechten Seite wurde von Guillermo Flanco verlängert und von dem am zweiten Pfosten stehenden Omar Bravo unbedrängt [ins Tor gesetzt]$_{ScoreGoal}$.[15]

The relation our system infers for this example is *prepares*(*Freekick*, *ScoreGoal*). However, this solution does not reflect the intervening *BallDeflection* which is due to erroneous results at the stage of event extraction.

Example (11), which results in the relation *prepares*(*Pass*, *ScoreGoal*) is similar to (10) insofar as in both cases intervening events—*Interception* in (11) and *BallDeflection* in (10)—are not recognized. What distinguishes (11) from (10), however, is that the intervening event has to be inferred as an implicit contribution in the former case whereas it is explicitly mentioned, though not successfully recognized on the level of event extraction, in the latter. Apart from errors relating to the event extraction module, (11) represents one of the most problematic issues for our approach to discourse analysis, since world knowledge about the prototypical relation between certain events is overridden[16] in the given discourse.

[15]Pavel Pardo's freekick from the right wing was redirected by Guillermo Flanco, and put into the back of the net by Omar Bravo at the far post.
[16]This topic is elaborated in more detail in Asher and Lascarides (2003) and Hartung (2006).

(11) Ein schlimmer Fehler vom letzten Mann in Ghanas Defensive, Samuel Kuffour, beschert den Italienern ihr zweites Tor: sein [Rückpass]$_{Pass}$ zum Torwart fällt zu kurz aus, der eingewechselte Vincenzo Iaquinta nimmt das Geschenk dankend an, umrundet den Keeper und [schiebt den Ball ins leere Tor]$_{ScoreGoal}$.[17]

(12) In einer Partie, die von den Ecuadorianern von der ersten Minute an dominiert wurde, [erzielte Carlos Tenorio die frühe Führung]$_{ScoreGoal}$. Es war zugleich sein zweiter [Treffer]$_{ScoreGoal}$ in diesem Turnier.[18]

Coreference resolution is another problem our method is affected by. Rules of the form $sportevent\#ScoreGoal(e_1) \wedge sportevent\#ScoreGoal(e_2) \wedge e_1 \prec e_2 \Rightarrow smartsumo\#elaboration(e_1, e_2)$ are, on the one hand, necessary in numerous cases like (12), but also frequently overgenerate in examples such as (13).

(13) Doch Toni ließ mit seinem zweiten [Treffer]$_{ScoreGoal}$ nach schöner Vorarbeit von Zambrotta die ukrainischen Halbfinalträume endgültig platzen (69.). Italien hat somit weiter im gesamten Turnierverlauf bisher nur einen [Treffer]$_{ScoreGoal}$ kassiert, das Eigentor gegen die USA.[19]

(14) Nur drei Minuten später zappelte der Ball dann [im Netz]$_{GoalScore}$- doch auf Seiten der Spanier, als Tunesien den ersten gefährlichen [Angriff]$_{Attack}$ des Spiels mit einem [Tor]$_{ScoreGoal}$ abschloss. Zied Jaziri setzte sich wunderbar gegen Carlos Puyol durch und zog im Strafraum drei spanische Verteidiger auf sich, bevor er das Leder zu Jaouhar Mnari [passte]$_{Pass}$, der im zweiten Versuch Torwart Iker Casillas aus kurzer Entfernung [überwand]$_{ScoreGoal}$.[20]

The only relation our system generates for (14) is *prepares*(*Pass*, *ScoreGoal*). Although this result is absolutely correct from a perspective of precision, it sheds light on a deficit of our discourse module with regard to recall: Since our algorithm is restricted to checking pairs of events, we are only capable of generating local relations between events as opposed to build up complete hierarchical discourse structures in an SDRT fashion (see Asher and Lascarides, 2003). For (14), such a complete discourse structure would indicate an *attack* as the top node of the discourse, which all the subsequent events elaborate on.

Nevertheless, with respect to the overall evaluation results reported in Section 6, the approach to discourse analysis as described here can be considered effective. In fact, local relations between events turned out to cover a reasonable number of prototypical cases and thus serve their intended purpose of enhancing SOBA's capability to process complex queries relating to anaphorically connected events.

6. Evaluation

In order to measure the accuracy of the information extraction and the quality of the generated knowledge base, we performed a number of evaluation experiments. As the type of information extracted from tables differs significantly from the one extracted from text, we devised two rather different evaluation strategies as explained in the next sections.

For the evaluation of information extraction from tables we relied on an "*a posteriori*" manual evaluation, whereas for information extraction from text we used an "*a priori*" constructed manual benchmark for automatic evaluation of extracted entities and events in combination with an "*a posteriori*" manual evaluation of the extraction of more complex event chains. In the following sections we first present a manual evaluation of the information extraction from tabular data (see Section 6.1). The evaluation of the information extracted from text in terms of precision and recall with respect to a hand-annotated gold standard is presented in Section 6.2, while the discourse analysis component is evaluated in Section 6.3.

6.1. Evaluation of facts extracted from tables

The evaluation of fact extraction from the FIFA tables could only be carried out by validation with respect to other publicly available football knowledge sources. We therefore manually verified the knowledge base with respect to

[17]A terrible mistake by Samuel Kuffour, the last man in the Ghanaian defense, gives Italy the 2nd goal: his pass back to the goalkeeper is too short and Vicenzo Iaquinta picks it up thankfully, moves past the keeper, rolling the ball into the empty net.

[18]In a match dominated by Ecuador from the first minute on, Carlos Tenorio scored the early lead. This was at the same time his second goal in the tournament.

[19]But Toni ended the Ukrainian dreams of the semi-final with his second goal after Zambrotta set him up superbly (69.). As a result, Italy continues to have conceded only one goal throughout the whole tournament, namely the own-goal against the USA.

[20]Only three minutes later the ball was in the back of the net—however, on the Spanish side, when Tunisia finished their first dangerous attack with a goal. Zied Jaziri superbly dribbled past Carlos Puyol and drew attention from three Spanish defenders before passing on the ball to Jaouhar Mnari, who beat goalkeeper Iker Cassillas from close range with his second attempt.

Table 1
Number of entities extracted from tabular match reports

Championship	# Entities	# Relations
1930	1628	3790
1934	1883	4154
1938	1915	4349
1950	2701	4964
1954	2536	7226
1958	2966	8784
1962	2694	7831
1966	2755	7976
1970	3082	8845
1974	3489	10 349
1978	3538	10 429
1982	5019	14 622
1986	5166	15 015
1990	5189	15 074
1994	5456	15 880
1998	7193	20 511
2002	7527	21 308
2006	7645	21 757
Total	67 272	196 913

Table 2
Number of goals, red cards and yellow cards extracted with difference between extracted and real numbers

	Avg. diff. goals	Yellow (real)	Yellow (SOBA)	Diff.	Red (real)	Red (SOBA)	Diff.
1930	0.5	0	0	0	1	1	0
1934	0.17	0	0	0	1	1	0
1938	0.58	0	0	0	4	4	0
1950	0.91	0	0	0	0	0	0
1954	0.19	0	0	0	3	3	0
1958	0.14	0	0	0	3	3	0
1962	0	0	0	0	6	6	0
1966	0	0	20	20	5	5	0
1970	0	45	33	12	0	0	0
1974	0	83	85	2	5	5	0
1978	0.27	67	57	10	3	2	1
1982	0	100	98	2	5	5	0
1986	0.08	135	133	2	8	8	0
1990	0	163	160	3	16	16	0
1994	0	228	235	7	15	15	0
1998	0.08	250	258	8	22	22	0
2002	0.21	266	271	5	17	17	0
2006	0	326	345	19	28	28	0
All	0.17 (avg.)	1412	1695	4.74 (avg.)	142	141	0.06 (avg.)

benchmark information provided by the Soccer Hall of Fame web site.[21] In particular, we verified how many of the facts that we extracted from the FIFA tables were actually incorrect as well as in how many cases our knowledge base contained a correct fact which was not listed on the Soccer Hall web site. This was verified by consulting other external sources such as Web.de and Wikipedia. Table 1 shows the number of entities (players, scores, yellowcard events, etc.) as well as relations extracted from the tabular match reports for each wold championship. In general, the extracted information was highly accurate due to the fact that the wrapper had been tuned by hand iteratively until reasonable results were achieved. Most errors in the extracted data could be directly traced back to errors in the FIFA pages. In general, our system can be regarded as having an almost 100% accuracy on the facts extracted from tabular reports. When analyzing the number of events such as goals, yellow cards and red cards extracted, and comparing these numbers to the ones compiled from other

[21] http://www.soccerhall.org/history/WorldCup_[1930...2002].htm; last access on 06.07.2008.

Table 3
Evaluation of extraction from text (test data set)

SWIntO event types	Instances	Precision (%)	Recall (%)	F-Measure (%)
Diving	1	100	100	100
Hattrick	1	100	100	100
Handball	1	100	100	100
OwnGoal	5	80	80	80
ShowingYellowRedCard	23	86.7	56.5	68.4
ScoringOpportunity	127	75.8	54.3	63.3
Block	26	92.3	46.2	61.5
FreeKick	69	88.9	46.4	61
Volley	13	85.7	46.2	60
GrassCutter	12	100	41.7	58.8
Save	5	100	40	57.1
CornerKick	53	83.3	37.7	51.9
Miss	272	77.4	39	51.8
Challenge	15	100	33.3	50
Header	58	67.6	39.7	50
ShowingYellowCard	13	80	30.8	44.4
Equalizer	27	72.7	29.6	42.1
Rebound	6	50	33.3	40
Cross	95	80.6	26.3	39.7
Trap	45	70.6	26.7	38.7
Shot	213	53.3	30	38.4
Parry	180	78.6	24.4	37.3
Pass	118	78	22	34.4
Ban	4	50	25	33.3
PenaltyKick	19	41.7	26.3	32.3
Assist	20	44.4	20	27.6
HandBall	7	100	14.3	25
Foul	42	100	14.3	25
ShowingRedCard	13	33.3	15.4	21.1
ScoreGoal	206	55.3	12.6	20.6
BallDeflection	35	80	11.4	20
Clear	46	100	10.9	19.6
SendingOff	28	100	10.7	19.4
PunchOut	12	100	8.3	15.4
Substitution	79	66.7	7.6	13.6
Dribble	31	50	3.2	6.1
LeadingGoal	64	25	3.1	5.6

sources such as Web.de and Wikipedia, a number of divergences appear which are summarized in Table 2. The table shows the average difference in the number of goals, yellow cards and red cards for each world championship compared to the numbers as specified by other sources. Overall, the average differences are really negligible (0.17 difference in goals on average, 4.74 on yellow cards and 0.06 on red cards). These differences are obviously due to the facts that (i) there is never a total agreement between different sources and (ii) humans introducing the data also make errors. Summarizing, we can indeed say that the quality of our automatically compiled world championship knowledge base is very high.

6.2. Evaluation of the extraction of entities and events from text

For the evaluation of entities and events extracted from text we could not rely on similar benchmarks as the ones used for the extraction from tables since we had a focus particularly on those events (and entities involved) that were not covered by the FIFA tables. We therefore had to construct a manually annotated benchmark of FIFA textual match reports. At the same time we used a similar set of FIFA textual match reports for development purposes. Both sets were annotated by a domain expert who had been given guidelines on how and what to annotate. The focus of the manual annotation was on events as defined by the SWIntO ontology including the events covered by the tables (ScoreGoal, Substitution, Penalty, ...) as well as those events that were only mentioned in the textual match reports (Header, Assist, Foul, ...). Additionally, entities were annotated with their roles: `CommittedBy` or `CommittedOn` for `PlayerActions`, `penalizedPlayer` for `RefereeActions` and `inPlayer` for `Substitutions`.

Table 4
Evaluation of extraction from text (development data set)

SWIntO event types	Instances	Precision (%)	Recall (%)	F-Measure (%)
Diving	2	0	0	0
Hattrick	–	–	–	–
Handball	1	0	0	0
OwnGoal	5	100	20	33.3
ShowingYellowRedCard	4	67	50	57.1
ScoringOpportunity	46	93.3	60.9	73.7
Block	11	87.5	63.6	73.7
FreeKick	27	100	59.3	74.4
Volley	3	100	66.7	80
GrassCutter	6	75	50	60
Save	6	100	50	66.7
CornerKick	24	100	41.7	58.8
Miss	82	95.9	57.3	71.8
Challenge	2	0	0	0
Header	17	66.7	47.1	55.2
ShowingYellowCard	4	100	50	66.7
Equalizer	4	0	0	0
Rebound	1	0	0	0
Cross	15	100	46.7	63.6
Trap	19	90.9	52.6	66.7
Shot	72	76.7	45.8	57.4
Parry	64	89.3	39.1	54.3
Pass	33	100	36.4	53.3
Ban	2	0	0	0
PenaltyKick	8	100	50	66.7
Assist	17	100	47.1	64
HandBall	1	0	0	0
Foul	12	100	25	40
ShowingRedCard	2	100	100	100
ScoreGoal	59	76	32.2	45.2
BallDeflection	21	100	52.4	68.8
Clear	14	100	14.3	25
SendingOff	7	0	0	0
PunchOut	3	100	33.3	50
Substitution	9	50	11.1	18.2
Dribble	14	100	21.4	35.3
LeadingGoal	19	100	21.1	34.8

Table 5
Evaluation in terms of precision, recall and F-Measure for selected relations (roles)

SWIntO event types	Role	Precision (%)	Recall (%)	F-Measure (%)
Diving	committed_by	0	0	0
Handball	committed_by	100	100	100
OwnGoal	committed_by	0	0	0
ShowingYellowRedCard	penalized_player	50	4.5	8.3
ScoringOpportunity	committed_by	0	0	0
	committed_on	100	2.2	4.3
Block	committed_by	100	5.9	11.1
	committed_on	100	37.5	54.5
FreeKick	committed_by	100	21.2	34.9
	committed_on	100	28.6	44.4
Volley	committed_by	100	7.7	14.3
	committed_on	0	0	0
GrassCutter	committed_by	100	25	40
	committed_on	0	0	0
Save	committed_by	100	20	33.3

Table 6
Overall averaged results (micro/macro) for type and roles

	Precision (%)	Recall (%)	F-Measure (%)
Macro-average (types)	51	23	31
Macro-average (roles)	38	6	11
Micro-average (types)	72	26	38
Micro-average (roles)	88	6	12

The manually annotated evaluation data set consists of 57 match reports,[22] covering 2132 event instances. Tables 3 and 4 show detailed results of the evaluation on event types for the test set and correspondingly for the development set,[23] whereas Table 5 shows evaluation results on the test set for extraction of roles (attributes) for selected event types (top 10 of Table 3).

Precision and recall for a class c are defined as in Sebastiani (2002):

$$P_c = \frac{tp_c}{tp_c + fp_c}$$

$$R_c = \frac{tp_c}{tp_c + fn_c}$$

where tp_c are the true positives, i.e. the correct extractions, fp_c are the wrong extractions and fn_c are the (missed) extractions.

Note that, as shown for events in Tables 3 and 4, the distribution of event instances (and role instances) is very uneven. This is why a simple macro-average over the extraction results for the different event types is insufficient. Therefore, besides reporting macro-averaged results, we also computed micro-averaged results, thus taking into account the number of event instances for each event type as shown in Table 6. In particular, micro- and macro-averaged precision and recall are also defined in line with Sebastiani (2002):

$$P_{\text{macro}} := \frac{\sum_{c \in C} P_c}{|C|} \tag{1}$$

$$R_{\text{macro}} := \frac{\sum_{c \in C} R_c}{|C|} \tag{2}$$

$$P_{\text{micro}} := \frac{\sum_{c \in C} tp_c}{\sum_{c \in C} (tp_c + fp_c)} \tag{3}$$

$$R_{\text{micro}} := \frac{\sum_{c \in C} tp_c}{\sum_{c \in C} (tp_c + fn_c)} \tag{4}$$

where C is the set of classes and $|C|$ is the total number of classes.

The F-Measure is then the harmonic mean between precision and recall (macro-averaged or microaveraged), i.e.:

$$F_X = \frac{2 P_X R_X}{P_X + R_X} \tag{5}$$

where $X \in \{macro, micro\}$.

Micro-average results show that precision on both types and roles across event types is relatively high. On the other hand, recall is very low. Although improvement is clearly needed, we nevertheless regard this result as acceptable in an automatic setting in which precision is much more important than recall. The system proposes with relatively high precision only those instances for which it has enough information.

A comparison with evaluation results by other systems evaluated with respect to standard data sets such as provided by the ACE program[24] or the MUC conferences (Grishman and Sundheim, 1996; Hirschmann, 1998) is difficult as different types and numbers of classes are used. Moreover, our system is focused on a specific domain, in contrast to the domain-independent nature of the ACE program. Nevertheless, the results yielded by our system are clearly encouraging.

[22]For an example match report see: http://www.fifa.com/worldcup/archive/germany2006/news/newsid=25460.html; last access 06.07.2008.

[23]Events, such as Hattrick, which are not covered by the development data set can still be recognized in the test data set because of a basic rule that maps predicates in a sentence to lexically corresponding event types.

[24]http://www.nist.gov/speech/tests/ace/; last access on 06.07.2008.

Table 7
Results of the evaluation of discourse rules

Relation	Correct	Intra-sentential	Incorrect	Spurious extraction	Acc1 (%)	Acc2 (%)
Prepares	48	9	25	10	65.75	76.19
Result	83	55	56	35	59.71	79.81
Elaboration	43	3	46	28	48.31	70.49

6.3. Evaluation of discourse analysis

In this subsection, we present the results of the evaluation of our discourse analysis component. There were 78 rules available in the system to infer *prepares*, *result* and *elaboration* relations as discussed in Section 5.4. The rules were developed on a training set consisting of 20 texts reporting qualifiers for the 2006 WorldCup and were evaluated on an evaluation set consisting of 50 match reports from the 2006 WorldCup corpus. Overall, the total number of events in the evaluation set was 2041 and the rules fired 301 times. As the evaluation set merely consists of unannotated textual data, correctness was assessed by manual ("*a posteriori*") validation, i.e. we scrutinized manually whether the relations delivered by our discourse analysis component were indeed appropriate or not. Table 7 summarizes the results of our evaluation. It shows the number of discourse rules that fired *correctly* and *incorrectly*. For the correct cases, it also indicates the number of cases in which the relation was intra-sentential. The column labeled with *spurious extraction* denotes the cases in which an erroneous discourse relation was inferred due to spurious events delivered by the event extraction module. Distinguishing these cases allows us to calculate two types of accuracy, i.e. Acc(uracy)1, which also penalizes the systems for erroneous discourse relations inferred due to errors in the event extraction, and Acc(uracy)2, which disregards these cases. Accuracy2 is thus much higher than Accuracy1. Overall, the results of our discourse processing module are quite satisfactory, ranging between 70% and 80% correct answers.

7. Application within the SmartWeb system

In the previous sections we have discussed a number of aspects of SOBA, which is used in the SmartWeb system for automatic population of the SWIntO ontology with instances extracted from textual and semi-structured match reports. Here we will briefly discuss the application of the SOBA-generated knowledge base in answering world cup related questions.

The SOBA knowledge base consists of more than 100k entities that cover matches, players, countries, etc. as well as events such as "getting a red card". In the following, we give a few example questions which show how the information extracted by the SOBA system can be used for the purpose of question answering. We give the natural language question as well as the corresponding translation into a formal query to the knowledge base in F-Logic. While the translation is performed automatically by the SmartWeb system, it is not a focus of this paper. Let us consider the following question:

(15) Welche Spieler haben eine Karte gesehen?
('Which players were carded?')

```
FORALL Text, Showing, Player, Name, Role, Denom <- Text[media#talksAbout ->
Showing:sportevent#ShowingCard[sportevent#committedOn ->
Player[sportevent#hasUpperRole ->
Role[sportevent#impersonatedBy ->
Denom[externalRepresentation@(de) -> Name]]]]].
orderedby Text, Name
```

The knowledge base contains answer objects for this question that were derived from semi-structured data but also a number of answers that were extracted from textual match reports, e.g.:

(16) Luis Valencia sieht für eine rüde Grätsche gegen Ballack im Mittelfeld die Gelbe Karte.
('Luis Valencia receives for a rude attack against Ballack in the midfield a yellow card')
(17) Doch nachdem Jean-Paul Abalo später die Rote Karte gesehen hatte...
('But after Jean-Paul Abalo later got a red card...')

A number of other questions can only be answered on the basis of information extracted from text, e.g.

(18) Welche Tore wurden durch einen Freistoss vorbereitet?

('Which goals were prepared through a freekick?')

```
FORALL Text, Prep, Goal <- Text[media#talksAbout ->
Prep:sportevent#FreeKick[smartsumo#prepares -> Goal:"sportevent#ScoreGoal]].
orderedby Text, Goal
```

The knowledge base contains answer objects for this question that were extracted from text passages like this one:

(19) Nachdem ihm drei Minuten zuvor der Treffer noch verweigert worden war, knackte Henry den Abwehrriegel schliesslich doch. Zidane schlug einen Freistoss in den Strafraum, und der Stürmer von Arsenal stürmte unbewacht zum langen Pfosten vor und bugsierte den Ball volley ins Netz.

('After he had not been able to score three minutes before, Henry broke the defense line after all. Zidane shot a freekick into the penalty area, and the Arsenal striker ran unopposed to the long post and shot the ball in the net.')

Finally, as a byproduct of extracting information from image captions, pictures are annotated and stored in the knowledge base. The following query asking for pictures depicting a "Save"-event indeed returns 7 pictures of save situations for the world championship 2006:

(20) Zeige mir Bilder von Saves.

('Show me pictures of saves.')

```
FORALL Picture, Save <- Picture[media#shows -> Save] AND Save:sportevent#Save.
orderedby Picture, Save
```

The pictures resulting from this query are shown in Fig. 1. Indeed, this question corresponds to the example we have used in order to motivate our approach in the introduction. For a more detailed end-to-end user evaluation of the SmartWeb system, the interested reader is referred to Mögele and Schiel (2007).

8. Related work

In this section, we discuss work related to ontology-based information extraction and knowledge base generation, integration of deep and shallow linguistic analysis as well as information fusion.

The case for ontology-based information extraction has been most clearly stated in Nirenburg and Raskin (2004). In fact, from a natural language understanding point of view, the value of extracting information without a formal semantics as specified by an ontology is unclear, i.e. extracted information needs to have a formal interpretation in order to allow for meaningful postprocessing. The framework of ontological semantics also allows to exploit background knowledge and reasoning for deep semantic interpretation. Our approach is in line with ontological semantics in the sense that it relies on a given ontology to formalize the meaning of extracted information. However, our focus has been on the integration of information extracted from various sources and on building tight interfaces between linguistic processing and the background ontology.

A comparable ontology-based information extraction system is the Artequakt system (Alani et al., 2003), which is similar to SOBA with respect to the aim of extracting as much information as possible about an entity (a specific painter in the case of Artequakt) from relevant web pages. The information extracted is then formalized with respect to an artist ontology and used to generate a user-tailored summary of the biography of the artist in question. Besides focusing on information extraction and generation aspects, Artequakt has also focused on *information consolidation*, which consists in detecting and resolving inconsistencies and merging information. For example, the system would detect, using some heuristics, that "Rembrandt" and "Rembrandt van Rijn" are the same person and would attempt at merging the information extracted from both.

Also related are approaches to information fusion as used in knowledge engineering. Hunter and Liu (2006) and Hunter and Summerton (2006), for example, have been concerned with the development of a framework for defining fusion rules specifying how conflicting values coming from different resources can be aggregated. Though we have not focused on information fusion in this sense, it would be an important aspect of future work to include a component for the fusion of different values in case of conflicts. The fusion capabilities of our system are essentially restricted to recognizing whether a certain entity is already present in the knowledge base relying on strict matching of certain key properties. Thus, if the values of these properties are specified using orthographic variations, the system will not recognize the duplicate information but assume that the entities in question are incompatible. This strict matching could be enhanced by relying on string distance metrics determining the similarity of different strings. Two strings could then be regarded as equivalent if

their distance is below a certain threshold. A crucial question is then which string similarity measure and threshold to choose. We have not pursued this question in our current work and therefore it constitutes an interesting option to explore in future work. A similar problem of identifying mentions of the same entity within and across documents is currently tackled in the context of the ACE Program.[25] However, our treatment of the problem is different in that we merge entities on the basis of their properties as modeled in the knowledge base rather than on the basis of textual information only. While the problems are definitely related, they are tackled at different levels from a conceptual point of view. Our system is not able to detect logical inconsistencies due to the fact that the languages we consider (RDFS and F-Logic) are not expressive enough to produce any conflicts. Moving to a more expressive language such as OWL would allow for inconsistencies and thus would require some strategies for resolving them (compare Haase and Qi, 2007).

A key feature of our approach is that it makes use of integrated shallow and deep linguistic analysis methods. A number of methods have been proposed for the integration of "deep" and "shallow" grammar processing models in so-called *hybrid NLP architectures* that combine the robustness of shallow processing tools with the higher precision and fine-grainedness of deep linguistic analysis (cf. Crysmann et al., 2002; Frank et al., 2003, 2004). While these earlier proposals tried to integrate "lower-level" information provided by robust shallow processing tools to improve the robustness and coverage of "deeper" analysis systems, the architecture realized in SOBA is designed to work in the opposite direction. It relies on the formalism and methods of the SProUT extraction system, and allocates additional higher-level information about syntactic analysis in a very focused way. The advantages of this model are that it fully retains the robustness, coverage and efficiency of the shallow system, and allows integration of alternative parsers in a very modular way.

Concerning the application of wrapper-like techniques, there is much related work on automatically inducing wrappers given some structured training data (e.g. Muslea et al., 2001). Freitag and Kushmerick (2000) show how wrapper induction methods can be extended to unrestricted textual data. It would have been interesting to apply such techniques to automatically generate our wrappers, but automatic generation has not been our focus of research. In recent years, the information extraction community made considerable progress in the field of *adaptive information extraction* (cf. Turmo et al., 2006, for an overview). Initial work from this area, aiming at the automatic induction of extraction rules from an annotated training data set, is reported in Ciravegna (2001). More recent research (e.g. McLernon and Kushmerick, 2006; Surdeanu et al., 2006) is geared towards semi-supervised methods, i.e. using small amounts of annotated training data as seeds for the acquisition of extraction patterns. While the effort spent on developing a rule-based extraction system as presented in this paper is high (the definition of SProUT grammar rules took roughly 6 person months, discourse analysis rule definition took roughly 1 person week), the effort of building a suitable training corpus should not be underestimated. The important aspect certainly is who is expected to customize the system to a certain domain: an engineer (possibly with NLP background) or an end user. While an end user can be possibly expected to provide annotated data, he/she can for sure not be expected to write extraction or discourse rules. Assuming that an engineer is supposed to customize the system, an approach as presented in this paper certainly allows much more tuning and control over the system than adaptive approaches with less supervision. In future work we aim at developing learning methods for the semi-automatic acquisition of argument structure-based extraction rules and the induction of argument-to-role mappings, using a limited set of general extraction rule types.

Finally, in our approach, we have essentially treated images as black boxes insofar as captions are used to approximate the content of the image and annotate it. Quite recently, several approaches have emerged to extract higher-level semantic features (in contrast to color, texture, etc. features) from images (compare Papadopoulos et al., 2006; Petridis et al., 2006). Such approaches are still under research, but they could straightforwardly extend our approach as information would not merely be extracted from the image captions, but the images proper. This would require an extension of our system in order to treat semantic annotations of images as one more source of information to be integrated into the knowledge base by our consolidation component.

9. Conclusion

Advanced question answering functionality which allows answering questions that require inferencing, counting or aggregation seems only feasible in restricted domains where background knowledge can be modeled and factual knowledge can be acquired with a reasonable degree of completeness. We have presented a system, SOBA, which acquires factual knowledge for a certain domain on the basis of a given ontology. SOBA has been applied in the context of the SmartWeb system for the automatic acquisition of a relatively complete body of factual knowledge about football which can be used as a basis for advanced question answering in the football domain.

Original aspects of SOBA include at least the following. First, SOBA extracts information from heterogeneous resources such as tabular reports, plain text and images with captions. In order to consolidate the knowledge derived from these different information sources with recourse to an ontology, a flexible approach based on declaratively described rules has

[25] http://www.nist.gov/speech/tests/ace/

been developed for the transformation and integration of structures produced by the linguistic analysis components into ontology-conform knowledge structures. Second, SOBA combines shallow and deep natural language processing: wrappers, finite-state technology and deep parsing are combined in a seamless way within one integrated architecture. Third, we have presented a method for detecting duplicate information based on querying the knowledge base in order to avoid that information is inserted twice into the knowledge base. This is crucial considering that one of our goals is to support the answering of aggregate questions involving counting of entities. Finally, we have shown that while the recall of SOBA on the football data set could be definitely higher, the precision is reasonably high to allow for the system to be used in an automatic fashion for knowledge base generation (i.e. population of the underlying ontology).

We are not aware of any system which combines all of these above mentioned aspects into one system. While our system adopts relatively simple (but principled) solutions in some cases, we strongly believe that SOBA can be regarded as a first blueprint of a system for extraction and ontology-based integration of information from heterogeneous sources.

Acknowledgements

This research has been supported by Grant 01 IMD01 of the German Ministry of Education and Research (BMB + F) for the SmartWeb project as well as by the European Commission as part of the Information Society Technologies (IST) programme under EC Grant number IST-FP6-026978 (project X-Media) and the DFG-funded project Multipla. We would like to thank our (current and former) student assistants Thomas Eigner, Günter Ladwig, Matthias Mantel, Alexander Schutz, Atsuko Shimada, Kathrin Spreyer, Tuvshintur Tserendorj, Corinna Weber, Nicolas Weber and Honggang Zhu for preparing and evaluating the data and for implementing parts of the system. Anette Frank was affiliated with DFKI GmbH, Saarbrücken when she contributed to the research described in this paper.

References

Alani, H., Kim, S., Millard, D., Weal, M., Hall, W., Lewis, P., Shadbolt, N., 2003. Automatic ontology-based knowledge extraction from web documents. IEEE Intelligent Systems 18 (1), 14–21.

Asher, N., Lascarides, A., 2003. Logics of Conversation. Cambridge University Press, Cambridge, MA.

Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G., 2002. The TIGER treebank. In: Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol, Bulgaria.

Brickley, D., Guha, R., 2004. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, February. Available at: ⟨http://www.w3.org/TR/rdf-schema/⟩.

Buitelaar, P., Sintek, M., Kiesel, M., 2006. A lexicon model for multilingual/multimedia ontologies. In: Proceedings of the 3rd European Semantic Web Conference (ESWC06), Budva, Montenegro.

Buitelaar, P., Declerck, T., Frank, A., Racioppa, S., Kiesel, M., Sintek, M., Engel, R., Romanelli, M., Sonntag, D., Loos, B., Micelli, V., Porzel, R., Cimiano, P., 2006. Linginfo: design and applications of a model for the integration of linguistic information in ontologies. In: Proceedings of the OntoLex Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies, collocated with LREC 2006, Genoa, Italy.

Buitelaar, P., Declerck, T., Nemrava, J., Sadlier, D., 2008. Cross-media semantic indexing in the soccer domain. In: Proceedings of the 6th International Workshop on Content-Based Multimedia Indexing (CBMI), London, UK, June 2008.

Callmeier, U., Eisele, A., Schäfer, U., Siegel, M., 2004. The DeepThought core architecture framework. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC) 2004, Lisbon, Portugal, pp. 1205–1208.

Ciravegna, F., 2001. Adaptive information extraction from text by rule induction and generalization. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1251–1256.

Crysmann, B., Frank, A., Kiefer, B., Müller, S., Neumann, G., Piskorski, J., Schäfer, U., Siegel, M., Uszkoreit, U., Xu, F., Becker, M., Krieger, H.-U., 2002. An integrated architecture for deep and shallow processing. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Pittsburgh, PA.

Decker, S., Erdmann, M., Fensel, D., Studer, R., 1999. Ontobroker: ontology based access to distributed and semi-structured information. In: Database Semantics: Semantic Issues in Multimedia Systems. Kluwer, Dordrecht, pp. 351–369.

Drozdzynski, W., Krieger, H.-U., Piskorski, J., Schäfer, U., Xu, F., 2004. Shallow processing with unification and typed feature structures—foundations and applications. Künstliche Intelligenz (1), 17–23.

Dubey, A., 2005. What to do when lexicalisation fails: parsing German with suffix analysis and smoothing. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, MI, USA.

Frank, A., Becker, M., Crysmann, B., Kiefer, B., Schäfer, U., 2003. Integrated shallow and deep parsing: TopP meets HPSG. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan, pp. 104–111.

Frank, A., Spreyer, K., Drozdzynski, W., Krieger, H.-U., Schäfer, U., 2004. Constraint-based RMRS construction from shallow grammars. In: Proceedings of the HPSG Conference, Workshop on Semantics in Grammar Engineering. CSLI Publications, CA, pp. 397–417.

Freitag, F., Kushmerick, N., 2000. Boosted wrapper induction. In: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), pp. 577–583.

Grishman, R., Sundheim, B., 1996. Message understanding conference 6: a brief history. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING).

Gruber, T., 1993. Toward principles for the design of ontologies used for knowledge sharing. In: Formal Analysis in Conceptual Analysis and Knowledge Representation. Kluwer, Dordrecht.

Haase, P., Qi, G., 2007. An analysis of approaches to resolving inconsistencies in DL-based ontologies. In: Proceedings of the International Workshop on Ontology Dynamics (IWOD'07).

Hartung, M., 2006. Die Ausnutzung von Diskurswissen zum Zwecke der Informationsextraktion: Zur Gewinnung impliziter Information aus Texten. Magisterarbeit, Universität Heidelberg.

Hirschmann, L., 1998. The evolution of evaluation: lessons from the message understanding conferences. Computer Speech and Language 12 (4), 281–305.

Hunter, A., Liu, W., 2006. Fusion rules for merging uncertain information. Information Fusion 7 (1), 97–134.

Hunter, A., Summerton, R., 2006. A knowledge-based approach to merging information. Knowledge-based Systems 19, 647–674.

Kifer, M., Lausen, G., Wu, J., 1995. Logical foundations of object-oriented and frame-based languages. Journal of the ACM 42, 741–843.

Krieger, H.-U., Schäfer, U., 1994. TDL—a type description language for constraint-based grammars. In: Proceedings of the 15th International Conference on Computational Linguistics (COLING), pp. 893–899.

Krieger, H.-U., Drozdzynski, W., Piskorski, J., Schäfer, U., Xu, F., 2004. A bag of useful techniques for unification-based finite-state transducers. In: Proceedings of the 7th Conference on Natural Language Processing (KONVENS), pp. 105–112.

McLernon, B., Kushmerick, N., 2006. Transductive pattern learning for information extraction. In: Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006).

Mögele, H., Schiel, F., 2007. Summative evaluation of the SmartWeb prototype 1.0. SmartWeb Technical Document 12, September 2007.

Muslea, I., Minton, S., Knoblock, C., 2001. Hierarchical wrapper induction for semistructured information sources. Journal of Autonomous Agents and Multi-Agent Systems 4, 93–114.

Neumann, G., Sacaleanu, B., 2005. Experiments on cross-linguality and question-type driven strategy selection for open-domain question answering. In: Proceedings of the 6th Workshop of the Cross-Language Evaluation Forum (CLEF'05), pp. 429–438.

Nirenburg, S., Raskin, V., 2004. Ontological Semantics. MIT Press, Cambridge, MA.

Noy, N.F., Klein, M.C.A., 2004. Ontology evolution: not the same as schema evolution. Knowledge Information Systems 6 (4), 428–440.

Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Schmidt, C., Weiten, M., Loos, B., Porzel, R., Zorn, H.-P., Micelli, M., Sintek, M., Kiesel, M., Mougouie, B., Vembu, S., Baumann, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Burkhardt, F., Zhou, J., 2007. DOLCE ergo SUMO: on foundational and domain models in SWIntO (SmartWeb Integrated Ontology). Journal of Web Semantics 5 (3), 156–174.

Papadopoulos, G.T., Mylonas, P., Mezaris, V., Avrithis, Y., Kompatsiaris, I., 2006. Knowledge-assisted image analysis based on context and spatial optimization. International Journal on Semantic Web and Information Systems 2 (3), 17–36.

Petridis, K., Bloehdorn, S., Saathoff, C., Simou, N., Dasiopoulou, S., Tzouvaras, V., Handschuh, S., Avrithis, Y., Kompatsiaris, I., Staab, S., 2006. Knowledge representation and semantic annotation of multimedia content. IEEE Proceedings on Vision Image and Signal Processing 153 (3), 255–262.

Reithinger, N., Herzog, G., Blocher, A., 2007. Smartweb—mobile broadband access to the semantic web. Künstliche Intelligenz (KI) Mai (2), 30–33.

Schäfer, U., Beck, D., 2006. Automatic testing and evaluation of multilingual language technology resources and components. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC).

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Computing Surveys 34, 1–47.

Strzalkowski, T., Harabagiu, S. (Eds.), 2006. Advances in Open Domain Question Answering. Text, Speech and Language Technology, vol. 32, Springer, Berlin.

Surdeanu, M., Turmo, J., Ageno, A., 2006. A hybrid approach for the acquisition of information extraction patterns. In: Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006).

Turmo, J., Ageno, A., Catala, N., 2006. Adaptive information extraction. ACM Computing Surveys 38 (2), 1–47.