A set of tools for integrating linguistic and non-linguistic information

Thierry Declerck[1]

Abstract. In this position paper we describe the actual state of the development of an integrated set of tools (called SCHUG) for language processing supporting interaction with disparate sources of information, making thus Natural Language Processing (NLP) and Human Language Technology (HLT) even more relevant for Information Technology (IT) applications. The set of tools is realizing the communication with non language-based devices and services via XML machine readable annotations. Non-linguistic information, in most of the cases domain-specific knowledge, can thus be straightforward included in the linguistically analysed texts, and so contribute to a knowledge markup of textual documents. The basic language technology guiding this markup is Information Extraction (IE) and the added information can be made visible by means of automatic hyperlinking and visualization techniques.

1 Introduction

In this paper we describe the actual state of the development of an integrated set of tools (called SCHUG) for language processing supporting the interaction with various sources of information, making thus Natural Language Processing (NLP) and Human Language Technology (HLT) even more relevant for Information Technology (IT) applications. The set of tools is realizing the communication with non language-based resources, devices and services and their integration into textual documents via XML machine readable annotations and protocols, the standard underlying all Web services. It is thus important that all information providing devices deliver an XML output, or at least have an output format that can be easily transformed into XML. Ontologies providing an hierarchical description of domain specific knowledge are very good candidates for interacting with natural language processing tools – as Information Extraction tasks have already shown, since ontologies can be easily described in XML based representation languages and mapped onto XML encoded results of linguistic analyses.

2 The Chunk Parser used: SCHUG

SCHUG (Shallow and CHunk-based Unification Grammar tools) has been designed in such a way that it can read results from various language processing tools (at any level of NL processing up to the detection of Grammatical Functions) and transform those into an XML document conforming to our basic (shallow) linguistic DTD[2], which is shown below:

```
<?xml version="1.0" encoding="iso-8859-1"
   standalone="yes"?>
<!ELEMENT NE ( W+ ) >
<!ATTLIST NE SUBTYPE NMTOKEN #REQUIRED >
<!ATTLIST NE TYPE ( 1 | 12 | 2 | 3 | 4 | 5 | 6 | 8 )
   #REQUIRED >
<!ELEMENT NP ( NE | W )* >
<!ATTLIST NP TYPE NMTOKEN #FIXED "1" >
<!ELEMENT PARAGRAPH ( NE | NP | PP | VG | W | SC )* >
<!ELEMENT SC (NE | NP | PP | VG | W )+  >
<!ELEMENT PP ( NE | W )* >
<!ATTLIST PP TYPE NMTOKEN #FIXED "2" >
<!ELEMENT SPPC_XML ( PARAGRAPH+ ) >
<!ELEMENT VG ( W+ ) >
<!ATTLIST VG TYPE NMTOKEN #FIXED "3" >
<!ELEMENT W ( #PCDATA ) >
<!ATTLIST W COMP CDATA #IMPLIED >
<!ATTLIST W INFL CDATA #IMPLIED >
<!ATTLIST W POS CDATA #IMPLIED >
<!ATTLIST W STEM CDATA #IMPLIED >
<!ATTLIST W TC NMTOKEN #REQUIRED >x
```

This simple DTD just states that the basic linguistic analysis of a document will deliver a tree consisting of an arbitrary number of paragraphs, each containing an arbitrary combination of single words (W) nominal and prepositional phrases (NP, PP), verbgoups (VG, being a list of verbs), Named Entities (NE, being persons, companies, date and time expressions etc.) and subclauses (SC, being defined for W, NE, NP, PP and VG). NP, PP, NE and VG contain at least one word. The element word is associated with a list of attributes: COMP (result of compound analysis), INFL (information about the inflectional properties of the word), STEM (the lemma of the word) and POS (the syntactic category of the verb). Examples are given below. For the time being the SC element is defined quite sloppily and doesn't state that the subclause should consist of a least one word (subordination or coordination word) and one verbgroup. This condition, which is valid for German, might be too specific at this place. This DTD can be extended for the purpose of more detailed linguistic analyses or for specialized applications.

The internal grammar machinery of SCHUG first maps the XML structure of the available shallow linguistic analysis onto a generic feature structure, which is reflecting the original XML tree annotation of the document. And appropriated rules (defined by regular patterns over annotations) can then activated within the (shallow) unification formalism used in SCHUG for the further processing of the linguistic data. An advantage of this strategy is that it allows us on the one hand to use well-defined unification and subsumption operations on the linguistic data, and on the other hand to use the unification algorithm for integrating available non-linguistic, which can be put in relation with the linguistically annotated terms. The feature structure is internally realized as an hash table, which is also offering the advantage of efficient random access.

[1] DFKI GmbH, 66123 Saarbruecken, Germany

[2] This DTD, SPPC_DTD, has been designed for the SPPC system (see [7]), whose results are further processed by SCHUG.

For German texts the basic XML structure is delivered by the SPPC (Shallow Processing Production Center) System, which is performing tokenisation, morphological analysis, POS tagging, Named Entities detection and analysis of basic chunks (see [7]). Chunks can be defined as the non-recursive components of basic phrases, like NPs or PPs.[3] An example of this mapping is given below, where the XML encoding of a PP ("Fuer die Angaben" – *for the data*) as delivered by the SPPC system is mapped into a feature structure of SCHUG[4].

```
<PARAGRAPH>
<S>
 <PP TYPE="2">
  <W TC="22" POS="23" STEM="fuer"
     INFL="[102]">Fuer</W>
  <W TC="21" POS="7" STEM="d-det"
     INFL="[2 5 20 6 13 23 9 16]">die</W>
  <W TC="22" POS="1" STEM="angabe"
     INFL="[6 7 8 9]">Angaben</W>
 </PP>
...
                The SPPC XML structure
```

```
para[0] => {
  frag[0] => {
    TAG = PP
    SENT = BOS
    sub_frags => {
        item[0] => {
          STRG = Fuer
            features => {
            INFL = [102]
            STEM = fuer
            POS = 23
            TC = 22
          }
        }
        item[1] => {
          STRG = die
          features => {
            INFL = [2 5 20 6 13 23 9 16]
            STEM = d-det
            POS = 7
            TC = 21
          }
        }
        item[2] => {
            STRG = Angaben
              features => {
              INFL = [6 7 8 9]
              STEM = angabe
              POS = 1
              TC = 22
            }
         }
      }
                The SCHUG feature strucutre
```

On the base of this feature structure SCHUG then applies in a cascaded manner various Natural Language operations, if they are needed, pos tagging, named entity recognition, chunking, detection of Grammatical Functions and reference resolution for free text. We are adopting here the general model of cascaded chunk processing, a it was defined by [2], proposing solutions at the levels of processing where enough information is available for generating correct linguistic structures. Furthermore we also include a "retagging" procedure: the XML results of some systems we want to integrate contain sometimes (regular) errors, which can be corrected by SCHUG, avoiding thus the cumulative propagation of wrong linguistic annotations. The retagging procedure is also enriching the annotations provided by the underlying systems. So in the case of the SPPC example given above, SCHUG enriches the analysis with missing information, like for example dependence structure or the result of agreement check:

```
para[0] => {
  frag[0] => {
    NP_HEAD = Angaben
    TAG = PP
    SENT = BOS
    PP_AGR = [102]
    STRUK = 23_7_1
    NP_SPEC = die
    STRING = Fuer die Angaben
    NP_AGR = [6 9]
    PP_NP_AGR = [9]
    TYPE = 2
    NP_SPEC_AGR = [6 9]
    PP_HEAD = Fuer
    sub_frags => {
        item[0] => {
          STRG = Fuer
            features => {
            INFL = [102]
            STEM = fuer
            POS = 23
            TC = 22
          }
        }
     }
...
            The enriched SCHUG feature strucutre
```

In the enriched feature structure above, the reader can see that SCHUG has added to the mother node of the PP constituent information about so-called "head_complement" and "head_modifier" structure, introducing thus a dependence structure into the shallow analysis. The head of the PP is the preposition "Fuer", whereas the head of the NP complement is "Angaben". Also an agreement check has been performed, and the results are given in additional features. It is important to have these additional agreement features for supporting in further processing steps the detection of Grammatical Functions (Subject, direct or indirect Object etc.) and for the resolution of references. It should be noted that the detection of grammatical functions is a very important step towards the attachment of semantic or extra-linguistic information to texts. The detection of grammatical functions offers some guidance in deciding if some information should be attached at the place where certain terms are occuring in texts: one might decide to attach external information only if the terms are in the subject position of a sentence, or if the sentence is not in a passive mode etc. The resolution of references (pronominal, anaphorical, ellipses) is also important since it gives more evidence for integrating non-linguistic information: in case an anaphor like "she" or "he" can be resolved to a referential expression, the system will get more evidence that the document is about a specific topic.

SCHUG is actually processing two languages: German and Spanish, where the use of Spanish is for the time being limited to the base chunks NP, PP and verbgroups.

At the end of the processing SCHUG delivers all the resulting information again in XML, providing thus an increased amount of annotations for the original documents.

[3] For more details on chunk parsing, see [1].

[4] For reason of processing efficiency, some values are encoded as a figure, so for example the POS "Prep" is encoded as '23' and inflectional properties of the words are encoded as lists of figures, each representing an instantiated feature structure over relevant morphological propoerties, like GENDER, NUMBER, CASE. We don't go into more details here.

3 The integration of external information in the textual documents

At the various levels of linguistic processing (cascades) or at the end of this process, external non-linguistic information or code can be added to (unified with) the linguistic description, supporting thus a scalable integration of disparate information sources (i.e. domain ontologies, multimedia material or program codes for automatic hyperlinking) into the Natural Language Processing chain. The well known procedures acting on feature structures, unification and subsumption, allow a descriptive mapping between (for example) domain ontologies and the results of NL processing.[5] The resulting feature structure is mapped back into an (enriched) XML structure and so available for further processing. Some of the the added annotations can be used as "semantic" index for a content-based search. Alternatively, one can add the relevant nodes (or some local paths) of the ontology that have been detected as relevant for the text into the Metadata list associated with the document, extending thus the core Metadata to a contentful one, which can be easily scanned by search engiene, facilitating thus the constitution of the Semantic Web. So NL processing guide the detection and presentation of additional and associated information and knowledge, which might be available at some other places in a net of information and present it in a XML structure. So for example once in a document an occurence of a proper noun is found, a search can be started within other documents (structured or not), extract relevant information about the entity refered to by the proper noun and present it in a structured way to the reader. The technology responsible for this is often called *automatic hyperlinking* and is central in the context of document enrichment. This technolgoy also helps in order to incrementally create specialized database on entities or events. One unique document can be enriched (annotated) by different types of annotations, depending for example on the underlying terminology, thesaurus etc.

The integration of (domain-specific) knowledge during the NL processing can improve the results of the linguistic analysis, since decision about syntactic disambiguation and attachment of linguistic chunks can in certain cases be supported by non-linguistic information.

4 An example of an application: the MUMIS project

The design and the ongoing implementation of SCHUG has been done initially for supporting the information extration (IE) task in the context of the EU project MUMIS decicated to the indexing of Multimedia material[6].

MUMIS develops and integrates basic technologies for the automatic indexing of multimedia programme material. The domain of application is soccer. Various technology components operating offline are generating formal annotations of events in the data material processed. These formal annotations (in XML) constitute the basis for the integral online part of the MUMIS project, consisting of a user interface allowing the querying of videos. The indexing of the video material with relevant events is done along the line of time codes extracted from the various documents.

For this purpose the project makes use of data from different media sources (textual documents, radio and television broadcasts) to build a specialized set of lexicons and an ontology for the selected domain (soccer). All are available in XML and are integrated into the IE processing components. It also digitizes non-text data and applies speech recognition techniques to extract text for the purpose of annotation.

The core linguistic processing for the annotation of the multimedia material consists of advanced information extraction techniques for identifying, collecting and normalizing significant text elements (such as the names of players in a team, goals scored, time points or sequences etc.) which are critical for the appropriate annotation of the multimedia material in the case of soccer.

Due to the fact that the project is accessing and processing distinct media in distinct languages, there is a need for a novel type of merging tool in order to combine the semantically related annotations generated from those different data sources, and to detect inconsistencies and/or redundancies within the combined annotations. The merged annotations (in XML) are stored in a database, where they are combined with relevant metadata.

Actually we are investigating how domain-specific annotations, gained on the base of the merging of linguistic and domain-specific knowledge, can be included in the MPEG-7 standard, using for this the slot foreseen for "Textual Annotation". The main issue of this investigation will be to check to which extent textual annotations can be combined with low-level video features in order to achieve better content indexing (and searching) of video material.

5 Integration of various types of documents for an incremental IE

As we have seen above, MUMIS makes uses of various types of sources for the generation of content annotations. MUMIS also operates a distinction within the textual documents it consults, and applies different processing techniques in dependence of the type of textual document:

1. Reports from Newspapers (reports about specific games, general reports) which is classified as free texts
2. Tickers, close captions, Action-Databases which are classified as semi-formal texts
3. Formal descriptions about specific games which are classified as formal texts

Since the information contained in formal texts can be considered as a database of true facts, they play an important role within MUMIS. But nevertheless they contain only few information about a game: the goals, the substitutions and some other few events (penalties, yellow and red cards). So there are only few time points available for indexing videos. Semi-formal texts, like live tickers on the web, are offering much more time points sequences, related with a higher diversity of events (goals scenes, fouls etc,) and seem to offer the best textual source for our purposes. Nevertheless the quality of the texts of online tickers is often quite poor. Free texts, like newspapers articles, have a high quality but the extraction of time points and their associated events in text is more difficult. Those texts also offer more background information which might be interesting for the users (age of the players, the clubs they are normally playing for,

[5] The mapping procedure between linguistic features and non-linguistic knowledge can very probably be executed within the sole frame of XML and associated semantic represetentational languages, but the use of feature structures for our purposes has shown various advantages, being the higher level of declarativity for the description of mapping (unification) rules and the higher efficiency (random access) in accessing sub-structures of the results of the linguistic analysis.

[6] MUMIS is an on-going EU-funded project within the Information Society Program (IST, number 1999-10651) of the European Union, section Human Language Technology (HLT). See for more information http://parlevink.cs.utwente.nl/projects/mumis/

etc.). Figures 1 and 2 show examples of 2 (German) formal texts on one and the same game, and 4 gives an example of a semi-formal text on the same game.

England - Deutschland 1:0 (0:0)

England: Seaman (2,5) - G. Neville (3,5), Keown (3), Campbell (2), P. Neville (4,5) - Ince (3,5), Wise (5) - Beckham (4), Scholes (3) - Shearer (3), Owen (5) - Trainer: Keegan

Deutschland: Kahn (2) - Matthaeus (3) - Babbel (3,5), Nowotny (2,5) - Deisler (3), Hamann (2,5), Jeremies (3,5), Ziege (3,5) - Scholl (5) - Jancker (4), Kirsten (5) - Trainer: Ribbeck

Eingewechselt: 61. Gerrard fuer Owen, 72. Barmby fuer Scholes - 70. Rink fuer Kirsten, 72. Ballack fuer Deisler, 78. Bode fuer Jeremies

Tore: 1:0 Shearer (53., Kopfball, Vorarbeit Beckham)

Schiedsrichter: Collina, Pierluigi (Viareggio), Note 2 - bis auf eine falsche Abseits-Entscheidung souveraen und sicher

Zuschauer: 30000 (ausverkauft)

Gelbe Karten: Beckham - Babbel, Jeremies

Figure 1. Example of a so-called formal text, where one can see that only 5 distinct time points can be extracted, concerning the player subsitutions ("Eingewechselt") and one goal ("Tore").

Aufstellungen:

England: 1 Seaman (Arsenal London/36 Jahre/59 Laenderspiele) - 2 Gary Neville (Manchester United/25/38), 6 Keown (Arsenal London/33/32), 4 Campbell (Tottenham Hotspur/25/35), 3 Phil Neville (Manchester United/23/28) - 7 Beckham (Manchester United/25/33), 14 Ince (FC Middlesbrough/32/52), 8 Scholes (Manchester United/25/26), 17 Wise (FC Chelsea London/33/18) - 9 Shearer (Newcastle United/29/62), 10 Owen (FC Liverpool/21/21) Deutschland: 1 Kahn (Bayern Muenchen/31 Jahre/26) - 2 Babbel (Bayern Muenchen/27/51), 10 Matthaeus (New York Metro Stars/39/149), 6 Nowotny (Bayer Leverkusen/26/21) - 18 Deisler (Hertha BSC/20/5), 14 Hamann (FC Liverpool/26/26), 16 Jeremies (Bayern Muenchen/26/26), 17 Ziege (FC Middlesbrough/28/52) - 7 Scholl (Bayern Muenchen/29/28) - 19 Jancker (Bayern Muenchen/25/8), 9 Kirsten (Bayer Leverkusen/34/50/49 fuer die DDR) Schiedsrichter: Collina (Italien)

Figure 2. A second example of a so-called formal text, where one can see that different informatin providers give distinct information: here for example the number of games for the national team.

Since the formal texts require only few linguistic analysis, but rather an accurate domain-specific interpretation of the symbols used, a module has been defined within SCHUG, which in a first step maps the formal texts onto a XML annotation[7], giving the domain semantic of the expressions in the text. In a second step SCHUG *merges* all the XML annotated formal texts about one game. Figure 3 shows a part of such merged annotations:

```
<TEAM>
  <NAME>Deutschland</NAME>
    <TRAINER>
      <TEAM_FUNCTION>#Trainer</TEAM_FUNCTION>
      <TRAINER_NAME>#Ribbeck</TRAINER_NAME>
    </TRAINER>
    <PLAYERS>
      <PLAYER>
      <PLAYER_NAME>Kahn</PLAYER_NAME>
      <PLAYER_NOTE>#(2)</PLAYER_NOTE>
      <PLAYER_POSITION>1</PLAYER_POSITION>
      <PLAYER_NUMBER>##1</PLAYER_NUMBER>
      <PLAYER_OLD>##31</PLAYER_OLD>
      <PLAYER_CLUB>##Bayern Muenchen</PLAYER_CLUB>
      <PLAYER_NO_PLAYS>##26</PLAYER_NO_PLAYS>

...

<REFEREE_INFORMATION>
   <REFEREE_NAME>#Collina, Pierluigi
        ##Collina</REFEREE_NAME>
   <REFEREE_ORIGIN>#Viareggio
        ##Italien</REFEREE_ORIGIN>
   <REFEREE_NOTE>#2</REFEREE_NOTE>
   ...
   </REFEREE_INFORMATION>
```

Figure 3. Merged annotations generated from formal texts. Information extracted from the first text is marked with "#", from the second text with "##". No special marker is provided if both texts give the same information.

Those merged annotations are generated at a level that requires only few linguistic analysis, and reflect basically domain specific information about actors and events involved in the text. The SCHUG module applied at this level also extracts metadata information: name of the game, date and time of the game, intermediate and final scores etc. This is quite inmportant, since the metadata will guide the use of the annotations produced so far for supportig linguistic analysis and Information Extraction applied to more complex document, like the ticker shown in 4. Let us take as an example the line beginning with the time code "16." The word "Ziege" can be interpreted as being a soccer player on the base of the available annotations generated from the formal texts. Without this, the default reading (*goat*) would have been selected. The other soccer terms like "flankt", "Freistoss" etc. are getting interpreted on the base of a multilingual soccer thesaurus semi-automatically developed within the MUMIS consortium. In this thesaurus terms in three distinct languages, Dutch, English are German are put in relation with soccer concepts. So "flankt" is put into relation with the concept "cross". With the help of those document external information, partially dynamically generated, the line starting with the time code "16." in figure 4, for example, can be successfully analysed and following event annotations can be generated:

```
2-event_1_PLAYER = Ziege
1-event_LOC = Goal-line::Goal-area
1-event_1_PLAYER = Scholes
3-event_EVENT_CLASS = goal_scene_fail
3-event_TYPE = Save
3-event_TIME = 16:00
2-event_TIME = 16:00
1-event_TIME = 16:00
1-event_TYPE = Cross
DOM = SOCCER
```

But also the already available information about the player "Ziege" (or about the player "Scholes") is made available at this level, mixed with linguistic information:

```
OLD = ##28
TAG = NP
NP_AGR = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 .. 23]
NOTE = #(3,5)
NP_STRUK = 25
NP_STRING = Ziege
OBJ_AGR = [1 3 4 5 7 8 9 11 12 14 15 16 18 19 21 22 23]
POS = #4 ##3
NP_HEAD = Ziege
PLAYER = Ziege
GF = SUBJ/DAT_OBJ/AKK_OBJ/NP_MOD_GEN
SUBJ_AGR = [2 10 17]
NUMBER = ##17
NR_PLAYS = ##52
TEAM = Deutschland
CLUB = ##FC Middlesbrough
TYPE = NP_PLAYER
DOM = SOCCER
```

[7] Following a DTD resulting from the analysis of all available formal texts in our soccer corpus.

This basic information can also be very useful for reference resolution. So for example, if in a sentence it is written “The 28 year old midfield player of Middlesbrough ..”, SCHUG can consult the dynamically generated annotations and then point to “Ziege”. SCHUG is actually also adding to the “Ziege entry” all the events it detects in the semi-formal texts. The updated set of annotations will be of use for the subsequent analysis of free texts.

All the generated (XML) annotations on events, with the information available about the actors involved, are passed to a MUMIS module in charge of integrating text annotations and the video stream of the game, so that this video can be queried on the base of such events and actors, which are also put into relation. The MUMIS searching environment allows queries of the form: “Give me all the goal scenes in the second half of the game, if Ziege is involved.”

Gruppe A: England - Deutschland 1:0 (0:0)
7. Ein Freistoss von Christian Ziege aus 25 Metern geht ueber das Tor.
12. Ziege flankt per Freistoss in den Strafraum und Jeremies versucht es per Kofball, verfehlt den Kasten jedoch deutlich.
16. Scholes flankt gefaehrlich von der Torauslinie in den Fuenfmeterraum, doch Ziege hat aufgepasst und kann klaeren.
18. Hamann versucht es mit einem Distanzschuss aus 20 Metern, aber Seaman ist auf dem Posten.
23. Scholl mit einer Riesenchance: Nach Zuspiel von Hamann rennt er in den englischen Strafraum, wird jedoch gleich von drei Seiten bedraengt und kommt nur zu einem unplazierten Schuss, den Seaman sicher abfangen kann.
27. Jancker spielt auf Ziege, dessen Schuss von der Strafraumgrenze kann von Seaman abgefangen werden.
35. Michael Owen kommt nach Flanke von Philip Neville voellig frei vor dem deutschen Tor zum Kopfball, doch Kahn kann zum ersten Mal sein Koennen unter Beweis stellen und rettet auf der Linie.
43. Kahn zum zweiten: Beckham flankt auf Scholes, der zieht ab in den rechten Winkel, aber der deutsche Keeper verhindert erneut die englische Fuehrung.
47. Christian Zieges Freistoss aus 20 Metern geht einen halben Meter ueber das Tor.
53. Beckham flankt per Freistoss an der deutschen Abwehr vorbei auf den Kopf von Alan Shearer, der voellig freistehend zum 1:0 fuer die Englaender verwandelt.
58. Scholl wird von Matthaeus bedient, aber sein Schuss geht aus halbrechter Position um Zentimeter am Tor vorbei.
65. Seaman kann nach einem Eckball vor Kirsten klaeren, der Nachschuss von Jancker geht knapp am Tor vorbei. Riesenmoeglichkeit fuer die DFB-Elf.

Figure 4. Example of a so-called semi-formal text, where one can see that here more time points are available, and that those can be complementary to the time points to be extracted from formal texts. So, already at this level, a unification or merging of extracted time can be done.

6 CONCLUSION

We have shown that (shallow) multilingual linguistic procedures can be very helpful for a whole range of IT applications, since it is supporting the integration of various sources of information and Knowledge Markup of textual documents. Within the SCHUG system it is possible to associate non-linguistic information at various levels of linguistic analysis, as required by the application under consideration. The XML representation has proven to be an easy and useful mean for communicating between disparate sources of information. The SCHUG tools can capture related knowledge for a document on the base of robust but accurate NLP and of the ontology driven IE supported by the system. This knowledge is visualized to the reader via the automatic hyperlinking feature included in SCHUG, which also allows to semantically annotate the documents and also to make the underlying conceptual structure visible at any place of the documents.

We will in the future have to look at how to integrate our approach in a general XML architecture or Knowledge Markup editing tools.

ACKNOWLEDGEMENTS

The major part of the work reported here has been done within the context of the EU-funded Project MUMIS (IST-1999-10651). We would like to thank also Mireia Farrus for her help on the spanish grammar, Claudia Crispi and Mihalea Hutanu for their work on the SCHUG platform

REFERENCES

[1] Steven Abney, ‘Parsing by chunks’, in *Principle-Based Parsing*, eds., Steven Abney Robert Berwick and Carol Tenny, Kluver Academic Publishers, (1991).

[2] Steven Abney, ‘Partial parsing via finite-state cascades’, in *Workshop on Robust Parsing, 8th Europen Summer School in Logic, Language and Information (ESSLLI*, (1996).

[3] Doug E. Appelt, ‘An introduction to information extraction’, *AI Communications*, **12**, (1999).

[4] Thierry Declerck and P. Wittenburg, ‘Mumis – a multimedia indexing and searching environment’, in *Proceedings of the 1st International Workshop on MultiMedia Annotation, MMA-2001*, Tokyo, (2001).

[5] ISO/IEC JTC1/SC29/WG11. Mpeg-7 overview. http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm.

[6] MUC, ed. *Seventh Message Understanding Conference (MUC-7)*, http://www.muc.saic.com/, 1998. SAIC Information Extraction.

[7] Jakub Piskorski and G. Neumann, ‘An intelligent text extraction and navigation system’, in *Proceedings of the 6th Conference on Recherche d’Information Assistée par Ordinateur, RIAO-2000*, (2000).