XML and NLP: Their interaction and their role for HLT Applications

Declerck Thierry and Peter Wittenburg
DFKI GmbH and Max Planck Institute for Psycholinguistics,
Stuhlsatzenhausweg 3, D-66123 Saarbruecken, Germany and
Wundtlaan 1,PB 310, 6500 AH Nijmegen, The Netherlands
declerck@dfki.de and Peter.Wittenburg@mpi.nl

Abstract

We describe in this paper some of the interactions NLP and XML have or can possibly have and show the role such interaction can play for Human Language Technology (HLT) applications. One of the example being the MUMIS Project (Multimedia Indexing and Searching Environment)[1] where linguistically motivated XML annotations, coupled with domain-specific information, are used for the indexing and the searching of multimedia content. Closely related to this is the automatic hyper-linking of information in text, also making HLT interesting for Knowledge Management in general. As a background for the whole presentation, we will at a first place describe more generally the use of Standard Markup languages for deep or shallow linguistic analysis.

1 Markup Languages for deep linguistic analysis

In this section we describe experiences done in the context of the LSGRAM (Large Scale Grammars) project, a project funded by the EU between 1994 and 1996. A central motivation of LSGRAM was the investigation of the reusability of linguistic resources and the development of language analysis components for 9 languages, using for this the ALEP platform (see (Schmidt et al. (1996))). ALEP is an open and generic environment, providing a usable, programmable framework for the integration of various language engineering (LE) components into a powerful NLP environment. ALEP was foreseen for the development of high-level (typed) unification-based grammars, supporting the multiple inheritance of linguistic information.

Typical problems with such systems for deep analysis lie in their lack of efficiency and robustness. The design of the ALEP platform takes this fact in consideration and foresees a module (the Text Handling to Linguistic Description module[2]) for including the result of various pre-processing steps into the high-level parsing machinery. This module also supports the description of the lexical component, extending it to expressions not belonging to the classical HPSG or LFG (or other linguistic formalisms) lexical descriptions (the so-called “messy details”). LSGRAM integrated the pre-processing treatment of those phenomena within the general NLP processing chain and could improve both the efficiency and the robustness of the parsing component, while considerably extending its coverage. Typical examples of “messy details” are numbers, codes or other (sequences of) word-forms (so-called “fixed phrases”) which can occur in many variations, making impossible a comprehensive treatment by traditional linguistic means. Word level phenomena are usually the most frequently occurring messy details, including such things as dates,

[1] MUMIS is an EU-funded project within the Information Society Programme (IST) of the European Commission, section Human Language Technology (HLT).

[2] For more details, see (Declerck (1997)).

numbers and proper names (see (Bredenkamp et al. (1996))). For any realistic NLP application these types of construct must be processed efficiently, the alternative being coding them individually in some lexicon and/or implementing sets of grammar rules for parsing them syntactically.

As already mentioned, the ALEP platform provides the user with a Text Handling (TH) component which allows a textual preprocessing of inputs. Input texts will first go through a SGML-based tagging of their elements. The default setup of the system defines the following processing chain: the text is first converted to an EDIF (Eurotra Document Interchange Format) format. Then a subsequent recognition process is provided: paragraph recognition, sentence recognition and word recognition. The output of those processes consists in the tagging of the recognized elements: 'P' for paragraphs, 'S' for sentences, 'W' for words (in case of morphological analysis, the tag 'M' is provided for morphemes) and 'PT' for punctuation signs. Some specialized features are also provided for the tagged words, allowing to characterize them more precisely, so for example 'ACRO' for acronyms or 'D' for digits and so on. So the simple input "John loves Mary." after being processed by the default TH component will take the form:

```
<P>
 <S>
  <W>John</W>
  <W>loves</W>
  <W>Mary</W>
   <PT>.</PT>
 </S>
</P>
```

The (normalized) output of the TH component for the word level will have following format:

```
... <W TYPE="WORD" CASE="FIRST">john</W>
    <W TYPE="WORD" CASE="NO">loves</W>
    <W TYPE="WORD" CASE="FIRST">mary</W>
...
```

This structure will be mapped onto the linguistic description of the system, thus associating the 'W' textual information with a specific linguistic information. The syntax of the mapping rules is as follows:

```
tsls_rule( <ld>,
  <tag_name>, [<features>], <tag_content> )
```

where:
`<ld>` is a (partial) Linguistic Description (LD). `<tag_name>` is the name of a sgml tag (e.g. `<S>`, `<W>`). `<features>` is a list of feature-value descriptions of the tag's features. `<tag_content>` is the atomic content of the string within the tag (optional).
This kind of mapping allows the flow of information between text structures and linguistic descriptions. At this place we integrated POS information resulting from a tagger, annotated in SGML (but as we can see, exactly the same format as XML in this case), and investigated how this step can support the efficiency and robustness of the subsequent linguistic parsing steps. The following sentence "Große Bereiche der Dasa leiden unter dem Rückgang des einst lukrativen Rüstungsgeschäfts. (Large areas of the DASA are suffering from the decline of the once lucrative arms trade.) will be tagged in the following way:

```
<W TYPE="WORD" CASE="FIRST"
   POS="ADJ">grosse</W>
<W TYPE="WORD" CASE="FIRST"
   POS="N">bereiche</W>
<W TYPE="WORD" CASE="NO"
    POS="DEF">der</W>
<W TYPE="NE" CASE="FIRST">dasa</W>
...
```

Extending slightly the mapping rules shown above, it is possible to map the POS enriched SGML tags into the linguistic descriptions of the grammar module of the parsing component, thus keeping the parser free of process intensive tasks (for unification-based formalisms) like agreement check, POS disambiguation etc. Basically: the parser does not have to calculate the categorial information of the context of the words at parse time, since it is already given by the (efficient) preprocessing.

Concerning the messy details in general, we mapped those into so-called "generic entries", which are generic "lexical" descriptions of set of expressions (like currency expressions, or Named Entities). With this step it was possible to considerably extend the lexical coverage of the HPSG based grammars.

The gain of efficiency was considerable, as can be seen in the following figures, considering two types of parser used in the ALEP platform, the input being the sentence "Grosse Bereiche der Dasa leiden unter dem Rueckgang des einst lukrativen Ruestungsgeschaefts." (see above):

```
Results for grammar without POS information
   basic parser (backtracking):
     total      BaWordSeg BaLift    BaAna
     40.550     0.190     0.020     40.340
   record parser (subsumption of analyses):
     total      RWordSeg  RLiftAna
     6.260      0.200     6.060

Results for grammar with POS information
   basic parser (backtracking):
     total      BaWordSeg BaLift    BaAna
     13.820     0.190     0.020     13.610
   record parser (subsumption of analyses):
     total      RWordSeg  RLiftAna
     4.860      0.190     4.670
```

On the left side the total processing time is given. The processes involved are morphological segmentation (Seg), Lifting of information from text processing to linguistic description (Lift) and syntactic parsing (Ana). In the case of the record parser, Lifting and Analysis are coupled (LiftAna). The results of these experiments demonstrated that SGML/XML annotations can support and improve high-level NL analysis. Moreover it has been shown that Markup languages are ideal (since simple and well understood) for putting in relations different types of NL processing. So we could easily extend the Text Handling to Linguistic Description interface of the ALEP platform to an interface between Shallow Linguistic Processing to Deep Linguistic Processing.

2 Markup Languages for shallow processing

More recently, in the context of the MUMIS project (described below), concerned with Information Extraction (IE), we took the XML output of a finite state toolkit[3] for shallow NL processing, including tokenization, morphology, POS tagging, named entity recognition and chunking, as the base of more complex syntactic and semantic processing. An example of the XML annotation delivered by the SPPC toolkit for shallow processing, applied to the example given above, is given below:


```
<?xml version="1.0" standalone="yes"?>
<SPPC_XML>
<PARAGRAPH>
<S>
<NP TYPE="1">
  <W TC="22" POS="5" STEM="gross"
     INFL="[26 27 28 29 30 31 32 33 34 35
           36 37 38 39 40]">Grosse</W>
  <W TC="22" POS="1" STEM="bereich"
     INFL="[18 20 21 23]">Bereiche</W>
</NP>
<W TC="21" POS="7" STEM="d-det"
   INFL="[17 3 4 21 7 14]">der</W>
<W TC="22">Dasa</W>
<VG TYPE="3">
  <W TC="21" POS="5 2 1 1"
     STEM="leid leid leid leiden"
     INFL="[41 42 43 ... 82 83 84]
           [92 93 94 95 96 97 98 99 100 101]
           [13 14 15 16]
           [10 11 12 ... 15 16]">leiden</W>
</VG>
<PP TYPE="2">
  <W TC="21" POS="1 5 17 16 23"
     STEM="unter unter unter unter unter"
     INFL="[17 18 19 20 21 23] [25] [] []
           [24 102]">unter</W>
  <W TC="21" POS="7" STEM="d-det"
     INFL="[18 11]">dem</W>
  <W TC="22" POS="1" STEM="rueckgang"
     INFL="[17 18 19]">Rueckgang</W>
</PP>
<NP TYPE="1">
  <W TC="21" POS="7" STEM="d-det"
     INFL="[1 0]">des</W>
  <W TC="21" POS="17" STEM="einst">einst</W>
  <W TC="21" POS="5" STEM="lukrativ"
     INFL="[41 42 43 44 45 46 47 48
            ...49 83 84]">lukrativen</W>
  <W TC="22" POS="1"
     COMP="ruestung s geschaefts"
     STEM="geschaeft"
     INFL="[291]">Ruestungsgeschaefts</W>
</NP>
<W TC="1" POS="21" INFL="[440]">.</W>
</S>
</PARAGRAPH>
</SPPC_XML>
```


The XML tags are giving information about the status of the fragments of the analyzed text: <P> for paragraphs, <S> for sentences, <W> for words, <NP>, <PP> and <VG> for linguistic chunks (corresponding to Noun Phrase, Prepositional Phrase and Verb groups). Within the word tags additional features are encoding further information: "TC" stays for the tokenizer, all other

[3]This toolkit is called SPPC and has been developed and implemented at DFKI in Saarbrücken. For more details, see (Piskorski and Neumann (2000)).

features are related to morpho-syntax information, if available. Most of the values of the word level features are encoded in figures for reason of efficiency. We do not go into details here. On the base of this example, one can see the potential degree of ambiguity of the German language.

Contrary to the approach described in section 1, we do not map the XML annotated text into the linguistic descriptions of a linguistic formalism for further processing, but define the more complex syntactic and semantic rules directly on the base of the (shallow) linguistic information delivered by the XML annotations, accessing all possible information encoded there. In order to do this more efficiently, we first map the annotated document onto an internal and genereric feature structure (implemented as a hash table) and define more complex linguistic operations (which eventually can not be described straightforward by means of finite state machineries) on this hash. The system that has been implemented (in Perl) for supporting those processes is called SCHUG (Shallow and CHunk based Unification Grammar tools). Within this system we implement rules for specifying linguistic fragments attachment, for the disambiguation of fragments, for the detection of grammatical functions or for the resolution of references[4].

On the top of this it is then quite straightforward to define a mapping between the linguistic information and some domain specific knowledge, merging thus XML annotated linguistic data and XML encoded domain ontologies and knowledge. This is providing a clean and efficient support for the Information Extraction task as such, and can as well provide for some content annotations that can be used for enriching the annotation of corpora, supporting eventually corpus linguistics work[5].

To conclude and summarize this section: we are here dealing with a grammar implementation that is acting on XML annotated documents, if we abstract from the fact that we first map the XML document onto a feature structure (hash) for reason of processing efficiency (this mapping is in any case structure preserving). So it seems that XML can be a precious representation formalism not only for language resources, but also for the definition of syntactic and semantic analyses. A nice property of such an approach is that it allows to integrate already at a low level of the NL processing all kind of available relevant information encoded in XML and to fusion this information with linguistic information. One application of this approach is described in the next section.

3 The MUMIS Project

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

[4]This special task has been described in more details in (Declerck and Neumann (2001)).

[5]This issue has been discussed in more details in (Declerck and Neumann (2000)).

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 for filtering the queries in the database of videos.[6]

3.1 Generation of Formal Annotations

The formal annotations are reflecting the typical output of the IE systems being used in MUMIS, i.e. instantiated domain-specific templates. The slots to be filled by the systems are basically entities (player, teams etc.), relations (player_of, opponents etc.) and events (goal, substitution etc.), which are all derived from a domain ontology. All the templates associated with an event are including a time slot to be filled if the corresponding information is available in a least one of the sources consulted during the IE procedure. Those templates are realized as XML encoded event tables.

The IE systems include a lookup procedure in the domain lexicon (the corresponding monolingual part of it). The systems will have to apply to distinct sources and so there is a need for checking consistency in the IE result on distinct sources about the same event (game): this is the task of the merging tool mentioned above.

The IE system does not extract only information on events but also on *metadata*. Those are information concerning the naming of the game, the teams involved, the place, the date, the document source, the type of document etc.

At the end of the process, only those formal annotation will be delivered which are relevant for the purpose of querying the videos, but one has to be aware of that all the other information used and provided by the IE systems is also at some place available and can be used for annotating as well textual documents with linguistic and domain-specific information.

As an example consider the following lines starting a "ticker"-like report on a game of the European Championship 2000:

```
Gruppe A: England - Deutschland 1:0 (0:0)

7. Ein Freistoss von Christian Ziege
   aus 25 Metern geht ueber das Tor.
....
```

The first line is naming the game, giving also information about the intermediate and the final score. The second line[7] is a typical example of a ticker. First a time code and then a single sentence describing an important event from the point of view of the journalist. Our IE system is extracting both the metadata information and the event relation and maps those onto our annotation scheme, which is a instantiated event table in the case of the event:

```
<NP TYPE="GROUP_OPPONENTS_SCORE"
    DOM="SOCCER" GROUP = "A"
    TEAM_2 = "Deutschland" TEAM_1 = "England"
    TEXT_TYPE = "ticker" TEXT_SOURCE = "Kicker"
    FINAL_SCORE = "1:0" HALF_TIME_SCORE = "0:0"
</NP>

<EVENT_1>
  TIME = "7:00" PLAYER = "Christian Ziege"
  DISTANCE = "25" PRED = "geh(1)"
  EVENT_CLASS = "goal_scene_fail(1)"
  EVENT = "freekick(1)"
</EVENT_1>
....
```

The reader can see how parts of the original texts are mapped onto domain-specific slots. This is done on the base of shallow (but detailed) linguistic analysis, the use of a domain specific lexicon including a specialized "subcategorization frame" for the verbs (actually pointing not only to arguments but also to all possible modifications of the verb in the context of the specific domain), and a domain ontology. The definition of the latest allows the mapping of the predicate-argument-structure of the sentence not only

[6] The project has been described in more details in (Declerck et. al (2001)).

[7] A word by word translation could be something like: "A freekick by Christian Ziege from 25 meter goes above the goal".

to the event “freekick”, but also to an event-class located higher in the domain hierarchy: “goal_scene_fail”. So in the subsequent (ontology-based) query to the video material, it is possible to discriminate the search for successful goals from goal scene in general.

3.2 User Interface Building

The user first will interact with a web-portal[8] to start a MUMIS query session. An applet will be down-line loaded in case of showing the MUMIS demonstration. This applet mainly offers a query interface. The user then will enter a query that either refers to metadata, formal annotations, or both. The MUMIS on-line system will search for all formal annotations that meet the criteria of the query. In doing so it will find the appropriate meta-information and/or moments in some media recording. In case of meta-information it will simply offer the information in scrollable text widgets. This will be done in a structured way such that different type of information can easily be detected by the user. In case that scenes of games are the result of queries about formal annotations the user interface will first present selected video keyframes as thumbnails with a direct indication of the corresponding metadata.

The user can then ask for more metadata about the corresponding game or for more media data. It has still to be decided within the project whether several layers of media data zooming in and out are useful to satisfy the user or whether the step directly to the corresponding video fragment is offered. All can be invoked by simple user interactions such as clicking on the presented screen object. Playing the media means playing the video and corresponding audio fragment in streaming mode requested from a media server.

3.3 Standards for Multimedia Content

MUMIS is looking for a compliance with existing standards in the context of the processing of multimedia content on the computer and will adhere to the emerging standard MPEG7, which is about defining standards for annotations which can be seen as multimedia objects or events[9]. MPEG-7 seems to be quite promising, since on the one side it supports XML annotations and on the other side it explicitly foresees the integration of textual annotation (in XML format). MUMIS will explore the use of XML in the context of MPEG-7 for interrelating distinct kinds of information: linguistic, domain knowledge and features derived from the image processing.

4 Automatic Hyperlinking

As we saw above, the intensive use of standard markup languages in the context of NLP allows on the one side to both improve NL processing and to integrate other kind of knowledge into the NL processing chain. On the other side, linguistic information can guide the detection and presentation of additional and associated information and knowledge, which might be available at some other places in an information network and present it in a XML structure. So for example once in a document an occurrence 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 very valuable for all kind of Knowledge Management systems and applications.

This functionality has been successfully integrated into the SCHUG system (briefly described in section 2). The search in the information network can be guided by a thesaurus that models the specific knowledge. The linguistic component of SCHUG is in charge of detecting all relevant terms (including their variants) in documents. Pointers to associated information are added to the detected terms in the texts, and presented to the reader in a structured menu. The user can

[8] A stand alone prototype of the demonstrator is already available.

[9] For more details on the MPEG-7 standard, see the good overview in (Salambier P. (2001)).

click on menu items for opening external windows or URLs. For the time being we have been adding to the terms of the texts only JavaScript code for presenting the added information in the document to the reader. We are currently working on mapping these information structures onto the XML Linking Language.

5 Conclusions

In our everyday work as computational linguist or language engineer, we have been often experiencing the fruitful relationship between NLP technologies and the SGML/XML Markup languages. Promising developments are to be found in the domain of Multimedia and Knowledge Management. XML annotated information and knowledge can be integrated quite straightforward into the NL processing chain. And linguistic analysis can help in associating subset of (XML structured) information and knowledge bases in the processed documents.

References

BIM/SEMA, *ALEP System Documentation.* 1993. CEC Luxembourg.

Simpkins, N.K., Groenendijk, M., Cruickshank G. *ALEP–1 User Guide.* 1993. CEC Luxembourg.

Alshawi H., Arnold D. J., Backofen R., Carter D. M., Lindop J., Netter K., Pulman S., Tsuji J., Uszkoreit H. 1991. *Eurotra ET6/1: Rule Formalism and Virtual Machine Design Study (final report).* CEC Luxembourg.

Bredenkamp A., Declerck T., Fouvry F., Music B. 1996. *Efficient Integrated Tagging of Word Constructs.* COLING 96. pp. 1028–1031.

Declerck T. 1997. *An Interface between Text Structure and Linguistic Description.* DALF 97. Kolding, Denmark.

Declerck Thierry and G. Neumann 2000. *Using a parameterisable and domain-adaptive information extraction system for annotating large-scale corpora?.* Proceedings of the Workshop Information Extraction meets Corpus Linguistics, LREC-2000. Athens, Greece.

Declerck Thierry and G. Neumann 2001. *A Distributed Shallow Approach to Reference Resolution in the context of Information Extraction.* RANLP-2001 Tzigov Chark, Bulgaria.

Declerck Thierry, P. Wittenburg and H. Cunningham 2001. *The Automatic Generation of Formal Annotations in a Multimedia Indexing and Searching Environment.* Proceedings of the 2001 ACL/EACL Workshop on Human Language Technology and Knowledge Management. Toulouse, France.

Piskorski, Jakub and G. Neumann 2000. *An Intelligent Text Extraction and Navigation System.* RIAO 2000. Paris.

Schmidt P., Rieder S., Theofilidis A., Declerck T. 1996. *Lean formalisms, linguistic Theory and Applications. Grammar Development in ALEP.* COLING 96. pp. 286–291.

Salembier Philippe 2001. *An overview of MPEG-7 multimedia description schemes and of future visual information analysis challenges for content-based indexing.* CBMI 2001. Brescia, Italy.

Sperberg-McQueen C.M., Burnard L. (editors) 1994. *Guidelines for Electronic Text Encoding and Exchange (TEI P3).* Chicago, Oxford.