

Language Technology for eLearning - a multilingual approach from the German perspective

Eelco Mossel**, Lothar Lemnitzer*, Cristina Vertan**

* University of Tübingen

Wilhelmstr. 19, 72074 Tübingen, {lothar}@sfs.uni-tuebingen.de

** University of Hamburg, Natural Language Systems group
Vogt-Köln-Str. 30, 22527 Hamburg, {mossel,vertan}@informatik.uni-hamburg.de

1 Introduction

With the increasing importance of eLearning in education, a substantial number of Learning Management Systems (LMS) have been developed. Until now, such systems have focused on providing communication between LMS users, and management of learning objects (LOs) and their metadata, based on manual annotation of the corresponding files. As the manual annotation process is not trivial and time-consuming, and therefore often neglected, part of the potential of an LMS remains unexploited. An example is the annotation of the LOs with keywords. The lack of information about the LOs has a negative impact on the search and retrieval of content in a LMS.

Language Technology techniques provide promising solutions both for the automation of the annotation process as well as for the search of LOs. However, up to now little research has been done in this direction. The project LT4eL¹ (Language Technology for eLearning) aims to fill this gap, and to demonstrate, on the basis of a prototype implementation, the benefits of NLP technology for eLearning systems. In the project, a keyword extractor is used for the automatic detection of keywords for the learning objects. The keyword extractor relies on automatic linguistic annotation of learning objects. A second important aspect which we are investigating is the automatic detection of definitory contexts for domain specific terms. This process relies on the linguistic annotation of the LOs as well as on a grammar developed in the project. The search of LOs is supported in the LT4eL project by an ontology which is built from the extracted keywords.

The learning objects are also annotated with semantic information (parts of the ontology) and as a consequence, the search process can deliver not only the relevant LOs but also information about interaction between the LOs. The LT4eL project places the eLearning scenario in a multilingual context, as we are working with LOs in 9 languages from three different language families. This has

¹ The LT4eL project is supported by the European Community under the Information Society and Media Directorate, Learning and Cultural Heritage Unit.

consequences for the keyword extractor and glossary candidate detector as well as for the retrieval of LOs. In order to enable cross-lingual retrieval, terminological lexicons in all 9 languages are mapped onto a domain-ontology.

In this paper we present the main results achieved so far. We will focus on the linguistic annotation of the German learning objects and on the results of the glossary candidate detector.

As one result that is available for further research, we present a corpus of German learning objects which has been annotated with linguistic information. The domain covered by the LOs is computer science for non-computer-science specialists. The annotation is recorded in an XML format developed for the purposes of the project. The format of the LOs and of the annotations as well as the process of linguistic annotation for the German language is explained in section 2. Section 3 describes the keyword extractor and the glossary candidate detector. The approach used for both tools is supervised learning. Section 4 presents the evaluation methodology and results for the tools. Finally in section 5 we explain further work to be performed in the project as well as expected results.

2 Creation of the German corpus

2.1 Data flow

In the context of our application, a learning management system, we have to deal with learning objects in various formats, including DOC, RTF, PDF and HTML, in a wide variety of languages. At the end of the processing chain we get linguistically annotated documents including those features of the layout which are relevant for further processing.

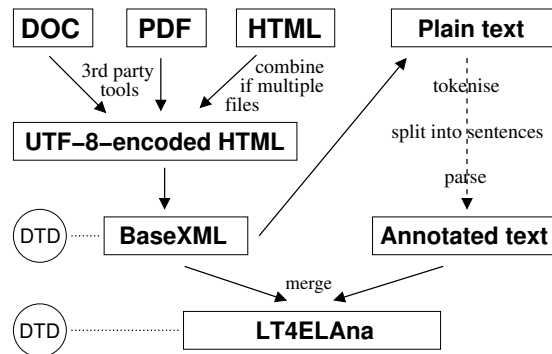


Fig. 1. Data flow for the processing of learning objects

The processing of the German documents, displayed in Figure 1, therefore comprises:

- Conversion of the original documents of all formats to tidy HTML. In this step, we are using third party tools;
- Removal of layout information which is not needed in our application, and conversion of the reduced HTML text to XML. This format is determined by a DTD called BaseXML, which has been developed within the project;
- Extraction of the stripped text from this XML file;
- Linguistic annotation of the pure text with the WCDG parser (This processing step is of course different for each language);
- Merging of the linguistically enriched text with the layout information which is stored in the BaseXML file.

We arrive at documents in a format called LT4ELAna, which are ready for input into the information processing and extraction tools (see below) .

2.2 The LT4ELAna document format

The LT4ELAna DTD determines the format of the fully annotated files that are used in the project. The DTD has the following features:

- it is derived from the XCESAna DTD, which is a de-facto standard for linguistically annotated corpora. We changed the DTD slightly and added some elements which are specific to the project, namely keywords, defined terms and defining texts;
- the DTD structures the documents into paragraphs, sentences, chunks and tokens, where the concept of a chunk is used in the sense defined by Abney (1991).
- the text of the document is encoded as textual content of the *tok* elements, the layout and linguistic information is encoded as attribute-value pairs.

2.3 An example

Conversion to BaseXML The following is an example fragment of an HTML file after converting it from PDF² and cleaning the output.

```
<div class="T0C"><font size="+1">
<ol><li>
<p lang="de-DE" class="western" style="margin-bottom: 0cm">
Schreiben Sie eine <b>E-Mail</b>.</p></li></ol></font></div>
```

Our HTML-BaseXML converter removes the elements and attributes which are not required and converts the HTML into clean XML, conforming to our BaseXML DTD³.

```
<ol><li><p> Schreiben Sie eine <b>E-Mail</b>.</p></li></ol>
```

² We used Adobe's online converter, http://www.adobe.com/products/acrobat/access_onlinetools.html

³ The DTD is available at <http://nats-www.informatik.uni-hamburg.de/~mossel/LT4eL/LT4ELBase.dtd>

Linguistic annotation with WCDG In order to add part-of-speech and morphosyntactic information to the text, we analyse it with the open-source WCDG Parser ⁴ using a dependency grammar for unrestricted German text⁵. The analysis output is a stand-off annotation in XML format.

The WCDG system generates a dependency structure for every sentence of the input text. It performs natural language analysis within the paradigm of constraint optimization, where the analysis that best conforms to all rules of the grammar is returned. The dependency structure is a tree, where every token can modify another token. The structure is not binary: a token can be modified by zero, one or more tokens.

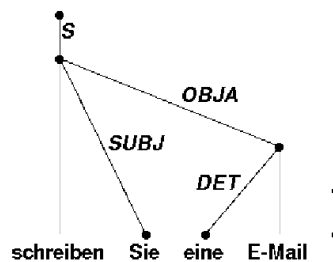


Fig. 2. Graphical representation of a dependency structure

For every token in the dependency structure, its lemma, part-of-speech and various morpho-syntactic features are determined. The values of the part-of-speech feature are names of categories specified by the Stuttgart-Tübingen Tagset (STTS)⁶.

We selected the following frequently occurring features to be included in the linguistically annotated text: number, gender, case, person, degree, tense, stress (occurring with adjectives and verbs) and subcat (occurring with proper nouns). Other less frequently occurring features are removed because they do not help us to constrain our language models (see section 3.1).

We use the following information from the parse result: lemma⁷, POS, morphosyntactic features (see above), simple NPs as inferred from the dependency structure. Note that the results of the linguistic analysis are not corrected manually. We simulate a real-world situation where manual correction is not feasible. Our information extraction tools have to cope with these errors (see section 4).

⁴ cf. Schröder (2002), URL: <http://nats-www.informatik.uni-hamburg.de/view/CDG/WebHome>.

⁵ cf. Foth et al. (2005)

⁶ cf. Schiller et al. (1999).

⁷ To improve the lemmatisation results, we used Helmut Schmid's SFST tools, cf. Schmid (2006).

Automation of the processing steps for linguistic annotation In order to parse entire texts with the CDG system and add the results to the original texts, we wrote a few pre- and post-processing scripts for use in the LT4eL project. They perform the following four steps (see Figure 1):

1. Extract plain text from BaseXML.
2. Tokenise the text, split into sentences, send each sentence to the parser and collect the results.
3. Extract the selected information from the stand-off annotation.

```

<annotation id="s114" lattice="s114" nowords="5" noedges="10">
...
  <arc from="1" to="2" word="Sie">
    <tag case="nom"/>
    <tag cat="PPER"/>
    <tag gender="bot"/>
    <tag number="pl"/>
    <tag person="third"/>
    <dep level="SYN" label="SUBJ" modifiee="1"/>
  </arc>
...

```

Fig. 3. Fragment (the second token) of the stand-off annotation of the sample sentence

The dependency structure is encoded by the *modifiee* attribute of the *dep* element of each token. Every edge from a word to the word it modifies has a label with its role (SUBJ in the example). An NP can include structures like a PP or a relative clause. In view of the use of NP annotation in our project, namely as an aid to recognising possible keywords and definitions, we decided to extract only simple NPs: we take everything that is a (direct or indirect) modifier of a common noun and precedes it as well. Thus, relative clauses or PPs following the noun are not included.

Creating XML files conforming to the LT4eLana DTD First, an inline annotation is created on the basis of the tokeniser output and the stand-off annotation (the input and the output of the parser, respectively). The created inline annotation conforms to the LT4eLana DTD with two exceptions: there are no paragraph elements, and the tokens do not have the *rend* attribute for layout information. Therefore, this file is further modified by matching its textual contents to the text of the BaseXML, and inserting layout information, found at the corresponding places from the BaseXML. The BaseXML elements *p*, *h1*, *h2*, *h3*, *h4*, *h5*, *h6*, *code*, *table* are changed to `<par name="p/h1/h2/h3/h4/h5/h6/code/table">`. Other layout elements, such as `` and `<i>`, are encoded as a comma-separated list, and used as the value of the

rend attribute for the token elements. For instance, in the following example: `<i>two words</i>`, the token *two* will have the attribute `rend="i"`, whereas the token *words* will have the attribute `rend="i,b"`.

The following figure presents our example sentence in LT4ELAna format, which contains linguistic information as well as layout information. The latter is encoded in the `rend` attribute of each token. The format conforms to the LT4ELAna DTD⁸.

```
<par id="p63" name="p"><s id="s114">
  <tok base="schreiben" ctag="VFIN" id="t1091"
    msd="pl,0,0,third,0,present,0,0" rend="ol,li">Schreiben</tok>
  <tok base="Sie" ctag="PPER" id="t1092"
    msd="pl,bot,nom,third,0,0,0,0" rend="ol,li">Sie</tok>
  <chunk category="NP" id="c247">
    <tok base="eine" ctag="ART" id="t1093"
      msd="sg,fem,acc,0,0,0,0,0" rend="ol,li">eine</tok>
    <tok base="E-Mail" ctag="NN" id="t1094"
      msd="sg,fem,bot,third,0,0,0,0" rend="ol,li,b">E-Mail</tok>
  </chunk>
  ...
</s>
...
</par>
```

Fig. 4. LT4ELAna example. Legend: par = paragraph; s = sentence, tok = token; base = lemma; ctag = part of speech; msd = morpho-syntactic description of the word, in the form of a feature vector; rend = layout information.

2.4 Results

In the LT4EL project, we compiled a corpus of 36 LOs, consisting of around 230 000 words in total. Within the files, 1000 keywords and 350 definitions are manually selected and annotated. Most of the texts are free of third-party IPR claims and therefore are freely available for research. Furthermore, the DTDs and annotation manuals which have been developed for these tasks are available.

3 The information extraction tools

3.1 Keyword Extractor

We give a detailed description of the keyword extractor in Lemnitzer and Degórski (2006). We will therefore mention only the most important architectural features

⁸ The DTD is available at <http://nats-www.informatik.uni-hamburg.de/~mossel/LT4eL/LT4ELAnaProject-v32.dtd>

of this tool: a) it recognises single word keywords as well as multiword keywords; b) for the ranking, it uses some metrics based on tf*idf; c) it has a language independent core and language specific extensions (*language models*). The further development of the tool is open to the community, as a SourceForge project⁹.

3.2 Glossary candidate detector

The second information extraction task of the project is the identification of definitions, which are assumed to be glossary candidates. A glossary can be seen as a small lexical resource which supports the reader in understanding the central concepts of a text. It can be built on the definitory contexts which are contained in the learning objects themselves.

Our approach is to define rules which match grammatical patterns that are assumed to be definitions (e.g. *NP ist NP*). The first step is to define language dependent rules, because grammatical patterns for definitory contexts arguably differ from language to language. We are therefore currently developing a grammar for each individual language. In the future, we intend to compare the grammars of all the languages in the project, grouped into their respective language families, with the goal of defining a language independent core and to separate this core from language specific extensions. If this approach is successful, it will be easier to add grammars for new languages of the language families we are now covering (Germanic, Romance, Slavic).

In the literature about extraction of definitory contexts, good results for the rule-based approach have been reported for English¹⁰, Dutch¹¹ and German¹².

We apply the grammars to files in the LT4eLAna format, using the tool *lxtransduce*¹³, an XML transducer intended for use in NLP applications.

Manual annotation and grammar development We follow an empirical and iterative approach. First, the definitions occurring in the LOs were manually selected and annotated. On the basis of the definitions found, we manually write grammar rules. Obviously, if rules were to be generated automatically while allowing the surface text form to be used, a 100% recall and precision could be achieved by overfitting, which is not useful for new LOs. However, when writing the rules manually, we try to generalise and write rules that are not too complicated, yet effective. Full forms of words are used sparingly, only if they are a strong indicator that something is defined, e.g. certain forms of *sein* (to be) or other verbs.

The German grammar for definitions draws heavily on the linguistic annotation of the texts. Much used are sentence starts, NP chunks, part-of-speech, and

⁹ Cf. <http://sourceforge.net/projects/lt4e1/>.

¹⁰ cf. Klavans and Muresan (2001).

¹¹ cf. Fahmi and Bouma (2006).

¹² cf. Storrer and Wellinghoff (2006).

¹³ cf. Tobin (2005).

singular/plural for nouns. We currently have 15 rules and subrules, covering the following patterns of definitions which occur most frequently in our corpus:

- defined term - some form of to be - optional adverb - NP - rest of sentence
- defined term - open-parenthesis - defining text - close-parenthesis
- defined term - one of certain priming words - rest of the sentence

With the grammar, definitory contexts are identified and extracted automatically. These automatically extracted definitions are compared with the manually annotated definitions in the same texts. Based on the results of this comparison, the grammar is refined. An example of how the evaluation result helped us improve an aspect of the grammar is the following. A number of automatically annotated sequences which did not have corresponding manual ones, started with a singular noun without an article, such as: *Voraussetzung ist ...* ('Prerequisite is ...') They were found by the grammar because a single singular noun forms an NP. We could now add the condition that an NP must either start with an article, or must be plural.

To increase recall, new rules will be written to cover more kinds of definitions. Also, some restrictions in existing rules can be loosened, but obviously this can also lead to a drop in precision. To increase precision, existing rules can be adapted to be more specific. If in a later stage of the project the grammars work well for the current LOs (the "training" set), they can be applied to new LOs.

4 Evaluation strategy and results

4.1 The strategy

The evaluation of the tools will proceed in three steps. The first method is an automatic procedure: First, the manually extracted keywords and definitions are matched against the automatically extracted keywords and definitions and a summary score is generated. A qualitative analysis helps us to refine the language models and the grammars.

The second method is based on human evaluation of the quality of automatically extracted keywords and definitions. Test users will rate those words / definitions which have been ranked highest by the tools. Based on their knowledge of the text, the test persons rate the quality of each keyword and definition. This evaluation helps us to assess the quality of the extracted keywords and definitions - which can be acceptable even though they do not match the manually annotated ones, because the annotator had just missed it.

The third method is a step towards a gold standard for keyword evaluation. It is based on the assumption that individuals will widely diverge in their choice of keywords for a certain document, but that there might be a core of keywords on which most annotators of a text would agree. We will therefore let test persons annotate the same text with a limited number of keywords. The keywords which are collected from all annotators will be ranked by the frequency with which they have been selected. The keywords which are chosen by the majority of

annotators will be considered to be a gold standard for that document. This experiment also allows us to measure the average inter-annotator agreement and to check where the agreement between the automatic extractor and any annotator ranks relatively to the average agreement.

In the following, we will focus on the evaluation of the glossary candidate detector.

4.2 Evaluation of the glossary candidate detector

We looked for definitions in all 36 LOs. 358 definitions with a total of 8655 tokens, spread out over 612 sentences (some definitions consist of more than one sentence), were found and annotated manually. The automatic annotation (the result of applying the grammar to the LOs), covers 2111 tokens. 870 tokens were shared by the two annotations. Using the number of annotated tokens as the only quantitative evaluation criterion, this results in a recall of 10% and a precision of 41%. However, this measure gives longer definitions a higher weight. We will develop the evaluation measure further and make it more meaningful by giving the defined term more importance.

At present, all manual definitions that comprise more than one sentence have a negative effect on recall, since the grammar always selects at most one sentence. In the running system, the user will see the extracted definition in a larger context, e.g. with one preceding sentence and one following sentence. Therefore, it is not essential that the full definition is captured, as long as the "core" is captured; a partially matched definition can safely be counted as a successful match.

Causes of not-found definitions (negative influence on recall) There are several reasons why certain definitions are not found by the grammar, for example:

- The search pattern depends on the start of a sentence; some sentence starts are not correctly annotated by the automatic processing.
- Some tokens have incorrect POS tags. For example, the grammar looks for an NP with an article, but the article has been annotated as a cardinal number.
- Tokens that the grammar looks for are not found because of tokenisation problems arising from errors in the input text (e.g. *LAN(Local* should be recognised as three tokens but is recognised as one token, because a space is missing).

Causes of falsely recognised definitions (negative influence on precision) There are several reasons why certain patterns are detected which are not considered to be definitions, for example:

- The first word of a sentence is falsely annotated as an unknown word (e.g. because of an unexpected character, such as a hyphen, in the middle of the

- word) or as a proper noun (e.g. the abbreviation "Ggf."); because of this, it becomes a definition candidate.
- The grammar finds something that can indeed be considered a good definition, it was just not discovered during manual annotation; this does not occur often and is not really harmful. The manual annotation should be adapted in such cases.

5 Conclusions

We outlined the aims and achievements of the LT4eL (Language Technologies for eLearning) project, which will run until mid-2008. Based on a richly annotated corpus, we have developed and implemented two natural language processing tools for information extraction from learning objects. These tools will be integrated into our reference learning management system as well as being provided as web services. Both these tools and the corpus will be made available to the research community. Tool-based and task-based evaluation of the functionalities will be of central importance in the next phase of the project.

The extracted keywords (see section 3.1) are the basis for the domain ontology which will be developed in the second part of the project. The German grammar for the glossary candidate detector (section 3.2) will be extended and compared with the grammars for the other languages involved in the project. The aim of this comparison is the identification of a possible core of language independent rules. We will also investigate the possibility of improving the glossary candidate detector by means of other methods, as presented for example in Storrer and Wellinghoff (2006).

Bibliography

- Abney, S. (1991). Parsing by Chunks. In Berwick, R., Abney, S., and Tenny, C., editors, *Principle-Based Parsing*. Kluwer Academic Publishers, Dordrecht.
- Fahmi, I. and Bouma, G. (2006). Learning to Identify Definitions using Syntactic Features. In *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*.
- Foth, K., Daum, M., and Menzel, W. (2005). Parsing unrestricted German text with Defeasible Constraints. In Christiansen, H., Skadhauge, P. R., and Villadsen, J., editors, *Constraint Solving and Language Processing*, volume 3438 of *Lecture Notes in Artificial Intelligence*, pages 140–157. Springer-Verlag, Berlin.
- Klavans, J. and Muresan, S. (2001). Evaluation of the DEFINDER System for Fully Automatic Glossary Construction. In *Proc. of AMIA Symposium 2001*.
- Lemnitzer, L. and Degórski, Ł. (2006). Language Technology for eLearning – Implementing a Keyword Extractor. Paper presented at the fourth EDEN Research Workshop "Research into online distance education and eLearning. Making the Difference", 25-28 OCTOBER, 2006 in Castelldefels, Spain. <http://www.sfs.uni-tuebingen.de/lothar/publ/LemnitzerDegorskiKeywords.pdf>.
- Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, University of Stuttgart / University of Tübingen.
- Schmid, H. (2006). SFST tools. Website: <http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/SFST.html>.
- Schröder, I. (2002). *Natural Language Parsing with Graded Constraints*. PhD thesis, Dept. of Computer Science, University of Hamburg, Germany.
- Storrer, A. and Wellinghoff, S. (2006). Automated detection and annotation of term definitions in German text corpora. In *Proceedings of LREC 2006*, pages 22–28, Genoa, Italy.
- Tobin, R. (2005). Lxtransduce, a replacement for fsgmatch. Website: <http://www.cogsci.ed.ac.uk/richard/ltxml2/lxtransduce-manual.html>.