



Project no. 027391

Project acronym: LT4eL
Project title: Language Technology for eLearning

Instrument Specific Targeted Research Project

Thematic Priority Information Society Technology

D2.3b Validated glossary candidate detector – first cycle

Due date of deliverable: 30-11-2007
Actual submission date: 21-12-2007

Start date of project: 1-12-2005

Duration: 30 Months

Organisation name of lead contractor for this deliverable: Tübingen University (UTU)

Revision [1]

| Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006) | | |
|--|---|---|
| Dissemination Level | | |
| PU | Public | x |
| PP | Restricted to other programme participants (including the Commission Services) | |
| RE | Restricted to a group specified by the consortium (including the Commission Services) | |
| CO | Confidential, only for members of the consortium (including the Commission Services) | |

Glossary Candidate Detector Deliverable

Contents

- 1 Title
- 2 Summary
- 3 Improvement of the glossary candidate detector
 - 3.1 Divide and conquer
 - 3.2 Setting the baseline
 - 3.3 Improving the pattern grammars
 - 3.4 Adding machine learning to post-process the extracted definitions
- 4 Actions taken in response to user experience
- 5 Documentation
- 6 Scientific papers on the Glossary Candidate Detector

1 Title

D2.3b Validated glossary candidate detector – first cycle

2 Summary

Glossaries can play an important role within eLearning, since they are lexical resources which can support the learner in decoding the learning object he is confronted with and in understanding the central concepts which are being conveyed in the learning material.

In classical text books, read offline, new terms are typically introduced at the beginning of the text or when they are needed first. In eLearning, texts are not often read linearly. They are split into smaller learning objects which can be digested independently of one another. Authors can not longer rely on the learners having read his definition when they first encounter a term. It is therefore necessary to collect the central terms and their definitions in a glossary which accompanies a (set of) learning object(s).

Our glossary candidate detector helps authors to create glossaries by presenting definitions from their learning objects. Authors can accept, reject, or modify the definitory contexts which are presented to them by the glossary candidate detector. The accepted definitions will be written to a glossary which accompanies a learning object and which can be edited by the author at any time.

On the review meeting in January 2007 we presented a prototype of the glossary candidate detector as a standalone tool which demonstrated the basic functionality of the extraction of definitory contexts using early versions of our pattern grammars. The tool was accessible via a simple interface which was useful to demonstrate this basic functionality.

The main activities in the reporting period wrt to the keyword extractor have been the evaluation and the integration of the language technology tools and their improvement in response to evaluation and validation (cf. Technical annex, section 7.2 and milestones for WP2, p. 27; for the evaluation cf. D2.4).

In the reporting period, we made progress in the development of the tool and in particular in its integration in a learning management system and its embedding into the learning process, which is represented by the use of the tool in some usage scenarios. While these

usage scenarios are described in detail in deliverable D5.1b, we will outline in the following the progress we made in developing and integrating the tool. In particular we will describe:

- how we distinguished and classified different types of definitory contexts.
- how we established a baseline for monitoring the development of the tool
- the use of machine learning techniques in a post-processing step to increase the precision of the tool
- the integration of this tool in a learning management system and its embedding into learning processes which reflect the real needs of the users.

Based on the iterative evaluation of the tool and its validation in use scenarios, we are constantly improving the glossary candidate detector. The improvements we made will be the major topic of this deliverable.

There is some interaction between the tool development and the development of resources in WP3. The extracted definitions for relevant keywords have been integrated into the domain ontology (for further details, cf. There is also some interaction between the development and the evaluation and validation of the tools. Evaluation and validation inform the tool development and improvement process. The evaluation of the results showed that the pattern grammars yield an acceptable recall rate for the major types of definitions. The precision, however, is less than optimal. These findings led to the decision to introduce machine learning as a post-processing step in order to improve the precision of the extraction process. We report about machine learning experiments which we performed for Dutch and Polish, and about their results, which are encouraging.

For the evaluation process, cf. deliverable D2.4. For the validation process, cf. deliverable D5.1b.

3 Improvement of the glossary candidate detector

3.1 Divide and conquer

Many learning objects are relatively small in size, thus our approach has not only to favor precision but also recall, that is we want to make sure that all possible definitions present in a text are proposed to the user for the creation of the relevant glossary. Therefore, the extraction of definitions cannot be limited to sentences consisting of a subject, a copular verb and a predicative phrase, as is often the case in question-answering tasks, but a much richer typology of patterns needs to be identified than in current research on definition extraction.

In the first versions of the pattern grammars, we tried to capture all structural kinds of definitory context at once. This proved to be not very effective. However, there is an imbalance between certain kinds of definitory contexts. Some structural types are used for the majority of definitions, while others are much less frequent. Following hints in the literature about definitions, we distinguish six types of definitory context:

- definitions with a copula verb (e.g. *Instant Messaging is a tool of direct communication among the Internet users*) or an other copula element (like *to* in Polish)
- definitions with a verb other than the copula verb (e.g. *Characters are normally represented as strings of seven bits each in an encoding called ASCII (American Standard Code for Information Interchange)*)
- definitions with a punctuation connecting the definiendum and the definiens and no verb (e.g. *CAI -- Computer Aided (Assisted) Instruction - a package for learning in a subject or topic (e.g. mathematics or handling a spreadsheet)*)
- definitions where layout information is used to mark the the definiendum and the definiens, and no verb (e.g. *A CMS: • enables large amounts of information to be dealt with...* -- The full definition is spread over a list of bullet points)

- definitions where the definiendum is represented by a pronoun which refers to an antecedent noun phrase (e.g. *VRML (Virtual Reality Modelling Language) (ISO/IEC 14772) is the current standard. It is a file format for 3D objects and scenes.*)
- other types of definitory context. These can be sentences in which words and phrases like Dutch *ofwel* ("or") and *dat wil zeggen* ("which means") are used as indicator or in which an uncommon connector verb is used. The unclassifiable sentences are relatively often part of a multisentence definitory context.

The manual annotated definitions in the learning objects have been classified following this scheme. The quantitative profiles of the definition sets revealed that, with the exception of German, the vast majority of definitions falls into one of the first two classes (cf. deliverable D2.4 for details). In the German case we assume that the distribution of definitions in our corpus is atypical, also in the light of the findings reported in Storrer & Waldenberger, 2006. We therefore decided to follow the same strategy as the other groups and to optimise the grammars for the first two types of definitions.

Given the variety of definition patterns present in our learning objects, we believe that the rule-based approach is the most appropriate to use to detect them. Previous research has shown that grammars that match the syntactic structures of the definitory contexts are the most successful approaches when deep syntactic and semantic analysis of texts is not available (cf. Liu, Chin & Ng 2003 and Mureasan and Klavans 2002), which is indeed the case in our project, since parsers (and chunkers) are not available for all the languages for which we have developed the glossary candidate detector. Furthermore, experiments with a deep parser for Dutch (Alpino) revealed that deep syntactic analysis does not improve the extraction results significantly.

3.2 Setting the baseline

Most advanced experiments with the pattern-based approach have been conducted for Polish, so these results will be used as an example.

Three baseline grammars have been constructed:

- **B1** marking all sentences as definition sentences
- **B2** marking all sentences containing
 - a possible copula (*jest, są, to*)
 - the abbreviation *tj.* ("i.e.")
 - the word *czyli* ("that is", "namely")
- **B3** marking as definitions all sentences containing any of the 27 very simple patterns (mostly single-token keywords) manually identified on the basis of manually annotated definitions. These include, among others, all patterns in B2

It is arguable that **B3** should still be called a baseline, in fact it is a very simple and permissive grammar, reaching the recall rate of 88% (compared to 59% by the best scoring *real* grammar), of course at the cost of low precision (less than 11%, while the best one reached around 19%).

Other language partners used simpler baselines, for instance the Portuguese baseline marked a sentence as a definition if it contained a verbal form of the verb "to be" in the third person singular or plural of the present or future past or in gerundive or infinitive form.

3.3 Improving the pattern grammars

The XML transducer *ltransduce* developed by language technology group at the University of Edinburgh is used to match the grammar against files in the LT4eLAna format. *Ltransduce* has been developed especially for use in NLP applications. It supplies a format for the development of grammars which are matched against either pure text or

XML documents. The grammars must be XML documents which conform to a DTD. In each grammar, there is one 'main' rule which calls other rules by referring to them. The XPath-based sub-rules are matched against elements in the input document. When a match is found, a corresponding rewrite is done.

The grammars were developed on the basis of the training corpus, in many iterations (e.g. for Polish, more than 100), where in each iteration the grammar was improved and/or the results were automatically evaluated quantitatively and manually evaluated qualitatively. In case of Polish, the finished grammar was additionally tuned in a separate cycle of a few iterations, using the held-out data, to improve performance on this data. The Calimera guidelines were used as the held-out corpus, as this document was slightly different in style and structure than the other learning objects.

Most of the grammars are split into layers; the Dutch and Polish grammars that will be presented as examples have 4 layers, with rules of each layer possibly calling only rules of the same and previous layers.

Bulgarian grammar

The Bulgarian grammar is built as follows:

- the first layer contains rules making reference to particular strings, punctuation, words and classes of words.
- the second layer wraps this, i.e. it matches and marks complete definitory contexts.

The grammar matches is-definitions, verb-defintions, punctuation definitions, definitions with a missing explicit defined term and layout definitions. It contains 62 rules in a 21k file.

Czech grammar

The Czech grammar consists of 148 rules structurally divided into several categories:

- 33 basic rules used for Part-of-Speech labeling as well as classification of non-alpha characters into special classes (punctuation, symbol, quotes, brackets, etc.)
- another 33 rules for construction of phrases from basic word types (noun phrases, prepositional phrases, etc.)
- 57 rules for capturing different types of definiendum (defined object) and definines (defining text).
- 25 top-level rules for single and multi sentence definition extraction

There are 14 different top-level rules for single-sentence and 6 rules for multi-sentence definition extraction. Due to limits of the Ixtransduce tool, input files must by preprocessed by a script in order to enable multi-sentence definition processing.

Dutch grammar

The Dutch grammar is built as follows:

- in the first layer, the part-of-speech information is used to make rules for matching separate words (e.g. verbs, nouns, adverbs).
- the second layer consists of rules to match chunks (e.g. noun phrases, prepositional phrases). A chunker was not used to be able to define the possible patterns of the chunks manually.
- the third layer contains rules for matching and marking the defined terms
- in the last layer the pieces are put together and complete definitory contexts are matched.

In total, this grammar consists of 67 rules (part 1: 24 rules; part 2: 5 rules; part 3: 20 rules and part 4:18 rules) in a 35K file. The rules were made as general as possible to

prevent overfitting to the training corpus.

English grammar

The English grammar is built as follows:

- in the first layer, the part-of-speech or lemma information is used to construct basic rules to capture categories of words (nouns, verbs, etc.)
- the second layer captures phrase structures that were noted as useful for the task
- the third layer places these phrase structures into sequences which would match definitions according to the categories specified and marks the definitory terms
- the final layer calls these different rules and marks the definitional sentence itself.

In total there are 70 rules. At layer 3 there are 7 rules for the `is_def` category, 3 rules for the verb category (containing various verbal options), and 3 rules for the punctuation category. These top layer rules call various other rules at layer 2, which in turn call rules at layer 1. The file is 21KB.

German grammar

The German grammar is built as follows:

- the first layer contains rules making reference to particular strings and words.
- the second layer contains rules which refer to chunks (the German corpus is the only one where NP chunks are annotated).
- the third layer contains rules which identify and mark the defined term
- the fourth layer wraps it all up, i.e. it matches and marks complete definitory contexts.

The grammar has three subparts, i.e. highest level rules: one matching is-definitions, one matching verb-definitions and one matching punctuation definitions.

In total, the grammar consists of 25 rules in a 12K file.

Polish grammar

The Polish grammar is built as follows:

- the first layer contains rules making reference to particular orthographic forms and base forms.
- the second layer contains linguistically justified rules identifying nouns, NPs, PPs etc.
- the third layer contains auxiliary rules, e.g. identifying a possible term or a parenthetical expression
- the fourth layer contains top-level rules corresponding to various types (general patterns) of definitions

In total, the grammar consists of 48 rules (layer 1: 7 rules, layer 2: 17 rules, layer 3: 10 rules and layer 4: 13 rules) in a 20K file.

Portuguese grammar

The Portuguese Grammar is composed by a file containing rules, the grammar, and a file containing lexical information, the lexicon. The grammar file is composed by 57 rules structured as follows:

- The first group of rules are built to match words or simple chunks (e.g. compound tense or parenthetical)
- The second group contains rules for matching specific sentence structures for different types of definitions

- The third group consists of rules for matching the defined term.
- Finally the last group consist of top-level rules each one matching a different type of definitions.

The first group contains 35 rules, the second 15, the third 4 and the last 4.

The lexicon file contains 74 entries, each entry correspond to a verb. Three different category were identified, characterizing the syntactic behavior of definitory verbs. Each verb was marked as belonging to one of more of these categories.

Romanian grammar

The Romanian grammar is built as follows:

- the first layer contains rules that identify different part of speech or punctuations (6 rules).
- the second layer combines the rules in the first group identifying complex chunks (e.g. definite or indefinite noun phrases, prepositional phrases, etc.) (7 rules).
- top-level rules for different types of definitions (6 rules).

The grammar identifies all types of definitions, but for the `pron_def` and `other_def`, although there are several manually annotated definitions, the rules considered aren't accurate enough to identify definitions in unknown contexts. It contains 19 rules in a 8k file.

3.4 Adding machine learning to post-process the extracted definitions

In order to improve the results obtained with the pattern-based grammar approach, machine learning techniques have been applied on the Dutch, Polish, Portuguese and English results mainly to filter out incorrectly extracted definitions. The plan is to use machine learning as a way of post-processing the results which the pattern grammars yield. Machine learning is mainly used to filter extracted sentences which cannot be considered definitions. In the following, we report about the Dutch and Polish experiments where the progress made was the largest.

While the Dutch experiment worked exactly that way, the Polish group tried first to evaluate the machine learning approach when applied directly to the learning object. It has been investigated whether an machine learning approach yields comparable or even better results than the pattern grammar approach when applied to the same data.

For the Dutch experiment, the Naive Bayes machine learning algorithm has been used as a fast and easy applicable classifier based on the probabilistic model of text. The data set used is relatively small, therefore a 10-fold cross validation for better reliability of the classifier results. Several combinations of attributes have been used to find the highest gain in F-measure for the results when compared to the human annotation. The 10 attribute settings were tested for the two most frequent definition types: the to be-patterns and the punctuation patterns extracted by the grammar. With the best combination of attributes, the F-measure for the copula-definitions could be raised from 42.1 to 73.2. For the punctuation type of definitions, the F-measure could be raised from 17.3 to 44. This significant improvement is due to a rise in precision, but it comes only at the cost of decreasing recall. The recall decreased by about 19 % for the copula type and 31 % for the punctuation type of definition. In our context, this means that the user will loose some suggestions which are indeed acceptable definitions. The user-centered validation which is run in WP5 should reveal whether a high-precision result is to be preferred over a result with a higher recall but a much lower precision. For further details and figures cf. Westerhout and Monachesi 2007, the paper can be found in the appendix.

The Polish group developed the classifier which is independent from the grammars and evaluates the results directly on the corpus. In this regard this experiment differs from the Dutch experiment reported above. The goal was to investigate what performance (in terms

of recall) can be obtained using ML methods. Only if we achieve the recall comparable to the recall obtained for grammars, it is worth combining both approaches. The rationale is that if we apply the classifier with recall lower than the grammars yield the final recall will be equal or lower than the classifier recall. In other words we may increase the precision but also decrease the recall, which is critical to our application. Four types of classifiers (Naïve Bayes, IB1, ID3 and C4.5) have been tested in two ways: a) using 10-fold Cross Validation on the training and the held-out sets b) using the training and the held-out set to train the classifier and a testing set to evaluate the results. We have tested combinations of 4 types of features. Additionally, we experimented with different ratios of negative to positive examples by subsampling the number non-definition sentences.

One of the conclusions of our experiments is that using trees classifier (ID3 or C4.5) and subsampling of negative instances we can obtain recall comparable to the grammar results (about 70% for classifier, and 75% for GR' grammar evaluated on the training corpus). Therefore it can be expected that when we combine both methods we can improve the final result. The classifiers have been evaluated quantitatively using the WEKA experimental environment (<http://www.cs.waikato.ac.nz/ml/weka>). For further details and figures, cf. Adam Przepiórkowski, Michał Marcińczuk and Łukasz Degórski 2007, the paper can be found in the appendix.

Malta will be implementing machine learning for definition extraction in English, starting in January 2008. Their task will look at different learning techniques, namely evolutionary algorithms, which have never been applied to the task of definition extraction previously. Through this work, we will be able to compare the results of these techniques to already tested techniques.

The Portuguese group experimented different algorithm, namely Naïve Bayes classifier and the ID3 tree classifier. As the data set is small a 10-fold cross validation was used in order to ensure the reliability of results. The data set was composed not by the entire corpus, but by the sentences marked as definitions by the previously developed grammar. In this case we used a base-line grammar that ensures a high recall (0.98) and a low precision (0.13). We started to test this approach with copula definitions (definition introduced by the verb "to be"), feeding the algorithm with the syntactic representation of sentences. The better results was archived with ID3 algorithm with a precision of 0.46. This result is a clear improvement not only with respect to the baseline grammar but also with respect to our best grammar that shows a precision of 0.32 precision. For the next year we are planning to try different set of attributes and different algorithms and to extend this experiment to the others type of definitions.

4 Actions taken in response to user experience

The following improvements of the glossary candidate detector have been initiated in response to feedback from users:

- A context of two sentences around the extracted definition has been added to the text which is presented to the user. We are experimenting with contexts of larger sizes.
- The defined term is presented in an extra box, the defined term is given in its base form

The focus of the third phase of the project will be on improvements which are triggered by the feedback of users from the validation scenarios of WP5.

5 Documentation

Documentation of the glossary candidate detector goes alongside the development and integration of the tools. Currently, a description of all classes and methods are available as JAVADOC documents. As such, the documentation is easy to maintain. It is targeted to

software developers who want to use the tools and integrate them into their applications. The language of documentation is English. The documentation in its current state can be downloaded from the portal server in Iasi.

6 Scientific papers on the Glossary Candidate Detector

The following papers have been written and submitted or presented on conferences and published. The full papers are in the appendix. In the following, the bibliographical data are listed, together with a short summary.

- Eline Westerhout, Paola Monachesi: Extraction of Dutch definitory contexts for eLearning purposes. CLIN proceedings 2007.
 - In this article, the first version of the Dutch pattern grammar is presented
- Eelco Mossel and Lothar Lemnitzer and Cristina Vertan: *Language Technology for eLearning -- a multilingual approach from the German perspective.*. Appeared in: Georg Rehm / Andreas Witt / Lothar Lemnitzer (eds): Data Structures for Linguistic Resources and Applications. Tübingen:Narr 2007, pp. 125-134.
 - In this paper, the first version of the German pattern grammar is presented
- Adam Przepiórkowski, Łukasz Degórski, Miroslav Spousta, Kiril Simov, Petya Osenova, Lothar Lemnitzer, Vladislav Kubon, Beata Wójtowicz: Towards the automatic extraction of definitions in Slavic. In the proceedings of the BSNLP (<http://langtech.jrc.it/BSNLP2007/>) workshop at ACL 2007.
 - In this paper, evaluation results for Polish, Czech and Bulgarian are presented. In addition, an inter-annotator agreement experiment is outlined.
- Rosa Del Gaudio, António Branco: Supporting e-learning with automatic glossary extraction: Experiments with Portuguese. RANLP 2007 workshop: Natural Language Processing and Knowledge Representation for eLearning Environments. Borovets, 26. September 2007.
- Rosa Del Gaudio, António Branco: Automatic Extraction of Definitions in Portuguese: A Rule-Based Approach. In Proceedings of 2nd Workshop on Text Mining and Applications, Guimarães, Portugal, 2007.
 - In these two papers, the Portuguese pattern grammar is presented.
- Adam Przepiórkowski, Łukasz Degórski, Beata Wójtowicz: On the evaluation of Polish definition extraction grammars. At LTC 2007 (<http://www.ltc.amu.edu.pl>) and RANLP workshop.
 - This paper presents an alternative approach to inter-annotator agreement and describes the development of the Polish pattern grammar
- Eline Westerhout, Paola Monachesi: Combining pattern-based and machine learning methods to detect definitions for eLearning purposes. RANLP 2007 workshop: Natural Language Processing and Knowledge Representation for eLearning Environments. Borovets, September 2007.
 - Machine learning experiments for the post-processing of pattern grammar results are presented with the example of Dutch
- Adrian Iftene, Diana Trandabăț, Ionuț Pistol: Grammar-based Automatic Extraction of Definitions and Applications for Romanian. RANLP 2007 workshop: Natural Language Processing and Knowledge Representation for eLearning Environments. Borovets, 26. September 2007.
 - In this paper the Romanian pattern grammar and its use in various project are presented.

- Claudia Borg, Mike Rosner and Gordon Pace. Towards Automatic Extraction of Definitions. In proceedings of the Computer Science Annual Workshop 2007, Malta.
- Claudia Borg: Discovering grammar rules for Automatic Extraction of Definitions at the Doctoral Consortium at Eurolan Summer School 2007, Iasi, Romania.
 - In these two papers, the development of the English pattern grammar is presented and an approach to machine learning using genetic algorithms is outlined.
- Eline Westerhout, Paola Monachesi: Semi-automatic glossary creation from learning objects. CLIN 2007.
- Laska Laskova. "(Semi)automatic glossary detection for Bulgarian". BIS21++ Infoday at RANLP 2007, 28.09.2007, Borovets, Bulgaria.
 - In this presentation, the Bulgarian grammar was presented
- Adam Przepiórkowski, Michał Marcińczuk and Łukasz Degórski. Definition Extraction: Partial Parsing and Machine Learning. (submitted to LREC 2008)
 - This paper reports about machine learning experiments which are combined with and compared to the Polish pattern grammar
- Adam Przepiórkowski and Łukasz Degórski. Automatic Extraction of Glossary Candidates - Evaluation and Inter-Annotator Agreement. (submitted to CILing2008)
- Claudia Borg, Gordon Pace. Automatic Definition Extraction using Parser Combinators. (submitted to LREC 2008)
- Eline Westerhout, Paola Monachesi. Creating glossaries using pattern-based and machine learning techniques. (submitted to LREC 2008)