

# Combining pattern-based and machine learning methods to detect definitions for eLearning purposes

Eline Westerhout and Paola Monachesi  
Utrecht University  
Trans 10  
Utrecht, The Netherlands  
*Eline.Westerhout,Paola.Monachesi@let.uu.nl*

## Abstract

One of the aims of the Language Technology for eLearning project is to show that Natural Language Processing techniques can be employed to enhance the learning process. To this end, one of the functionalities that has been developed is a pattern-based glossary candidate detector which is capable of extracting definitions in eight languages. In order to improve the results obtained with the pattern-based approach, machine learning techniques are applied on the Dutch results to filter out incorrectly extracted definitions. In this paper, we discuss the machine learning techniques used and we present the results of the quantitative evaluation. We also discuss the integration of the tool into the Learning Management System ILIAS.

## 1 Introduction

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. Therefore, existing glossaries or wikipedia entries can be linked to learning objects, but an obvious shortcoming of this approach is that the learner would be confronted with many definitions for the term he is looking for and not only with the definition which is appropriate in the given context.

A better alternative is to build glossaries based on the definitions of the relevant terms which are attested in the learning objects. By doing so, the exact definition that the author of a certain document uses is captured; in many cases, this definition overrides a more general definition of the term. By providing the most appropriate definition to the learner for the concept he is not familiar with, we facilitate the learning process.

One of the aims of the European project Language Technology for eLearning (LT4eL)<sup>1</sup> is to show that Language Technology can provide a solution to the task of creating appropriate glossaries by developing a glossary candidate detector. More generally, the goal of the project is to show that the integration of Language Technology based functionalities and Semantic

Web techniques will enhance Learning Management Systems (LMS) and thus the learning process.

Definition extraction is the topic of much current research and techniques have been developed to this end within the Natural Language Processing and the Information Extraction communities mainly based on grammars that detect the relevant patterns and machine learning methods: in the LT4eL project, we adapt these techniques for eLearning purposes.

The glossary candidate detector we have developed, extracts definitions in all the eight languages represented in our consortium, that is, Bulgarian, Czech, Dutch, English, German, Polish, Portuguese and Romanian [17], [16]. In this paper, however, we focus only on the definitory contexts attested in the Dutch learning objects and the approach we have used to identify them. First, a substantial amount of definitions is selected and annotated manually in the learning objects which are the asset of this project. From these examples, a grammar is developed in order to extract possible definitions (cf. [18] for a similar approach). After the extraction of the definition patterns, machine learning techniques are applied on the extracted definitions to improve precision (cf. also [7] for Dutch).

The rest of the paper is organized as follows. Section 2 introduces related work on the area of definition extraction. The details of our approach are presented in section 3, in particular we discuss the corpus we have assembled, the grammar we have developed to detect definitions from our corpus of learning objects and the machine learning techniques employed to narrow down the set of definitions. Section 4 evaluates the results obtained. In section 5, we discuss the embedding of the glossary candidate detector within the Learning Management System ILIAS<sup>2</sup> and its function within an eLearning context while section 6 contains our conclusions and suggestions for future work.

## 2 Previous work

Research on the detection of definitions has been pursued in the context of automatic building of dictionaries from text, question-answering and recently also within ontology learning.

In the area of automatic glossary creation, the DEFINDER system [18] combines shallow natural lan-

<sup>1</sup> <http://www.lt4el.eu>

<sup>2</sup> <http://www.ilias.de>

guage processing with deep grammatical analysis to identify and extract definitions and the terms they define from on-line consumer health literature. The system is based on two modules, the former one uses cue-phrases and text markers in conjunction with a finite state grammar to extract definitions while the latter one uses a grammar analysis module based on a statistical parser in order to account for several linguistic phenomena used for definition writing. Their approach relies entirely on manually crafted patterns.

Research on definition extraction has been pursued very actively also in the area of Question-Answering. The answers to ‘What is’-questions are usually definitions of concepts. A common approach in this area is to search the corpus for sentences consisting of a subject, a copular verb and a predicative phrase. If the concept matches the subject, the predicative phrase is returned as answer, also in this case research relied initially almost totally on pattern identification and extraction and only later, machine learning techniques have been employed.

In [21], both the analysis of document structure as well as dependency parsing are explored. Definitions consisting of a subject, a copular verb and a predicative phrase are extracted from Dutch texts in order to provide answers to questions in the medical domain. The texts used are often encyclopedias and wikipedia entries which are well structured and thus layout information is a reliable feature to detect definitions in a text, this is however not the case for other types of texts. Therefore, for texts that are not well structured the parsing approach is more promising. However, medical questions often require answers which are larger than a single sentence while parsing techniques are typically applied to sentences.

Thus, a better alternative might be to combine the two approaches and [7] is an attempt in that direction. They propose an approach to definition extraction which operates on fully parsed text and machine learning techniques (cf. also [2], [14] for the use of machine learning methods in definition extraction). Also in this case, a rather well structured corpus is employed such as the medical pages of the Dutch version of Wikipedia. Therefore, first candidate definitions which consist of a subject, a copular verb and a predicative phrase are extracted from a fully parsed text by using their syntactic properties. Second, machine learning methods are applied to distinguish definitions from non-definitions and to this end a combination of attributes have been exploited which refer to text properties, document properties, and syntactic properties of the sentences. They show that the application of machine learning methods improve considerably the accuracy of definition extraction based only on syntactic patterns.

Research on definition extraction has been carried out also in the area of ontology learning. For example, within the German HyTex project [20], 19 verbs that typically appear in definitions were distinguished and search patterns have been specified based on the valency frames of these definitor verbs in order to extract definitions. Furthermore, semantic relations have been extracted from these definitions. Even though this information has been employed for the automatic generation of hypertext views that support both reading

and browsing of technical documents, one could imagine employing the same technique to actually update and enlarge existing formalized ontologies.

Work in this direction is that of [23] that proposes a rule-based method for extracting and analyzing definitions from parsed text on the basis of a semantically oriented parsing system. The results are then employed to improve the quality of text-based ontology learning. Also this approach relies on pattern extraction techniques to detect definitions and doesn’t employ machine learning. A difference with respect to previous systems is its use of semantic information in the identification of patterns.

### 3 The glossary candidate detector

The extraction of definitions for glossary creation for eLearning purposes constitutes a novel application of current techniques which presents some interesting challenges.

The most relevant one is constituted by the corpus of learning objects which includes a variety of text genres and also a variety of authors writing styles that pose a real challenge to computational techniques for automatic identification and extraction of definitions together with the headwords. Our texts are not as structured as those employed for the extraction of definitions in question-answering tasks which include encyclopedias and wikipedia entries, thus layout information plays in our context a marginal role.

Furthermore, some of our learning objects are relatively small in size, thus our approach has not only to favor precision as is often the case in the approaches discussed in the previous section 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.

Despite the challenges that the eLearning application involves, we believe that the techniques for the extraction of definitions developed within the Natural Language Processing and the Information Extraction communities can be adapted and extended for our purposes. In particular, our approach is similar to that of [18] since we use a grammar to identify a wide variety of possible definition patterns. However, we follow [7] in applying machine learning techniques to improve the precision of the definition extracted and distinguish definitions from non-definitions.

#### 3.1 The grammar component

As already mentioned, the first step in the detection process of definitions is the development of a grammar which is able to identify the relevant patterns.

In order to detect the most common patterns in our corpus and write appropriate rules for their extrac-

Type	Example sentence
is_def	Gnuplot is een programma om grafieken te maken <i>‘Gnuplot is a program for drawing graphs’</i>
verb_def	E-learning omvat hulpmiddelen en toepassingen die via het internet beschikbaar zijn en creatieve mogelijkheden bieden om de leerervaring te verbeteren . <i>‘eLearning comprises resources and application that are available via the internet and provide creative possibilities to improve the learning experience’</i>
punct_def	Passen: plastic kaarten voorzien van een magnetische strip, die door een gleuf gehaald worden, waardoor de gebruiker zich kan identificeren en toegang krijgt tot bepaalde faciliteiten. <i>‘Passes: plastic cards equipped with a magnetic strip, that can be swiped through a card reader, by means of which the identity of the user can be verified and the user gets access to certain facilities.’</i>
layout_def	RABE Een samenwerkingsverband van een aantal Duitse bibliotheken, die gezamenlijk een Internet inlichtingendienst bieden, gevestigd bij de gemeenschappelijke catalogus, HBZ, in Keulen. <i>‘RABE Cooperation of a number of German libraries, that together provide an Internet information service, residing at the common catalogue, HBZ, in Cologne’</i>
pron_def	Dedicated readers. Dit zijn speciale apparaten, ontwikkeld met het exclusieve doel e-boeken te kunnen lezen. <i>‘Dedicated readers. These are special devices, developed with the exclusive goal to make it possible to read e-books.’</i>

**Table 1:** Examples for each of the definition types

tion, we have manually annotated 21 files with definitory contexts which delivered 303 definitions most of which can be divided into five categories (Table 1 shows examples for each of the categories). However, 12.5 % (38 definitions) of the extracted definitions does not fit into the existing categories. Therefore we created an extra category ‘other’ to label these sentences. These can be sentences in which words and phrases like ‘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 first category (i.e. *is\_def*) are the definitory contexts in which a form of the verb *zijn* (‘to be’) is used as connector verb. These are the most straightforward definitions.

The second group (i.e. *verb\_def*) is formed by the definitory contexts in which other verbs are used as connector (e.g. *betekenen* (‘to mean’), *wordt ... genoemd* (‘is called’), *wordt gebruikt om* (‘is used to’)). Together with the first group, the second group comprises over 50 % of our definitions.

The third type (i.e. *punct\_def*) are the definitory contexts having specific punctuation features (e.g. *;*, *(..)*).

In the fourth group (i.e. *layout\_def*) are the definitory contexts in which the layout plays an important role (e.g. in tables, defined term in margin, defined term in heading).

The last category (i.e. *pron\_def*) contains the definitory contexts in which relative and demonstrative pronouns (e.g. *dit* (‘this’), *dat* (‘that’), *deze* (‘these’)) and words like *hiermee* (‘with this’), *hierdoor* (‘because of this’) are used to point back to a defined term that is mentioned in a preceding sentence. The definition of the term then follows after the pronoun, so these are often multisentence definitory contexts. Table 1 shows for each of the categories an example definition.

Grammar rules have been developed to detect all definition types, except for the layout definitions. The reason for this is that not many examples have been found for this type of definitions (i.e. only 7 definitions, that is 2.1 % of all definitions) and in addition grammar rules are not the best way to detect them.

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 [18], [13].

We have extracted definitions from a corpus of learning material which has different formats, such as HTML, PDF or DOC. All these formats are via several steps converted into XML conforming to the LT4eLAna DTD, which is an adapted version of the XCES DTD for linguistically annotated corpora [8]. For our purposes, the XCES DTD has been enriched with elements that are relevant for our project. Besides the content of the original files (that is, information about layout and the text itself), the DTD allows encoding information about part-of-speech, morphosyntactic features and lemmas. The Wotan tagger presented in [5] has been used for the annotation of the Dutch learning objects with part-of-speech information and morphosyntactic features whereas the CGN lemmatizer discussed in [3] was used for the lemmatization. It should be noticed that the rules of the grammar for the extraction of the definitory context patterns make use also of the the information encoded in the LT4ELAna format.

The XML transducer *lxtransduce* developed by [22] is used to match the grammar against files in the LT4eLAna format. *Lxtransduce* is an XML transducer, especially intended 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 (`lxtransduce.dtd`, which is part of the software). In each grammar, there is one ‘main’ rule which calls other rules by referring to them. The XPath-based rules are matched against elements in the input document. When a match is found, a corresponding rewrite is done.

The grammar contains rules that match the grammatical patterns described above. It is split into 4 layers, with rules of each layer possibly calling only rules of the same and previous layers. 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). We did not use a chunker to parse our data, because we wanted to be able to define the possible patterns of the chunks ourselves. The third layer contains rules for matching and marking the defined terms and in the last layer the pieces are put together and the complete definitory contexts are matched. The rules were made as general as possible to prevent overfitting to our training corpus.

In total, the 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.

An alternative approach could have been to parse the corpus syntactically with Alpino, a robust wide-coverage parser for Dutch [4], as proposed in [7]. However, the experiments showed that we do not need the level of deep syntactic representation produced by Alpino and that a shallower representation, as that produced by our grammar suffices for our purposes. Furthermore, since parsers (and chunkers) are not available for all the languages for which we have developed the glossary candidate detector, a shallow approach was the most promising one.

### 3.2 The machine learning component

After the detection of a large number of definitory context candidates with the grammar (1098 for the various types of definitions), another step follows to filter incorrectly extracted sentences which cannot be considered definitions. For the filtering, the Naive Bayes machine learning algorithm has been used. The Naive Bayes classifier is a fast and easy applicable classifier based on the probabilistic model of text [15]. It has often been used in text classification tasks ([11], [12]). It is also one of the classifiers used in [7] for the classification of definitions. Because our data set is relatively small, we used 10-fold cross validation for better reliability of the classifier results.

We aim at finding the best attributes for classifying definition sentences. We experimented with combinations of the following attributes (cf. also [7]).

**Text properties:** bag-of-words, bigrams, and bigram preceding the definition. Punctuation is included as [9] observe that it can be used to recognize definitions (i.e. definitions tend to contain parentheses more often than non-definitions). We include all bigrams in a sentence as feature. The use of the bigram preceding the definition is similar to the use of n-grams by

setting	description
1	using bag-of-words
2	using bigrams
3	combining bag-of-words and bigrams
4	adding bigram preceding definition to setting 3
5	adding definiteness of article in marked term to setting 3
6	adding presence of proper noun to setting 3
7	adding bigram preceding definition & definiteness of article in marked term to setting 3
8	adding bigram preceding definition & presence of proper noun to setting 3
9	adding definiteness of article in marked term & presence of proper noun to setting 3
10	using all attributes

**Table 2:** Configurations used for the Machine Learning experiment

[1] who add n-grams (n being 1, 2 or 3) occurring frequently either directly before or after a target term.

**Syntactic properties:** type of determiner within the defined term (definite, indefinite, no determiner). [7] investigated the use of determiners in definition sentences. They found out that for their data the majority of subjects in definition sentences have no determiner (62 %), e.g. ‘Paracetamol is een pijnstillend en koortsverlagend middel’ (*Paracetamol is an pain alleviating and a fever reducing medicine*), while in non-definition sentences subject determiners tend to be definite (50 %), e.g. ‘De werkzame stof is acetylsalicylzuur’ (*The operative substance is acetylsalicylic acid*).

**Proper nouns:** presence of a proper noun in the defined term. [7] observed a significant difference in the distribution of this feature between definition and non-definition sentences. Definition sentences tend to have more proper nouns in their subjects (40.63 %) compared to non-definition sentences (11.58 %).

[7] also used the document property of the position of a sentence in the document. For their type of texts (i.e. Wikipedia) this is a relevant feature, however, for our texts which are of a totally different structure this is not relevant. Another feature they used, which is difficult to simulate in our experiment, is the position of the subject in the sentence because we do not have the syntactic structure of sentences but only the part-of-speech information.

We have experimented with ten combinations of these attributes. In the first setting, only a bag-of-words has been used by the classifier and in the second setting only bigrams have been used. The third setting combines unigrams and bigrams. All other settings (4 - 10) use bigrams and the bag-of-words together, in combination with one or more other attributes. Table 2 summarizes the 10 settings. Weka, a collection of machine learning algorithms for data mining tasks, was used to perform the experiments [25].

## 4 Evaluation

### 4.1 First step: using the grammar

As already mentioned, the grammar was used to detect definitions on the basis of syntactic patterns and we have calculated precision (P), recall (R) and F-score (F) for each of the types identified by the grammar to evaluate its performance. The manual annotation of definitions was used as gold standard against which precision and recall were measured. We should be aware of the fact that it is possible that the human annotator missed correct definitions or selected definitions which were not correct according to other humans. The sentence was identified as the most appropriate unit to evaluate the performance and therefore we report the results obtained when using the sentence as a unit [19].

type	P	R	F
is_def	28.10	86.52	42.41
verb_def	44.64	75.76	56.18
punct_def	9.91	68.18	17.31
pron_def	9.18	41.30	15.02

**Table 3:** Performance of the grammar

We refer to [24] for more details on the performance of the grammar. [6] present the work done for Portuguese using the same methodology. Their grammar slightly outperforms ours in recall, whereas our precision and F-score are much higher.

### 4.2 Second step: filtering the results using machine learning methods

The precision of the results obtained with the grammar is low, which means that a user who wants to use the glossary candidate detector to create a dictionary is presented with many incorrect definitions. In order to increase precision, we trained a Naive Bayes classifier and applied it on the results obtained with the grammar.

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. There were 274 *to be*-patterns extracted, of which 77 were real definitions. This means that we have a precision of 28.1 %. For the punctuation patterns, there were even more incorrect sentences contained. This set includes 454 sentences, of which 45 are correct definitions (precision of 9.9 %).

In classification experiments, often only the accuracy is reported. This is the proportion of correctly classified instances. However, for our purposes the recall and precision of the definitions are more important than the precision and recall of the non-definitory contexts. It is possible, that the accuracy is high, whilst the recall of the definitions is very low; this occurs when the classifier categorizes a large number of non-definitions correctly. Such a large difference between accuracy and recall is clearly present in the results for the punctuation patterns. Therefore, table 4 and table

	Accuracy	P	R	F
1	82.1168	69.4	64.9	67.1
2	81.3869	66.3	68.8	67.5
3	86.8613	76.6	76.6	76.6
4	86.8613	76.6	76.6	76.6
5	87.2263	77.6	76.6	77.1
6	86.8613	76.6	76.6	76.6
7	87.5912	78.7	76.6	77.6
8	86.4964	76.3	75.3	75.8
9	87.9562	78.9	<b>77.9</b>	78.4
10	<b>88.3212</b>	<b>80.0</b>	<b>77.9</b>	<b>78.9</b>

**Table 4:** Performance of Naive Bayes classifier on the *to be*-patterns. Precision, recall and F-score are given for the definitory context class.

	Accuracy	P	R	F
1	88.9868	43.2	35.6	39.0
2	86.7841	31.7	28.9	30.2
3	88.9868	45.1	51.1	47.9
4	89.4273	46.8	48.9	47.8
5	88.9868	45.3	53.3	49.0
6	90.0881	50.0	53.3	51.6
7	<b>90.3084</b>	<b>51.1</b>	53.3	52.2
8	90.0881	50.0	53.3	51.6
9	90.0881	50.0	<b>57.8</b>	<b>53.6</b>
10	90.0881	50.0	53.3	51.6

**Table 5:** Performance of Naive Bayes classifier on the punctuation patterns. Precision, recall and F-score are given for the definitory context class.

5 report also the precision, recall and F-score for the definitory contexts.

For the *to be*-patterns, the accuracy is highest when all attributes are used. The precision, recall and F-score give also best results with this configuration. However, the differences between the settings are small for settings 3 to 10. Only for the first two settings, the scores are remarkably lower.

The accuracy and precision are highest for the punctuation patterns when configuration 7 is used for training the classifier. In this setting, the bigram directly appearing before the defining text and the definiteness of the article are taken into consideration. Recall and F-score are best for setting 9, the setting in which the definiteness of the article and the presence of a proper noun in the marked term are used as attributes.

Although the accuracy scores of the *to be*-patterns and the punctuation patterns are comparable (both around 90), precision, recall and F-score for the classification of definitory contexts are remarkably lower for the punctuation patterns. This has to do with the fact that there are far more non-definitions for the punctuation patterns whereas we are interested in the classification of the definitions.

setting	P	R	F
1	69.44	56.18	62.11
2	66.25	59.55	62.72
3	76.62	66.29	71.08
4	76.62	66.29	71.08
5	77.63	66.29	71.52
6	76.62	66.29	71.08
7	78.67	66.29	71.95
8	76.32	65.16	70.30
9	78.94	67.42	72.73
10	80.00	67.42	73.17

**Table 6:** Final results for the *to be*-patterns.

setting	P	R	F
1	43.24	24.24	31.07
2	31.71	19.70	24.30
3	45.10	34.84	39.32
4	46.81	33.33	38.94
5	45.28	36.36	40.34
6	50.00	36.36	42.11
7	51.06	36.36	42.48
8	50.00	36.36	42.11
9	50.00	39.39	44.07
10	50.00	36.36	42.11

**Table 7:** Final results for the punctuation patterns.

### 4.3 Discussion

It should be noticed that the recall values reported in the previous section are calculated in relation to the number of correct definitions extracted by the grammar. In order to identify the actual recall values, it is necessary to calculate the scores in relation to the manually annotated set of definitions, thus the final recall values calculated after applying the grammar and the machine learning classifier differ from the recall values reported in the previous section.

The final recall is calculated with the formula:

$$\text{recall} = \frac{\text{final \# correct definitions found}}{\text{\# manually annotated definitions}} \times 100$$

The precision obtained after the machine learning filtering already represents the final precision values, because it shows the proportion of correctly classified definitions in relation to the total number of sentences classified as definition.

Therefore, the final precision values are the same as the values reported in table 4 and 5. In table 6 and table 7 we report all final results, that is precision, recall and F-score.

When we compare these results to the results obtained by the grammar, we should keep in mind that there is a restriction inherent to our approach: recall cannot improve with respect to the results obtained by the grammar, because we use these results as input. The correct definitions that were not detected by the grammar are definitively lost. As a consequence, it is inevitable that the recall decreases. However, the

better the classifier performs, the smaller the loss will be.

For the *to be*-patterns, using the Naive Bayes classifier leads to an improvement of precision of 51.9 % for the best setting (setting 10). Recall drops for this same setting with 19.1 %, which means that 17 correct definitions of the 77 extracted by the grammar are lost during the classification step.

For the punctuation patterns, the precision increases with maximal 41.2 %. The recall decreases with 31.8 %, which means that 21 definitions are lost during the classification step. However, the F-score increases with 26.8 %.

There is a trade-off between precision and recall. Before using the classifier, we had better recall whereas after using the classifier the precision was much better. For the 21 files we used, 1098 definitions were extracted by the grammar of which 209 were correct (19.0 %). This means that on average 52 definitions are proposed for a file of which only 10 are correct. For the user who has uploaded a learning object and wants to generate a glossary related to it, this means that he has to check the proposed sentences very carefully and that 80 % of them have to be thrown away, if we rely only on pattern-based methods to identify correct definitions.

At the moment, we have only employed machine learning methods to filter out results for the *to be*-patterns and the punctuation patterns. For these categories the grammar extracted 728 sentences of which only 122 were correct (16.8 %). After using the Naive Bayes classifier, the number of definitions presented to the user has decreased to 127 of which 86 are correct (67.7 %). This means that the user uploading a file is presented with on average 6 possible definitions per file which have to be checked for these two categories. Out of these 6 definitions, 4 are real ones. However, the counter effect of using machine learning after applying the grammar to detect definition patterns is that on average 2 correct definitions per file are lost for these categories. Given that our goal is the automatic development of glossaries for eLearning purposes, it remains to be evaluated whether a pure pattern-based approach for definition extraction might be more appropriate than one in which it is combined with machine learning techniques, as discussed in more detail in the section below.

It is difficult to compare our results with those achieved in the area of definition extraction for automatic building of dictionaries, question-answering and within ontology learning given the different setup, languages involved, applications and aims. Perhaps, the only work we could compare our results with is that of [7] given the similarity of tasks, methodology and language. Their results with respect to accuracy are slightly better than ours since their best accuracy is 90.26% for the Naive Bayes classifier with respect to the *to be*-pattern while in our case the best result is 88.32%. However, it should be noticed that they have employed a much bigger and more structured corpus than ours.

On the other hand, [7] could not measure the effect of using machine learning on recall, because they did not annotate the definitions in their corpus manually and could therefore not compare the results obtained

Include in Glossary <input checked="" type="checkbox"/>	
Term	Hand-outs
Definition	Hand-outs zijn uitgetikte bladen met informatie, die je bij het begin van de presentatie onder het publiek verspreidt.
Context	je rug naar het publiek staat. s Hand-Outs in een groot aantal wetenschappelijke disciplines zijn hand-outs de meest gebruikte vorm van visuele ondersteuning, bv. in de psychologie en de linguïstiek. <b>Hand-outs zijn uitgetikte bladen met informatie, die je bij het begin van de presentatie onder het publiek verspreidt.</b> Het gebruik van hand-outs heeft drie grote voordelen: ten eerste kan de luisteraar terugbladeren naar dingen die eerder in het verhaal naar voren zijn gebracht.

**Fig. 1:** User Interface for the glossary candidate detector

to the set of manually annotated definitions. Thus, we cannot evaluate how we compare to them in this respect.

## 5 Embedding into ILIAS

The glossary candidate detector we have presented, is one of the functionalities which have been integrated in the ILIAS Learning Management System. One of the aims of the LT4eL project is to show that the automatic development of glossaries, on the basis of definitions attested in the learning objects, should help the student in its learning process.

Even though the glossary candidate detector has been integrated into ILIAS, it should be possible to enhance other LMS with it since this functionality has been offered as web service.

Figure 1 shows the first integration of the glossary candidate detector into the ILIAS system. Users can: a) select the option to generate a glossary for the learning object he has uploaded, this implies that the glossary candidate detector will become active and a list with terms and associated definitions will be produced; b) select the term and associated definitions from this list (shown in figure 1); c) generate a glossary on the basis of this list. The possibility of adding additional definitions should also be envisaged. It should be noticed that glossary generation is an interactive task, since the user can decide which definitions are appropriate and which should be removed.

In the previous section, we have discussed a quantitative evaluation of the performance of the glossary candidate detector which is crucial to verify that the tool produces state of the art results. To this end, various techniques have been explored. However, we believe that the best way to evaluate the glossary candidate detector is in the context of its use within ILIAS. Therefore, a scenario based evaluation, which will take user satisfaction into account, might be the best way to decide whether we should privilege recall or precision in the case of our application and thus a pure pattern-based method or a pattern-based method in combination with machine learning filtering.

## 6 Conclusions

One of the functionalities developed within the LT4eL project is the possibility to derive glossaries automatically on the basis of the definitory contexts identified within the learning objects.

A pattern-based approach is employed to identify the definitory contexts. The current grammar is able to identify most types of definitory contexts and we obtain an acceptable recall while precision should be improved. To this end, machine learning techniques have been employed which have shown that precision can be improved considerably, with the consequence that recall decreases.

Improvements can be envisaged to find a better balance between precision and recall. To this end, we plan to evaluate other classifiers and to include additional features, including semantic ones. Furthermore, we plan to extend the use of machine learning techniques to all types of definitions and not only to the most frequent ones.

Finally, a scenario based evaluation of the glossary candidate detector is envisaged. While a quantitative evaluation might be useful to establish whether the tool produces state of the art results, we wonder whether a qualitative evaluation might not be a better way to evaluate our results. Given the eLearning context in which we operate, it might be thus more relevant to evaluate the degree of satisfaction of the users. These are both the content providers who will exploit this functionality in order to develop glossaries semi-automatically and they can thus select among the proposed definitions those that they consider the most appropriate as well as the learners who thanks to this functionality will have glossaries at their disposal that should facilitate their learning process

## References

- [1] I. Androutsopoulos and D. Galanis. A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 323–330, 2005.
- [2] S. Blair-Goldensohn, K. R. McKeown, and A. Hazen Schlaikjer. *New Directions In Question Answering*, chapter Answering Definitional Questions: A Hybrid Approach. AAAI Press, 2004.
- [3] A. v. d. Bosch and W. Daelemans. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, ACL '99*, pages 285–292, 1999.
- [4] G. Bouma, G. van Noord, and R. Malouf. Alpino: Wide coverage computational analysis of Dutch. In W. Daelemans, K. Sima'an, J. Veenstra, and J. Zavrel, editors, *Computational Linguistics in the Netherlands 2000*, pages 45–59, 2001.
- [5] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. Mbt: A memory-based part of speech tagger generator. In E. Ejerhed and I. Dagan, editors, *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 14–27, 1996.
- [6] R. Del Gaudio and A. Branco. Automatic extraction of definitions in Portuguese: A rule-based ap-

- proach. In *Workshop proceedings RANLP 2007*, to appear.
- [7] I. Fahmi and G. Bouma. Learning to identify definitions using syntactic features. In R. Basili and A. Moschitti, editors, *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*, 2006.
- [8] N. Ide and K. Suderman. XML Corpus Encoding Standard, document XCES 0.2. Technical report, Department of Computer Science, Vassar College, and Equipe Langue et Dialogue, New York, USA and LORIA/CNRS, Vandoeuvre-lés-Nancy, France,, 2002. <http://www.cs.vassar.edu/XCES/>.
- [9] J. J. Klavans and S. Muresan. Dender: Rule-based methods for the extraction of medical terminology and their associated denitions from on-line text. *American Medical Informatics Association*, 2000.
- [10] L. Lemnitzer, C. Vertan, A. Killing, K. Simov, D. Evans, D. Cristea, and P. Monachesi. Improving the search for learning objects with keywords and ontologies. In *Proceedings of ECTEL 2007*, 2007.
- [11] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
- [12] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 3–12. ACM/Springer, 1994.
- [13] B. Liu, C. W. Chin, and H. T. Ng. Mining topic-specific concepts and definitions on the web. In *Proceedings of WWW-2003*, 2003.
- [14] S. Miliaraki and I. Androutsopoulos. Learning to identify single-snippet answers to definition questions. In *Proceedings of COLING 2004*, pages 1360–1366, 2004.
- [15] T. M. Mitchell. *Machine learning*. McGraw-Hill, 1997.
- [16] P. Monachesi, D. Cristea, D. Evans, A. Killing, L. Lemnitzer, K. Simov, and C. Vertan. Integrating language technology and semantic web techniques in elearning. In *Proceedings of ICL 2006*, 2006.
- [17] P. Monachesi, L. Lemnitzer, and K. Simov. Language technology for elearning. In W. Nejdl and K. Tochtermann, editors, *Proceedings of EC-TEL 2006*, pages 667–672. Springer LNCS, 2006.
- [18] S. Muresan and J. Klavans. A method for automatically building and evaluating dictionary resources. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2002)*, 2002.
- [19] A. Przepiórkowski, L. Degórski, M. Spousta, K. Simov, P. Osenova, L. Lemnitzer, V. Kubon, and B. Wójtowicz. Towards the automatic extraction of denitions in Slavic. In *Proceedings of BSNLP workshop at ACL*, 2007.
- [20] A. Storrer and S. Wellinghof. Automated detection and annotation of term definitions in German text corpora. In *Proceedings of LREC 2006*, 2006.
- [21] E. Tjong Kim Sang, G. Bouma, and M. de Rijke. Developing offline strategies for answering medical questions. In D. Mollá and J. L. Vicedo, editors, *Proceedings AAAI 2005 Workshop on Question Answering in Restricted Domains*, 2005.
- [22] R. Tobin. Lxtransduce, a replacement for fsgmatch, 2005. <http://www.ltg.ed.ac.uk/~richard/ltxml2/lxtransduce-manual.html>.
- [23] S. Walter and M. Pinkal. Automatic extraction of definitions from German court decisions. In *Proceedings of the workshop on information extraction beyond the document*, pages 20–28, 2006.
- [24] E. N. Westerhout and P. Monachesi. Extraction of Dutch definitory contexts for elearning purposes. In *Proceedings of CLIN 2006*, 2007.
- [25] I. Witten and E. Frank. *Data mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman Publishers, 2005.