



Project no. 027391

Project acronym: LT4eL
Project title: Language Technology for eLearning

Instrument Specific Targeted Research Project

Thematic Priority Information Society Technology

D2.4 Evaluation methodology of NLP based functionalities

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)	

Evaluation Deliverable

Contents

- 1 Title
- 2 Summary
- 3 Evaluation Rationale
- 4 Quantitative Measurement of the LT4EL corpora
 - 4.1 Introductory remarks
 - 4.2 Corpus size
 - 4.3 Type-token ratio per language
 - 4.4 Size of annotated corpus
 - 4.5 Level 3 annotation
- 5 Linguistic Annotation Chain
- 6 Quantitative Evaluation of the KWE
 - 6.1 Introduction
 - 6.2 Test 1: measuring performance of the keyword extractor against the human annotation
 - 6.3 Test 2: Inter-Annotator Agreement
 - 6.4 Test 3: Testing the adequacy of KWE-selected keywords
- 7 Quantitative Evaluation of the GCD
 - 7.1 Introductory remarks
 - 7.2 Test 1: comparison against manual annotation
 - 7.3 Test 2: Inter annotator agreement
 - 7.4 Test 3: Judging the adequacy of extracted definitions
- 8 Use cases
 - 8.1 Author annotates learning object with keywords (File resource), WP2
 - 8.2 Author generates glossary for learning object, WP2
 - 8.3 Author or learner searches for learning object, WP2/WP3
- 9 Scientific papers on the evaluation of tools and resources

1 Title

D2.4 Evaluation methodology of NLP based functionalities

2 Summary

Within WP2, the first year of the project was devoted to the conceptual planning, development and implementation of the language technology tools. This was done partially in parallel to and partially following the acquisition and linguistic annotation of the corpora. We mainly

- Developed a DTD, derived from the XCESAna DTD for the linguistic annotation of corpora
- Acquired, converted and annotated corpora along common guidelines which have been developed in WP2
- implemented a first prototype of a keyword extractor
- implemented a first prototype of the glossary candidate detector.

On the review meeting in January 2007 we presented a prototype of these tool as a standalone tool which demonstrated their basic functionalities. The reviewers argued that a more extensive and formal evaluation of the corpora as well as the tools is necessary to appreciate the quality of the data and tools independent of their integration in a learning management system. It has therefore been decided that the evaluation activities, which have been planned to be a substantial part of the WP2 activities in the reporting period, are described in an additional M24 deliverable.

In this report, we therefore explain in detail:

- the rationale of our evaluation strategy

- the quantitative evaluation and profiling of our corpora
- the three-step evaluation of the keyword extractor
- the two-step, iterative evaluation of the glossary candidate detector

Based on the iterative evaluation of the tool and its validation in use scenarios, we have been constantly improving the corpora as well as the language technology tools.

There had been some interaction of the evaluation with the corpus development and annotation - some of corpora, i.e. the English, Bulgarian, German, Polish and Portuguese ones, have been extended or re-annotated in response to the evaluation of the keyword extractor output. There has been a clear division of labour between evaluation of data and tools in WP2 and the validation of the tools which is done in WP5. The division of labour is outlined in the M18 deliverable of WP5. For the validation process, cf. deliverable D5.1a and D5.1b.

3 Evaluation Rationale

Evaluation of natural language and information extraction tools is quite common nowadays. Natural Language Processing, as an area, traditionally relies on quantitative evaluation regimes and statistical measures as a means of checking and comparing performance levels of new approaches, systems, and technologies. Tool developers have to prove that these tools do what they are supposed to do adequately and according to the state of the art.

The settings of this project makes the evaluation nevertheless a challenging tasks. Our NLP tools, the keyword extractor and the glossary candidate detector have not been developed as stand alone tools. They should extract information of certain types, keyword and definition candidates, in the context of larger applications, i.e. Learning Management Systems and they will be embedded in the larger context of eLearning (cf. the use cases below). The output of the tools is checked and approved or rejected by the user - the tools interact with the user. This has to be taken into account when evaluating the tools.

Another aspect which complicates evaluation is that we deal with eight languages. In most cases the evaluation of information extraction processes is performed for one language only.

The evaluation strategy we followed was:

- **formative** in the sense that it accompanied the process of development and tuning of the tools for each of the language. Therefore we ran several cycles which followed the development cycles of the tools
- **summative** in the sense that at a certain point we will finish the development of the tools and report the performance at that point in time
- **intrinsic** in the sense that we measure the performance independent of a specific application
- **extrinsic** in the sense that the contribution of the tools to the eLearning process, their added value, is assessed. This is done by Work Package 5 and is therefore out of scope of this workpackage / deliverable.

The following additional aspects had to be taken into account in the evaluation strategy

1. **Multilinguality:** The data and language models for the languages involved might differ significantly. The evaluation must therefore inform the developers of the NLP tools to find the optimal settings of the tools for each individual language. The evaluation also led to significant improvements (i.e. extension and / or re-annotation) of the corpora for some of the languages (see below)

2. **inter-annotator agreement:** In a first approach to evaluating the keyword extractor, we used the manual annotation of keywords and definitions as a gold-standard against which the output of our tools has been assessed. It turned out that a) the annotation was of varying quality across languages and b) annotators disagree significantly in which terms to choose as keywords. Even with definitions there was a higher variance than was expected beforehand. We therefore conducted inter-annotator-agreement experiments to assess where our tools stand wrt to the average human annotator

3. the tasks, in particular the keyword annotation, was not well defined. This has to do with the hybrid character of the annotated words, which have also been used as terms for the construction of the domain ontology. Therefore, the number of marked terms is higher than the normal average number of keywords assigned to a document (see the figures below).

All these aspects made the evaluation a challenging, non standard process. A significant amount of time has been invested in this process.

4 Quantitative Measurement of the LT4EL corpora

4.1 Introductory remarks

It has been outlined in the technical annex of the contract that for each of the languages of the project at least 200.000 running words (tokens) of learning objects should be collected. For most of the languages significantly more tokens have been acquired. Still the corpora are relatively small in comparison to the base population, which is assumed to be all the learning objects, or, even wider, all instructive text in a language. The base population and its size are rather vague and impossible to measure precisely. The corpora which have been collected can therefore not be claimed to be representative. The individuals text have been chosen by the responsible partners with their original use of learning objects and their relation to the chosen domains of the project in mind. The texts which have been collected can therefore be regarded as **exemplary** for the text type of learning objects as well as for the chosen domains.

We are aware that the moderate size of the corpora reveals another problem which is closely connected to the information extraction tasks: the data sparseness. Data sparseness can be described formally as a low ratio of types to tokens or, in other words, as a relatively high vocabulary growth rate, i.e. a relatively small distance between two occurrences of new tokens. We have chosen the statistics to measure keywordiness carefully as those for which the data sparseness does not have a negative impact. Another problem which arises from the relatively modest size of the corpora is the difficulty to find an adequately large number of definitions in these documents. The problem has been solved for some languages by adding new LOs to the corpus.

The Calimera document (cf. <http://www.calimera.org>) has been chosen to form the parallel part of the corpora. Calimera is a set of guidelines in the fields of information technology for library sciences. The document has been translated by the CALIMERA consortium into several languages, among them all the languages of our project, except Dutch, for which parts of the document are available as part of another project (Pulman, cf. <http://www.pulmanweb.org/>), and Maltese. We decided to incorporate part 3 of these guidelines into our corpus, because this part fits best the domain of our project. This part comprises, for most languages, between 70000 and 80000 running words. This part of the corpus plays an important role for the cross-lingual aspects of the project, and in particular for the ontology building task.

In the following, we present an overview of the corpus profiles to illustrate the points made above. The profile comprises, per language: a) the number of learning objects, b) the number of running words, c) the type-token ratio or and d) the number of tokens which form the parallel part of the corpus, i.e. the Calimera document and e) the number of documents / tokens which have been annotated manually with keywords and definitions.

Terminology: In the following we call

- the BaseXML level which preserves some layout features of the original documents *Level 1*
- the linguistic annotation level (parts of speech and morphosyntactic annotation of tokens) *Level 2*
- the additional manual annotation of keywords and definitions *Level 3*.

4.2 Corpus size

At the current state of the project (i.e. the end of the second year), the collection of learning objects (Los) available on the Lt4eL resource server (http://consilr.info.uaic.ro/uploads_lt4el) totals over 5.5 million words. All LOs were annotated following at least three main standards (sxml: a basic xml format, wp2xml: a XML format with some linguistic information, and axml: format with keywords and definitions marked manually). In addition to these formats, there are 5 other intermediate formats, with several Los being available in 5 or more annotated formats.

In the following, we give the token counts for the linguistically annotated corpora.

Language	Number of documents	Number of tokens	Size of CALIMERA part (number of tokens)
Bulgarian	54	211500	76793
Czech	45	962103	88757
Dutch	77	442559	42867
English	102	1340792	76244
German	47	362385	76907
Polish	46	463091	84288
Portuguese	39	296673	92825
Romanian	66	467298	108308

4.3 Type-token ratio per language

In general our data are very sparse: in terms of standard corpora used for NLP, our dataset is rather small. Furthermore, sparseness is affected by morphosyntactic and orthographical features of different languages (cf. Sarkar and De Roeck 2004).

The average number of tokens per type reveals the sparseness of each corpus. The lower the number of tokens per type, the sparser the corpus is. With a low average token number, we can expect to see only one or two "examples" for the majority of types. The performance of our tools, in particular of the keyword extractor, must be seen on the background of the profile of the corpus which forms the language model of the keyword extractor.

Language	Number of tokens per type
Bulgarian	9.65
Czech	18.37
Dutch	14.18
English	39.10
German	8.76
Polish	9.04
Portuguese	12.27
Romanian	12.43

4.4 Size of annotated corpus

We record the size of that part of the resp. corpora on which level 3 annotation (keywords and definitions) has been performed. While for some language exactly the same subsets have been used, annotators of other languages used different subsets in both annotation task. This is indicated in the following table

Language	Number of documents	Number of tokens
----------	---------------------	------------------

Bulgarian (definitions)	53	207865
Bulgarian (keywords)	54	211500
Czech (keywords and definitions)	19	321686
Dutch (definitions)	35	392237
Dutch (keywords)	72	497719
English (definitions)	14	223707
English (keywords)	35	343075
German (definitions)	39	328022
German (keywords)	19	154671
Polish (definitions)	40	304262
Polish (keywords)	25	189972
Portuguese (definitions)	30	273407
Portuguese (keywords)	29	265915
Romanian	41	138391

4.5 Level 3 annotation

The Level3 annotation of the documents for WP2 comprises of keywords and definitions. These data are used for training and evaluation of the tools. Another type of annotations is conceptual annotation which is used for semantic search. This is covered in the Workpackage 3 deliverable.

Language	Number of keywords	Number of definitions	Average number of keywords per document	Average number of definitions per document
Bulgarian	3236	726	61	13.5
Czech	1640	1261	86.3	66.3
Dutch	1706	528	23.7	15.1
English	1174	736	33.5	52.6
German	1277	342	67.2	8.2
Polish	1049	563	42	16.1
Portuguese	1033	619	35.6	20.7
Romanian	2555	213	62.3	5.2

5 Linguistic Annotation Chain

So far, the learning objects have been annotated on the language partners' sites with their tools. Therefore the corpora cannot be extended easily with new learning objects. This is, however, necessary to make the prototype a usable extension of a learning environment. For this reason we decided to integrate a linguistic processing chain into the language technology server. This is currently done for three languages (Czech, English, Romanian) for which we have both the rights to use the necessary annotation tools and the work-power to integrate these tools.

The Linguistic Processing System ALPE, developed by the Romanian partner, will be used for this purpose. It is intended to facilitate workflows which encompass a set of modular annotation processes on linguistic data, allowing for the integration of resources and tools. As a standalone linguistic processing environment, the user is presented with a visual

representation of a hierarchy of annotation formats. The user has basically three main choices: a) add a new resource to the hierarchy, b) add a new processing tool or c) compute and use a processing chain.

In ALPE, processing chains are automatically computed, therefore requiring no human intervention. Moreover, they can be created between any two formats defined in the hierarchy, provided the modules required are available. Also, ALPE deals with multilinguality, as it has a module that performs language identification automatically for each input file, then selects the matching tools and language resources, if available.

In an initial LPC integration step we will add analysis and annotation functionalities for the English, Czech and Romanian language. This phase will be completed by the end of February 2008. Since one of the offered functionalities is the possibility of automatically adding new tools in a LPC and creating new LPSc for other languages, we foresee that by the end of the project we will be able to offer full processing capabilities for those LT4eL languages which dispose of publicly available annotation tools.

Note that this is additional project work which is not mentioned in the technical annex.

6 Quantitative Evaluation of the KWE

6.1 Introduction

As has already been mentioned, it is not easy to establish which is the best way to evaluate the keyword extractor. We have mainly used statistical measures which are usually employed for term extraction but our application is different from the construction of a domain ontology or a terminological lexicon. In the case of the latter types of applications, precision is usually measured by dividing the extracted terms which are appropriate for a given domain by the number of accepted terms.

On the other hand, this approach cannot be used to evaluate our keyword extractor given the application envisaged in our project. Recall that the identification of appropriate keyword that describe the document will be employed for the semi-automatic metadata annotation of the learning objects. Thus, appropriate keywords will be much more restricted in number than appropriate terms for a given domain. In addition, the choice of keywords for a given document is often determined by the context of its use and we thus expect to be variation among annotators in determining which keywords are appropriate for a given document.

The tool, which will be integrated into the learning management systems should be optimized for this application context and thus a methodology has been developed to verify its appropriateness and to eventually improve its performance. Therefore, it is necessary to develop an evaluation methodology which should take into account the peculiarity of the application envisaged and the complexity of the task, that is the use of the tool for several languages.

There are certain parameters which have been taken into account in this process: a) the language(s) of the learning objects and the corresponding language models which influence the pre-selection of keyword candidates; b) the maximal length of keyphrases to be extracted; c) several distributional statistics. We have also implemented or are currently implementing the weighting of additional features of potential keywords, namely their layout features and their membership in lexical chains. The various and divergent features have to be combined into a complex statistics. Experiments in this direction are currently running, cf. deliverable D2.2b for details.

Therefore, the verification must be formative in a sense that it is repeated several times in the development cycle. It informs the optimization process for each language and verifies that certain changes or adjustments have a positive impact on the performance of the tool. The verification has also to be summative in the sense that at the end of the optimization process the overall performance for each language should be assessed.

In the rest of the section, we describe three tests which we have foreseen to evaluate the keyword extractor.

6.2 Test 1: measuring performance of the keyword extractor against the human annotation

This step of the keyword extractor evaluation is based on the keywords which have been selected and annotated manually. More specifically, for each language, at least 1000 keywords have been manually selected and marked in the corpus of learning objects by one annotator. Table \ref{table2} gives additional information on how many documents were annotated per language, how many keywords were annotated per document and average number of keywords per document.

In this step of the evaluation, automatically extracted keywords have been matched with the manually selected ones. Thus, the manually selected keywords are used as gold standard. Recall and precision of the keyword extractor are measured against this standard in the following way:

- For each document d_i , let $WM = wm_1 \dots wm_n$ be the set of manually selected keywords. Let N be the number of these keywords. For each i, j , if $i \neq j$, then $wm_i \neq wm_j$.
- Let $WA = wa_1 \dots wa_m$ be the keywords selected by the keyword extractor and M the number of these keywords, such that $M = N$. For each i, j , if $i \neq j$, then $wa_i \neq wa_j$.
- Both WM and WA contain two subsets: WMS and WAS , the subsets of single word keywords, and WMM and WAM , the subsets of multi word keywords.
- For each element in WMM and WAM , the length is calculated as the number of words. If wm_k is a two word keyword, then $L_{wm_k} = 2$. For each single word keyword wm_l , $L_{wm_l} = 1$.

Recall and precision are calculated as follows:

- For each $i : 0 \dots M$, check whether wa_i matches any $wm \in WM$. If this is the case and the match is exact, add a match value of one. All exact matches are summed up to a total value of EMV .
- If the match is partial, divide the length of the shorter keyword by the length of the longer keyword. If wa_l partially matches wm_k and $L_{wa_l} = 1$ and $L_{wm_k} = 3$, then the match value is $\frac{L_{wa_l}}{L_{wm_k}} = 1/3$.
- All exact matches and partial matches are summed up to a total value MV .
- Recall R : the recall of the keyword extractor is calculated as $\frac{MV}{N}$
- Precision P : the precision of the keyword extractor is calculated as $\frac{EMV}{M}$
- F2: the F2 measure is calculated as $\frac{2pr}{(p+r)}$, i.e. no higher preference is given to either recall or precision.

These calculations take into account that for the user it is better to be presented a part of a good multi word keyword than nothing at all. For precision, though, only the exact matches count. The following table gives an overview of the performance of the keyword extractor for the various languages.

The various statistical measures employed, that is TFIDF, ADRIDF and RIDF, were tested on the various languages and results show that, in general, TFIDF and ADRIDF nearly produced the same results. ADRIDF performs better than TFIDF only in the case of Bulgarian and Polish,

in all the other cases, performance is either the same or worst. RIDF performed worst for almost all settings; therefore, the statement of Church that residual inverse document frequency of a term is a good indicator of its keywordiness could not be proved. Simple frequency of occurrence of a term in a document plays a much more important role (cf. deliverable D2.2a for details about the statistics).

We also tested the impact of multiwords on results and we noticed that results improved for all languages if multi-word keywords up to a length of 3 words were included. This is at least partially due to the fact that a higher proportion of multi word keywords increases the number of partial matches.

Bulgarian			
Method	Recall	Precision	F-Measure
ADRIDF	0.60	0.30	0.40
RIDF	0.57	0.29	0.38
TFIDF	0.60	0.30	0.39
Czech			
Method	Recall	Precision	F-Measure
ADRIDF	0.22	0.17	0.18
RIDF	0.14	0.10	0.11
TFIDF	0.23	0.17	0.18
Dutch			
Method	Recall	Precision	F-Measure
ADRIDF	0.34	0.24	0.27
RIDF	0.25	0.19	0.21
TFIDF	0.36	0.25	0.29
German			
Method	Recall	Precision	F-Measure
ADRIDF	0.16	0.14	0.15
RIDF	0.15	0.12	0.13
TFIDF	0.18	0.15	0.16
Polish			
Method	Recall	Precision	F-Measure
ADRIDF	0.42	0.19	0.26
RIDF	0.29	0.15	0.19
TFIDF	0.42	0.19	0.25
Portuguese			
Method	Recall	Precision	F-Measure
ADRIDF	0.30	0.17	0.21
RIDF	0.21	0.12	0.15
TFIDF	0.31	0.18	0.22
Romanian			
Method	Recall	Precision	F-Measure
ADRIDF	0.26	0.12	0.15

RIDF	0.24	0.12	0.15
TFIDF	0.26	0.11	0.15

6.3 Test 2: Inter-Annotator Agreement

In order to assess the difficulty of the task that the keyword extractor has to perform, that is how much variation there is among users in the assignment of keywords to a text, an evaluation of inter-annotator agreement (IAA) on the keyword selection task has been performed.

In this sections we report about the results and findings of an inter-annotator agreement experiment for the task of keyword selection. The experiment has been performed for all language of the project except Maltese for which there are no linguistically annotated learning objects.

As explained later on, we used two different ways of measuring inter-annotator agreement.

For each language, a document of moderate size -- around 10 pages -- was chosen for keyword selection. The content of the paper was chosen such that the test persons could understand it without too much difficulty. Most of the language partners chose their language version of of chapter 3, part 7 of the Calimera document. This document deals with Multimedia.

All language partners (with the exception of English) recruited a minimum of 12 test persons for this experiment. These persons where given instructions, which had been produced centrally in English and translated by the partners into their languages. The instructions contained two aspects which are important for the following investigations:

- The annotators where asked to select not more than 15 keywords (in the pilot experiment for German the limit was set to 50, but this limit turned out to be too high)
- The annotator where asked to mark for each keyword how sure they are that this is a good keyword. A scale was given from 1 (very sure) to 3 (not so sure).

The selected keywords and their confidence ranking had to be filled into Excel spreadsheets by the annotators. These data have been sent to the work package leader who calculated the inter annotator agreement for each language and document.

For German and Romanian, two experiments where performed. For German, the same text was given to two groups, a group of novices and a group of experienced scientists. We wanted to investigate whether experienced scientists achieve a higher inter-annotator agreement than novices. The Romanian group ran the experiment with two different texts to check whether characteristics of the text influence inter annotator agreement.

IAA calculated according to Bruce and Wiebe

We used the approach of Bruce and Wiebe to calculate the inter-annotator agreement for this task. This means that we model it in a way that for every token in a text it is recorded whether an annotator decided that this word is a keyword or not.

Let $A = a_1 \dots a_A$, where A is the number of annotators. Let D be the Document to be annotated and T_D be the number of tokens in the document. For any two annotators t_i and t_j , where $t_i, t_j \in A$ and $i \neq j$, and a text D , we record the following values:

- $t_i \wedge t_j$, the number of words chosen by both annotators
- $t_i \wedge \neg t_j$, the number of words chosen the first annotator, but not the second
- $\neg t_i \wedge t_j$, the number of words chosen by the second annotators but not the first
- $\neg t_i \wedge \neg t_j$, the number of words chosen by neither annotator

It follows that $(t_i \wedge t_j) + (t_i \wedge \neg t_j) + (\neg t_i \wedge t_j) + (\neg t_i \wedge \neg t_j) = T_D$.

Let $n_{11} = t_i \wedge t_j$, $n_{12} = t_i \wedge \neg t_j$, $n_{21} = \neg t_i \wedge t_j$, $n_{22} = \neg t_i \wedge \neg t_j$ the observed values. These values can be filled in a contingency table with for cells from which marginal sums can be computed.

n_{++}	n_{1+}	n_{2+}
n_{+1}	n_{11}	n_{21}
n_{+2}	n_{12}	n_{22}

From this contingency table, the following formula for inter-annotator agreement will be derived, following the approach of Buce and Wiebe.

$$\kappa = \frac{\sum_i \frac{n_{ii}}{n_{++}} - \sum_i \frac{n_{i+}}{n_{++}} \frac{n_{+i}}{n_{++}}}{1 - \sum_i \frac{n_{i+}}{n_{++}} \frac{n_{+i}}{n_{++}}}$$

κ is 1 if the agreement is perfect, and zero if the agreement is that by expected by chance. Negative values are possible by if the agreement is lower than that expected by chance.

We proceed in the following way. For each language, we calculated for each pair of annotators, the agreement value of these two annotators measured by the kappa statistics.

For each annotator a_i , we furthermore calculate the average agreement of this annotator with each other annotator as $avg_kappa_{a_i} = \frac{\sum_j \kappa_{a_i, a_j}}{A - 1}$ where $i \neq j$ and $a_i, a_j \in A$.

We did that for all human annotators. In a further step, we extracted keywords from the same text with our keyword extractor. We used the three different statistics, Residual Inverse Document Frequency (RIDF), Adjusted Residual Inverse Document Frequency and Term Frequency by Inverted Document Frequency (TFIDF). These measures are described in more detail in the documentation of the keyword extractor. We counted and averaged the keywords chosen by the group of annotators and selected the r highest ranked keywords returned by the KWE, where r is the average number of keywords selected by the annotators. We then measured the inter annotator agreement of these three machine agents with the human annotators.

The experiment yielded the following results per language:

Language	Average human annotator agreement	KWE agreement with human annotators (using optimal settings)
Bulgarian	0.10	0.37
Czech	0.33	0.45
Dutch	0.16	0.28
English	0.09	0.43
German	0.25	0.23
Polish	0.23	0.24
Portuguese	0.18	0.19
Romanian	0.20	0.26

IAA calculated according to Gwet

As an alternative, we used the so-called AC1 measure proposed by Kilem Gwet (cf. Gwet, Kilem, *Handbook of Inter-Rater Reliability: How to Estimate the Level of Agreement Between Two or Multiple Raters*, 2001) and elaborated by Debra Haley in her PhD thesis (cf. Debra Trusso Haley, *Using a New Inter-rater Reliability Statistics*, 2007; <http://computing-reports.open.ac.uk/index.php/2007/200716>).

Gwen and Haley investigate Cohen's kappa for inter-annotator agreement, of which the formula Bruce and Wiebe used is a variant, and argue convincingly that under certain conditions this formula leads to unreliable results.

In particular, κ (kappa) is affected by skewed distributions of categories (the prevalence problem) and by the degree to which the coders disagree (the bias problem). In our experiments we deal with a skewed distribution. In particular, we have a large portion of elements (words) which are marked by neither annotator as a keyword. It was therefore worth to recalculate the figure using Gwets AC1 statistics.

The setting is the same as described in the last section, only the formula changes slightly. The AC1 formula, applied to our "two annotators and two categories" setting is as follows. Using the following 2 times 2 contingency table:

<i>A</i>	<i>B</i>	<i>B1 = A + B</i>
<i>C</i>	<i>D</i>	<i>B2 = C + D</i>
<i>A1 = A + C</i>	<i>A2 = B + D</i>	<i>N</i>

we arrive at the following equation for AC1

$$AC1 = \frac{\frac{A+D}{N} - 2 \frac{\frac{(A1+B1)}{2}}{N} \left(1 - \frac{A1+B1}{N}\right)}{1 - 2 \frac{\frac{(A1+B1)}{2}}{N} \left(1 - \frac{A1+B1}{N}\right)}$$

Using this metrics, we get much more convincing results in terms of average inter-annotator agreement, presented in the following table:

Language	Average human annotator agreement	KWE agreement with human annotators (using optimal settings)
Bulgarian	0.63	0.98
Czech	0.71	0.78
Dutch	0.67	0.72
English	0.62	0.82
German	0.64	0.63
Polish	0.63	0.67
Portuguese	0.58	0.67
Romanian	0.59	0.61

Conclusion: While with the Bruce and Wiebe statistics we get much lower agreement values than with the AC1 statistics, and AC1 seems to yield the more realistic agreement rates, some of the results are consistent over both statistics. In particular, the keyword extractor seems generally to be in better agreement with the human annotators than the human annotators among themselves. This seems to be encouraging towards the results which are generated by the key word extractor.

6.4 Test 3: Testing the adequacy of KWE-selected keywords

An evaluation which is fairly standard and which is e.g. performed by Velardi et al. (2007) on a similar task is to expose user to a) a given document and b) a set of automatically extracted keywords and let them judge how adequate the selected words are as keywords.

In this sections we report about the results and findings of this experiment which has been run for all languages except Maltese.

For each language, a document of moderate size -- around 10 pages -- was chosen. All language partners chose their language version of of chapter 3, part 7 of the Calimera document. This document deals with Multimedia. For each language, 6 - 12 test persons took part in the experiment

A ranked list of keywords was automatically generated by our keyword extractor with the optimal settings for each language.

We presented test persons with the 20 highest ranked keywords and asked them to judge their adequacy on a scale from 1 to 4:

1 = very relevant (would be a definite searching term) 2 = quite relevant (would be a secondary searching term) 3 = not relevant to the document 4 = not a valid term

Additionally, a value could be given in the case that the test person was not confident enough to decide

5 = not sure

In addition, test persons where given the opportunity for adding keywords which where missing in the list. Not all language group used this opportunity though.

The following table summarizes the average values for each language:

Language	Average score for the first 20 keywords	Number of additional keywords suggested
Bulgarian	2.21	21
Czech	2.22	none
Dutch	1.93	12
English	2.15	22
German	2.06	none
Polish	1.95	45
Portuguese	2.34	7
Romanian	2.14	none

In the following table, the average scores for the first 20, the first 10 and the first 5 keywords are compared:

Language	Average score for the first 20 keywords	Average score for the first 10 keywords	Average score for the first 5 keywords
Bulgarian	2.21	2.54	.12
Czech	2.22	1.96	1.96
Dutch	1.93	1.68	1.64
English	2.15	2.52	2.22
German	2.06	1.96	1.96
Polish	1.95	2.06	2.10

Portuguese	2.34	2.08	1.94
Romanian	2.14	1.80	2.06

Conclusions: First, the results are acceptable for all languages, with some room left for improvement. Second, there is a tendency towards better scores for the first 10 keywords, in contrast to the score for the first 20 keywords and for some languages even compared to the score for the first 5 keywords. From these results we can infer that it is a good decision to present the user, in the real system, the first ten keywords, with the option to look at further keywords if they want to.

The lists of additional keywords are used by the developers and language groups as an advice to optimizing the tool. They also prove the usefulness of allowing, in the real system, users to add their own keywords in addition to those which are suggested by the system and approved by them.

7 Quantitative Evaluation of the GCD

7.1 Introductory remarks

In general, the evaluation strategy for the glossary candidate detector is similar to that of the keyword extractor. We also carried out or are currently carrying out the same evaluation steps. In the following, we mention the most important differences:

- For the evaluation of the glossary candidate detector, a sentence-level model seems to be more sensible than the token level. The token-level measurement assumes a probabilistic model where the annotator throws a (weighed) coin for each token to decide whether it will be marked or not. This would lead to marking many short (mostly one token long) definitions, rather than fewer longer ones, as it is in reality. The sentence-level measurement assumes a probabilistic model where the annotator throws a (weighed) coin for each sentence to mark it as a definition or not. While this model is still not perfect (definitions might be shorter or longer than exactly one sentence), it is much better than the token-level one. Additionally, sentence boundaries are marked as a part of the linguistic annotation of our learning objects, so we can draw on this information.
- in the course of evaluation it turned out that it is better to divide all annotated definitions into certain types and to optimize our local grammars towards those structural definition patterns which are most common. In the following table, we present the structural types of definitions together with their relative share for each language (we give the percentage values, shares of less than 1 per cent are left out):

Type of definition	Shortcut	BG	CZ	NL	EN	DE	PL	PT	RO
def. with copula verb	is_def	38	68	24	25	26.0	34.4	25	64.2
def. with other verbs	verb_def	31	30,7	28	30	7.9	27.1	44	35.8
def. with punctuation instead of verb**	punct_def	5	1	13	27	3	19.4	14	
def. with layout marking instead of verb*	layout_def				2	61	3.3		
def. with a pronoun as defined term	pron_def	1	1	16	6			1	
other patterns	other_def	25			10		15.9	16	

Partners adjusted and optimized their local grammars to capture the more frequent patterns first, even for the price of some decrease in performance for the less frequent patterns.

7.2 Test 1: comparison against manual annotation

In this experiment we compared the automatically extracted definitory contexts with the manually annotated chunks.

- let SM be the set of sentences in the manually annotated subcorpus

- let SA be the set of sentences in the automatically annotated corpus

We consider a sentence as *marked* if at least one token is marked manually as belonging to a definition.

We count it as a match if the sentence $sm_i \in SM$ and the sentence $sa_j \in SA$ (and $j = i$) are both marked as definitions. Recall, precision and F-measure are calculated as usual.

The grammar development partners worked in circles: they ran this evaluation scheme several times and adjusted their grammar as a result of the evaluation procedure. In the following we report only the best results which have been achieved for each language and each type of definition.

Bulgarian

Type of definition	Shortcut	Best performance
def. GENERAL	all_type	R - 0.643, P - 0.182, F(2) - 0.348;
def. with copula verb	is_def	R - 0.666, P - 0.277, F(2) - 0.454;
def. with other verbs	verb_def	R - 0.679, P - 0.236, F(2) - 0.418;
def. with punctuation instead of verb	punct_def	R - 0.25, P - 0.101, F(2) - 0.16;
def. with layout marking instead of verb	layout_def	R - 0.220, P - 0.080, F(2) - 0.139;
def. with a pronoun as defined term	pron_def	R - 0.8, P - 0.082, F(2) - 0.204;
other patterns	other_def	

Czech

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	P = 29.5% R = 48.1% F2 = 39.8%
def. with other verbs	verb_def	P = 9.0% R = 25% F2 = 15.8%
def. with punctuation instead of verb	punct_def	P = 21% R = 61.5% F2 = 37.4%
def. with layout marking instead of verb	layout_def	P = 0% R = 0% F2 = 0%
def. with a pronoun as defined term	pron_def	
other patterns	other_def	

Dutch

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	R: 73.81; P: 22.63 (Training). R: 91.80; P: 20.97 (Test)
def. with other verbs	verb_def	R: 75.76; P: 44.64 (Training). R: 41.46; P: 25.76 (Test)
def. with punctuation instead of verb	punct_def	R: 54.35; P: 5.71 (Training). R:75.00; P: 2.31 (Test)
def. with layout marking instead of verb	layout_def	
def. with a pronoun as defined term	pron_def	R: 40.74; P: 6.15 (Training). R: 41.30; P: 9.18 (Test)
other patterns	other_def	

English

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	(P) 0.17; (R) 0.58; (F2) 0.32
def. with other verbs	verb_def	(P) 0.34; (R) 0.32; (F2) 0.33
def. with punctuation instead of verb**	punct_def	
def. with layout marking instead of verb*	layout_def	
def. with a pronoun as defined term	pron_def	
other patterns	other_def	

German

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	Recall: 0.55 Precision: 0.37 F(2): 0.47
def. with other verbs	verb_def	Recall: 0.20 Precision: 0.32 F(2): 0.23
def. with punctuation instead of verb	punct_def	Recall: 0.17 Precision: 0.017 F(2): 0.041
def. with layout marking instead of verb*	layout_def	
def. with a pronoun as defined term	pron_def	
other patterns	other_def	

Polish

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	P = 22.2% R = 74.1% F2 = 41.6%
def. with other verbs	verb_def	P = 12.6% R = 40.3% F2 = 23.2%
def. with punctuation instead of verb	punct_def	P = 4.2% R = 63.9% F2 = 11.0%

def. with layout marking instead of verb*	layout_def	
def. with a pronoun as defined term	pron_def	
other patterns	other_def	P = 27.9% R = 56.4% F2 = 42.1%

Portuguese

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	Dev. r=0.69 p=0.30 F2=0.48 Test r=0.66 p=0.32 F2=0.49
def. with other verbs	verb_def	Dev. r=0.73 p=0.12 F2=0.32 Test r=0.65 p=0.14 F2=0.33
def. with punctuation instead of verb	punct_def	Dev. r=0.64 p=0.19 F2=0.35 Test r=0.47 p=0.28 F2=0.38

Romanian

Type of definition	Shortcut	Best performance
def. with copula verb	is_def	P: 0.5366, R: 1.0, F2: 0.7765
def. with other verbs	verb_def	P: 0.7561, R: 1.0, F2: 0.9029

7.3 Test 2: Inter annotator agreement

In order to put the results of the glossary candidate detector into context, we performed some experiments with human annotators to explore how well human annotators are able to distinguish definitory contexts from non-definitory contexts and how well they agree in this binary classification tasks. We performed experiments for Dutch, which are presented in Westerhout and Monchesi, 2007, for Czech and Polish, which are presented in Przepiórkowski et al. 2007, and for Polish, presented in Przepiórkowski, Degórski and Wójtowicz 2007. In the Dutch and Czech case, however, the agreement was measured on token level only. We argued above why we now think that this is inadequate. In the following, we therefore concentrate on the results of the Polish experiment presented in Przepiórkowski, Degórski and Wójtowicz, 2007, for which the sentence level is used.

One of the findings across all experiments is that the task of finding definitory contexts in instructional texts is relatively ill-defined: such definitory contexts often provide definitions of terms in implicit or indirect way, lacking structural definitional clues. Consequently, the annotator's decision may be controversial.

For Polish, we asked two annotators to mark definitions in 10 files (containing over 83000 tokens) and gave them the same set of instructions. The results differed significantly: while the first annotator marked 158 definitions, the other one marked 595.

The agreement was measured on the sentence level. We counted marked and not marked sentences - where a marked sentence is a one that contains at least one marked token.

The results are:

		A		TOTAL
		marked	not marked	
B	marked	127	419	546
	not marked	39	2968	3007
TOTAL		166	3387	3553

We calculated the level of agreement between both annotator using Cohen's kappa:

The sentence-level kappa for this Polish experiment was 0.307, which is very low for a classification task, in particular if this is a binary classification task.

However, Cohen's κ does not take into account prevalence and bias effects, both conspicuous in the table above: non-definitory tokens/sentence are strongly prevalent and there is a strong bias between annotators, with one annotator classifying over twice as many tokens (over three times as many sentences) as definitory than the other.

For these reasons, we considered it to be more adequate to compare κ to the maximum value κ could attain given the actual proportions of decisions by annotators. Thus we can remove prevalence effects, but retain the bias effects which are relevant for the outcome of the experiment.

In order to calculate such κ_{max} as much as possible weight should be moved from the disagreement cells in a contingency table to the agreement cells, but without changing the marginal totals (cf. Sim and Wright, 2005, for further details)

Such maximal κ value which we could have been achieved in this experiment is 0.425 (sentence-level), to be compared with the 0.307 that has really been achieved. Even if we put this value now in a different relation, the agreement is still very low. This is to be taken into account when evaluating the performance of the glossary candidate detector.

7.4 Test 3: Judging the adequacy of extracted definitions

This experiment is still running, so no results can currently be reported. Please see deliverable D5.1b for validation experiments which have been run and which include the assessment of extracted definitions (in particular the "WP2 tutor" scenario).

8 Use cases

The following use cases are the basis of the usage scenarios which have been developed in the context of WP5, please refer to D5.1b for details. These use cases also informed the development of the keyword extractor and the glossary candidate detector. Please consult deliverables D2.2a, D2.2b, D2.3a and D2.3b for further details.

8.1 Author annotates learning object with keywords (File resource), WP2

Brief Description of Context and Goal: An author wants to provide a new learning object within the learning management system. He uploads a PDF file as a learning object. ILIAS+LTTools will assist him to assign useful keywords to the learning object.

Related LT-Functionality and WP: Semi-automated keyword annotation (WP2)

Primary Actor: Author

Preconditions: Author is within ILIAS repository and has permission to create file objects.

Postconditions: A new file object has been created, the author has chosen appropriate keywords from a list of candidate keywords, provided by ILIAS+LTTools.

Main Success Scenario

1. Author selects File in the new resource selection list and hits Add
2. ILIAS+LTTools displays a form including input fields for title, language and filename
3. Author enters title and language, selects a local .pdf file and hits Upload File
4. ILIAS+LTTools displays the (LOM) metadata input form, including a list of suggested keywords
5. Author selects some of the suggested keywords, enters some new keywords and hits Save
6. ILIAS+LTTools saves the metadata

8.2 Author generates glossary for learning object, WP2

Brief Description of Context and Goal: An author wants to provide a glossary for an existing learning object. ILIAS+LTTools will assist him to find candidates for terms and definitions of the glossary.

Related LT-Functionality and WP: Glossary term and definition detection (WP2)

Primary Actor: Author

Preconditions: Author is within ILIAS repository and has permission to create glossaries.

Postconditions: A new glossary has been created, the author has chosen terms and definitions from a list of candidate terms and definitions provided by ILIAS+LTTools.

Main Success Scenario

1. Author opens an existing learning object in Edit mode
2. ILIAS+LTTools displays learning object in Edit mode including a feature Generate Glossary
3. Author hits Generate Glossary
4. ILIAS+LTTools displays the a list of candidate terms and definitions
5. Author selects a set terms and definitions, makes some manual changes, enters a glossary name and hits Save Glossary
6. ILIAS+LTTools saves the new glossary

8.3 Author or learner searches for learning object, WP2/WP3

Brief Description of Context and Goal: An author wants to search for a learning object by keywords, e.g. for re-use within a course.

Related LT-Functionality and WP:

Primary Actor: Author

Preconditions: Author is logged into ILIAS.

Postconditions: ILIAS+LTTools lists a set of learning objects that match to the keywords entered by the user.

Main Success Scenario

1. Author hits Search in the main menu
2. ILIAS+LTTools displays a search form
3. Author enters search terms and hits Search
4. ILIAS+LTTools displays a list of learning objects that match the search terms

9 Scientific papers on the evaluation of tools and resources

The following papers have been written, 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.

- Lothar Lemnitzer, Paola Monachesi: Keyword extraction for metadata annotation of Learning Objects. In proceedings of RANLP 2007 workshop: Natural Language Processing and Knowledge Representation for eLearning Environments. Borovets, 26. September 2007.
 - In this paper, we focus on the evaluation of the keyword extractor results.

- Lothar Lemnitzer, Cristina Vertan, Alex Killing, Kiril Simov, Diane Evans, Dan Cristea, Paola Monachesi: Improving the search for learning objects with keywords and ontologies. Appeared in: Duval, Erik; Klamma, Ralf; Wolpers, Martin (Eds.) Creating New Learning Experiences on a Global Scale. Second European Conference on Technology Enhanced Learning, Lecture Notes in Computer Science , Vol. 4753, pp. 202-216
 - In this paper we describe, among others, the architecture of the keyword extractor and the distributional measures used.
- Lothar Lemnitzer, Paola Monachesi: Evaluating a multi-lingual keyphrase extractor in an eLearning context; Presentation at the CLIN conference, Nijmegen, December 2007
- 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.
- Adam Przepiórkowski, Łukasz Degórski, Beata Wójtowicz: On the evaluation of Polish definition extraction grammars. In proceedings of 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. In proceedings of 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. The influence of the machine learning approach to recall and precision is quantified.