

Keyword extraction for metadata annotation of Learning Objects

Lothar Lemnitzer and Paola Monachesi
Tübingen University and Utrecht University
lothar@sfs.uni-tuebingen.de Paola.Monachesi@let.uu.nl

Abstract

One of the functionalities developed within the LT4eL project is the possibility to annotate learning objects semi-automatically with keywords that describe them. To this end, a keyword extractor has been created which can deal with documents in 8 languages. The approach employed is based on a linguistic processing step which is followed by a filtering step of candidate keywords and their subsequent ranking based on frequency criteria.

Two tests have been carried out to provide a rough evaluation of the performance of the tool and to measure inter annotator agreement in order to determine the complexity of the task and to evaluate its performance with respect to human annotators.

1 Introduction

eLearning aims at replacing the traditional learning style in which content, time and place are predetermined with a more flexible, customized process of learning. While in traditional learning, the instructor plays an intermediate role between the learner and the learning material, this is not always the case within eLearning since learners have the possibility to combine learning material and to create their own courses. However, a necessary condition is that content should be easy to find and metadata plays a crucial role to this end. It provides a common set of tags that can be viewed as data describing data. Metadata tagging enables organizations to describe, index, and search their resources and this is essential for reusing them.

In the eLearning community, various metadata standards have emerged to describe eLearning resources with the IEEE LOM¹ being the most widespread and well-known standard. Providing metadata, however, is a tedious activity and it is not widely accepted by content providers and authors as part of their work. This has, however, the highly undesirable consequence that content becomes less visible and more difficult to retrieve.

One of the goals of the LT4eL project² is to show that language technology can provide significant support for this task. The solution we offer is to provide a Language Technology based functionality, that is a keyword extractor which allows for semi-automatic metadata annotation of the learning objects within a

Learning Management System (LMS). Keyword extraction is the process of extracting a few salient words or phrases from a given text and using these words to represent the content of the text. Keyword extraction has been widely explored in the natural language processing and information retrieval communities and in our project we take advantage of the techniques and the results achieved in these areas and adapt them to the eLearning context. More specifically, our approach employs statistical measures in combination with linguistic processing to detect salient words which are good keyword candidates.

It should be noticed, however, that keyword and keyphrase extractors have been provided mainly for English, cf. [13], [5], [16],[18], [17], [14], [10], [6]. One innovative aspect of our project is that we provide this functionality for all the eight languages represented in our project, that is Bulgarian, Czech, Dutch, English, German, Polish, Portuguese and Romanian and we embed this significant result within the eLearning context. Another innovative feature is that keyphrases are extracted in addition to keywords. This responds to findings that users frequently use keyphrases to describe a document, cf. [7].

More generally, the main objective of the LT4eL project is to show that the integration of Language Technology based functionalities and Semantic Web techniques will enhance the management, distribution and retrieval of the learning material within Learning Management Systems.

The rest of the paper is organized as follows. In section 2, we outline the architecture of the keyword extractor, including the methods we are using for ranking keywords and we point out the innovative features of our tool. The quantitative evaluation of the tool is discussed in section 3 and results obtained are analyzed. The keyword extractor has been integrated into the learning management system ILIAS. We show the result in section 4. Finally, section 5 contains our conclusions and plans about future work.

2 The Keyword Extractor

The task of a keyword extractor is to automatically identify a set of terms in a document that best describes its content. Keyword extractors have been employed to identify appropriate entries for building an automatic index for a document collection and have been used to classify texts. Keyword extraction has also been considered in combination with summarization ([16], [10], [18]). An additional use is to identify automatically relevant terms that can be employed

¹ <http://ltsc.ieee.org/doc/wg12/LOM3.6.html>

² www.lt4el.eu

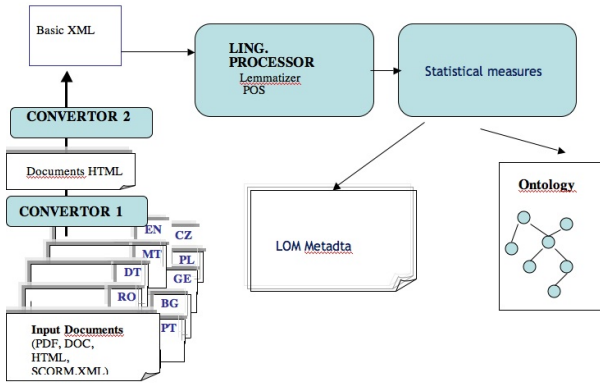


Fig. 1: Architecture of the keyword extractor

in the construction of domain-specific dictionaries or more recently of domain ontologies ([13]).

In the LT4eL project, we have adapted current techniques for term extraction in order to develop a keyword extractor which is employed for the semi-automatic metadata annotation of the learning objects. We have privileged a simple approach which is based on a frequency criterion to select the relevant keywords in a document which has been complemented with a linguistic processing step.

This method was found to lead to poor results, as claimed in [10] and consequently alternative methods were explored in the literature. They are mainly based on supervised learning methods, where a system is trained to recognize keywords in a text, based on lexical and syntactic features. However, given the specific application which has been envisaged in our project, that is the extraction of relevant keywords for metadata generation, we have chosen an approach which could be easily adapted to several languages. In the LT4eL project, we use the same algorithm for all the languages under consideration while we encode the language specific differences in the language model. It should be noticed that a machine learning approach didn't seem a possible option: given the small corpus of learning objects available for each language, we wouldn't have had enough training data at our disposal.

The keyword extractor accepts linguistically annotated input and outputs a list of suggested keywords to be included in the LOM metadata (cf. figure 1).

More specifically, the input for the keyword extractor is constituted by learning objects of various formats, e.g. PDF and DOC which are converted into HTML. From this intermediary representation an XML format is generated which preserves basic layout features of the original texts. Linguistic information is added to this format. The process yields a linguistically annotated document in an XML format which is derived from the XCESAna standard for linguistically annotated corpora. The linguistic annotation comprises: a) the base form of each word; b) the part of speech of this base form; c) further morphosyntactic features of the word form which is used in the text.

This linguistic information, which is extracted from the corpus of learning objects, is added to the *language model* for the specific language which consists of three

parts:

- **Lexical units:** they represent the combination of a lemma and a part of speech tag. They are the basic units on which statistics are calculated and they are returned as possible keywords. Only those lexical units that can occur as possible keywords are retained – mainly nouns, proper nouns and unknown words.
- **Word Form Types:** they represent the actual form of the lexical unit in the input file in combination with their morphological information.
- **Documents:** they represent the documents which constitute the corpus including their names and domains. The two domains of our documents are information technologies for the non-expert and eLearning.

Potentially interesting sequences of words are extracted using the suffix array data structure [15] but a condition is that they must appear at least twice in the document. Afterwards, filtering occurs on the basis of language specific information and sequences longer than a certain threshold are discarded. In general, sequences comprising up to 3 words are retained.

The list of candidate keywords is ranked by their saliency and to determine it an approach based on frequency has been adopted.

As already mentioned, keywords are those terms that best identify the text and represent what the text is about (i.e. the topics of a text). They tend to occur more often in that text than could be expected if all words were distributed randomly over a corpus.

A well-established way to measure the distribution of terms over a collection of documents is TFIDF, cf. equation 1.

$$TFIDF \quad \text{where} \quad IDF = \log_2 \frac{N}{df} \quad (1)$$

Church argued that Poisson distributions or mixtures of Poisson distributions of words in texts are quite useful statistics (cf. [4] and equation 2).

$$\pi(k; \theta) = \frac{e^{-\theta} \theta^k}{k!} \quad (2)$$

While the distribution of e.g. function words like *of*, *the*, *it* is close to the expected distribution under the Poisson distribution model, good keyword candidates deviate significantly from the expectation. The score of this deviation can be used as a statistics by which the lexical units are ranked ([3]). The deviation of the observed distribution of a word from the expected distribution under the Poisson model, i.e. predicted IDF (cf. equation 3) is called Residual IDF (short: RIDF, cf. equation 4).

$$-\log_2(1 - e^{-\theta}) \quad \text{where} \quad \theta = \frac{cf}{N} \quad (3)$$

$$IDF - PredictedIDF \quad (4)$$

A closer look at the formula reveals that RIDF does not take the term frequency in the analysed document

into account. It measures the overall distribution of a term in a collection of documents, but not the frequency of a term in an individual document. Since we like to know more about the role which a term plays in a particular document, we extended this metric with a factor which measures the frequency of the term in that document and arrived at a statistics which we call Adjusted Residual IDF (short: ADRIDF. cf. equation 5).

$$ADRIDF = RIDF\sqrt{tf} \quad (5)$$

In our project, we have implemented and evaluated the appropriateness of these statistical measures in ranking the most relevant keywords from our multilingual learning objects. In section 3, the results are discussed in detail.

The keyword extractor is built to deal with a wide range of languages: Bulgarian, Czech, Dutch, English, German, Polish, Portuguese and Romanian. This is a relevant result since techniques to extract keywords and keyphrases have been usually tested on English and never on such a variety of languages.

On the one hand, the management of such a wide range of languages makes it necessary: a) to build a common annotation format for all annotated corpora and b) to keep the language specific components of the tool as lean as possible. On the other hand, the multilingual aspect of the development gives us the chance to broadly evaluate the performance of the tool and its underlying information extraction methods, as discussed in detail in the next section.

The evaluation strategy must be both formative – i.e. inform the development of the tool, in particular the language-specific settings – and summative – i.e. assess the performance of the final tool. It has to be both intrinsic – i.e. assess the performance of the tool in isolation – and extrinsic – i.e. assess the performance of the tool as part of a learning environment.

As discussed more at length in section 3, the novelty of our application makes it difficult to adapt current evaluation tests for our purposes. On the other hand, we do need to assess the performance of the tool and verify that it achieves acceptable results before integrating it in the Learning Management System. Therefore we have to accept the limitation of these tests and work towards the development of new ones more fit to the purpose. It should be noticed that a non-optimal performance of the tool in the intrinsic evaluation might still lead to an appropriate behavior of the keyword extractor in the extrinsic evaluation.

In the development of the keyword extractor, special attention has been devoted to multiword terms. A first analysis of the manually selected keywords revealed, that for some languages a substantial amount of them is multi word. E.g. for Polish we have 67 % keyphrases of two or more words. For other languages, e.g. German, multi word key phrases do not play a significant role, see table 1 for details.

We therefore put some effort to properly deal with these items and several tests have been carried out to detect the most appropriate length for multiword keywords and possible variation due to language. We followed the approach of Yamamoto and Church, cf. [15], to effectively identify and extract recurrent multi-word

language	Keywords (%)	Keyphrases (%)
Bulgarian	57	43
Czech	73	27
Dutch	75	25
English	38	62
German	90	10
Polish	33	67
Portuguese	86	14
Romanian	70	30

Table 1: Percentages of keywords and keyphrases per language

key phrases up to a predefined length. Additionally, we used linguistic information to further restrict this set of multi-word key phrases, e.g. to exclude phrases which end in a preposition. Statistically, multi-word phrases are treated as single words.

Providing users with multiword keywords raises the issue of which should be the best way to represent them. We have noticed that, at least for some languages such as Polish, a sequence of base forms looks quite unnatural. Therefore we have decided that the selected multi-word keywords are represented by their most frequent attested forms.

We refer to [9] for additional details on the use of the keyword extractor within the LT4eL project.

3 Evaluation of the keyword extractor

As already mentioned, it is not easy to establish which is the best way to evaluate the keyword extractor. In our approach, 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 these 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 there to be variation among annotators in determining which keywords are appropriate for a given document.

Ultimately, the best way to validate the keyword extractor might be in the context of the Learning Management System, that is by authors or content providers which will employ it to annotate learning objects with LOM metadata semi-automatically. On the other hand, the keyword extractor which will be integrated into the LMS should be optimized for this task and thus a methodology should be developed to verify its appropriateness and to eventually improve

language	# annot. doc.	# annot. KWs	tokens / doc.	KWs / doc.
Bulgarian	55	3236	3980	77
Czech	465	1640	672	3.5
Dutch	72	1706	6912	24
English	36	1174	9707	26
German	34	1344	8201	39.5
Polish	25	1033	4432	41
Portuguese	29	997	8438	34
Romanian	41	2555	3375	62

Table 2: Percentages of keywords and keyphrases per document

its performance.

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 preselection of keyword candidates; b) the maximal length of keyphrases to be extracted; c) several distributional statistics. We also envisage to employ additional features to select and rank keywords, as discussed in section 5 and once these features have been implemented, their impact has also to be evaluated.

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 planned to evaluate the keyword extractor.

Test 1 In order to have a rough idea of the performance of the tool, we have measured recall and precision of the keyword extractor for each language and each appropriate parameter setting. A gold standard has been established on the basis of manually annotated keywords (i.e. 1.000 keywords for each corpus of learning material). This part of the evaluation has been performed automatically.

Test 2 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.

Test 3 In order to assess the appropriateness of the keyword extractor in the context of the semi-automatic metadata annotation of the learning objects, we present test persons for each language with a document and a limited set of keywords which have been extracted from this document. Each member of this set of keywords is assessed by the test person with respect to the adequacy to represent the text. This is an ongoing part of the evaluation for which we cannot report results yet.

3.1 Test 1: measuring performance of the keyword extractor

This evaluation of the keyword extractor 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 2 gives additional information on how many documents were annotated per language, how many keywords were selected per language (= keyword types), average length of these documents 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 $wm_i \neq wm_j$ (i.e., there are no duplicates in the keywords lists).
- Let $WA = wa_1 \dots wa_m$ be the keywords selected by the keyword extractor and M the number of these keywords. We abridge these keyword lists 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 $i : 0 < i < M$, 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 $\frac{EMV}{M}$. Note that for the calculation of the P value only the exact matches are taken into account.
- 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.

Table 3 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 worse than TFIDF. 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 proven. Simple frequency of occurrence of a term in a document plays a much more important role.

With respect to precision and recall, results varied significantly across languages. If we only consider TFIDF, the best result is 48 % for recall reached for English and the worst is 18 % for German, while with respect to precision, the best result is again obtained for English with 26 % while the worst is obtained for Romanian with 11 %. These values are influenced by two factors: a) the quality of the human judgment when selecting the keywords; b) the quality of the linguistic annotation of the corpora.

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.

As already mentioned, this test was performed only to get a rough impression of the performance of the keyword extractor as well as to determine which statistical measure performed best and to determine the maximum length for multiword keywords. More generally, its major purpose lies in informing the developers by presenting those keywords which did not match with the manually annotated ones and by presenting those manually selected keywords which have not been extracted by the tool. Note that not all keyword candidates which do not match manually selected keywords are necessarily bad keywords.

In fact, we believe that there might be some variation among users in identifying keywords and it is for

this reason that we have performed an experiment to measure inter annotator agreement which is described in detail in the following section.

Bulgarian			
Method	Recall	Precision	F-Measure
ADRIDF	0.38	0.18	0.25
RIDF	0.36	0.18	0.23
TFIDF	0.39	0.19	0.25
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
English			
Method	Recall	Precision	F-Measure
ADRIDF	0.47	0.28	0.33
RIDF	0.33	0.18	0.22
TFIDF	0.48	0.26	0.32
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

Table 3: Performance of the keyword extractor for the various languages

3.2 Test 2: Inter annotator agreement

In order to assess the intrinsic difficulties of the keyword selection task and to verify the performance of the keyword extractor compared to human annotators, we have investigated the inter annotator agreement on at least one document for each language. More specifically, we wanted to investigate where the performance of our keyword extractor stands relative to the performance of a group of human annotators. In the test described in the previous section, we have compared the output of the keyword extractor to the choices of

one single human annotator. We have thus relied completely on the performance of this individual annotator. The experiments which we describe in this section reveals how reliable the human judgement is and where the judgments of the keyword extractor stands relative to the human judgments.

A document of average size – around 10 pages – has been chosen for manual keyword selection. The content of the learning object was chosen so that it would be easy to understand for the test persons: it was a document dealing with Multimedia belonging to chapter 3, part 7 of the Calimera corpus.³ This material is available for all the languages under consideration.

A minimum of 12 test persons have been recruited for the experiment (with the exception of English and Czech, where we had only 9 and 6 judgments). In the instructions, which have been written in English and have been translated into the eight languages we have tested, the annotators were asked to select not more than 15 keywords and to mark for each keyword how sure they were that this is a good keyword. A scale was given from 1 (very sure) to 3 (not so sure).

For German and Romanian, two experiments were performed. For German, the same text was given to two groups, a group of students who were not familiar with the topic and a group of experienced scientists. We wanted to investigate whether experienced scientists achieve a higher inter-annotator agreement than students who are not familiar with the topic. The Romanian group ran the experiment with two different texts to check whether characteristics of the text influence inter annotator agreement.

We used the approach of [2] to calculate the inter-annotator agreement for this task. This means that we have modeled 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 a_i and a_j , where $a_i, a_j \in A$ and $i \neq j$, and a document 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 by the first annotator, but not by the second one
- $\neg t_i \wedge t_j$, the number of words chosen by the second annotator but not by the first one
- $\neg t_i \wedge \neg t_j$, 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 from which marginal sums can be computed.

From this contingency table, the following formula for inter-annotator agreement can be derived, following the approach of Bruce and Wiebe:

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

Table 4: Contingency table with marginal sums

$$\kappa = \frac{\sum_i i \frac{n_{ii}}{n_{++}} - \sum_i i \frac{n_{i+}}{n_{++}} \frac{n_{+i}}{n_{++}}}{1 - \sum_i i \frac{n_{i+}}{n_{++}} \frac{n_{+i}}{n_{++}}} \quad (6)$$

κ is 1 if the agreement is perfect, and zero if the agreement is that 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 calculated the average agreement of this annotator with the other annotators as:

$$avg_kappa_{a_i} = \frac{\sum_j j \kappa_{a_i, a_j}}{A - 1} \quad (7)$$

where $i \neq j$ and $a_i, a_j \in A$.

In a follow up experiment, we extracted keywords from the same text with our keyword extractor. We used the three different statistics – RIDF, ADRIDF and TFIDF – which we have described above. We selected the n highest ranked keywords, where n is closest to the average number of keywords selected by the human annotators, and measured the inter annotator agreement of these three machine agents with the human annotators.

Language	Human	TFIDF	RIDF	ADRIDF
Bulgarian	0.10	0.25	0.28	0.37
Czech	0.23	0.33	0.28	0.39
Dutch	0.16	0.22	0.14	0.28
English	0.09	0.29	0.29	0.43
German	0.25	0.23	0.23	0.23
Polish	0.28	0.15	0.15	0.20
Portuguese	0.18	0.16	0.15	0.19
Romanian	0.20	0.24	0.22	0.26

Table 5: Results of inter annotator agreement experiments

The results are reported in table 5. From this experiment, we can conclude that inter annotator agreement is not very high therefore this is in general not a task on which humans easily agree. What is relevant for us though, is that the keywords extractor performs for most languages better than humans. Furthermore, on the basis of the results obtained for the additional experiment performed for Romanian and German, we can conclude that experts do not seem to behave more consistently than non-experts.

The experiments might also provide useful feedback which might help improve the performance of the keyword extractor. It might be relevant to look at

³ <http://www.calimera.org/>

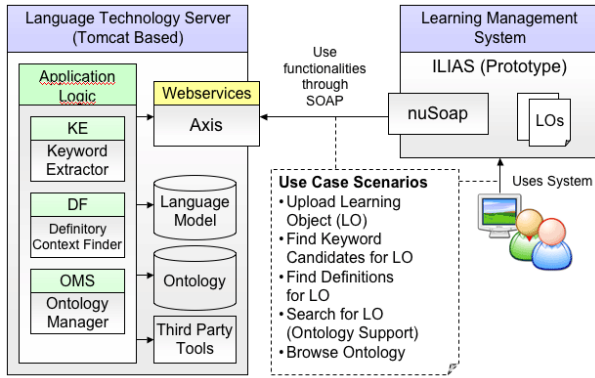


Fig. 2: Architecture of the Language Technology enhanced LMS

the words on which the majority of human annotators agree and to check why they are not captured by the key word extractor and adapt the keyword extractor if this is possible.

Another interesting finding is that, while in test 1 TFIDF performed slightly better than ADRIDF, in test 2 ADRIDF produced lists which resulted in a better agreement with the human annotators. First, the results cannot be compared directly, because in test 1 we refer to only one human annotator and a collection of documents, while in test 2 we refer to a group of annotators and only one document. Since ADRIDF did not perform much worse than TFIDF in test 1, we are inclined to favour ADRIDF.

4 Integration into ILIAS

The keyword extractor is a functionality which has been integrated in a learning management system to support the semi-automatic metadata annotation of the learning objects. It should assist authors of learning material to find and assign appropriate keywords to their learning objects. In the context of the LT4eL project, the tool has been integrated in the ILIAS learning management system even though it should be possible to enhance other LMS with it.

The tools and data reside on a dedicated server and are approached from inside the Learning Management System via Web Services. Figure 2 shows the major components of the integration setup. The language technology server on the left provides the keyword extractor and other NLP components (cf. [9] for more details). The functionalities can be accessed directly on the webserver for test purposes or they can be used by the learning management system through the web service interface. Figure 3 shows the first integration of the keyword extractor into the ILIAS learning management system. The function is embedded into the existing LOM metadata handling of ILIAS to enable a semi-automatic generation of keywords. Users can: a) run the keyword extractor and get a list showing a number of suggested keywords for a document; b) select keywords from this list and c) add their own keywords. The interactivity is an important feature of this tool. It will not be used to completely perform the task of keywording a documents, but to make in-



Fig. 3: User interface to the Keyword Extractor

formed suggestions which the user has to approve or reject.

5 Conclusions and future work

One of the functionalities developed within the LT4eL project is the possibility to annotate learning objects semi-automatically with keywords that describe them, to this end a keyword extractor has been created. The approach employed is based on a linguistic processing step which is followed by a filtering step and keyword ranking based on frequency criteria.

Two tests have been carried out to provide a rough evaluation of the performance of the tool and to measure inter annotator agreement in order to determine the complexity of the task and to evaluate the performance of the keyword extractor with respect to human annotators.

The results are promising also considering that the task has been carried out for 8 different languages. However, there are possible ways in which the results of the keyword extractor could be improved.

A further distributional characteristic of keywords is what has been called their *burstiness* (cf. [8]). Good keywords tend to appear in clusters within documents. It can be observed that, once a word appeared in a text, it tends to appear in the following section of the text in much shorter intervals as would be the case if all the occurrences of this word were distributed evenly throughout the text. When the burst ends, the occurrence intervals return to normal, until the keyword appears - and probably bursts - again. A method to capture this behaviour of key words is described in [12]. This distributional behaviour reflects the fact that keywords represent topics, and reference to a certain topic is a local phenomenon: many texts deal with many (sub)topics and in some texts a topic is resumed after a while. We are currently investigating the im-

fact of term burstiness on the extraction of keyword sets on a subset of the languages under consideration.

Keyword candidates tend to appear in certain salient regions of a text. These are the headings and the first paragraphs after the headings as well as an abstract or summary. Salient terms might also be highlighted or emphasised by the author, e.g. by using italics. Investigations of the manually annotated keywords have shown that a word with a salient position or marked by layout features is at average twice as probable to be marked as keyword as those words which do not bear these features. Therefore, we plan to use layout information as additional filter in proposing salient keywords (cf. [13]).

Another approach to improve the extraction of appropriate keywords is to look at words which are related syntactically and/or semantically. Sets of related lexical units represent the topic(s) of a text better than individual words can do. In general, there are two methods to build structures of related words from text. First, one can relate words which show a higher probability of co-occurrence that could be expected if words would be distributed randomly. Second, lexical-semantic networks can be used to find pairs of words or concepts which are linked by lexical-semantic relations. We envisage to extract lexical chains from texts (cf. [11], [1]), using wordnets for those language in the project for which they are available. Lexical chains are rich structures of semantically related words. Keywords are assumed to participate in long and / or dense chains.

References

- [1] Barzilay R. and Elhadad M. 1997. *Using Lexical Chains for Summarisation*. In: ACL/EACL-97 summarisation workshop. pp. 10-18.
- [2] Bruce R. and J. Wiebe. 1999. *Recognizing subjectivity: A case study of manual tagging*. In: Natural Language Engineering. Vol. 5, No 2, pp 187–205.
- [3] Church, K. and W. Kenneth and W. Gale. 1995. *Inverse Document Frequency (IDF): A Measure of Deviations from Poisson*. In: Proc. of Third Workshop on Very Large Corpora.
- [4] K. Church and W. Gale. 1995. *Poisson mixtures*. In: Natural Language Engineering. Vol. 1, No 2, pp 163–190.
- [5] Frank, E. and Paynter, G. and Witten, I. and Gutwin, C. and Nevill-Manning, C. 1999. *Domain-specific Keyphrase Extraction*. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence 1999, pp. 668-673
- [6] Hulth, A. 2003. *Improved automatic keyword extraction given more linguistic knowledge*. In: Proceedings of EMNLP2003.
- [7] Jones S. and G. W. Paynter. 2006. *An Evaluation of Document Keyphrase Sets*. In: Journal of Digital Information, Vol. 4, No 1, <http://journals.tdl.org/jodi/article/view/jodi-107/92>
- [8] Katz, S.M. 1996. *Distribution of content words and phrases in text and language modelling*. In: Natural Language Engineering 2(1996)1. pp. 1559.
- [9] Lemnitzer L. and C. Vertan and A. Killing and K. Simov and D. Evans and D. Cristea and P. Monachesi. 2007. *Improving the search for learning objects with keywords and ontologies*. In: Proceedings of the ECTEL 2007 conference. Springer Verlag.
- [10] Mihalcea R. and P. Tarau. 2004. *TextRank: Bringing Order into Texts*, In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, July 2004.
- [11] Morris, J. and Hirst, G. 1991. *Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text*, In: Computational Linguistics, 17 (1), pp. 21-45.
- [12] Sarkar, A. and P. H. Garthwaite, and A. De Roeck. 2005. *A Bayesian Mixture Model for Term Re-occurrence and Burstiness*. In: Proc. of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005). Ann Arbor, Michigan. ACL. pp 48–55.
- [13] Sclano, F. and P. Velardi, TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities. To appear in Proc. of the 3rd International Conference on Interoperability for Enterprise Software and Applications I-ESA 2007, Funchal, Madeira Island, Portugal, March 28-30th, 2007.
- [14] Turney, P. D. 2000. *Learning algorithms for keyphrase extraction*. Information Retrieval, 2:303-336.
- [15] Yamamoto, M. and K. Church. 2001. *Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus* In: Computational Linguistics 27(2001),1. pp 1–30.
- [16] Wan X. and J. Yang and J. Xiao. 2007. *Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction*. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, 2007, pp. 552–559.
- [17] Witten, I.H. and G. W. Paynter and E. Frank and C. Gutwin, and C. G. Nevill-Manning. 1999. *KEA: Practical automatic keyphrase extraction*. In: Proceedings of Digital Libraries 99 (DL'99), pp. 254-256.
- [18] Zha, H. Y. 2002. *Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering*. In: Proceedings of SIGIR2002, pp. 113-120.