Using Textual CBR for e-Learning Content Categorization and Retrieval

Luis Rodrigues[2], Bruno Antunes[1], Paulo Gomes[1], Arnaldo Santos[2], Jacinto Barbeira[2] and Rafael Carvalho[2]

[1] AILab - CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal
[2] PT Inovação SA, Aveiro, Portugal

Abstract. The importance of e-learning systems is increasing mainly due to the growing number of companies that need to train their employees. But companies need to make the process of creating e-learning contents more efficient, this can be achieved reusing e-learning materials. This happens especially in big companies with a considerable amount of contents developed and stored. This paper presents an approach to indexing and retrieval of e-learning contents based on Textual Case-Based Reasoning. We describe how we represent contents as cases and how the indexing and retrieval mechanisms work. We also describe experimental work that defines a first setup for the reasoning mechanisms implemented.

1 Introduction

Nowadays, most of the medium and large-size companies invest a considerable amount of resources in the training and education of their employees. In order to do it in an efficient way, most of these companies use e-learning systems. Although the use of these type of system is widespread, one problem that the e-learning users face is the time that takes to create an e-learning content. It can take most of the time of the teachers and the development team of the e-learning system. This problem is especially important in big organizations, where there are a huge amount of different e-learning contents. Authoring and searching tools are needed to make the content creation and maintenance process more efficient. One of the kind of tools needed, are reuse mechanisms. Teachers must be able to easily reuse materials from e-learning contents. In our work, we are interested in creating these mechanisms for e-learning systems.

This paper presents a mechanism for indexing and retrieving e-learning contents, based on a Textual Case-Based Reasoning (TCBR [1, 2]) approach. This mechanism is implemented in PEGECEL, a Learning Content Management System developed with the collaboration between PT Inovação and the AILab of the University of Coimbra. A case in PEGECEL represents an e-learning content, with categories associated to it. TCBR techniques are used to help users classifying contents (indexing) and in the retrieval of similar contents.

The next section describes PEGECEL and explains its architecture. Section 3 presents the way contents are represented using cases. Section 4 describes how the indexing mechanism works and the next section shows the retrieval mechanism. In section 6, we describe the experimental results of this work and finally section 7 concludes the paper.

2 PEGECEL

The PEGECEL project is a collaboration between the Artificial Intelligence Laboratory of CISUC and PT Inovação. Its main goal is the development of an e-learning content manager for FORMARE [3], the e-learning platform developed by PT Inovação. PEGECEL provides tools to help reusing e-learning contents. This paper focus in the reuse of e-learning contents. We explore three different points: the case representation (since we have selected a Case-Based Reasoning (CBR) [3, 4] approach for e-learning content representation), the indexing and retrieval of contents.

We have chosen a CBR and Natural Language Processing (NLP) [5, 6] approach to solve the problems of classification and searching of e-learning contents. Natural language processing (NLP) techniques are used to extract information from the documents inside the e-learning contents, which then enables the use of CBR to predict the categories of a new content. CBR can also retrieve a list of contents similar to a user's search query.

The two main concepts in PEGECEL are: contents, which correspond to e-learning contents; and content areas, which are logic containers for contents. There are three different users in the system: the administrator that has full privileges in the system, which enables her/him to configure the system and manage other users; the content manager, which is responsible for several content areas; and the student (or normal user) that as access to specific content areas, in a limited way. PEGECEL architecture comprises three layers (see figure 1): the presentation layer, responsible for the interface with the users; the logic layer that comprises all the managers and specific modules, including the reasoning modules; and the data layer that comprises the information that supports the system, which is stored in a database.

The presentation layer is web-based and comprises three different versions, depending on the user privileges. The administrator has all the functionalities available to her/him. This user profile represents the users that are in charge of managing the system. The content manager profile is responsible for managing the e-learning contents in the central repository, as well as the content areas. The normal user or student, can only manage contents that s/he created or that someone gave her/him managing permissions. Most of the times, the normal user has a personal content area, where s/he may store and manage personal contents and no editing permissions in other content areas.

The logic layer is the core of PEGECEL and comprises three different types of modules: the core modules, which provide the more complex functionalities of

[3] http://www.formare.pt

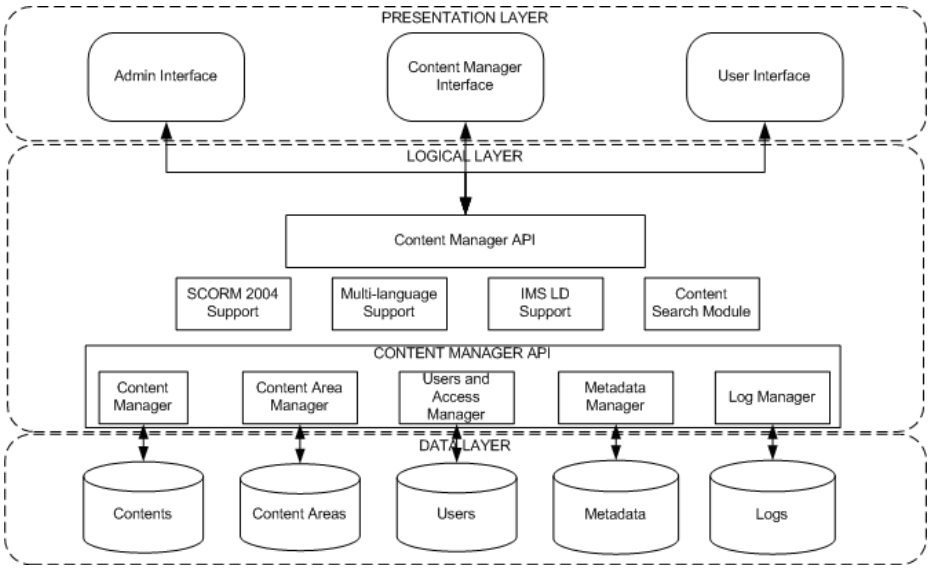

Fig. 1. The architecture of PEGECEL, based in three layers: presentation, logical and data.

the system; the content manager API, which provides access to the core modules, from the interface point of view; the data manager modules, that enables the direct access to the data layer, making the bridge between the core modules of the logic layer and the data layer. The core modules comprise four important sub modules: the SCORM[4] support module, which enables the system to handle e-learning contents in SCORM format; the multi-language support, which enables PEGECEL to handle contents in several languages; the IMS LD[5] support module, which enables the system to handle e-learning contents in IMS LD format; and the content search module, which is responsible for indexing and retrieving e-learning contents. The remaining of this paper focus on the content search module.

The data layer comprises several information that is manipulated by the system. This information comprises: e-learning contents, content areas, information about users, metadata and logging information. The metadata information comprises the case representation used for the system reasoning. The next section presents how a case represents an e-learning content.

[4] A standard format for e-learning contents, see [7].
[5] Another standard format for e-learning contents, see [8].

3 Representing e-Learning Contents

As said before, PEGECEL uses a CBR approach, in which the case representation is a basic concept. A case in PEGECEL represents an e-learning content, which comprises several files organized in a hierarchy defined in a manifest file. The case problem description comprises a list of documents (files) each of which has a list of words associated with it (see figure 2). These words are extracted from the content documents, the next section describes this process in greater detail. The case solution description comprises a set of categories (or topics, we use both words as synonyms), which can be words extracted from the documents, but can also be words given by the user. These categories represent the e-learning content topics and are assigned by the user that classified the e-learning content.

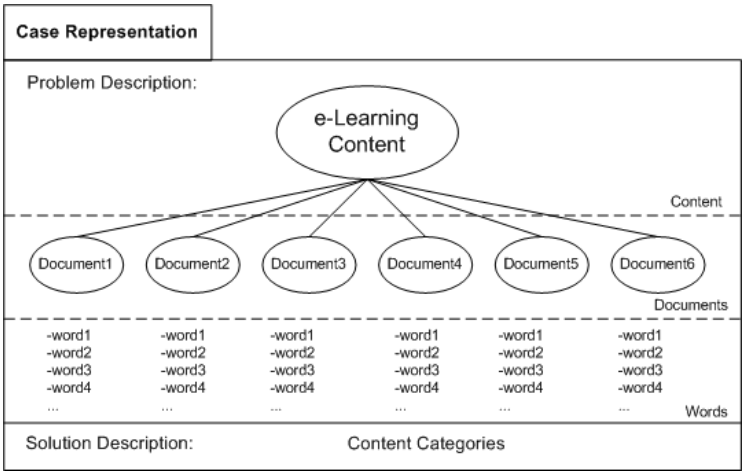

Fig. 2. The case representation in PEGECEL.

PEGECEL uses cases for two tasks: the suggestion of categories to the user, during the classification of an e-learning content, and the retrieval of similar contents given a user query. The next two sections describe these processes in greater detail.

4 Indexing e-Learning Contents

The content search module is responsible for the indexing of cases, which is a process comprising two phases: word extraction and content indexing (or building of a new case). The term extraction is a process that comprises several steps: scanning each document in the e-learning content for text, removal of stopwords and stemming. In the end, there is a list of words by document, and associated to each document there is information of word frequency in the document

(TFxIDF see [9]). If the document is in HTML format, which is the majority percentage of documents in e-learning contents, PEGECEL extracts formatting information, like: title, heading level, bolds, italics, and underlines. This formatting information is used to increase the word frequency associated with each word.

Indexing only occurs when a user adds a content to the system. The user must assign categories (or topics) to the content. PEGECEL suggests a list of topics based on two sources: the words extracted from the content, organized by document; and the topics of similar contents. The most frequent words extracted from the content (see previous paragraph) are added to the list of suggested topics. Note that words in titles and headings are given more importance.

PEGECEL also looks for topics from similar contents, which are represented in the system as cases. So, the system retrieves the most similar cases (see the retrieval and ranking in the next section) and selects the most frequent topics. These topics, which came from the most similar contents, are added to the list of topics to be suggested to the user. After which, the user selects the ones that are important for the new content. Then the system creates a new case, corresponding to the added content indexed by the topics selected by the user. Figure 3 shows an example of the topic suggestion process.

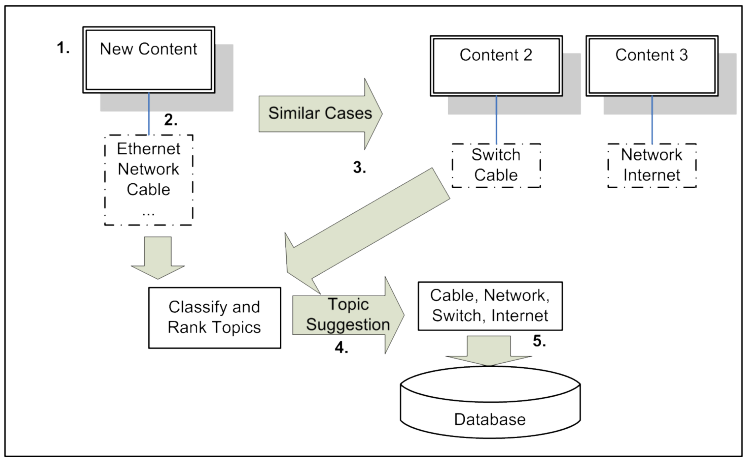

Fig. 3. An example of topic suggestion in PEGECEL.

The creation of the list that comprises the suggested topics is based on the selection of the nine more relevant words in the content (this number is based on [10]) and the topics in the most similar cases. The number of similar cases to be considered can be established by the system administrator as a system parameter. Then, for each topic is assigned a score, based on a weighted sum between the frequency of the topic if it is present in the content, and the similarity score

that the case in which the topic is present. If a topic is suggested both from a word in the content and from a similar case, then it will have a higher score, as opposed to a topic coming from only one source. It is also presented to the user a tag cloud (see figure 4) with the most used topics in indexing new contents, which can help the user choosing the words. It also gives her/him a sense of what is being indexed in the system.

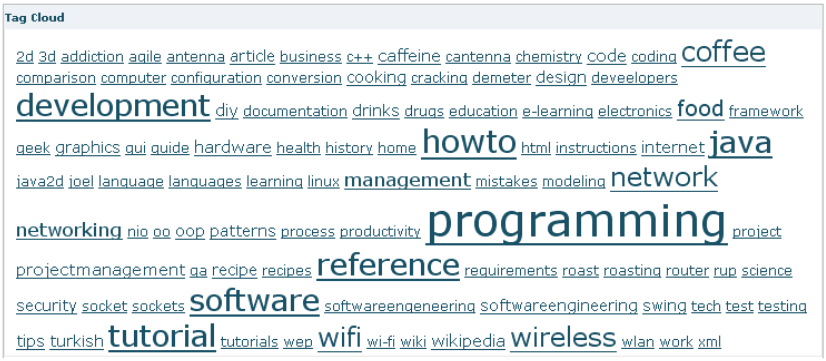

Fig. 4. Topic presentation in PEGECEL as a tag cloud.

5 Retrieval and Ranking of e-Learning Contents

The retrieval process is based on the words that index the cases describing the e-learning contents. The output of retrieval is a list of cases, which has some common words with the user query. The cases in this list are then ranked based on the similarity that they have with the user query. The retrieval algorithm searches the case base for cases indexed by words in the user query.

The ranking process can use three different similarity metrics, depending on the goal of the system (see the experimental work section for a comparison between these metrics). The similarity metrics used are: cosine similarity [9], a standard metric that computes the cosine of the two vectors representing the cases being compared; the word count similarity [9], that counts the number of words in common between the description of the two cases; and the Jaccard similarity [6], which is computed dividing the number of common words in both cases by the number of words in both cases, or simply, "intersection divided by union".

In the list of similar contents, the user can navigate through the topics of the retrieved contents, thus browsing the contents by similar topics (see figure 5).

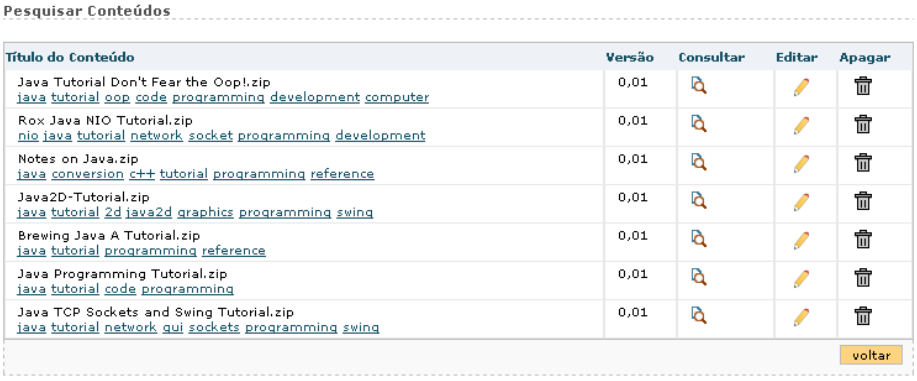

Fig. 5. Content presentation and navigation.

6 Experimental Work

This section describes part of the experimental work performed with PEGECEL, in particular the indexing and retrieval mechanisms. This section presents two types of experiments: indexing experiments, evaluating the capability of the system to suggest categories to the user; and retrieval experiments, evaluating the capacity of the system to find relevant contents. In the experiments, 40 e-learning contents were used, in four different main subjects: Java, Coffee, Networks and Software Engineering. Each one of these subjects has 10 different e-learning contents with a set of categories associated to them. These 40 contents comprise the case base used in the experiments. For each main subject there are several sub topics, which are distributed in the following way: Java 24, Coffee 27, Networks 30, and Software Engineering 38. The average number of topics by e-learning content are: Java 5.6, Coffee 4.8, Networks 6.4 and Software Engineering 7.

6.1 Indexing Experiments

The indexing experiments evaluate the accuracy of the indexing mechanism to suggest the correct categories to the user, when a new e-learning content is added to the system. Remember that the system extracts the categories from two sources: the words in the content documents and the categories of similar contents (using the case similarity metric). For these experiments, 30 new contents (testing contents) were used. These contents were pre-categorized, enabling the comparison between these categories and those suggested by the system. The testing contents were distributed by the same main subjects as the ones in the case base.

Figure 6 presents the average topic suggestion accuracy[6]. There are two important parameters: P1 and P2, which are complementary as their sum is one.

6 $\frac{SuggestedCategories \cap RelevantCategories}{SuggestedCategories \cup RelevantCategories}$

Parameter P1 represents the categories suggested from the similar cases, and P2 represents the importance given to the words extracted from the content. This figure has also a variation parameter, which is the similarity threshold, which is the minimum similarity value for taking a similar case into account. By the graph, it can be concluded that the best results with the current case base are achieved with P1 = 0.9, P2 = 0.1, and the similarity threshold set to 0.05 (the threshold is low, but this is due to the low similarity values between cases, which represent different contents). These experiments were performed using the cosine metric.

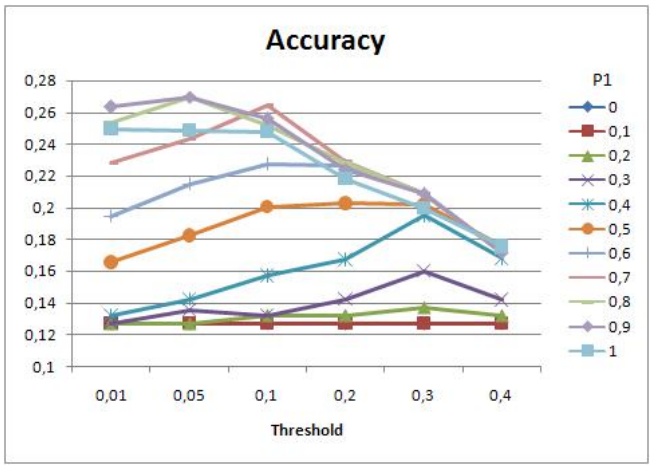

Fig. 6. Average topic suggestion accuracy by similarity threshold and by parameter P1, (average value for the three similarity metrics).

We then tested the similarity metrics: cosine, word count and Jaccard's. The results are shown in figures 7 (threshold = 0.05) and 8 (threshold = 0.4). It can be seen from the values that the best accuracy value occurs using the cosine metric and, with P1 = 0.9 and threshold = 0.05. But with the threshold = 0.4, the best value occurs using the word count similarity with P1 = 0.9.

6.2 Retrieval Experiments

The retrieval experiments are based on 30 queries defined by four different users, within the four domain subjects. These queries were used as retrieval queries to search for the most similar contents. Each one of these queries has a relevant set of cases associated to it, which were selected by the users that defined the queries. Several performance measurements were gathered: average retrieval time (see figure 9), precision values, recall values and F measure values (see figure 10).

p1	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
p2	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	0%
co-seno	12,71%	12,71%	12,71%	13,55%	14,22%	18,25%	21,50%	24,31%	26,92%	26,97%	24,86%
wordcoun	12,71%	12,71%	12,71%	12,71%	12,71%	12,71%	12,71%	14,89%	16,77%	22,33%	18,70%
jaccard	12,71%	12,71%	12,71%	12,71%	13,01%	14,31%	14,31%	14,58%	15,87%	22,34%	22,66%

Fig. 7. Average topic suggestion accuracy by parameter P1 and similarity metric, for a similarity threshold of 0,05.

p1	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
p2	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	0%
co-seno	12,71%	12,71%	13,22%	14,17%	16,79%	17,55%	17,17%	17,17%	17,17%	17,17%	17,55%
wordcoun	12,71%	12,71%	12,71%	12,71%	13,01%	15,42%	17,25%	20,53%	21,93%	23,24%	20,17%
jaccard	12,71%	12,71%	12,71%	12,71%	14,31%	14,31%	14,31%	14,31%	14,31%	14,31%	14,31%

Fig. 8. Average topic suggestion accuracy by parameter P1 and similarity metric, for a similarity threshold of 0,4.

Figure 9 shows the average retrieval times (in seconds) by case base size, on a Intel Dual Core machine with 4 GB of RAM, using Microsoft Windows XP operating system and the Microsoft SQL Server 2005 database. From these values it can be seen that while the cosine and Jaccard's metrics increase the retrieval time with the number of cases in the case base, the word count metric remains stable. Figure 10 the average values for the F measure, where the cosine metric presents the best results, with the word count metric in second place. Figures 9 and 10 show a clear trade-off between accuracy and retrieval time, with word count presenting a good compromise in both aspects.

	Case base size			
	10	20	30	40
cosine	0,1173127	0,1847264	0,2069119	0,2414715
wordcount	0,080001	0,0702972	0,0704938	0,0711947
jaccard	0,3169285	0,5102682	0,5724344	0,7136739

Fig. 9. Average retrieval time by similarity metric and by case base size.

7 Conclusions and Future Work

We have described an approach to e-learning content representation, indexing and retrieval based on Textual Case-Based Reasoning. This work is integrated in

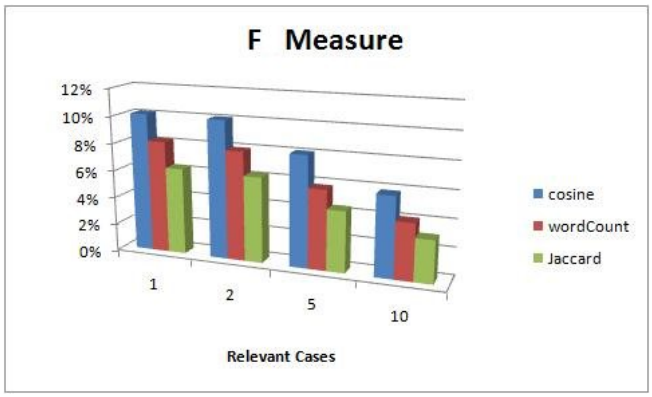

Fig. 10. Average values for the F measure by retrieval set size and similarity metric.

an e-learning platform helping content developers and teachers to reuse course materials. Our first experiments with the indexing and retrieval mechanisms, show a clear importance of the content words in the category suggestion, and a trade off between the cosine and the word count similarity metrics for retrieval. We think we have identified a first mechanism setup for being used in a real environment. Future work also includes the development and exploration of a case representation based on ontologies and Semantic Web technologies. An improvement that we want to explore is the reuse of parts of contents, in special documents and SCOs.

References

1. Lenz, M., Hübner, A., Kunze, M.: 5. textual CBR. In: Case–Based Reasoning Technology: From Foundations to Applications, Springer (1998) 115–137
2. Rosina Weber, K.A., Brüninghaus, S.: Textual case-based reasoning. The Knowledge Engineering Review **20** (2006) 255–260
3. Kolodner, J.: Case-Based Reasoning. Morgan Kaufman (1993)
4. Aamodt, A., Plaza, E.: Case–based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications **7**(1) (1994) 39–59
5. Jackson, P., Moulinier, I.: Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization. John Benjamins (2002)
6. Manning, C., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
7. ADL: The SCORM Overview. ADL (2001)
8. Consotrium, I.G.L.: IMS Learning Design Specification. IMS (2006)
9. Weiss, S.M.: Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer (2005)
10. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review **63** (1956) 81–97