A WordNet based Search Engine for the Moodle Learning Environment

Paper

[1]Valdemar Vicente Graciano Neto, Federal University of Goiás, Brazil.

[2]Ana Paula Ambrósio, Federal University of Goiás, Brazil.

[1]vgracianoneto@gmail.com

[2]ambrosio.anapaula@gmail.com

Abstract: Most Learning Management Systems (LMS) have search engines to help learners find important documents. However, they are not effective in their retrieval task. This happens because LMS Search Engines do not usually treat the query's context, ignoring common information retrieval problems such as synonymy, polysemy and ambiguity of natural language. Furthermore, small and large texts are treated equally, which privileges the larger texts, diminishing the chances of retrieval for small texts. This is especially worrisome in LMS, where an important part of the content is composed of small and medium texts. This paper proposes solutions to these problems presenting a new strategy to recover documents within the Moodle LMS, based on query contextual treatment, document length normalization and automation of some tasks.

Keywords : Moodle, Search Engine, WordNet.

A WordNet based Search Engine for the Moodle Learning Environment

1. Introduction

Leaning Management Systems (LMS) have acquired increasing importance in education, where they offer support to on and off class activities. These environments provide services that offer an efficient support for solid relations and collaboration between students and teachers, including tools that allow knowledge to be created at an individual level to be then transformed into group knowledge. They aim to create a hospitable interaction space, promoting socialization and trust between its members. They offer solutions for the access, categorization and sharing of knowledge, within an environment that facilitates the creation, exchange, retention and reuse of knowledge. These environments introduce a new dimension to relations, allowing members to work as a team, collaboratively, increasing their individual and collective knowledge.

Most LMS offer resources that facilitate and promote social interaction and information sharing. Communication is supported by the use of social software and includes: Email, Instant Messaging, Chats, Forums, Blogs, Wikis, Video Conferencing, etc. In addition, LMS also offers users the option to create data within the environment or upload data and media containing different types of information that must be managed by the environment. Knowledge made available by LMS users, explicitly or implicitly (through different communication methods or by the creation or upload of documents), must be formalized, captured and stored in information repositories.

In a learning context, conversation management influences significantly the knowledge creation process [5]. Conversation is an excellent way of exchanging ideas, experiences, opinions and personal belief, making it a fundamental human activity. As they foster group discussion and induce the sharing of new insights, conversations, if well managed, help in the efficiency of relationships and solicitude among collaborators. Efficient conversations provide a greater level of creativity, stimulating learning.

Conversations in LMS are mostly in text form. Texts are the basic unit of a virtual learning environment and, because of that, accessing them should be a simple and intuitive task. But this is usually not the case. Lewis et al. [6] observed that five out of the nine Learning Management Systems they analyzed presented a search engine to help learners to find documents faster. However, the techniques used to retrieve documents inside these systems are not as good as desirable.

Moodle is an Open Source Course Management System (CMS) [10]. It has become very popular among educators around the world as a tool for creating online dynamic web sites for their students. When Moodle is installed, there is a basic configuration available (with file management system, forum search, grades and activities management, etc.) and it is possible to add functionalities through modules.

Content Management in Moodle – as in most LMS - offers a file management system that organizes information in indexing structures defined by the teachers. College [2] presents a set of steps a teacher needs to perform to, for example, upload a resource in Moodle. It is possible to create a new directory and inside that, put an uploaded file, such as a Power Point presentation, a text, a figure, etc. After uploading a file, it is necessary to

link the resource, adding it to a specific course week, making it visible for student access.

However, some critical problems have been observed in the Moodle environment. Regarding its Search Engine: 1) resources uploaded and posted by the teacher, are not recovered by the Moodle Search Engine - only posts inside forums are retrieved -, 2) retrieval is not satisfactory because it returns only texts where the exact keyword used in the query or a word that contains that keyword appears, 3) synonymy and polysemy aspects are not treated 4) many retrieved texts may not correspond to the learner's expectation, and important texts that are not contained in a forum (like resources available in the course weeks) are not retrieved, 5) complete texts are returned, making navigation through the results a hard task, 6) this navigation difficulty is intensified because of the large amount of documents returned, a problem caused by the inefficient results filter.

This paper proposes solutions to the above problems. The solutions proposed are: 1) all kind of textual information must be retrieved, not only forums posts; 2) contextual treatment of the query, making it more precise and unambiguous; 3) improvement of retrieval results doing a document length treatment and an information retrieval process.

2. Information Retrieval in Learning Environments

In the context of virtual learning environments, Information Retrieval (IR) deals with the representation, organization and access to learning objects [12] - textual information contained in LMS will be called **Textual Learning Objects (TLO)**. When queried, the system should retrieve all the learning objects that are relevant to the learner while retrieving as few non-relevant learning objects as possible. Harman [3] says that achieving this goal is highly dependent on the learning object's metadata and the way the query (retrieval specification) is specified.

The IR process begins with an indexing task, where one or more descriptors are used to classify document content [12]. Most of these descriptors are nouns. Although natural language sentences are composed by nouns, pronouns, articles, verbs, adjectives and connectives, nouns are the grammar class that contains the highest semantic information while other grammar classes have phrasal complementary functions [7].

Terms that express content inside a text are called **concepts.** Indexing techniques are usually based on statistical measures that count the concepts existing in texts mapping them into an index structure called *inverted list* (also called reverse list or inverted index)[1]. To identify concepts, pre-processing techniques may be used such as eliminating *stopwords* - words with no meaning (articles, pronouns, etc.) or any word that does not belong to the class of nouns or cannot act as a noun in any kind of context/sentence - and *stemming* - that aims to identify the root concepts for grouping (Ex. Root of 'generation', 'generating' and 'generate' is 'generat').

Recovery is done consulting the inverted index to identify the texts containing the largest recovery coefficient, which usually includes the largest amount of occurrences of the query terms. A problem with such measures is that these techniques work better in large sized texts. According to Singhal et al, longer documents have a higher chance of being judged relevant to a user-query, compared to shorter documents [11].

Ignoring text size in LMS may be a problem because a considerable part of

texts existing inside these environments are medium and small, like chats, emails, forums posts and, considering this scenario, texts' size are not proportional to the text's importance. Small texts can contain useful definitions, an observation about some subject, a solution to a problem and all this features are important to a learner.

Metadata is data about data [3]. It describes certain important characteristics of its target. In that context, educational metadata describes any feature of the learning object, syntactical or semantic [3]. Metadata can be obtained by a TLO indexing process, for example. One metadata goal is to enable the learners to search and use learning objects.

Syed [12] and Harman [3] suggest using the metadata to describe learning object's content. Harman presents two automatic approaches for representing the content of learning objects (or documents in general): *full text indexing* and *extracted keywords* [3].

In Full Text Indexing [3], document's content is traditionally represented through keywords, which are extracted from the document. The user types a query using just one keyword. Texts that contain exactly that keyword are retrieved. That is, if a TLO has a synonym term in its content (that can satisfy the user information need), but does not contain exactly that keyword, that TLO will not be retrieved.

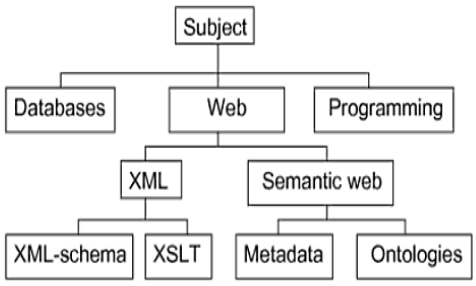


Figure 1: Taxonomy sample about ***learning objects*** *obtained in Harman. [3]*

In the Extracted Keywords approach, Harman proposes the use of metadata to represent documents content (index terms), and a **taxonomy** to represent the content of a course [3]. Typically these taxonomies contain terms describing certain topics. In this approach, the taxonomy is constructed by the teacher and provided to students. It is a set of words organized hierarchically and the learner can use one or more words contained in the taxonomy to enrich [13] his query, that is, including synonyms, antonyms, etc in his query. Then, the retrieval system returns the learning objects having one or more of the keywords that are present in the TLO description. Figure 1 presents a taxonomy sample.

3. A New Moodle Search Engine

We propose the development of a text based system for information retrieval using the electronic dictionary WordNet to improve Moodle Search Engine results. The intent is to provide a tool to improve information access inside a non- structured information environment as Moodle.

WordNet is a lexical ontology where concepts are connected to each other through specific types of relations [9]. Its structure is a graph where the vertices are *synsets* and the edges are relations. Each vertex contains a meaning (more properly called a *sense*) that expresses its semantics, and a list of words that can be used to refer to this meaning. Each synset represents a specific meaning and all the semantic relationships and terms involved in that sense. Thus, each synset defines a conceptual space or domain with its particularities, relevant terms and relations. Furthermore, WordNet can be perceived as an ontology since, besides having various types of lexical relations, the system is based on a taxonomic structure that employs the inheritance relationship (Hyponym / Hypernym).

Besides the inheritance relations, WordNet has several other semantic relations, as synonymy, composition relations (holonym, meronym), etc. Hypernym and hyponym are generalization and specialization relations (inheritance), while meronym and holonym are part-whole relations (composition).

The improvements we propose for the Moodle search engine are: 1) providing the learning object description automatically, selecting just the nouns inside the learning object; 2) indexing process and searching process executed not only inside forums, but for all kind of textual information inside a LMS to facilitate future retrieval; and, finally, 3) a treatment to retrieve distinct texts size, using document length normalization [11].

To treat different text sizes *document length normalization* is used [11]. This means the use of an inversely proportional factor that increases short documents' importance and decreases long documents', balancing their importance. This approach is applied to small and large texts.

WordNet is used in this project to perform three tasks: it provides all possible senses of a keyword – user can disambiguate his query with this - , it indexes the TLOs and builds the taxonomies automatically. So, after learner chooses the meaning closest to the information he needs, WordNet is used to automate taxonomy construction and return other semantically related words. Thus, user can enrich his query – that is, expand it with other words to specify it better.

In the indexing approaches presented by Harman [3], there are some problems that must be considered. Full Text Indexing does not treat synonymy. Extracted Keywords uses a taxonomy constructed by the teacher and needs to be done every time a new course is created. It is a time consuming task and error prone. Often teachers forget to include an important term or do not attain a good detail level in his specification. The solutions proposed treat these problems: synonymy and polysemy have their effects decreased with WordNet sense choosing and taxonomy creation is totally automated, acquitting the teacher of this onus.

3.1. Indexing Process

All the time a new textual learning object is posted or uploaded, the TLO indexing process starts. Text indexing is done using WordNet. The indexing process starts by dividing the text into tokens (units, pieces, words). Each token is checked to see if it is a stopword. To diminish the amount of accesses to WordNet, a list containing the most common stopwords (pronouns, connectors, etc.) is used. The words that do not belong to the stopword list are then submitted to WordNet. All stopwords found in TLOs are discarded. If the token can function as a noun, ie, if WordNet has a meaning for this token as a noun, then the token is added to the index.

The indexing process is performed only once for each text (before beginning to use the search engine), creating an **inverted index**. The reverse list is a plain text file that stores a record of all concept instances of all texts.

This list has the following tuple format:

$$\text{concept} - d_1(p_1), d_2(p_2) \ldots$$

where **concept** is a noun that was found during the evaluation process, d_1, d_2, ..., d_n are documents in which the concept appeared, and p_1, p_2, ... p_n represent the occurrence percentage of that word in each indexed text. The percentage is calculated by the expression:

$$tf_{ij} = q_{ij}/t_j*100$$

where tf_{ij} is the *term frequency factor* or, simply, *tf factor* [1]. This expression represents the term's appearing percentage in document j. tf_{ij} is the term i occurrence rate in document j, q_{ij} is the number of times term i appears in document j and t_j is the total number of concepts recovered from document j. Baeza says tf factor measures how well a term describes a document contents [1].

To illustrate the process, let's consider the following text d1: "It's known that when monks live in cells, their red cells become strange. The cells deliver an electric pulse like a cell-phone battery. This makes them feel like a thief being arrested and put in a cell".

Ten concepts appear in this text (since the word cell-phone is considered two words and plural and singular forms are considered equivalent). The word "cell" appears five times.

The tuples contained in Table 1 would be included in the inverted index. Table 2 shows how to calculate the **tf factor** for the text displayed above.

Table 1. Simulation of inverted index for test example
Battery – d1(0.1) Cell – d1(0.5) Monk – d1(0.1) Pulse – d1(0.1) Phone – d1(0.1) Thief – d1(0.1)

Table 2. Simulation of *tf factor* calculation
tor(battery/d1) = 1/10*100 = 10% tor(cell/d1) = 5/10*100 = 50% tor(monk/d1) = 1/10*100 = 10% tor(pulse/d1) = 1/10*100 = 10% tor(phone/d1) = 1/10*100 = 10% tor(thief/d1) = 1/10*100 = 10%

3.2. Queries

Figure 2 presents the Query Process. In the first step, the user types his query

using one word. The query is directed to WordNet, that executes the query (step 2) and returns all the possible meanings of that word. In the third step, the user chooses his intended sense. In step 4, WordNet generates the taxonomy and all the words semantically related to that chosen sense and returns it to the user. The user, in step 5, chooses the words that describe his information need, enriches his query – disambiguating and contextualizing it -, and passes his query to the search engine, that starts the retrieving process.

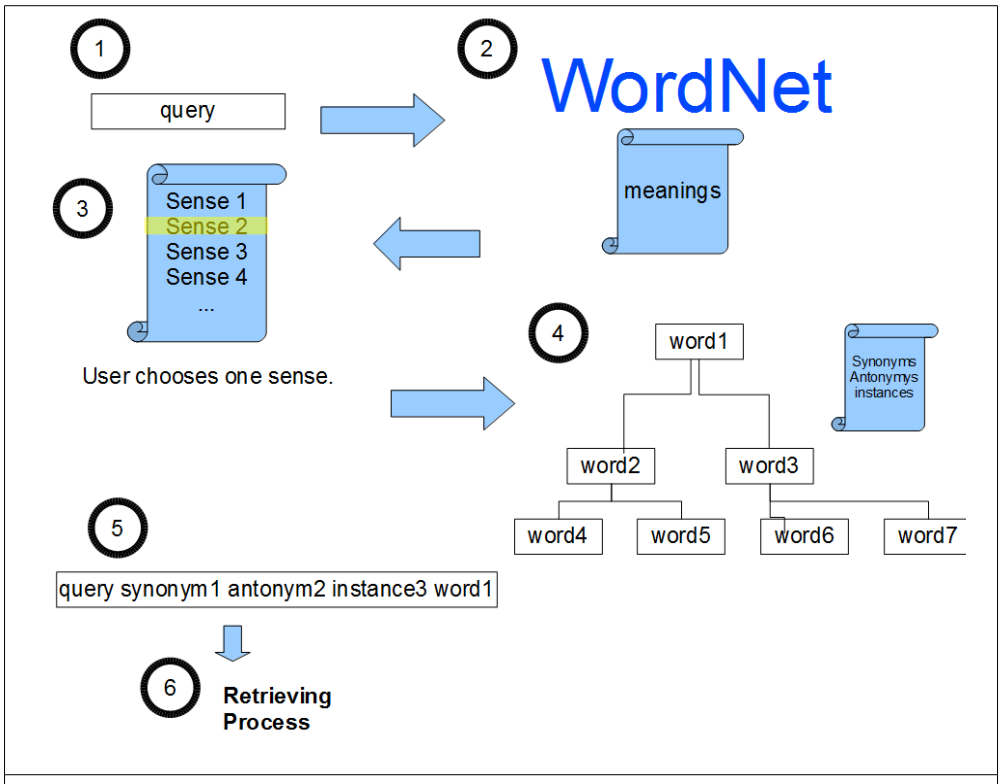


Figure 2: Query process.

3.3. Retrieving Process

The solution proposed for the *document length problem* [11] is to count the amount of words in all documents posted in the LMS and include that number of words in his TLO description/meta-data. Documents (TLOs) which have a word count greater than or equal to a certain threshold, will be considered large texts, and documents which have a word count less than that same threshold, will be considered small texts.

Document recovery starts comparing the text descriptors with the query using an adaptation of vector model [1], which gives weights to the index-terms and recovers documents through a threshold analysis. It uses **tf factor** to indicate the probable relevance of that term in the definition of the text's domain (tf factor will be the vector model weight). Texts that contain the highest term correlation with the user's query are recovered.

The result list contains a single occurrence for each document d_j in which one or more words belonging to the query appeared. A text **relevance index** (**ri**) is defined as:

$$ri = \sum_{i=1}^{n}(tf_{ij})$$

where tf_{ij} is the ratio q_{ij}/t_j . 100, where q_{ij} is the amount of times the term i appears in the text j and t_j is the total of concepts recovered from that text.

To illustrate this procedure suppose a query composed of the words: *prisoner, dungeon, detention, tank*. The system will then search the inverted index for the concepts in the query, recovering the tuples associated to those concepts (Table 3). The bold and underlined tuples $d_i(p_i)$ represent documents associated to more than one word in the query. This is observable in the query.

For example, document d_{113} contains both the word detention - with a percentage of 1.24 and prisoner with a percentage of 0.64. The sum of these percentages will result in the relevance index (ri) for document d_{113}, which is approximately 1.9. Applying the same procedure to d_{71}, its relevance index will be approximately 0.7. Therefore, the most relevant documents for the given query are documents d_{113} (ri ˜=1.9) and d_{79}, with ri ˜= 1.3.

After this analysis, is done the document length analysis. A *normalization factor* [11] is multiplied to the text relevance index. If the text is classified as a short text, its relevance index (ri) is multiplied by four. If the text is classified as a long text, its relevance index is divided by two. This is a suitable normalization factor because most texts relevance index are smaller than 10. When this treatment is done, it can provide equivalence to small and large texts importance. For example: large relevant texts in our results usually have a relevance index around 8.0 and small relevant texts have a relevance index around 1.0. When the normalization factor is applied, both large and small texts acquire similar relevance index (around 4.0).

Table 3: Tuples extracted from the index

Concept	*Ocurrences*
Detention	**$d_{71}(0.39)$,** $d_{72}(0.29)$, $d_{78}(1.06)$, $d_{79}(1.28)$, **$d_{113}(0.64)$**
Dungeon	**$d_{71}(0.19)$, $d_{113}(1.24)$**
Prisioner	$d_{16}(0.54)$
Tank	**$d_{71}(0.09)$**

3.4. Tests

To validate the system, a collection of 122 texts were uploaded to Moodle and indexed. These texts come from different domains, where *cell* appears with distinct senses and other texts where the word 'cell' does not occur. Texts with this word were chosen because "cell" has a notable polysemic feature. It therefore represents a word that generates problems linked to synonymy, ambiguity and morphologic variants, returning many irrelevant texts, depending on the approach applied in the retrieval task.

Miller used a set of 100 documents (subset of Brown corpus) to do his recovery experiments [8]. So, this is a good amount of texts to apply in our tests with Moodle.

According to WordNet, the word *cell* can have seven different meanings according to the domain in which it is used (Table 3): small compartment, organisms structural unit, electric cell, cadre, mobile, monks and nuns room, jail.

When the user submits the query to the system with the keyword *cell*, all the seven meanings of *cell* are returned to the user that will then choose the meaning that represents that domain he is interested in. The user chooses the best sense and it is returned to him the correspondent taxonomy and other words with semantic relations like synonymy, antonymy, composition, etc. This new query is submitted to the text base that calculates the relevance index of the recovered documents and returns the most relevant documents found (less or equal than 10 texts).

Table 4 shows a comparison between Moodle Search Engine results and the prototype results. These results are presented this way because the prototype does a contextual treatment, that is, while the keyword is applied in Moodle Search Engine and all possible results are retrieved and showed, the prototype works in that way described in query process and retrieving process: a specific group of texts is returned for each sense of the word cell since the user chooses the intended. There were ten relevant texts for each sense of cell within the repository.

Table 4: Comparison between Moodle Search Engine and Prototype Search Engine using the key word ***cell****.*

Parameter	*Moodle Search Engine Results*	*Prototype Results*
Results amount	82 texts retrieved	Honeycomb : 6 texts retrieved
		Biological : 6 texts retrieved
		Eletric cell : 7 texts retrieved
		Political Movement Core: 6 texts retrieved
		Mobile Phone: 4 texts retrieved
		Religious small room: 9 texts retrieved
		Jail, prison cell: 4 texts retrieved
Relevant items	Around 10, not necessarily in the first positions.	All the recovered.
Observations	Around 87,5% are not relevant to a specific sense of *cell.* Just texts forums texts were returned. Texts with the substring *cell* were returned, like a text about history that contained the word *Vecellio*.	Every document returned is relevant.

4. Conclusions

The results show that information recovery systems based on WordNet inside Moodle is an interesting solution to meet learner's information needs.

Query results presented by the prototype were more relevant because the proposed system does a semantic treatment on the query, automates the taxonomy construction, automates the TLO descriptors obtaining (indexing process), returns TLOs as near as possible of the learner information need and treats document length problems, equaling their relevance.

Finally the advantages that search engines can brings to the learning process

must be considered. Gary Small, in a recent research, concluded that searching the Web may help stimulate and possibly improve brain function [4]. So, provide a good search engine in a LMS as Moodle can contribute to learner's brain activity.

5. References

[1] BAEZA-YATES, R.; NETO, B. R. **Modern Information Retrieval**, volume 1. 1999.

[2] COLLEGE, G. A. **Moodle File Management**. Available in: <https://gustavus.edu/gts/Moodle_File_Management>. Access date: May 30 2009.

[3] HARMAN, K. **Learning Objects: Standards, Metadata, Repositories, and LCMS**. Published by Informing Science, 2007, ISBN 8392233751, 9788392233756, 395 pages. Used: 45 - 59).

[4] HASKINS, W. **Researchers Navigating the Web Boots Up Your Brain**. Available in: <http://www.technewsworld.com/story/Researchers-Navigating-the-Web-Boots-Up-Your-Brain-64830.html>. Access date: May 24 2009.

[5] KROGH, G. v., ICHIJO K, NONAKA I. **Enabling Knowledge Creation**. Oxford University Pres Inc. 2001.

[6] LEWIS, B. A.; MACENTEE, V. M. **Learning Management Systems Comparison**. Information Science + IT Education Conference Proceedings 2005. Available in: <http://proceedings.informingscience.org/InSITE2005/P03f55Lewis.pdf>. Access date: May 23 2009.

[7] MILLER, G. A., Ed. **WordNet: an on-line lexical database**. *International Journal of Lexicography* 3, 4 (Winter 1990), 235-312.

[8] MILLER, G. A.; LEACOCK, C.; RANDEE, T.; BUNKER, R. T. **A semantic concordance**. Cognitive Science Laboratory, Princeton University. ACM Portal. 1993.

[9] MILLER, G. **WordNet : A lexical database for the english language**.. On Communications of the ACM. 1995.

[10] MOODLE. **About Moodle**. Available in: <http://moodle.org/>. Access date: May 30 2009.

[11] SINGHAL A, SALTON G., MITRA M, BUCKLEY C. **Document Length Normalization**.. Information Processing and Management, Vol. 32, No. 5, pp. 619-633, 1996. Printed by Great Britain.

[12] SYED, R. M. **Strategic Applications of Distance Learning Technologies**. Chapter: Information Retrieval in Virtual Universities. Idea Group Inc (IGI), pp. 196 – 204. 336 pages. Year: 2008.

[13] TOMASSEN S. L. **Research on Ontology-Driven Information Retrieval**. Proceedings on Workshop on Ontology Content and Evaluation in Enterprise (OnToContent), OTM Workshops 2006. 1460-1468.