Searching and Ranking Educational Resources based on Terms Clustering

Marina A. Hoshiba Pimentel, Israel Barreto Sant’Anna and Marcos Didonet Del Fabro

C3SL Labs, Informatics Department, Federal University of Paraná, Curitiba, Brazil

Keywords: Educational Resources, Terms Clustering, Ranking.

Abstract: Open Educational Resources (OER) are important digital assets used for teaching and learning. There exists different repositories, but searching for such items is often a difficult task. On one hand, most part of the solutions implement engines with syntactic search based on term frequency metrics, or using only item’s metadata. On the other hand, the utilization of terms clustering (TC) have been used in other search and ranking contexts and they have shown to be effective. In this paper, we present an approach for searching and ranking for Open Educational Resources within a repository of objects, defining a set of tasks and an hybrid metric that integrates different ranking metrics obtained through terms clustering with the results of existing search engines (SE). We present an extensive implementation and experiments to validate our approach. The results empirically showed that our approach is effective to rank relevant OERs.

1 INTRODUCTION

We are living in an era of information abundance, available mostly through the internet. Digital repositories with their set of documents organized and available electronically are also part of this source of information (Lagoze et al., 2006). However, the great volume has implications in the process of organization, representation and management of all this variety of contents. The format and number of information has a direct impact on their retrieval process. It is not a trivial task to categorize them adequately to enable relevant information retrieval for those who are searching them in a digital environment (Aguiar et al., 2014).

In the context of Education, the searching and selecting relevant Open Educational Resources (OER) in digital repositories has been an exhausting and arduous task for teachers (Pontes et al., 2014). Large repositories can contain tens of thousands of different learning objects, making it difficult to find relevant objects. OER retrieval is usually a difficult task, mainly due to implementations of search algorithms based on metadata or keywords, which are common in these repositories. These techniques limit further the syntactic search process (de Souza et al., 2008). (Costa et al., 2013) shows how challenging is for teachers the search and selection of OERs available in digital repositories. The study shows that although search engines are heavily used, irrelevant content is returned to teachers.

In studies carried out by (Coelho et al., 2012), it was possible to verify that the search engines and the existing digital repositories present difficulties in the OER recovery process. The identified difficulties are long result lists, few relevant and often poorly ranked results, that reinforces the need to create an appropriate mechanism for OER recovery using other resources to facilitate the search process, such as the use of tags or keywords. It is worth mentioning that tags clustering has been exploited to improve search, navigation and recommendation services used on the internet (Gemmell et al., 2008)(Shepitsen et al., 2008)(Rafailidis and Daras, 2013)(Liu and Niu, 2014).

Existing solutions ((de Souza et al., 2008)(Patrocínio and Ishitani, 2009)(Costa et al., 2013)) show that the search services implemented in these repositories have limitations that return few meaningful results. Among the limitations we can highlight problems such as strictly syntactic search and those based only on metadata analysis (de Souza et al., 2008). In addition, if the search result is not well ranked, the problem is aggravated because, according to the surveys, users usually analyze only the first result page or the first ten results obtained (Silverstein et al., 1999).

For these reasons, the main objective of this work is to present a search model of OER in digital repositories that will handle the restriction of the syntactic


search, as well as to apply a good ranking to improve
the relevance of the returned resources.

The main contributions are a search model that combines the use of a well-known search technique/engine with a search process based on terms clustering, and a set of metrics adapted to OER to be integrated with search engines. We present a set of new metrics that are improvements from existing works, adapted to OERs. They have shown to be effective in a set of experiments realized in a real-world setting searching in a repository with 19,159 items, resulting in broader, diversified OER set with more relevant and better ranked items. The implemented components are integrated within the MECRED Portal.

This paper is organized as follows. First, we present the background for relevance calculation, data clustering and educational resources. Second, in Section 3 we present our approach for integrating a syntactic search with terms clustering, adapted for OERs. Section 4 contains the experiments. We finalize with the related work and conclusions.

2 BACKGROUND

Information retrieval (IR) is the process of finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored in computers) (Manning et al., 2008).

There are several components in an IR workflow, and the relevance calculation of the items is one central aspect. In this section, we introduce the base concepts of relevance calculation, data clustering and educational resources, which are the objects of study from our work.

2.1 Relevance Calculation

A search procedure for an IR system is developed whereby the answer to a question is obtained by maximizing an evaluation function of the system’s output in terms of the probability of relevance (Goffman, 1964). In repositories with large number of documents, the result of a search can return a number of documents that can easily exceed the human capacity to filter them, so it is essential that a search engine ranks the documents properly.

A searching procedure that allows users to type free text without using any type of operator (such as booleans) are popular on the web, and they handle the query as a set of words, calculating a weight of the terms that match the search terms (Manning et al., 2008).

Relevance can be calculated in a number of ways as Neural Networks (Benediktsson et al., 1990), Natural Language Processing (Blosseville et al., 1992), Boolean models (Lee et al., 1994) and Support Vector Machines (Thet et al., 2007).

One well-known and widely used weight assignment scheme is the term frequency. Denoted as \( TF_{t,d} \), it represents the number of occurrences of a term \( t \) in a document \( d \). The simplest approach is to assign to the weight the number of occurrences of the term \( t \) in document \( d \) (Manning et al., 2008). The more frequent, the more relevant.

\( TF \) as above defined, presents a critical problem: all terms are considered equally important. In fact, some terms have little or no discriminatory power to determine relevance. To mitigate this problem, the document frequency is adopted and is denoted as \( DF_t \). \( DF_t \) denotes the number of documents in the collection containing the term \( t \) and \( N \) is the number of documents in the collection. To adjust the term weight using the measure \( DF \), the inverse of the frequency in the documents is defined (\( IDF_t \)) as shown in Equation (1).

\[
IDF_t = \log \frac{N}{DF_t} \tag{1}
\]

\[
TF-IDF_{t,d} = TF_{t,d} \times IDF_t \tag{2}
\]

Term Frequency - Inverse Document Frequency (TF-IDF) calculates the relative frequency of terms in a specific document compared to the inverse proportion of that term over the entire document corpus (Ramos, 2003). This calculation determines how relevant a given term \( t \) is in a particular document \( d \) given by Equation (2). The TF-IDF assigns to term \( t \) a weight in document \( d \) that is (i) highest when \( t \) occurs many times within a small number of documents; (ii) lower when the term occurs fewer times in a document, or occurs in many documents and (iii) lowest when the term occurs in virtually all documents (Manning et al., 2008).

The table 1 represents an example of a TF-IDF calculation for four terms (car, auto, insurance, best) in three documents (\( d1, d2 \) and \( d3 \)) in a collection composed of 806,791 documents. The \( DF \) column denotes the number of documents in the collection in which each term occurs. In this way we can calculate the inverse of the frequency in the documents (Equation (1)) represented in the \( IDF \) column. The frequency of terms in each document is represented in the columns \( TF \). We can then calculate the weight given by TF-IDF (Equation (2)) for each term in each
of the documents as shown in the table. For example, the term car has a weight equal to 44.5 for the document d1; 6.6 for document d2 and 39.6 for document d3.

Table 1: TF-IDF calculation example.

<table>
<thead>
<tr>
<th>Term</th>
<th>DF</th>
<th>IDF</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>18.16</td>
<td>1.6</td>
<td>27</td>
<td>4</td>
<td>24</td>
<td>44.5</td>
<td>6.6</td>
<td>39.6</td>
</tr>
<tr>
<td>auto</td>
<td>6.72</td>
<td>2.0</td>
<td>3</td>
<td>33</td>
<td>0</td>
<td>6.2</td>
<td>68.6</td>
<td>0</td>
</tr>
<tr>
<td>insur</td>
<td>19.24</td>
<td>1.6</td>
<td>0</td>
<td>33</td>
<td>29</td>
<td>0</td>
<td>53.4</td>
<td>46.9</td>
</tr>
<tr>
<td>best</td>
<td>25.23</td>
<td>1.5</td>
<td>14</td>
<td>0</td>
<td>17</td>
<td>21</td>
<td>0</td>
<td>25.5</td>
</tr>
</tbody>
</table>

2.2 Data Clustering

Clustering algorithms group a set of objects into subsets or clusters, creating clusters that are coherent internally, but clearly different from each other (Manning et al., 2008). In other words, objects within a cluster should be as similar as possible (Aggarwal and Reddy, 2013).

Networks of nodes and links are powerful representations of datasets of interactions from a great number of different sources (Bohlin et al., 2014). One of the most relevant features of graphs representing real systems is community structure, or clustering, i.e. the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters. Such clusters, or communities, can be considered as fairly independent compartments of a graph (Fortunato, 2010).

We live in the era of Big Data and fortunately we have several tools for classifying data from many different sources. The challenge is to extract useful information. Therefore, tools for simplifying and highlighting important structures in networks are essential and such tools are called community detection methods and they are designed to identify strongly interconnected modules (Bohlin et al., 2014).

2.3 Educational Resources

Open Educational Resources describes any educational resources that are openly available for use by educators and students, without any need to pay royalties or licence fees (Butcher, 2015). A repository is a database or collection of OER.

Metadata, in the context of digital repositories, is the information about a given object. As the number of objects grows exponentially, the lack of information or metadata about objects places a critical and fundamental constraint on the ability to discover, manage, and use objects (Committee et al., 2002). A metadata instance for a learning object describes relevant characteristics of the learning object to which it applies. Such characteristics may be grouped in general, life cycle, meta-metadata, educational, technical, educational, rights, relation, annotation, and classification categories (Committee et al., 2002). Table 2 shows a simple example of an OER and its metadata.

Table 2: Simplified example of an OER.

<table>
<thead>
<tr>
<th>Metadata field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dc.contributor.author</td>
<td>Mooddigger</td>
</tr>
<tr>
<td>dc.date.created</td>
<td>2005-11-07</td>
</tr>
<tr>
<td>dc.description</td>
<td>Provides a close-up view of the constellation of Sagittarius</td>
</tr>
<tr>
<td>dc.title</td>
<td>Milky way 2</td>
</tr>
<tr>
<td>dc.type</td>
<td>Image</td>
</tr>
<tr>
<td>dcRights.license</td>
<td>Creative Commons Attribution ShareAlike 2.5 License</td>
</tr>
<tr>
<td>dc.subject.keyword</td>
<td>Astronomy</td>
</tr>
<tr>
<td>dc.subject.keyword</td>
<td>Constellation</td>
</tr>
<tr>
<td>dc.subject.keyword</td>
<td>Sagittarius</td>
</tr>
<tr>
<td>dc.subject.keyword</td>
<td>Space</td>
</tr>
<tr>
<td>dc.subject.keyword</td>
<td>Star</td>
</tr>
<tr>
<td>dc.subject.keyword</td>
<td>Universe</td>
</tr>
<tr>
<td>dc.subject.category</td>
<td>College education</td>
</tr>
</tbody>
</table>

3 SEARCHING AND RANKING OERs

This section presents our approach for searching and ranking OERs supported by terms clustering and TF-IDF searching. Our goal is to improve the search results of an existing search engine by increasing the number of relevant OERs related to the search terms. It is worth noting that in this work we are referring simply as terms all the keywords or tags that are associated with a given OER.

Figure 1 shows an overview of the tasks and inter-task flows required to perform an OER search and ranking supported by terms clustering. There are two large groups of processes, named Clustering Process and Search and Ranking Process. The main goal of the tasks from the first group Clustering Process is to form the terms clustering. And the tasks from the second group aim to perform the OER searching and ranking based on the cluster of terms resulted from the first group.

3.1 Clustering Process

The task Extracting OERs and its terms is responsible for extracting the list of all OERs and its respective terms. Suppose that an repository contains the OERs and the respective terms given by $S_r = \{ r_1 : ...$
The development of an clustering algorithm is not part of this work, therefore to perform the task **Terms clustering** we will use an available tool, the **Map equation/Infomap**, to form our term clusters from the graph $G$ generated on the previous task.

Map equation/Infomap (Bohlin et al., 2014) is a fast stochastic and recursive search algorithm, that follows closely the method presented by (Blondel et al., 2008), a heuristic method based on the optimization of modularity. Neighboring nodes are joined into modules, which subsequently are joined into supermodules and so on. First, each node is assigned to its own module. Then, in random sequential order, each node is moved to the neighboring module that results in the largest decrease of the map equation. If no move results in a decrease of the map equation, the node stays in its original module. This procedure is repeated, each time in a new random sequential order, until no move generates a decrease of the map equation.

Our term clusters are based on term pair co-occurrence and consist of undirected graph, from which we obtain clusters of similar and ranked terms. This way every cluster can offer additional semantically similar terms which users might not have thought at their search. With the formation of the clusters the tasks from the Clustering Process are concluded.
3.2 Searching and Ranking Process

The Searching and Ranking Process begins when a user enters a term or a set of terms to be searched. First, the search engine is invoked in the task **Searching via search engine** to perform the search for the terms the user has provided. At the end, a result set $RS_{se}$ of ranked OERs is returned. It is not the purpose of this work to develop a search engine, so we must use an available tool that accomplishes this functionality.

The task **Searching via terms clustering** performs the search supported by the clusters of co-occurrent terms provided at Subsection 3.1. The task first identifies which cluster the term belongs to, then the correlated terms are recovered. This way quantitative and semantic expansion can be achieved. Quantitative because it increases the number of search terms, and semantic because it considers similar and correlated terms. So it is possible to retrieve different OERs from those retrieved by the search engine.

Once the clusters of terms are formed, they do not change, but to calculate the found OER relevance and ranking for a specific search, we consider the term corresponding to the original search term the more relevant element in the cluster for this search. We denominate it main term. The relevance of the other correlated terms are normalized relatively to the main term. Their weights will be inversely proportional to the distance (difference) that they are in relation to the main term in the cluster. The greater the distance, less relevant the term.

The original terms weights in a cluster may suffer wide range of magnitudes as shown in Table 3. Simply calculating the difference between the original weights would not allow new weights to be obtained on a practical scale (e.g., between 0 and 1). To circumvent this problem we use the logarithmic scale.

First we obtain the measure of the distance between the main term and the other terms of the cluster, given by $dist_{t_{m}t_{i}} = |\log_{2}rank_{i} - \log_{2}rank_{m}|$, where $rank_{m}$ and $rank_{i}$ are the values of the original weights assigned to the terms $t_{m}$ and $t_{i}$ respectively. The lower the result's value $dist$, the closer and more similar the terms are considered. Besides that, let $max_{dist}$ be the value of the maximum distance between $t_{m}$ and $t_{i}$. The relevance (weight) of the term $t_{i}$ should be inversely proportional to the value of its distance from the main term $t_{m}$. To evenly distribute the distances in a scale, their values are normalized with the Min-Max Normalization, multiplying the $max_{dist}$ value by a scalar $1 + a$, avoiding that the most distant term to have relevance equals to 0, whereas it belongs to the same cluster as the searched term. The resulting equation is given by Equation (3) (the minimum distance possible is 0, so it's not represented in the equation).

$$rank_{i} = 1 - \left( \frac{dist_{t_{m}t_{i}}}{max_{dist} \times (1 + a)} \right) \quad (3)$$

Table 3 shows an example of this calculation, setting $a$ up to 0.5. Consider the search term “Sagittarius.” The first column shows all co-occurring terms. The second column shows the original weight of the terms and the third column shows the weight normalized by Equation (3).

Once the weights of the terms have been normalized, we can finally perform the search for the OERs. Based on the TF idea, for each OER found, the score is calculated as the sum of the weights of its co-occurring terms: $Rank_{OER} = \sum_{i=1}^{n} rank_{i}$, where $n$ is the number of cluster terms that the OER owns and $rank_{i}$ is the normalized weight of the term $i$.

At the end of this task, the result set $RS_{ct}$ is returned with all OERs ranked according to the total score of their terms. Consider the cluster which the term “Sagittarius” belongs to as shown in Table 3. Consider yet that the OER “Stars and HR Diagram” own the co-occurrent terms “Hertzsprung Russell Diagram,” “Satr,” “Dwarfstar” and “Luminous star”, so your score is given by $Rank_{OER} = 0.70 + 0.50 + 0.99 + 0.98 = 3.17$.

We start now the last task OER mix and reclasification. The input for the task are the two set $RS_{se}$ and $RS_{ct}$ resulted from the tasks **Searching via search engine** and **Searching via terms clustering** respectively. Since the weights from the two sets have different dispersions, a normalization of the OERs weights is made to adjust the scale of values, so enabling to join the two sets. To do this, an equation based on linear normalization is used. In the case of the result generated by the search engine, the normalization of the OERs score is given by Equation (4), where $min_{sc}$ and $max_{sc}$ are respectively the minimum and

---

Table 3: Co-occurrent terms for “Sagittarius”.

<table>
<thead>
<tr>
<th>Id</th>
<th>Term</th>
<th>Original weight</th>
<th>Normalized weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sagittarius</td>
<td>2.3147e+05</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>Dwarfstar</td>
<td>2.4271e+05</td>
<td>0.99</td>
</tr>
<tr>
<td>3</td>
<td>Luminous star</td>
<td>2.4271e+05</td>
<td>0.98</td>
</tr>
<tr>
<td>4</td>
<td>Peony</td>
<td>3.1146e-05</td>
<td>0.93</td>
</tr>
<tr>
<td>5</td>
<td>Shine</td>
<td>3.1146e-05</td>
<td>0.93</td>
</tr>
<tr>
<td>6</td>
<td>W5</td>
<td>1.5587e-05</td>
<td>0.91</td>
</tr>
<tr>
<td>7</td>
<td>Stars movement</td>
<td>1.4669e-05</td>
<td>0.89</td>
</tr>
<tr>
<td>8</td>
<td>The Cartwheel Galaxy</td>
<td>1.1827e-05</td>
<td>0.84</td>
</tr>
<tr>
<td>9</td>
<td>Recycle</td>
<td>1.1196e-05</td>
<td>0.83</td>
</tr>
<tr>
<td>10</td>
<td>Protoplanetary disk</td>
<td>6.4775e-06</td>
<td>0.71</td>
</tr>
<tr>
<td>11</td>
<td>Hertzsprung Russell Diag</td>
<td>6.1313e-06</td>
<td>0.70</td>
</tr>
<tr>
<td>12</td>
<td>Star</td>
<td>0.00021569</td>
<td>0.50</td>
</tr>
</tbody>
</table>
maximum scores found in the set $RS_{sc}$, $d$ is a coefficient to avoid values equal 0 and $boost$ is the impulse factor necessary to controlling the resources relevance in the ranking process. It’s a value given by the search engine to OERs depending on the occurrence of the exact searched terms (higher value) or syntactically close terms (lower value).

$$R_{m,sc} = boost \times \left( \frac{sc \times (1 + d) - min_{sc}}{max_{sc} - min_{sc}} \right)$$

(4)

The normalization of the resulting set $RS_{tc}$ generated by the support of the terms clustering is made by Equation (5), where $min_{sc}$ and $max_{sc}$ are respectively the minimum and maximum scores found in the set $RS_{sc}$, $d$ is a coefficient to avoid values equal 0 and $m_{boost}$ is the maximum boosting value $boost$ from the set $RS_{se}$. The major differential of this proposal is situated at this point, since the maximum boosting value ($m_{boost}$) is applied in the second set too, to boost the score of its relevant OERs, just like it is done in the search engine. Based on the premise that the OERs of the $RS_{tc}$ set were returned only because they are considered similar by the use of correlated terms, so it is reasonable to apply $m_{boost}$ to the OERs from $RS_{sc}$, so that there is no harm in their relevance in relation to the items from $RS_{sc}$ set.

$$R_{n,tc} = m_{boost} \times \left( \frac{sc \times (1 + d) - min_{sc}}{max_{sc} - min_{sc}} \right)$$

(5)

Next, another important measure adopted is based on other research findings: resources returned by both the search engine and the terms clustering should be considered more relevant. So the final OER score for OER that occurs in both $RS_{se}$ and $RS_{tc}$ is given by the sum of the scores assigned to the OER in these both sets.

Finally, the final result set $RS_{mix}$, that merges results from $RS_{se}$ and $RS_{tc}$, with a new ranking can be returned to the user. Table 4 shows a comparative example, where we can see the original ranking from the resulting sets $RS_{tc}$ and $RS_{se}$, as well the mixed and re-ranked final set $RS_{mix}$.

To summarize, the presented model increases the search results through the expansion of the original search terms given by the correlated terms identified by the clustered co-occurrence terms. In addition it combines the results found via terms clustering with the results from the traditional search engine. All the results are reclassified, so that greater relevance is attributed to the results that appear in both search approaches, as well as boosting the results obtained through the correlated terms.

## 4 EXPERIMENTS

The presented solution was implemented using the infrastructure of the MEC RED web portal of educational objects with mechanisms of social network, sending, searching and ranking objects, as well as mechanisms to follow collections and authors. At the experimental phase of this work, the portal repository contained 19,159 OERs and 23,808 terms. This portal uses only free and open source software\(^3\), in order to guarantee the dissemination of the knowledge produced and the possibility of cooperation in the construction of the platform itself\(^4\). OERs are stored in a DSpace\(^4\) portal and the social network information is stored in an instance of PostgreSQL\(^5\). The main search is performed by the Elasticsearch\(^6\) engine. The front end layer receives user connections and communicates with a Ruby/Rails API providing information needed by the users, and communicating with the Elasticsearch engine through the SearchKick API\(^7\).

To obtain the cluster of terms in the task **Terms clustering**, the Mapequation framework\(^7\) (Bohlin et al., 2014) was adopted. It is a set of tools for data clustering and visualization. From the graph generated in task **Graph generation**, represented in the specific format called PAJEK (.net), we used the Infomap tool to calculate and generate the clusters. Infomap, besides generating the clusters, provides the ranking of the terms of each cluster. The result is stored in the output file (.freet), being the basis of information to support the **Searching via terms clustering** task. Of the total of 23,807 terms and 173,038 identified co-occurrences, Infomap generated 8,568 clusters. Table 5 shows a sample of clustered terms with respective weights.

It is important to highlight the use of the explain parameter in Elasticsearch execution, because

---

\(^3\) Component sources: https://gitlab.c3sl.ufpr.br/portalmec/

\(^4\) Component sources: https://gitlab.c3sl.ufpr.br/portalmec/portalmec/tree/tag-clustering-task

\(^5\) www.dspace.org

\(^6\) www.postgresql.org

\(^7\) elasticsearch.co

\(^8\) www.mapequation.org
this way we can capture the boost value applied in the ranking process performed by the search engine. Remembering that this value is important to be applied also in the task OER mix and reclassification to boost relevant OERs found via terms clustering.

All experiments were performed with terms randomly chosen and were performed considering the retrieval of a list of at most ten results, because according to (Silverstein et al., 1999), users usually consider only the first page or the first ten results.

Table 6 shows the resulting set for the searching term “Sagittarius” performed via Searching via terms clustering. The column Term id represents the terms from the Table 3 owned by each OER. These terms are used to calculate the total weight of each OER, that is used to rank the OER resulting set. It is worth mentioning that in the Equation (3), according to our experiments, the value of a set to 0.5 provided satisfactory results.

Table 7 represents the resulting set of OERs found and ranked via Elasticsearch search engine. The columns Weight and Boost shows respectively the weight and the boost value assigned to the OER from the search engine. The maximum boost value returned was equal 10.

Considering results from Table 6 and 7, the final resulting set after performing the task OER mix and reclassification is showed in Table 8. In this experiment, only the first OER found via Elasticsearch can be considered relevant, therefore the final mixed resulting set, column Mixed Result, is composed by only one item from Elasticsearch and all other from the items found via terms clustering approach.

Table 9 shows the consolidated results of 10 experiments performed with terms randomly chosen. The first column shows the search terms used to evaluate the proposed model. The Original result column is composed by TC (number of OERs found via terms clustering) and SE (number of OERs found via search engine). The column Final result composition shows the composition of the final mixed result for the search. The column TC' shows the number of OERs from column TC that compose the final result, and the column SE' the number considered from SE. The
column TC+SE refers to the number of OERs that are returned by both (TC and SE).

Analyzing the search term corrosion from Table 9, we can interpret that 4 OERs were found via terms clustering, of which 1 compose the final mixed result; 10 OERs were found via search engine, of which 7 compose the final mixed result; and from the total 10 final result set, 2 were found by both, via terms clustering and search engine. In this case, we can consider that 30% \((3 = 1 + 2)\) of the OERs came from terms clustering. In 60% of the cases, the final mixed result is composed of at least half of the items returned via terms clustering (considering TC' and TC+SE). In 40% of the cases, where the search engine recovers few relevant results (because it considers only the main search term), the results via terms clustering compose more than 80% of the final result.

Considering the correlated terms to carry out the expansion of the original query made it possible, in some cases, to more than double the percentage of relevant results, as the experiments with term “Sustainable development”, “Phylogeny”, “Morphine” and “Sagittarius”. For other cases, where the search engine itself returns a good amount of relevant results, the evaluation is even more empirical and subjective, since only a specialist or a user with specific search purposes could evaluate with greater accuracy the relevance of the OERs as ranked.

(Knautz et al., 2010) presents a model based on the presentation of a tag cloud, where the user must click on the terms or the edges to access the documents related to the specific tag. In this proposal, a teacher would need to spend much more time and effort to get the results that in our approach are returned with just one query. Taking the example of the experiment with the term “DNA”, which has 11 correlated terms, the user would need to perform 12 queries in the model proposed by Knautz et al. to obtain similar results to those obtained by our model.

The ranking process, simply summing the scores of correlated terms, was also viable and presented good results. Less relevant results returned by the search engine were treated with little relevance in the final classification in relation to OERs considered more relevant because of the correlated terms.

5 RELATED WORK

(de Souza et al., 2008)(Patrocínio and Ishitani, 2009)(Costa et al., 2013) report some difficulties in the OER search process in digital repositories, such as searching process with syntactic restrictions, making it difficult to find relevant results. (de Souza et al., 2008) proposes a general purpose thesauri-based approach to semantic retrieval of learning objects. This model faces a practical limitation, which is the scarcity of thesauri. Using more generic thesauri, not specific to a particular knowledge area, often require the implementation of semantic similarity analysis techniques or the combination of the use of thesauri with ontologies.

(Patrocinio and Ishitani, 2009) proposes a mechanism for learning objects recovery, based on a directory service that integrates metadata used by the main Brazilian repositories and social annotation resources. The proposed model does not address the question of the ranking of the OERs.

A clustering framework called RankClus is proposed in (Sun et al., 2009) that generates clusters integrated with ranking. This work shows that ranking objects globally without considering which clusters they belong to often leads to dumb results.

The design of an information retrieval system based on tag co-occurrence and subsequent clustering is presented in the work of (Knautz et al., 2010). This system allows users to access digital data through a graphical/visual retrieval interface, providing an elaborate alternative to the conventional tag clouds. In addition, for these authors, tag clusters represent a new form of visualization-driven query expansion and thus to a new possibility of the application of human-computer interaction research in web-based information retrieval. This expansion happens as the user navigates through the vertices (tags) or the edges (relationship between tags) of the tag cluster. This need for interaction to compose the query may require a lot of time and effort from the user.

The ranking in (Knautz et al., 2010) is calculated in two ways: (i) absolute frequency of all tags is accumulated creating the ranking; (ii) with the Within Document Frequency, which takes the logarithms of the
The application of simple equations. Irrelevant results returned by the search engine were properly treated with little emphasis in the final classification, considering that other more relevant OERs were returned by the search. This way, we have concluded that it is not necessary to use complex models and calculations to obtain improvements in ranking. The equation results were normalized in order to not prioritize some specific search results. The relevance of the ranking was based on empirical analysis of the returned objects. Further analysis should be done to evaluate through some existing data set, if available. The main contribution of our presented approach is the increment of the number of OERs found by a searching process, as well as ranking the result set considering the relevance of all correlated terms.

As future work we can mention the application of Natural Language Processing techniques such as radicalization, lemmatization, removal of stopwords and disambiguation to improve the quality of the terms clustering and, consequently, to improve the search results. Another point that deserves additional research is the techniques to perform OER ranking, since depending on the applied equations and methods one can classify the final results in many different ways.

ACKNOWLEDGEMENTS

This work was funded by project (Pesquisa de redes sociais em nuvem voltadas para objetos educacionais), FNDE (Fundo Nacional de Desenvolvimento da Educação) and CNPq Universal.

REFERENCES


of the 15th annual international ACM SIGIR, pages 51–58. ACM.


Butcher, N. (2015). A basic guide to open educational resources (OER). Commonwealth of Learning (COL:).


