Automatic classification of documents in cold-start scenarios

Conference Paper · June 2013

DOI: 10.1145/2479787.2479789

CITATIONS
2

READS
244

5 authors, including:

Ricardo Kawase
Forschungszentrum L3S
69 PUBLICATIONS **609** CITATIONS
SEE PROFILE

Marco Fisichella
Risk Ident
30 PUBLICATIONS **164** CITATIONS
SEE PROFILE

Bernardo Pereira Nunes
Australian National University
99 PUBLICATIONS **512** CITATIONS
SEE PROFILE

Kyung-Hun Ha
ESCP Business School
7 PUBLICATIONS **86** CITATIONS
SEE PROFILE

Automatic Classification of Documents in Cold-start Scenarios

Marco Fisichella

Leibniz University of Hanover & L3S Research Center,
Appelstrasse 9, 30167 Hannover, Germany
E-mail: fisichella@L3S.de

Ricardo Kawase

Leibniz University of Hanover & L3S Research Center,
Appelstrasse 9, 30167 Hannover, Germany
E-mail: kawase@L3S.de

Ujwal Gadiraju

Leibniz University of Hanover & L3S Research Center,
Appelstrasse 9, 30167 Hannover, Germany
E-mail: gadiraju@L3S.de

Abstract:

Keywords: Algorithms, Classification, Competences, eLearning

1 Introduction

Currently, the World Wide Web is the largest source of information. Over the last decade, nearly every knowledge repository has moved its resources to online digital repositories. Consequently, the number of specific online disciplinary repositories has also increased significantly. Present-day online educational digital libraries are being deployed for a wide spectrum of topics, with a goal to facilitate easy discovery of relevant material on a particular topic. Since search engines like Google[a], Bing[b] and Yahoo[c], to name but a few, dictate information retrieval on the Web, digital libraries must offer an attractive differential for the users. The differential offered by these libraries is manifested in the form of focused topics, high quality resources and easy retrieval.

[a] http://www.google.com
[b] http://www.bing.com
[c] http://www.yahoo.com

The surge of the Open Archives Initiative[d], has resulted in abundant freely available data. Through utilization of the OAI-PMH protocol, a digital library can list the contents of several external repositories. However, digital libraries that rely on external content are hampered with the issue of assuring content quality since they do not own the actual documents. Nevertheless, these data accumulators also need to maintain a minimal threshold of quality, accessibility, and usability. Thus, it is crucial for any digital library to evaluate each new resource they receive by judging its quality and relevance to the collection. In most cases, evaluations are manually performed by curators who are familiar with the scope of the collection. However, as the amount of available content rises exponentially, it becomes an infeasible task for humans. This problem is even more significant in case of Open Archives, where a new repository may be added to the library with thousands of new documents at once.

To overcome the information overload problem and to maintain the quality of the collections, there is a lot of research focusing on the quality assurance of resources, as well as facilitation of access to information. For instance, the state-of-the-art work by Bethard et al. [1] proposed methods to automatically identify out-of-scope resources. Several other works approach the problem of automatically classifying documents [2, 3, 4, 5], thus identifying whether one document belongs to a collection or not .

A majority of previous works in this area, suggest methods which are built on top of machine learning strategies. They propose different solutions for the classic text classification problem, with the basic assumption of an existing training dataset. The work we present in this paper largely differs from previous works in the area. We consider the deep-rooted problem of absence of prior information pertaining to the corpus of the collection. To summarize, our proposed task is to classify an entirely unclassified collection. This issue is not exclusive to digital libraries. By modeling this problem as a recommendation task, the goal is to recommend a category to a document that has no prior connection with the collection (the so called cold-start problem).

In previous works, we proposed methods to automatically classify learning objects by exploiting the content from different but similar resources found outside the boundaries of a single content repository[6]. Our automatic classifying method is an extension of the state-of-the-art α-*TaggingLDA* for automatic tagging [7], which is based on the probabilistic topic model Latent Dirichlet Allocation [8].

The main difference between tagging and classifying is that, the task of tagging documents is not limited to a restricted vocabulary. Thus, with respect to tagging there is no completely right or completely wrong outcome. Tags may not be completely relevant to a document, yet they always attach additional information to the resource. On the other hand, the classification task requires a more precise and focused analysis since the outcomes must be within the boundaries of a fixed vocabulary. Contrary to tagging, categorization has a binary assessment that is either right or wrong. Additionally, misclassification has a greater impact for the user than misplaced tags. From the user's perspective, a misclassified document may bias the readers' understanding of the content, and even preclude the document from being discovered.

Regarding comprehension of the resources by learners, one essential feature is the use of competence metadata. A competence is the effective performance in a domain at different levels of proficiency. Educational institutions employ competences to gauge the level of ability or skill of a person. Thus, an educational resource enriched with competence

[d] `http://www.openarchives.org`

information allows learners to identify the resources to be consumed, in order to reach a specific competence target. Additionally, competence annotations are usually assigned alongside an expertise level. For example, the European Qualification Framework (EQF) has eight levels to describe a competence that ranges from beginner to expert.

In this paper, we extend our previous work by converting categories into competences. We present our work towards an automatic competence level assignment tool, taking into account the speed of development, exchange of educational resources, and the problems of ensuring that these materials are easily found and understandable. Our goal is to provide a mechanism that facilitates learners in finding relevant learning materials and to enable them to better judge the required skills to understand the corresponding content, through the interpretation of competence levels. We present a strategy that exploits knowledge from the wisdom of the crowds to automatically assign levels of expertise for LOs' competences. Finally, by providing a Web portal where learners, teachers and general users can browse the learning objects, we put the outcomes of our methods into practice.

2 Background

At the core of this work, lies the α-*TaggingLDA* method, a state-of-the-art LDA-based approach for automatic tagging introduced by Diaz-Aviles et al. [9]. α-*TaggingLDA* is designed to overcome new item cold-start problems by exploiting the content of resources, without relying on collaborative interactions.

In order to illustrate the method with an example, consider a novel LO entitled *Knowledge Technologies in Context*. Let us assume that this resource is new to the collaborative learning system and does not have any associated tag annotations. The absence of tags makes it difficult for the system to consider it as candidate for recommendations, for instance.

α-*TaggingLDA* first extracts relevant *textual content* such as the title, description or metadata (e.g., author) from the LO, and creates a document denoted by d_{LO}. Then, the LO is associated with a set of 'similar' documents, which we refer to as an *ad hoc corpus* for the LO, represented as $corpus_{LO}$.

The α-*TaggingLDA* method does not impose any restriction on the similarity measure used to associate the corpus with the LO. The similarity measure could be specified based on the nature of the resources, (e.g., text documents, multimedia items) and the textual content or the available metadata. For example, a particular implementation might rely upon a computationally inexpensive similarity measure or on a more complex clustering algorithm.

In our example here, the title of the LO is used to query an Internet search engine in order to retrieve the title and snippets of the n relevant results. This subset corresponds to $corpus_{LO}$.

The LO's textual content is extracted and the subset of the top n results constitutes the text collection $D = \{d_{LO}\} \cup corpus_{LO}$, which in turn serves as the input for LDA, together with the number of topics required. In this example, let us consider two topics, i.e., $|Z| = 2$. The set of tags to be used to annotate the LO is denoted by $TopN_{tags}(LO)$, and its size is set to six for this particular case, i.e., $|TopN_{tags}(LO)| = 6$.

Table 1 presents an example of the output produced by LDA as per the setting prescribed above. Topics are ordered based on the document-topic distribution $P(z \mid d)$, and within each topic, terms are ranked based on the topic-term $P(t \mid z)$ distribution.

Table 1 Example of two topics output by LDA. Topics are ordered based on the document-topic distribution $P(z \mid d)$, and within each topic, terms are ranked based on the topic-term $P(t \mid z)$ distribution.

$\boldsymbol{Topic_1}$		$\boldsymbol{Topic_2}$	
$P(z = 1 \mid d_{LO}) = 0.70$		$P(z = 2 \mid d_{LO}) = 0.30$	
Term t	$P(t \mid z = 1)$	Term t	$P(t \mid z = 2)$
technologies	0.45	phenomena	0.33
software	0.25	work	0.28
ecosystems	0.16	business	0.19
systems	0.11	researchers	0.15
representation	0.03	vendors	0.04
interpretation	0.01	people	0.01

For the construction of the final set of tags $TopN_{tags}(LO)$, α-*TaggingLDA* selects the first candidate tag from $Topic_1$'s top terms, the second tag from $Topic_2$'s top terms, the third tag, again from $Topic_1$'s top terms, and so forth. The final list of tag annotations for the LO in our example corresponds to $TopN_{tags}(LO) = \{$ *technologies, phenomena, software, work, ecosystems, business* $\}$. To harness further details of this strategy, we refer the reader to the work done by Diaz, et.al. [9].

2.1 α-TaggingLDA Evaluation

Prior to this work, we have evaluated the α-*TaggingLDA* method for automatic tagging of learning objects[7].

We have empirically demonstrated through a series of evaluations that the α-*TaggingLDA* method produces high quality metadata enhancement for learning objects. The evaluation compares the automatically generated tags against existing tag annotations performed by the authors. Furthermore, a user study compared the participants' preference for automatically produced tags against the authors' tags. Finally, the evaluation demonstrates that α-*TaggingLDA* tags are the best candidate terms for assisting users in the tagging process.

Results from evaluations with over 100 participants indicate an agreement of 38.4% between the automatically generated tags and those provided by the participants. More notable is the participants' preference for the automatically generated tags (67.5%) over the experts' tags (32.5%).

The most important revelation was found to be the potential benefits produced by information delivered through the automatic tagging method. We draw upon this information to build the automatic classification method.

3 Automatic Domain Classifier

In the following sections we present the work done towards an automatic domain classification method.

3.1 Related Work

Much research has been done to improve the task of automatically classifying documents. Typically, the classification task can be understood in two ways. First, in the sense of assigning classes (predefined terms) to a document. Second, as we approached in this work, strictly grouping documents into one class. In this area, important research has been conducted by Fisichella et al. in [10]; the authors assign each document to one class, using a soft clustering algorithm, which is described by a set of terms. In both cases, the final goal is to improve organization and information retrieval. A great part of the literature on text classification is based on machine learning approaches and rely on dimensionality reduction [11] or on probabilistic topic models [7]. These strategies begin with a large set of manually annotated documents (positive examples of classification) where algorithms find existing patterns in documents in each class. Then, in a second step, these patterns are automatically identified in non-classified documents [2, 3, 4].

Although there is vast literature in the area, the basic idea is immutable. Each algorithm exploits different features and implements unique strategies to identify patterns that can later be used to classify new documents.

In many studies, the well accepted approach to begin with text classification is TF-IDF weighting [3, 12, 13, 14]. This well known strategy turns documents into a list of weighted terms that facilitates the representation of the documents. It relies on the assumption that the most representative terms of a document occur many times in the document's text and, at the same time, occur only in a small set of the available documents. To the best of our knowledge, the most successful approaches for automatic classification are based on TF-IDF, usually combined with support vector machine (SVM) classifiers [12, 14, 1]. Standard SVM approaches try to predict, from input data, two possible classes maximizing their margin.

In all visited previous works, there is always the assumption of an existing training data. Our work distinguishes from the previous work on document classification in two ways. First, we do not build upon any pre-existing human annotated data. Second, we do not base our strategy on incremental machine learning algorithms. Since there is no training data that feed the method with confident positive examples, there is no learning strategy to build upon. As exposed in the following sections, our proposed method is composed of strategies that exploit existing knowledge of outside repositories, combined with the wisdom of the crowds and a straightforward heuristic approach.

3.2 Tag-Based Domain Classifier

We augmented an additional layer on top of the automatic tagging method presented in the previous section, in order to identify the most probable category that a document might belong to. This classification layer uses two different inputs. First, a ranked list of keywords that describes the resource to be classified and second, a list of domains (with a list of keywords describing each domain) to which the document can belong to. For the first input, as described earlier, we employ α-TaggingLDA that provides a ranked list of tags representing the main concepts in a document. As a second input, the list of topics used by the library and a few keywords that best describe each topic are required. Describing topics with keywords is a light-weight task when compared to the manual assignment of categories for each document in a collection.

Algorithm 1: Pseudocode for keyword-term matching method.

```
1  begin
2      for each document do
3          Get top N α-TaggingLDA keywords;
4          KeywordIndex=0; for each keywords do
5              KeywordIndex++; for each domain do
6                  Get domain's terms;
7                  for each domain's terms do
8                      if keyword == term then
9                          domain-score += 1/KeywordIndex;
10     return top scoring domain;
```

With these two inputs, the classification method assigns scores for each match found between the document's list of keywords and the domain's keywords. Since the document's keywords are already properly ranked, we apply a linear decay to the matching-score. This means that the domain's keyword that matches the first document's keywords has a greater score than those matching the document's keywords that are more highly positioned in the ranking. After the matching process, we compute the sum of the scores of each topic, assigning the top scoring to the document. The pseudocode (Algorithm 1) portrays the matching method.

We configured the α-*TaggingLDA* to return a maximum of 100 terms for each document. During the classification matching, if no correspondences are found the document is declared unclassified. To evaluate the proposed method (refer to Section 5), we utilized the OpenScout[e] project repository, a new digital library in the area of business and management that covers numerous topics. The project has its own domain classification that was proposed by experts in the field. We had access to the same experts who went on to build a list of keywords describing each domain, which we elucidate in Section 4.

3.3 Baseline

In order to draw a comparison between our approach and existing strategies, we chose well-known successful methods used in text classification. First, we calculate the TF-IDF values for all words in each document within the corpus. For each document, we remove words with less than 2 characters and words consisting of numbers from the text, because such terms are not useful when determining the category of an article. In addition, we remove the punctuation marks (e.g. –, ?, %, /, !, etc.) from the words and combine the remaining parts. Finally, we remove stop words and apply stemming.

Following the pre-processing, we store the top 15 remaining terms for each article according to the highest TF-IDF values in one vector. We use such vectors of words to represent a document as a surrogate. Then, we classify by computing the similarity (Jaccard) of the TF-IDF results and the relevant keywords of each domain. In addition, we perform the computation by using our proposed matching method previously presented. In Section 5, we evaluate three distinct strategies:

[e] `http://learn.openscout.net`

- TF-IDF + Jaccard
- TF-IDF + Matching
- α-TaggingLDA + Matching

4 Competences and Domain Classification

OpenScout[f] is an EU co-funded project which aims at providing skill-and-competence-based search and retrieval Web Services that enable users to easily find, access, use, and exchange open content for management education and training. Apart from connecting leading European Open Education Resources (OER) repositories, the project also integrates its search services into existing learning suites. Within the project, a management-related domain classification and competences have been developed (see Table 2) in order to support the learner while searching for appropriate learning resources that belong to a specific domain, e.g. marketing or finance, or a learning resource that suits the learner's knowledge (specific competence).

Furthermore, each identified domain was enriched by a list of descriptors (the most important keywords describing the domain) as accurately as possible. A step-by-step approach was adopted to develop the new OpenScout domain classification and its corresponding keywords.

As a first major step, a focus group was organized and moderated by an experienced OpenScout project coordinator. The focus group participants consisted of a sample of ten domain experts from Higher Education, Business Schools and Small-Medium Enterprises (SME), including two professors, six researchers, and two professionals to generate an initial domain classification based on experience and academic literature. After further deliberations, detailed discussions, and comparison with already existing domain classifications of other academic institutions, only those terms that best fit management education and the underlying project goals were finally retained by the focus group, yielding 15 fundamental domains.

A pretest with domain experts from higher learning institutions INSEAD[g], BRUNEL[h], EFMD[i] and VMU[j] was conducted to assess the content of the domain classification and to ensure content validity. Those terms that best fit management education in general, and thereby the content of the learning resources, were retained by the experts for the final domain classification.

Following the enrichment of each domain with a list of main keywords, as a second major step, eight researchers from the ESCP Europe Business School[k], with different research focus and knowledge about certain domains, were asked to provide a list of eight to ten terms that best fit their domains. The participants completed different diploma studies in Germany, the USA, UK, Australia, or China and on average had two years of work experience at the university. Three of these participants had also been previously employed on a full-time basis in several industries. Reflecting on the resulting keywords of each domain, all experts

[f] http://openscout.net
[g] http://www.insead.edu/home
[h] http://www.brunel.ac.uk
[i] http://www.efmd.org
[j] http://www.vdu.lt
[k] http://www.escpeurope.eu

Table 2 The domain/competence classification of the OpenScout repository and the respective examples of most relevant keywords.

Domains/Competences	Relevant Keywords
Organizational Behavior and Leadership	organizational,behavior,leadership,negotiation,team,culture…
Decision Sciences	decision,risk,forecasting,operation,modeling,optimization…
Marketing	marketing,advertising,advertisement,branding,b2b,communication…
Economics	economics,economy,microeconomics,exchange,interest,rate,inflation…
Finance	finance,financial,banking,funds,capital,cash,flow,value,equity,debt…
Strategy and Corporate Social Responsibility	strategy,responsibility,society,sustainability,innovation,ethics,regulation…
Accounting and Controlling	accounting,controlling,balance,budgets,bookkeeping,budgeting...
Management Information Systems	management,information,system,IT,data,computer,computation...
Technology and Operations Management	technology,operation,ebusiness,egovernment,ecommerce,outsourcing…
Entrepreneurship	entrepreneurship,entrepreneurs,start-up,opportunity,business…
Human Resource Management	resources,management,career,competence,employee,training,relation…
Language and Communication	languages,communication,message,grammar,nonverbal,verbal…
Project Management	management,monitoring,report,planning,organizing,securing…
Business and Law	law,legal,antitrust,regulation,contract,formation,litigation…
Others	-

emphasized that they could only provide a subjective assessment as each domain represents a broad field of knowledge. Despite this, and considering their long years of experience and ongoing education in their respective fields, we believe these experts fulfil the necessary criteria for providing the most relevant keywords.

5 Automatic Domain Classification Evaluation

In this section, we measure the benefits that can be reaped from automatic classification for an unclassified digital library with two distinct user studies. The remainder of this section describes each evaluation setting.

5.1 Dataset

We based our experiments on a dataset sampled from the OpenScout project collection [15]. According to the Open Archives Initiative, the project procures metadata information from learning resources located at various learning content repositories. For our evaluation, we selected all documents which were in the English language. We have collected 7,750 items in total, that should be classified under one of the 15 categories. None of the items had any information about the classification during the data collection stage. Consequently, every document was subject to automatic classification by each one of the methods, namely, TF-IDF+Jaccard, TF-IDF+Matching, and α-*TaggingLDA*+Matching.

5.2 Metadata Enrichment

For automatic classification in our experiments we use the TF-IDF+Jaccard, TF-IDF+Matching, and α-*TaggingLDA*+Matching methods. In case of the α-*TaggingLDA*, the

corpus builder is based on the search results accumulated by querying Google's search API. The titles and short text summaries (snippets) of the ten most relevant results thus returned are used to populate ten different textual documents. Apart from this, the final *ad hoc* corpus for the learning object consists of the textual content of the resource. Next, by applying LDA (with the Gibbs sampling implementation provided by the Machine Learning for Language Toolkit - MALLET[l]) to this corpus we extract the desired number of latent topics. As per the optimal setting specified in [9], the default number of topics considered was two. From these topics, the top tags were inferred and matched against the domain topics table presented in Section 4. The method produces a score for each topic in the classification and the topic corresponding to the highest score is chosen to classify the input document.

5.3 Evaluation I: User Classification

This study was carried out with an aim to procure evidence to evaluate whether or not the automatic classification actually matches the categories assigned by the users.

This evaluation is a user study where in each participant is presented with basic information (the title and an abstract varying from 60 up to 500 words) regarding a document. Each document is randomly selected from the dataset. The format of the original resource (e.g. video, image, presentation or document) is not made known to the participants in order to align the nature of the evaluation and avoid biased judgements of the classification relevance based on machine-incomprehensible information.

Each participant is then instructed to read the title and the description of the document and finally choose one of the categories in the proposed domain classification. On submitting the form, the participant is presented with a new object to be evaluated. Additionally, the participants are supported with the option of skipping a given document at any point during the analysis, in case they do not understand the meaning of the content or do not feel confident in judging it. The participants were requested to repeat the process for at least ten objects. However, we did not place a limit on the upper bound of their contributions to the study.

In order to assess the quality of results emerging from this evaluation, we measure the agreement between participants' choices and automatic classifications, and use recall, precision and F_1 measure, three widely used metrics. The metrics are defined in corresponding Equations 1 and 2.

- Recall for a given classification c is defined as:

$$recall = \frac{|ClassifiedDocs(c) \cap AutoClassifiedDocs(c)|}{|ClassifiedDocs(c)|} \tag{1}$$

- Precision for a given classification c is defined as:

$$precision = \frac{|ClassifiedDocs(c) \cap AutoClassifiedDocs(c)|}{|AutoClassifiedDocs(c)|} \tag{2}$$

where *ClassifiedDocs(c)* is the set of documents assigned to a category c by a participant and *AutoClassifiedDocs(c)* is the set of documents assigned to a category c by the automatic

[l] `http://mallet.cs.umass.edu`

classifier. The aggregated values of recall and precision are in turn used to compute their harmonic mean or f1 measure as defined according to Equation 3.

$$f1 = 2 \cdot \frac{recall \cdot precision}{recall + precision} \tag{3}$$

5.4 Evaluation II: User Agreement

The aim of this experiment is to evaluate the quality of the automatically assigned categories. Similar to Evaluation I, each participant in this user study is presented with the title and an abstract of a document which is randomly selected. Once again, the format of the original resource was not disclosed to participants in order to circumvent possible bias. In addition, participants were presented with a list of suggested topic classifications (see Table 2 for the list of possible classifications). Note that in this evaluation we only presented the classifications resulting from employing the proposed α-*TaggingLDA*+Matching method.

Participants are then instructed to read the title and description of the document and finally rate their level of agreement with the proposed classification, on a 5-point Likert scale. On submitting a form, participants are presented with a new object to be evaluated. Again, participants are supported with the option of skipping to a new document in case they do not understand the content or do not feel confident in judging it.

5.5 Participants' Behavior Analysis

We actively log and encapsulate the behaviour of the participants during their activity in Evaluation I and Evaluation II. We evaluate the degree of difficulty in the classification task, by tracking the time that each participant takes for analyzing each document in both the evaluations and the number of times that they tend to skip a given document.

5.6 Results

Our user study included a total of 81 participants (31 female and 50 male); 51 of them explicitly stated to be students and 18 were professionals in the area of education. The average age of the participants was found to be 32 years, ranging from 19 to 66 years. In total, participants evaluated 658 documents (405 unique) during the first part of the evaluation and 765 (478 unique) documents during the second.

With the data collected in the first part of the evaluation, we compared the participants' categorization with those automatically assigned by the different methods (Table 3). Our best performing method, α-*TaggingLDA*+Matching, produces a 32% improvement over the TF-IDF+Matching method.

The results for recall, precision and f1 are presented in Table 4. Despite the values of recall and precision not being exceptionally high, α-*TaggingLDA*+Matching provides a significant improvement over the other strategies.

From the feedback during the second phase of our user study we found that in 72% of the cases, participants either strongly agreed or agreed with the automatic classification assignments given by α-*TaggingLDA*+Matching (Table 5). These results imply that participants are inclined to accept the suggested categorization even though it may not be their first choice. We hypothesize that this is due to the complexity of the document classification task, which is more complex than merely judging whether a category is correct or not.

Table 3 The overlap of the classifications given by each combination of methods with the classifications given by the participants.

Evaluation 1 - Results		
Participants Classification	658	-
TF-IDF + Jaccard	103	15.7%
TF-IDF + Matching	157	24.0%
α-TaggingLDA + Matching	209	31.8%

Table 4 Precision, Recall and f1-score for each strategy.

Strategy	Precision	Recall	f1
TF-IDF + Jaccard	0.30	0.22	0.20
TF-IDF + Matching	0.26	0.26	0.25
α-TaggingLDA + Matching	0.37	0.35	0.33

Table 5 The results of the participants agreement with the automatic classification given by the α-TaggingLDA+Matching.

Evaluation 2 - Results	
Strongly disagree	9%
Disagree	12%
Neither agree or disagree	7%
Agree	40%
Strongly agree	32%

Participants face the proverbial paradox of choice when asked to choose the most representative category from several different categories, since in our evaluation setting each document can be assigned to only one exclusive category. During the stage which only requires a judgement of whether a category is relevant or not, participants took 27.0 seconds per document on average. During the category assignment task, the average time consumed by the participants per document is observed to be 36.6 seconds (35% higher). In addition to this, we compute the number of times participants skip a certain task. During the judgement stage we note 33 such occurrences of skips, while in the secondary phase this is nearly doubled to 65 skips in total.

6 Automatically Assigning Competences

In this section, we extend and extrapolate the concept of domains to competences and conduct experiments to validate the usefulness of the methods introduced.

6.1 Evaluation

In order to evaluate the method we introduced, we utilize the OpenScout dataset containing 21,768 learning objects. We prune this data and consider only objects that are in English,

with a description containing a minimum length of 500 characters. Such pruning thereby resulted in a set of 1,388 documents. We applied the competence assignment method on these documents. Considering that the dataset is relatively new and very few items have been assigned with competences, we propose an automatic method to evaluate the outcomes of the automatic competence assignments.

Our evaluation setup considers the similarity among the learning objects and a set of cases that we believe can validate whether the automatic competence assigner produces optimum results. To measure the similarity among the documents, we used MoreLikeThis, a standard function provided by the Lucene search engine library[m]. MoreLikeThis calculates the similarity of two documents by computing the number of overlapping words and assigning different weights based on TF-IDF [16]. MoreLikeThis runs over the relevant fields for comparison, that we specify (in our case the description of the learning object) and generates a term vector for each analyzed item (excluding stop-words).

To measure the similarity between documents, the method only considers words that contain more than 2 characters and that appear at least 2 times in the source document. Also, words that occur in less than 2 different documents are not taken into account for the corresponding calculation. To gauge the relevant documents, the method uses the 15 most representative words, based on their TD-IDF values, and generates a query with these words. The ranking of the resulting documents is based on Lucene's scoring function which in turn is based on the Boolean model of Information Retrieval and the Vector Space Model of Information Retrieval [17].

To formalize our notions, let $c(LO_i)$ be a function returning the competence for a specific learning object LO_i; let $s(LO_i, LO_j)$ be a function measuring the similarity between two resources LO_i and LO_j. Then, given the set of learning objects, the similarity scores $s(LO_i, LO_j)$ and the competence assignments $c(LO_i)$, we evaluate the results through four given cases:

- **Case 1:** If two LOs have the same competence and are similar to some extent, it is resonable to assume that the competence assigner is coherent. If $c(LO_1) == c(LO_2)$ and $s(LO_1, LO_2) >= 0.7$

- **Case 2:** If two LOs have been assigned with the same competence but are not similar, it is not completely implausible and means that the competence is broad. If $c(LO_1) == c(LO_2)$ and $s(LO_1, LO_2) < 0.7$

- **Case 3:** If two LOs have been assigned with different competences and are very similar, it implies a fault committed by the automatic competence assigner. Thus, the lower that the assignments fall in this case, the better are the results. If $c(LO_1)! = c(LO_2)$ and $s(LO_1, LO_2) >= 0.7$

- **Case 4:** Finally, for the cases where two LOs have been assigned with different competences and the LOs are not similar, correctness can not be directly derived but a high value demonstrates the coherence of the method. If $c(LO_1)! = c(LO_2)$ and $s(LO_1, LO_2) < 0.7$

[m] `http://lucene.apache.org/core/old_versioned_docs/versions/3_4_0/api/all/org/apache/lucene/search/similar/MoreLikeThis.html`

Table 6 Results of the automatic comepetence assignments according to the cases defined in Section 6.1, considering the top one and top two competences with similarity threshold at 0.7.

Rule	Tags(1)	Tags(2)
1) Same Competence Sim. $>= 0.7$	0.24	0.24
2) Same Competence Sim. < 0.7	9.60	10.63
3) Dif. Competence Sim. $>= 0.7$	1.02	0.95
4) Dif. Competence Sim. < 0.7	89.12	88.16

6.2 Evaluation Results

In Table 6, we plot the results depicting the number of occurrences that fall under each case. In addition, we alternate the number of competences considered in the evaluation. First, we use only the top scoring competence for a given LO and, in a second round of the evaluation, we consider the top two scoring competences.

The results show that very few items fall under the specified cases 1 and 3, meaning that most of the items do not meet the minimum threshold value of similarity, thus, showing that most of the classified documents are dissimilar. The low similarities also reflect the short textual descriptions available. Regarding the documents that are similar ($>= 0.7$), only around 1% of the items fall under case 3; given our assumptions in Section 6.1, we consider this 1% as a false allocation.

The results obtained show very few instances where different competences were assigned to very similar items. We interpret that as discernible evidence of the coherence and effectiveness of the proposed method, that may be applied to effectively enhance competence metadata for learning objects.

7 Competence Expertise

The main problem to be solved is the competence leveling. Given a LO and its competence (that is assumed to be correctly assigned), the goal is to automatically assign a level to this competence according to the European Qualification Framework (EQF). In other words, our method must assign a score between 1 (basic) and 8 (advanced).

7.1 Related Work

In recent years, many systems have been developed for the Technology-Enhanced Learning (TEL) with the goal of providing technological support for pedagogical purposes. In this sense, several learning object repositories (e.g., Stanford OpenCourseware[1], Merlot[2], Science Netlinks[3]) have been made available for retrieving educational resources on the Web. However, the process of retrieving educational resources is not straightforward due to the lack of descriptive metadata, such as competences and skills. Although standards of competence-based metadata to describe educational resources have been proposed in the literature [18], manual metadata filling is often an arduous and laborious task.

To deal with this problem, OpenScout [15] proposed a collaboration tool for describing its educational resources metadata [19]. In practice, competence and skill metadata could just be changed from few registered contributors, thus these metadata were not completely filled out.

In the same direction Auzende et al. [20] introduced the importance to visualize competences and sub-competences for educational resources. Authors developed interfaces to let teachers upload, create, search, and enrich metadata of LOs. In their work, a competence level was always assigned by humans to a resource. Then, according to users' feedbacks, such a competence level was refined. As introduced, this work does not handle with automatic competence classification, but once the competence is assigned to each resource, authors' work is limited only to update/modify the competence level.

Van Assche [21] introduces an approach for linking educational resources through competences according to curricula. In his method, he manually depicts the goals of curricula into competences allowing interoperability between different curricula and to support resources retrieval. This work is similar to ours, however we use an automatic approach to assign competences to educational resources in order to be easily retrieved and reused by lecturers and students.

An attempt to automate the competence assignment process is presented by Melis et al. [22, 23]. In order to facilitate the reuse of learning objects, they present a framework that maps different competence systems, such as PISA [24] and Blooms Taxonomy [25]. Our approach is complementary to that, since we automatically classify learning objects according to their competence type and level; these learning objects can be reused by such courses generator which may take advantage of the knowledge about the field of study and the competence level.

7.2 Competence Expertise Leveling Method

To accomplish competence leveling, our proposal is to transfer the knowledge from an external repository, like Wikipedia. Wikipedia is the largest repository of textual articles created and maintained by humans and arguably the most consulted, structured, and referenced repository. Our proposed method extracts the *authority* information of a Wikipedia article based on its link structure to calculate the competence levels.

Authority of a Wikipedia article is given by the *popularity* of an article as evidence of its complexity. We use the number of incoming links as the popularity measure for an article [26]. Wikipedia editors create these incoming links manually. Therefore, it is reasonable to assume that these articles are, to some extent, significant to the general public. The hypothesis is that the more popular an article is, the easier it is for the reader to understand (e.g., there are many more incoming links to the article *Finance* than to *Private equity*, which is a more specific term that requires more abstraction).

We use a dataset consisting of a snapshot of the entire Wikipedia corpus from October 2011. It contains more than 4.5 Million pages (all articles without redirect pages). In addition, we collect the list of Wikipedia categories from the contemporary period and corresponding statistical information of the most linked articles.

Then, our automatic competence level assigner is divided in the following steps. First, each document is semantically annotated (RDFa) using the DBpedia Spotlight[n] Web Service. The output returns the content of the document enriched by DBpedia resources (or Wikipedia articles). Then, for each link added to the content of the LO, we check the respective Wikipedia article and query for its authority value, i.e. the number of incoming links pertaining to each article.

The distribution of authorities follows a power law distribution, where a small number of dominant articles contain the larger part of all incoming links. In order to compensate

[n] http://dbpedia.org/spotlight

Table 7 Experts agreement with assigned competences levels.

Strongly Disagree	Disagree	Neither	Agree	Strongly Agree
7%	8%	3%	44%	38%

for this, we apply a logarithmic smoothing function before the proper normalization. In doing so, we still exploit the information but counterbalance the dominance of the few top authorities.

At this point, each LO contains the information of the authorities' values (number of incoming links) of each linked article. It is important to note that our competence leveling method regards only LOs that are assigned with one single competence. Thus, to compute the final level of a competence, we apply a linear combination of all the authorities' values for each linked term and normalize it to the European Qualification Framework scale.

7.3 Evaluation and Results

To evaluate the performance of the competence leveling, once again we use the OpenScout dataset containing 21,768 learning objects. We prune this data to consider only objects that are in English, and containing descriptions with a minimum length of 500 characters. We thereby arrive at a resulting set of 1,388 documents. Finally, we only consider resources that have at least 10 terms annotated by the DBPedia Spotlight service. In the end, we assign competence levels to 1051 learning objects.

As mentioned before, annotating resources with competences is a very time consuming and cumbersome task and is usually performed by experts from corresponding knowledge areas. Due to this, our ground truth to evaluate the results was limited. Out of the 100 resources that have been annotated with competences by an expert, 60 were in the English language and out of these 60, only 44 passed our directives of being at least 500 words long and having at least 10 terms mapped to Wikipedia articles.

The competence assignment in OpenScout includes lower and upper boundaries of necessary expertise. The outcomes of our evaluation showed that for 37 learning objects, out of the 44 considered as ground truth (84% of the cases), the competence level was automatically assigned within the boundaries given by the experts. The results are a strong indicator that exploiting Wikipedia's link structure to derive the expertise needed to understand an article - therefore a learning object - is valid.

In addition to the automatic evaluation, we perform a user evaluation to further assess the correctness of the assigned competence levels. Out of the 1051 learning objects assigned with competence levels, we randomly selected 100 and, with the participation of 4 experts in the field of business and management we evaluate the assignments. Each expert is presented with 25 learning objects (with the assigned competence). Then, they are instructed to evaluate the competence assignment and, finally, rate their agreement with the proposed competence level on a 5-point Likert scale. The results in Table 7 portray that in 82% of the cases the experts either agreed or strongly agreed with the competence levels automatically assigned by our method.

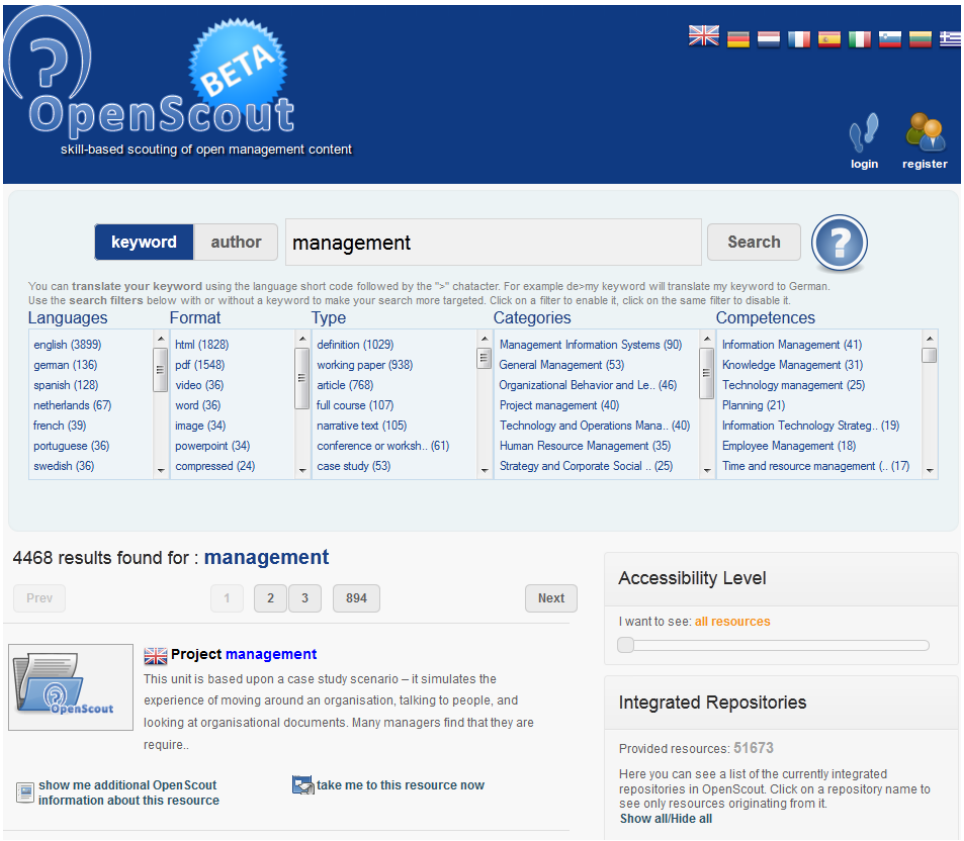

Figure 1 OpenScout Portal (Search).

8 OpenScout Portal

In this section, we present the current appearance and outlook of the OpenScout portal[o]. In the homepage users can select their language and perform core actions like login/register or view/edit their profile data. Furthermore, the index page uses an eye catching content slider to advertise the "most visited resources" apart from new important events and new features. Large icons are used to guide users to main areas of the portal like "Search", "Publish", and "Community".

Search is the most important action of the OpenScout portal. The search page, shown in Figure 1, is divided into logical areas, as follows:

Keyword - filter area: here users can select between keyword and author search, enter their query and translate it into another language, read search instructions and finally use the filtering mechanism to narrow down and focus their search.

Search module area: This section holds the accessibility slider which is used to narrow down search results based on their accessibility. The OpenScout portal has been developed with the aim to provide equal access to data and functionality to all users. More specifically the aim is that people with disabilities can perceive, understand, navigate, and interact

[o] http://http://learn.openscout.net/

within the portal. In order to offer the best experience possible, the architecture has been carefully curated and designed while following the associated standards. WAI-ARIA[p] has been implemented when possible; also pages have been validated against W3C standards. Since the features of the portal is bleakly out of the scope of this paper, we will not delve further into it and we leave the readers to explore the web portal.

Hereafter, we focus on the "Keyword - filter area". It is the most used search in OpenScout. After typing in a search term, the user can filter the results using the filters (also called facets), which are *Languages, Format, Type, Categories* and *Competencies*. When a value is selected, the results and numbers for the other facet values are directly updated. By clicking the selected value again, the user can un-select it. Furthermore, a user can select a specific *repository*, to curtail the learning resources from these specific repositories. Additionally, the user can decide to search for a specific author by selecting the *author* option at left of the search box. Since the main aim of this research is devoted to the automatic classification of documents, now we explore the "Competence filter". We applied the results of our research and a user can verify the results of the document classification using this filter. When a user experiences a Competence search by the "Competence filter", she needs to select a term from the Category filter, e.g. "Business and Law" or "Decision Sciences", which triggers the associated terms from the Competence filter to load. After the user selects a term from the Competence filter, e.g. "Contract Management", the learning objects tagged with these competences begin to load.

Finally, we maintain a web page with video tutorials in order to facilitate a better experience during the portal usage[q].

9 Conclusions

We successfully extend our previous work by extrapolating categories to competences. The automatic competence level assignment tool paves way for enriched usage of educational resources. Considering the rapid growth, development, and availability of educational resources, we provide a mechanism that supports learners in finding relevant learning materials and enables users to assess the required skills to understand the corresponding content, through the interpretation of competence levels. We have presented a strategy that exploits knowledge from the wisdom of the crowds in order to automatically assign levels of expertise for LOs' competences. Based on our evaluation and the consequent results, we find that our proposed strategy culminating in the automatic assignment of competences, significantly contributes to an enriched learning experience. Finally, the Web portal where learners, teachers and general users can easily browse the learning objects, is an additional contribution.

References

[1] Steven Bethard, Soumya Ghosh, James H. Martin, and Tamara Sumner. Topic model methods for automatically identifying out-of-scope resources. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '09, pages 19–28, NY, USA, 2009.

[p] `http://www.w3.org/WAI/intro/aria.php`
[q] `http://learn.openscout.net/video_tutorials.html`

[2] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features, 1998.

[3] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, 2002.

[4] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: A comprehensive study. In Sharon McDonald and John Tait, editors, *ECIR*, volume 2997 of *Lecture Notes in Computer Science*, pages 181–196. Springer, 2004.

[5] Sriharsha Veeramachaneni, Diego Sona, and Paolo Avesani. Hierarchical dirichlet model for document classification. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pages 928–935. ACM, 2005.

[6] Ricardo Kawase, Marco Fisichella, Bernardo Pereira Nunes, Kyung-Hun Ha, and Markus Bick. Automatic classification of documents in cold-start scenarios. In *WIMS*, page 19, 2013.

[7] Ernesto Diaz-Aviles, Marco Fisichella, Ricardo Kawase, Wolfgang Nejdl, and Avarï¿½ Stewart. Unsupervised auto-tagging for learning object enrichment. In *EC-TEL*, volume 6964 of *Lecture Notes in Computer Science*, pages 83–96. Springer, 2011.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[9] Ernesto Diaz-Aviles, Mihai Georgescu, Avaré Stewart, and Wolfgang Nejdl. Lda for on-the-fly auto tagging. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 309–312, New York, NY, USA, 2010. ACM.

[10] Marco Fisichella, Avaré Stewart, Kerstin Denecke, and Wolfgang Nejdl. Unsupervised public health event detection for epidemic intelligence. In *CIKM*, pages 1881–1884, 2010.

[11] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 81–90, New York, NY, USA, 2010. ACM.

[12] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, MA, USA, 2002.

[13] Aleksander Kolcz and Wen tau Yih. Raising the baseline for high-precision text classifiers. In Pavel Berkhin, Rich Caruana, and Xindong Wu, editors, *KDD*, pages 400–409. ACM, 2007.

[14] Pascal Soucy and Guy W. Mineau. Beyond tfidf weighting for text categorization in the vector space model. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 1130–1135. Professional Book Center, 2005.

[15] Katja Niemann, Uta Schwertel, Marco Kalz, Alexander Mikroyannidis, Marco Fisichella, Martin Friedrich, Michele Dicerto, Kyung-Hun Ha, Philipp Holtkamp, and Ricardo Kawase. Skill-based scouting of open management content. In *EC-TEL*, volume 6383 of *Lecture Notes in Computer Science*, pages 632–637. Springer, 2010.

[16] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[17] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[18] Demetrios G. Sampson. Competence-related metadata for educational resources that support lifelong competence development programmes. *Educational Technology & Society*, 12(4):149–159, 2009.

[19] Marco Kalz, Marcus Specht, Rob Nadolski, Yves Bastiaens, Nele Leirs, and Jan Pawlowski. OpenScout: Competence based management education with community-improved open educational resources. In Steve Halley, editor, *Proceedings of the 17th EDINEB Conference. Crossing Borders in Education and work-based learning, June 9-11, 2010*, pages 137–146, London, United Kingdom, June 2010. FEBA ERD Press.

[20] Odette Auzende, Hï¿½lï¿½ne Giroire, and Franï¿½oise Le Calvez. Using competencies to search for suitable exercises. In *ICALT*, pages 661–665. IEEE, 2009.

[21] Frans Van Assche. Linking content to curricula by using competencies. In David Massart, Jean-Noel Colin, and Frans Van Assche, editors, *LODE*, volume 311 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[22] Erica Melis, Arndt Faulhaber, Ahmad Doost, and Carsten Ullrich. Supporting Flexible Competency Frameworks. In Xiangfeng Luo, Marc Spaniol, Lizhe Wang, Qing Li, Wolfgang Nejdl, and Wu Zhang, editors, *Advances in Web-Based Learning ICWL 2010*, volume 6483 of *Lecture Notes in Computer Science*, chapter 22, pages 210–219. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2010.

[23] Erica Melis, Arndt Faulhaber, Anja Eichelmann, and Susanne Narciss. Interoperable competencies characterizing learning objects in mathematics. In Beverly Park Woolf, Esma Aï¿½meur, Roger Nkambou, and Susanne P. Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 416–425. Springer, 2008.

[24] A. Schleicher, Organisation for Economic Co-operation, Labour Development. Directorate for Education, Employment, Social Affairs. Statistics, Indicators Division, and Programme for International Student Assessment. *Measuring Student Knowledge and Skills: A New Framework for Assessment*. OECD Programme for international student assessment. Organisation for Economic Co-operation and Development, 1999.

[25] B. S. Bloom, M. B. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. *Taxonomy of educational objectives. The classification of educational goals. Handbook 1: Cognitive domain*. Longmans Green, New York, 1956.

[26] Jaap Kamps and Marijn Koolen. Is wikipedia link structure different? In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 232–241, New York, NY, USA, 2009. ACM.