Semantic-based recommender system with humain feeling relevance measure

David Werner[1], Thomas Hassan[1], Aurelie Bertaux[1], Christophe Cruz[1], and Nuno Silva[2]

[1] Université de Bougogne, LE2I, CNRS
{david.werner, thomas.hassan, aurelie.bertaux, christophe.cruz}@u-bourgogne.fr
[2] GECAD and School of Engineering Polytechnic of Porto
nps@isep.ipp.pt

Abstract. This work is a global project to develop a recommender system of economic news articles. Its objectives are threefold: (i) managing the vocabulary of the economic news domain to improve the system based on the seamlessly intervention of the documentalists (ii) automatically multi-classify the economic new articles and users profils the base of the domain vocabulary, and (iii) recommend the articles by comparing the multi-classification of the articles and profiles of the users. While several solutions exist to recommend news, multi-classify document and compare representations of items and profils they are not automatically adaptable to provide a mutual answer to previous points. Even more, existing approaches lacks substantial correlation with the human and and in particular with the documentalists perspective.

1 Introduction

The decision-making process in the economic field requires the centralization and intakes of a large amount of information. The aim is to keep abreast with current market trends. Thus, contractors, businessmen and salespersons need to continuously be aware of the market conditions. This means to be up-to-date regarding ongoing information and projects undergoing development. With the help of economic monitoring, prospects can be easily identified, so as to establish new contracts. Our tool is specialized in the production and distribution of press reviews about French regional economic actors. The overload of news information is a particular case of information overload, which is a well-known problem, studied by Information Retrieval and Recommender Systems research fields. News recommender systems already exist [23], Athena [13], GroupLens [26] or News Dude [4]. Some of these systems use domain knowledge to improve the recommendation task [23] [13]. To achieve this goal, a content-based recommender system is being developed. A recommender system is necessary for the item ranking and a content-based approach is required to analyze the content of each article to structure and preserve information content. The results of the analysis enable linking the domain knowledge to the articles to improve the recommendation task [23] [13]. Content-based recommender systems typically follow a two-step process:

(i) the indexing of articles and users (also known as *profiling*), which allows to describe them.
(ii) the comparison process which consists in comparing the description of the articles and the profiles of the users. The latter computes the article relevance with regards to the user profile.

In order to capture this economical context, we are moving towards a customized review for each user and towards an opinion survey on magazine readers that cover a broad array of subjects, including news services.

As consequence of this effort, the complete production process of the review is redesigned to produce and to automatically distribute a customized review for each user. So, the aim of the overall system is to manage all news articles produced, and provide the most relevant article for each customer.

In this paper we focus on :

(i) Content-based recommender system, and on the indexing subtask of textual items, based on a controlled vocabularies. Ontologies are central to our proposal, as they are used to represent and manage the controlled vocabularies, to describe profiles and articles, and finally to automatically multi-classify them via inference process. Also, we adopt a machine learning approach for generating a prediction model for supporting the automatic classification. This paper presents a proposal for enriching the documentalist-oriented ontology with the model prediction rules, which provides the necessary capabilities to the DL reasoner for automatic multi-classification.
(ii) Moreover, we are interested in the distinction between the *relevance* and the *similarity*, two notions that are often mixed up. We propose a new measure, *Relevance-Measure*, that allows us to capture the relevance of an article for a given user profile, based on their ontological descriptions.

This paper is organized as follows. First we present the background research work and related work. In the second section we present the system architecture followed by, in section 4 the description of our proposal to automate the indexing task and his valuation, in section 5 the indexing task applied to the particular case profiles and in section 6 the description of our proposal to compute relevance of articles according to profiles and his valuation. Finally, we summarize the contributions.

2 State of the art

The large amount of information on the web, company information systems, digital libraries, selling websites and so on, is a well-known fact. Recommender systems aims at providing for each user the better items according to his/her needs. Items can be websites, news articles, books, video, music, washing machine, etc. In the literature two paradigms are distinguished. First, *Content-Based Recommender Systems* try to recommend items similar to those a given user has liked in the past. Second, *Collaborative Filtering Recommender Systems* identify users whose preferences are similar to those of the given user and recommend items they have liked [2]. Some subtasks should be performed and the first is named the *indexing task*. It is possible to distinguish two cases, the indexing of items, by content analysis, and the indexing of the profiles, via profiles learning (which generally includes a study of the behavior for implicit feedbacks or proposed to the user to giving explicit feedbacks as *"I like"* button). Both can be seen as a multi-classification task. The second task: *comparison*, consists in filtering each item relative to a given profile [22]. The survey of K. Nagewara Rao [25] proposes a general comparison of the main advantages and drawbacks of each kind of Recommender System (e.g. content-based or collaborative filtering).

Knowledge management for humans and machines handling. As it is presented in the survey about controlled vocabularies from [20], more and more companies use controlled vocabularies in their information system. Several kinds of structure are used to manage vocabularies, i.e. from the lowest to the richest semantic definition: glossary, taxonomy, thesaurus, ontology. A lot of companies plan to use ontologies in their applications [20]. While acquiring, managing and maintaining controlled vocabularies are important yet relatively easy tasks for the documentalists, the ontology approach to model the domain knowledge is hardly accessible for them due to the complexity of the logical structure, but it is easily used by the machine because it is formal.

Indexing by multilabel-classification: In order to make the indexing process automatic, the system has to associate a set of labels from the taxonomic thesaurus to each article and profile. This cannot be done without the two following process: (i) a text analysis process to extract keywords and other features from texts and (ii) a machine learning process to learn from examples. Feature extraction processes range from simple term extraction process like tf-idf [6] to text-based semantic-aware processes, e.g. term extraction from text based on (i) information retrieval methods [28], or (ii) based on NLP works [9] [24]. It is possible to use several degrees of text processing tools (and preprocessing), to extract noun phrases (i.e. tokenizer, part of speech tagger, handmade patterns and even parsers). This process allows us to interlink a set of terms (i.e. *features*) from the article term extraction process (i.e. *content analysis*) and the set of taxonomic keywords (i.e. *labels*) for the articles indexing task.

Machine learning process has two primary goals: *Prediction* and *Description*. *Prediction* is concerned with using features of previously classified examples (e.g. documents or any other resources that can be analyzed and classified) to predict the unknown classification (i.e. labels). *Description*, on the other hand, focuses on finding human-interpretable patterns that describe the performed classification.

Two main categories of label-classification prediction can be enumerated: the single-label and the multi-label classification. Single-label classification aims to learn a prediction model from a set of examples that are related with a single label from a set of disjoint labels. In multi-label classification instead, the examples are associated with a set of labels [30]. Multi-label classification faced increased attention in the last decade, overcoming the single-label classification previous dominance, but it was only much recently that hierarchical multi-label classification (HMC) approaches received the desired attention [3] [10] [32]. Even so, some of the so-called HMC approaches do not follow a strict hierarchical semantics (in the sense of subsumption), but a clustering approach. This is the case of the state of the art "hierarchical" multi-label approach HOMER [29] and that of [31] that uses Predictive Clustering Tree (PCT) framework. However, unlike HOMER, the approach described in [31] is constrained by the taxonomy or Direct Acyclique Graph (DAG) underlying the training and testing datasets. This is also the case of other works, notably in the area of bioinformatics [12] [3].

Both [31] and [12] are very interested in the description of the predictions to the user. In [16] the authors propose an iterative and interactive (between AI methods and domain experts) approach to achieve prediction and description (which are usually hard to fulfill), considering domain expert knowledge and feedback. Unlike us however, in [16] the authors do not aim to automatically multi-classify the items but only to improve the ontology, which means that the resulting ontology is not used for automatic classification of items.

In [10] the authors propose an approach to build ontologies using data mining results upon databases. The result is the enrichment of the ontology with new concepts and datatype properties, which is far from the required specification of classes. Our goal instead is to enrich the already existing taxonomic thesaurus with constraints capturing the prediction model allowing the user to perceive the taxonomic thesaurus, rules and the adopted features.

In [7] the authors are concerned with automatically creating an ontology from the text documents without any prior knowledge about their content. For that they use an iterative and interactive 4-phases process. Unlike [32], [10] that constructs thesaurus from the learning examples, in this project/paper the thesaurus-based taxonomy already exists and should be applied both in the automatic classification and description.

The profiling is a particular case of indexing, most of existing approches try to nd correlations based on reading time and eventually scrolling [5]. [19] states that there is a lack of research literature about other methods. Reading time is generally correlated to the relevance of the resources, but this correlation is highly dependent on the reading condition (test protocols), and more importantly on the task complexity [17] [18]. Assuming it is possible to qualify these two characteristics in an application, it would be possible to determine a reading time threshold from which a document can be relevant. The use of scrolling, eyetracking and implicit methods in general are not sufcient, but these methods seem to improve the precision of the recommendations in conjunction with ratings [14] [15] [5]. [19] concludes by saying that if other actions are taken into account, they must be weighted, as saving or printing a document is probably more important than copying a part of it for example.

Comparison and relevance: Vector Space Model (VSM) [27] involves the representation of items to be recommended (e.g. articles), and sometimes the users needs (e.g. queries, profiles) as vectors. This presentation allows the use of different metrics for comparison. In this article we use the Cosine and Jaccard and Euclidean distance similarities. Many recommender systems based on content use them to make comparisons, either between items or between item and profile ([13] [23] [11] [4] [1]). This method of linear algebra has two main advantages: it not only provides a non-binary outcome, so for ordering the results of recommender systems, but also allows quick calculations and resistance to scalability.

Furthermore, methods of information retrieval can be used to take into account the knowledge from a knowledge base while using a vector modeling. The proposed approach [33] uses the WordNet lexical knowledge base [8] to improve the management of the heterogeneity of natural language and thus improve the understanding of user needs. The idea is to add information to user queries (*query expansion*). This method increases the *return* [3] with the aim of improving overall system performance. We adapt this method to the way which the domain knowledge in our ontology is modeled.

We found that the concepts of similarity and relevance are usually combined in systems using vector modeling. Determine suitability as a similarity does not take into account the different levels of specificity in the description of the requirement of users. This description is, however, made possible by the use of external knowledge system. We propose an evaluation of the relevance, *Relevance measure* using the concepts of similarity, but taking into account the perception of the relevance by the user.

3 The System

Our system is an ontology-driven content-based recommender system (also named semantic-based recommender system). An ontology is created and populated with the help of company experts, in order to model their domain knowledge in a knowledge base. As a classic content-based recommender system, our system is composed of two main tasks (fig. 1). The first one is *indexing*, to create a representation of each items content, and users needs. In our system the knowledge base is also populated during this task. The second task is *comparison*, to apply a comparison metric between items and profiles representations so as to measure each items degree of relevance for each profile. Items are ranked with the help of the relevance measure, before being provided to the user. These subjects are developed as follows.

The Knowledge Base. In order to capture the economical context, we move towards a customized review for each user and towards an opinion survey on magazine readers that covers a broad array of subjects, including news services. Criteria for a relevant customization of the review were identified as a result of this survey as well as expert domain knowledge. These criteria are economic themes (i.e. main economic events), economic sectors, temporal and localization

[3] Number of items correctly considered as relevant to the actual number of relevant articles.

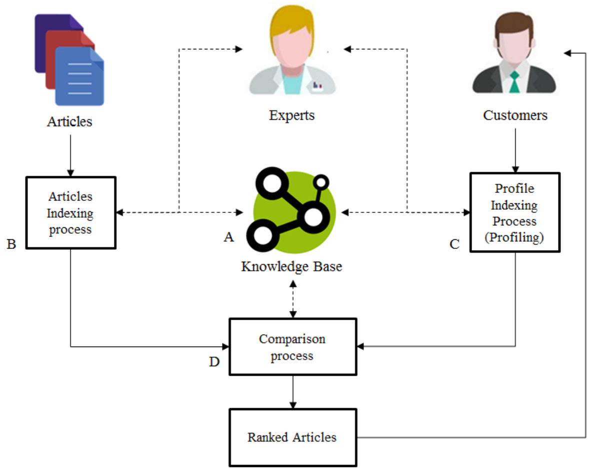

Fig. 1. Our ontology-based recommender system.

information. Vocabularies associated to these criteria are defined and structured by experts as thesauri. These thesauri are used as facets for the description of the items and profiles. An hybrid domain knowledge representation has been developed , in which the skeleton of the domain knowledge is delivered by an ontology that captures the process and application needs, complemented by a set of thesauri that capture the domain knowledge of the experts. These thesauri has been captured in a very light, expert-oriented fashion, with minimal formal semantics and consistency obligations. Once integrated in the ontology the strict semantics of OWL DL is followed. The relations in thesauri are integrated into the ontology to be used in the recommendation process as input for the evaluation of the semantic distance between articles and profiles. Furthermore, these thesaurus relations are very important for the experts' tasks as allow them to better and faster understand the context of the word/label to use in an article or profile classification.

To achieve the recommendation of articles to customers, the system needs are a representation of the content of each article, and representation of the needs of each customer. The index used in our system is the same for articles and profiles: the knowledge base (fig. 1 (A)). Articles and profiles are represented by instances in our knowledge base. The ontology contains several roles to model articles' content, and users' interests. To avoid manual classification of articles, which is a time consuming process, we propose to adopt a machine learning approach for generating a prediction model for supporting the automatic indexing. The documentalist-oriented ontology is enriched with the model prediction rules, which provides the necessary capabilities to the DL reasoner for automatic multi-classification. To ensure the same quality as manual indexing, the documentalists who know the economic context, perform the supervision and correction. This is discussed in the sections 4 and 5.

Comparison. The recommendation task is mainly based on the comparison between the user profile and the items available for recommendation. The system uses the knowledge base as an index, and profiles and articles are presented through a set of instances and relations. For each article validated by writers, the full content is inserted in the database and a representation of the article is created in the knowledge base. An instance of the concept article and an instance of each *isAbout* relation are created to link the article to its criteria. For each profile made by the sellers, the associated representation is created in the knowledge base. An instance of the concept profile is created, and each *isInterestedIn* relation between the profile instance and its criteria are instanciated. We present the comparison method used in the section 6.

4 Automatic indexing task

The automatic indexing task (fig. 1 (B)), is the automatic association of a set of labels from the taxonomic thesaurus to each article. This task needs the following process: (i) a text analysis process to extract keyword and other features from texts and (ii) a machine learning process to learn the classification process from examples.

This paper does not aim at improving the state of the art in multi-classification, nor in ontology learning from text, but instead to propose a method to semantically enrich the ontology by adopting machine learning processes in order to both classify and describe classification, so the gap between the experts perspective and the classification rules representation is filled. First we present the method and then its evaluation.

4.1 Four steps method for automatic indexing

Our method is based on four following steps.

1. The *vectorization* phase allows generating the matrix of term frequencies from a learning set.
2. The *resolution* allows creating logical constraints (rules) associated to the keywords of the taxonomy (controlled vocabulary) using named entities extracted in the vectorization phase. This phase generates a flat ontology.
3. The *hierarchization* allows to generate a class hierarchy of subsumption of the ontology used to label documents.
4. The *realization* allows searching and deducing the most specific classes of documents to be classified which consists in generating the multi-classification.

Phase 1: uses the indexing work already done by librarians and a text analysis process to extract keywords, to generate a matrix that presents the frequency of each word for each label.

Phase 2: uses the matrix to define rules able to define whether an item should be associated with a label based on the terms contained. Two frequency thresholds are defined, α and β. The words whose frequency is greater than the threshold α are considered as reliable clues. The presence of one of these words is considered as sufficient to consider that the document must be associated with the label. The β threshold is a lower frequency. In this case we need a combination of β-terms (having frequency greater then β) to confirm the label for a document. More information about the indexing rules can be found in the following publication [REF].

Phases 3 and 4: are done using standard reasoner: FaCT++, HermiT and Pellet.

Phase 3: the classification provides two types of results. The first is the discovery of the most specific subsuming class. The second allows to infer equivalence classes when the logical constraints are equivalent. On one hand, this means that when a document is labeled with a class that has subsumers, this document will also be labeled by subsumant classes. On the other hand, when a document is labeled with a class that has the equivalence classes then this document is also labeled with equivalent classes. These two elements can achieve a multi-labeling, knowing that the terms of the taxonomy are hierarchical. Accordingly, this is a hierarchical multi-label classification (HMC) process.

Phase 4: the realization phase consists in finding all the most specific classes of individuals. This phase is carried out by the inference engine which enables to deduce all the more specific classes. It also allows to manage multi-labeling while adding subsuming and equivalent classes. As consequence, a document is multi-labeled according to a hierarchy.

4.2 First evaluation of the approach

In this section we present a preliminary evaluation of the indexing approach. Due to the lack of real data for our platform, the evaluation is based on the *delicious* dataset available on the Mulan [4] project web site already used in some multilabel-classification works [9]. It was extracted from the del.icio.us social bookmarking site on the 1st of April 2007 and contains textual features and tags of webpages. This dataset is used to train a classifier for tag recommendation.

Dataset	**Exemples**		**Attributes**		**Labels**		
	Train	Test	Numeric	Nominal	Count	Cardinality	Density
— delicious —	12920	3185	0	500	983	19.020	0.019

Table 1. The dataset in some numbers.

[4] http://mulan.sourceforge.net/

With this dataset the (*phase 1*) manual multi-classification and the (*phase 2*) feature extraction tasks are not necessary: features and tags are already associated with documents and a sub-dataset is predefined for the (*phase 3*) learning of the prediction model. Our prediction model is the set of α and β-rules. The ontology is populated (*phase 4*), and reasoners are used to perform the multi-classification task. The results produced by the reasoner are not only a multilabel-classification of documents, but also a hierarchical reorganization of tags based on the equivalence rules. The table 2 presents a benchmark of the three best reasoners evaluated on different hardware.

α-rules	**FaCT++**	**HermiT**	**Pellet**
i7 4Go DDR3	50 s	n.e.m.[5]	n.e.m.
Xeon E3 24Go DDR3	-	8 h	18 h
α and β-rules	**FaCT++**	**HermiT**	**Pellet**
i7 4Go DDR3	n.e.m.	n.e.m.	n.e.m.
Xeon E3 24Go DDR3	n.e.m.	out[6]	out
Xeon E5 128Go DDR3	2 h / out[7]	out	out

Table 2. Reasoner time computation comparison with α and β-rules.

Table 2 shows that the β-type rules are much more time and memory consuming. We have only one result to show. Only FaCT++ produced a result with the best machine and an ontology without any instance (i.e. document). Within 2 hours the reasoner infers a hierarchical reorganization of tags based on the equivalence rules. Yet the ontology populated with documents and equivalent class rules seems very time consuming even for FaCT++. The ontology with β-type rules is not evaluated in the following steps due to the lack of results provided by reasoners.

Evaluation	**Precision**	**Recall**	**F1-Mesure**
Proposal with α-rules	30%	6%	10%
HOMER [29]	-	-	25%

Table 3. Evaluation and comparison with a similar work of multi-classification.

This precision-recall evaluation is only based on α-rules, because of the difficulty for reasoners to provide results with β-rules. The table 3 shows that the quality of the results are low. Another approach [29] with this dataset also shows low value for the F-measure.

[5] Not enough memory

[6] Too much time consumption (more than 3 days)

[7] Only the hierarchical reorganization of tags for the document less ontology

One of the consequences of our method to create rules (i.e. with an average of 10 terms for all rules) is the creation of some ruleless classes. With this method, for our 983 classes, only 427 have rules. There are 556 classes without labeling rule (obviously, theses classes should have had β-rules). So there are classes that the predictive model can not affect. This impacts very negatively the Recall.

In our α-rules, we consider that the presence of one of the selected terms for the rule is a sufficient clue. In fact, the terms selected for theses rules are in the majority of cases not frequent enough to be a sufficient evidence to qualify the class. So, the presence of only one of them is not enough for the decision making. This also affects very negatively the Precision. The solution could be β-rules. They allow to take a decision, based on a minimum of clues. β-rules are a small step to gain intelligence, but the impact on the computation time and memory used is very important. Using only β-rules can positively impact the Precision, but probably negatively the Recall.

The realization phase, shows interesting results such as the detection of semantic proximity between the terms tags "blogger", "blogging", "wp" (i.e acronym for wordpress) and "wordpress". . For instance,the semantic proximity is detected between the terms tags blogger, blogging, wp and wordpress, but these results will not be described further here.

5 Profiling and profil refinement

The previous section addresses the indexing of articles (fig. 1 (B)), profiling (fig. 1 (C)) (i.e. profiles indexing) is made similar way. Keywords are extracted from the articles that have been loved by users, as well as notes taken by the experts during the conversation with the customer. Manual profiling done by the experts can be used as a basis for the learning task. So automatic indexing is realiszed in the same way as it is described in the previous section. But manual indexing performed by the experts is not precise, it aims to launch the system in response to the cold start problem. A phase of refinement is necessary to identify more precisely the users needs.

To capture the user behavior, we integrate some sensors in the web application which presents the economic articles recommended. These sensors are of three types. The first one permits to capture the significance of an article due to the login of the user. When the user redirects an the information, his/her logs are updated and taken into account. The second sensor is the *time* spent to read an article. In order to read the complete article, the user has to click on a button with start a timer. The click on the button is itself an important information. The last sensor is a *news filter system* that allows to regenerate the web page regarding the most important information for the user. The navigation behavior is stored in our recommender system to qualify the profile refinement, using the three kinds of footprints: actions, reading time, and domain specic knowledge.

6 Automatic recommendation task

The recommendation task (fig. 1 (D)) is mainly based on the comparison between the user profile and the items available for recommendation. The semantic description of each article and profile is stored in an ontological knowledge base. In order to use the vector space model, it is necessary to transform these descriptions into vectors. This modelling is less expressive than an ontology because the dimensions being orthogonal in a vectorial model, every element of each vector is considered independantly ti the other [33].

According to the previous section, the description vectors of the articles and profiles only consist of the instances of the criteria which are directly related to them in the knowledge base (i.e. terms of each facet with which the article intance or profile is related to in the knowledge base).

We develop a *vector expansion* method based on the work of [13]. It is adapted to preserve the ontology knowledge in the vectors. The instances of each criterion are organized in a hierarchy in the knowledge base. For each added instance, its parent instances are also added.

6.1 Similarity versus Relevance

An article can more or less fit the needs of a user, that's why we use the vectorial model to estimate the relevance. Unlike classic approaches that mix up the concepts of similarity and relevance[27], we distinguish them.

Similarity: $Similarity(x, y) : I \times I \rightarrow [0, 1]$ is a function that evaluates the similiraty degree between two objects x et y. In our case, x is an article and y is a profile. This function must satisfy the three properties of *positivity*, *reflexivity* et *symmetry*. The similarity evaluation in a vectorial space can be done by different measures, such as cosine similarity, Jaccard similarity or euclidean distance. In this article, we will illustrate our comment with cosine similarity because it is the most used in the literature. The cosine similarity between two vectors $\overrightarrow{a}$ and $\overrightarrow{p}$ is based on the measure of the angle Θ between the two vectors.

Relevance: $Relevance(x, y) : I \times I \rightarrow [0, 1]$ is a function that measures the relevance degree of an article x for a profile y. This relevance measure must respects the properties of *positivity* and *reflexivity*, but not *symmetry*. Relevance is a concept widely used in the information retrieval field. In our case, the relevance is not binary.

Hierarchical relations between two related topics is a good example to highlight the difference between similarity and relevance: if a given user is interested in the city of Paris, he may be interested in more specific information (monuments, history...), but not in more general information about France. To solve this issue, we use an intermediate vector. The sub-vector $\overrightarrow{s_c}$ consists of the shared instances between the vectors of the article $\overrightarrow{a_c}$ and the profile $\overrightarrow{p_c}$. Therefore we define the relevance for a given criterion c in the following way :

$$Relevance_c(\overrightarrow{a_c}, \overrightarrow{p_c}) = \frac{\omega'_{1,c} \times Similarity_c(\overrightarrow{a_c}, \overrightarrow{s_c}) + \omega'_{2,c} \times Similarity_c(\overrightarrow{p_c}, \overrightarrow{s_c})}{\omega'_{1,c} + \omega'_{2,c}}$$

With S_c the sub-set of shared elements of the set of instances in relation with both the profile $I'_{p,c}$ and the article $I'_{a,c}$; $S_c = I'_{p,c} \cap I'_{a,c}$. $\forall i_{x,c} \in S_c$, the vector $\overrightarrow{s_c}$ consists of the elements of the set S_c; $\overrightarrow{s_c} =< i_{1,c}, i_{2,c}, ..., i_{t,c} >$. With this method, it is possible to balance the weight of the precision difference between profiles and articles. In our case, we use $\omega'_{1,c} = 1$ et $\omega'_{2,c} = 4$, considering that the precision loss of the profile compared to the article mustn't influence the result by more than 20%. Moreover, the precision loss of the article compared to the profile must influence greatly the result, i.e. by 80% in our case. The global $Relevance(\overrightarrow{a}, \overrightarrow{p})$ is the sum of the relevance measures for each criterion, eventually weighted. This measure is used in our prototype to sort the results(articles) proposed to the user, according to his profile:

$$Relevance(\overrightarrow{a}, \overrightarrow{p}) = \frac{\sum \omega_c * Relevance_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \omega_c}$$

6.2 Experiments

We have compared two different aspects (binary and rank evaluation) of the results of the articles recommendation via cosine similarity (C), cosine similarity with extended vectors (B) and *Relevance Measure* with extended vectors (A), which handles the precision difference between profiles and articles. For our evaluations, the set consists of 10 profiles and 70 articles corresponding to one day of articles production. For the binary evaluation, a manual selection of the relevant articles has been conducted for each profile by experts. For the rank evaluation, a manual ranking of the relevant articles has been conducted for each profile by experts. In both cases, the results of the different algorithms are compared to the work of experts, considered as an ideal recommendation.

Binary evaluation: To evaluate the binary recommendation we use the classical measures used in information retrieval, i.e. *precision*, *recall* and *F1-measure*, [21]. Every article with a correlation to the profile superior to 0.5 is kept. The results

Algorithms	Precision	Recall	F1-measure	Kendall Tau rank	Spearman's rank
A	**0.856**	**0.971**	**0.910**	**0.837**	**0.899**
B	0.916	0.453	0.607	0.830	0.894
C	0.883	0.181	0.301	0.713	0.694

Table 4. Result of binary and rank evaluation measures for each algorithm

of the binary evaluation of the recommendation presented in table 4 confirm the results of [33], as for the interest of *vector expansion*, and show that our *Relevance Measure* method, with dinstinction of the relevance and similarity gives the best result.

Rank evaluation: To evaluate the rank evaluation of the recommendation, we use two most popular linear correlation measures: Spearman's rank and Kendall Tau rank. The output score of these method ranges from -1 to 1, 0 being the lack of similarity, 1 the complete similarity and -1 the opposite. The results of the evaluation are presented in table 4. The interest of *vector expansion* is also confirmed here. Moreover, the results show that the use of our *Relevance Measure* enhances furthermore the performances of the system.

7 Conclusion

This article describes the architecture developped in the context of global project to develop a recommender system of economic news articles.
The first point addressed in this papier is utomatic indexing article and profiles based on controlled vocabularies contained in an OWL-DL ontology using reasoners : We describe the process of using an HMC approach to enrich an already existing ontology to be used for automatic multi-classification of economic news articles. We decided to capture the prediction model into the taxonomic thesaurus part of the ontology, thus transforming it into a more semantically rich ontology. Based on the early experiments, it was observed that the logical axioms/rules suggested the existence of several subsumption relations that were not present in the taxonomic thesaurus, giving rise to Direct Acyclic Graphs, i.e. a class can have more than one super-class. While this observation is potentially relevant for the refinement of the taxonomic thesaurus and therefore for the classification, a deeper and finer analysis and expert-based experiments have to be performed to better understand the advantages, disadvantages and potential applications. Moreover, our preliminary tests have highlighted the complexity of reasoning on ontology, even with a relatively small ontology.
The second point addressed in this paper is comparing articles and profiles indexed with vocabularies from the knowledge base using an approach based on the VSM. We present the adaptation of a standard VSM recommender system to our specific method of indexing (e.g. articles and profiles are semantically defined in the knowledge base via relations with the domain knowledge already defined in it). We explain the specific task of comparison that we adapted to our case and finally we evaluat our algorithms using both binary and graded evaluation and show that the results are improved.

References

1. Ahn, J.w., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open user profiles for adaptive news systems: help or harm? p. 11. ACM Press (2007), http://portal.acm.org/citation.cfm?doid=1242572.1242575

2. Balabanović, M., Shoham, Y.: Fab: Content-based, collaborative recommendation. Commun. ACM 40(3), 66–72 (Mar 1997), http://doi.acm.org/10.1145/245108.245124
3. Bi, W., Kwok, J.T.: Multilabel classification on tree- and dag-structured hierarchies. In: ICML. pp. 17–24 (2011)
4. Billsus, D., Pazzani, M.J.: A personal news agent that talks, learns and explains. In: Proceedings of the Third Annual Conference on Autonomous Agents. pp. 268–275. AGENTS '99, ACM, New York, NY, USA (1999), http://doi.acm.org/10.1145/301136.301208
5. Cantador, I., Bellogn, A., Castells, P.: News@hand: A semantic web approach to recommending news. In: Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, vol. 5149, pp. 279–283. Springer Berlin Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-70987-9_34
6. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
7. Elsayed, A.E., El-Beltagy, S.R., Rafea, M., Hegazy, O.: applying data mining for ontology building. 42nd Annual Conference On Statistics, Computer Science, and Operations Research 2007 (2007)
8. Fellbaum, C.: WordNet: an electronic lexical database. MIT Press, Cambridge, Mass (1998)
9. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the c-value/nc-value method (2000)
10. Garrido, A.L., Gomez, O., Ilarri, S., Mena, E.: An experience developing a semantic annotation system in a media group. In: Lecture Notes in Computer Science, vol. 7337, pp. 333–338. Springer (2012), http://dblp.uni-trier.de/db/conf/nldb/nldb2012.html#GarridoGIM12
11. Getahun, F., Tekli, J., Chbeir, R., Viviani, M., Yetongnon, K.: Relating RSS News/Items. In: Web Engineering, pp. 442–452. No. 5648 in Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2009)
12. Holden, N., Freitas, A.: Hierarchical classification of g-protein-coupled receptors with a pso/aco algorithm. IEEE Swarm Intelligence Symposium (SIS 06) 2006 (2006)
13. IJntema, W., Goossen, F., Frasincar, F., Hogenboom, F.: Ontology-based news recommendation. In: Proceedings of the 2010 EDBT/ICDT Workshops. pp. 16:1–16:6. EDBT '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1754239.1754257
14. Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of implicit and explicit feedback from an online music recommendation service. In: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems. pp. 47–51. HetRec '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1869446.1869453
15. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Trans. Inf. Syst. 25(2) (Apr 2007), http://doi.acm.org/10.1145/1229179.1229181
16. Johnson, I., Abcassis, J., Charnomordic, B., Destercke, S., Thomopoulos, R.: Making ontology-based knowledge and decision trees interact: An approach to enrich knowledge and increase expert confidence in data-driven models. In: Lecture Notes in Computer Science, vol. 6291, pp. 304–316. Springer (2010), http://dblp.uni-trier.de/db/conf/ksem/ksem2010.html#JohnsonACDT10

17. Kellar, M., Watters, C., Duffy, J., Shepherd, M.: Effect of task on time spent reading as an implicit measure of interest. Proceedings of the American Society for Information Science and Technology 41(1), 168–175 (2004), http://dx.doi.org/10.1002/meet.1450410119
18. Kelly, D., Belkin, N.J.: Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 408–409. SIGIR '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/383952.384045
19. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: A bibliography. SIGIR Forum 37(2), 18–28 (Sep 2003), http://doi.acm.org/10.1145/959258.959260
20. Kondert, .F., Schandl, T. and Blumauer, A.: do controlled vocabularies matter? surevey results pp. 17–35 (2011)
21. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: SIGIR 94, pp. 3–12. Springer London (Jan 1994), http://link.springer.com/chapter/10.1007/978-1-4471-2099-5_1
22. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Recommender Systems Handbook, pp. 73–105. Springer (2011), http://dblp.uni-trier.de/db/reference/rsh/rsh2011.html#LopsGS11
23. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological user profiling in recommender systems. ACM Trans. Inf. Syst. 22(1), 54–88 (Jan 2004), http://doi.acm.org/10.1145/963770.963773
24. Pantel, P., Lin, D.: A statistical corpus-based term extractor. In: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence. pp. 36–46. AI '01, Springer-Verlag, London, UK, UK (2001), http://dl.acm.org/citation.cfm?id=647462.726284
25. Rao, K., Talwar, V.: application domain and functional classification of recommender systemsa survey 28(3), 17–35 (2008)
26. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. pp. 175–186. CSCW '94, ACM, New York, NY, USA (1994), http://doi.acm.org/10.1145/192844.192905
27. Salton, G.: The SMART retrieval system - experiments in automatic document processing (1971)
28. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24(5), 513–523 (Aug 1988), http://dx.doi.org/10.1016/0306-4573(88)90021-0
29. Tsoumakas, G. and Katakis, .I.a.V.I.: effective and efficient multilabel classification in domains with large number of labels. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08) 2008, 30–44 (2008)
30. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. Int J Data Warehousing and Mining 2007, 1–13 (2007)
31. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Mach. Learn. 73(2), 185–214 (Nov 2008), http://dx.doi.org/10.1007/s10994-008-5077-3
32. Vogrincic, S., Bosnic, Z.: Ontology-based multi-label classification of economic articles. Comput. Sci. Inf. Syst. 8(1), 101–119 (2011), http://dblp.uni-trier.de/db/journals/comsis/comsis8.html#VogrincicB11
33. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR 94, pp. 61–69. Springer London (Jan 1994)