Method to Manage the Precision Difference between Items and Profiles

David Werner, Christophe Cruz
University of Burgundy, France
{david.werner , christophe.cruz}@u-bourgogne.fr

Abstract

Contractors, commercial and business decision-makers need economical information to drive their decisions. The production and distribution of a press review about French regional economic actors represents a prospecting tool on partners and competitors for the businessman. Our goal is to propose a customized review for each user, thus reducing the overload of useless information. Some systems for recommending news items already exist. The usefulness of external knowledge to improve the process has already been explained in information retrieval. The system's knowledge base includes the domain knowledge used during the recommendation process. Our recommender system architecture is standard, but during the indexing task, the representations of content of each article and interests of users' profiles created are based on this domain knowledge. Articles and Profiles are semantically defined in the Knowledge base via concepts, instances and relations. This paper deals with the relevance measure, a critical sub-task in recommendation systems and the relationships between relevance and similarity concepts. The Vector Space Model is a well-known model used for relevance ranking. Firth, the problematic exposed here is the utilization of the standard VSM method with our indexing method. Then, we show some experiments, classic precision recall, and more specifics evaluation methods to compare ranking results. Finally, we exposed evaluation results of our recommendation algorithms and discuss the case of multilabeled items.

1 Introduction

The decision-making process in the economic field requires the centralization and intake of a large amount of information. The aim is to keep abreast with current market trends. Thus, contractors, business men and sales persons 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 reviews sent are the same for each user, but personalized according to a geo-graphic area. All articles sent do not necessarily correspond to a person's needs, and can be a waste of time. To reduce the overload of useless information, we are moving towards a customized review for each user. Therefore, an opinion survey on magazine readers that covers a broad array of subjects, including news services, was undertaken. 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, major transverse projects, temporal and localization data about each element underlined. Therefore, the complete production process of the review was redesigned to produce and to automatically distribute a customized review for each user. Another drawback in the existing process is the produced information storage. News articles are stored as PDF file reviews (i.e. the same format that is sent to customers, natural language), but this unstructured format is hard to handle and reuse.

The aim of the architecture is to manage all news produced, and provide the most relevant article for each customer, using our domain knowledge. 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 [Middleton et al. 2004] [Getahun et al. 2009] [Liu et al. 2010] [Tanev et al. 2008] SCENE [Li et al. 2011] NewsJunkie [Gabrilovich et al. 2004] Athena [Ijntema et al. 2010] GroupLens [Resnick et al. 1994] News Dude [Billsus, Pazzani. 1999] et YourNews [Brusilosky et al. 2007]. Some of these systems use domain knowledge to improve the recommendation task [Ijntema et al. 2010] [Middleton et al. 2004].

To achieve this goal, a content-based recommender system is being developed. A recommender system is necessary for the item ranking. And content-base is required to analyze the content of each article to structure and preserve information content. The results of the analysis enable to link the domain knowledge to the articles. Because the domain knowledge can be reused to improve the recommendation task [Ijntema et al. 2010] [Middleton et al. 2004].

This paper is organized as follows: Section 2 presents a brief review of related work. In section 3, we outline the proposed solution of a recommender system. Section 4 and 5 presents respectively the similarity measure task and the relevance measure task, and our implementation in the VSM. Section 6 proposes an evaluation of algorithms presented in the section 4 and 5. In section 7 we discuss the relevance computation in the complex case of multi-label indexing. And finally, the Section 8 is the conclusion and future work.

2 Related work

Our work is related to several works in News recommender systems, SCENE [Li et al. 2011] NewsJunkie [Gabrilovich et al. 2004] Athena [Ijntema et al. 2010] News Dude [Billsus, Pazzani. 1999] et YourNews [Brusilosky et al. 2007]. The survey of K. Nagewara Rao [Nageswara 2008] proposed a general comparison of the main advantages and drawbacks of each kind of Recommender System (e.g. content based or collaborative filtering). The advantages of content-based recommender systems for news recommendation are also explained in [Liu et al. 2010] to improve the Google news platform. The main drawback of collaborative filtering systems is the recommendation of new items. Novelty is very important in the particular case of news; each new article must be quickly recommended. Waiting for enough users to have read it before recommending is a big waste of time. Furthermore, we need to be able to recommend for very particular user profiles, as some customer needs are unique.

There are many systems that work without Knowledge [Liu et al. 2010] [Billsus, Pazzani. 1999] [Resnick et al. 1994]. The advantages of using exterior knowledge for enhancing the recommendation were exposed by Ijntema [Ijntema et al. 2010]. He uses the name semantic-based recommender system to distinguish standard content-based systems from the systems using external knowledge (e.g. domain ontologies or lexical knowledge as WordNet [Fellbaum 1998]). Lexical knowledge is used by [Getahun et al. 2009] and domain knowledge by [Middleton et al. 2004]. Athena uses both [Ijntema et al. 2010]. Ontologies used by these systems already exist or are created by hand, and maintained. Unlike previous systems, the Knowledge base containing domain Knowledge is used as an index for articles and profiles, as it is explained in section 3. To compare profiles and articles, classic VSM is not directly usable, so we have adapted it, as presented in section 4.

3 Implementation

Our system is an ontology drive content based recommender system. An ontology schema is created and populated with the help of company experts, in order to model the domain knowledge in a knowledge base. In a classic content-based recommender system, we distinguish two main tasks. The first is indexing. The task is to create a representation of the users' needs, and item content. The Knowledge base will be populated during this task. The quality of content analysis is important for the knowledge base population and for indexing, so our system is semi-supervised by an expert. The second task is comparison. This task is the comparison with item representation so as to measure the degree of relevance for each profile. Items are ranked with the help of the similarity measure, after being provided to the user. These subjects are developed as follows.

3.1 The Knowledge base

The knowledge base $\mathcal{K}$ used for this system is composed of several ontologies [Werner et al. 2012]. An Upper level ontology is used to manage information shared by all application areas (in the case of an extension of the system to new fields of application). High level concepts like location, geospatial information, temporality, events, agents, etc. Domain ontology is used to manage domain-specific knowledge. Concepts of this ontology are mainly specialization of concepts from the upper level. Other ontologies are used to manage articles, profiles, and lexical resources used by a gazetteer. The lexical resource ontology is based on the ontology PROTON used on the KIM platform [Popov et al. 2003]. In this paper the knowledge base model defined by Ehrig et al. [Ehrig et al. 2004] and based on the Karlsruhe Ontology Model is used [Stumme et al.].

Definition 1
Ontology:

$$O = (C, T, \leq_C, \leq_T\ R, A, \sigma_a, \sigma_R, \leq_R, \leq_A)$$

WhereinC, T, R, A are disjoint sets of concepts, data types, relations and attributes, $\leq_C, \leq_T, \leq_R, \leq_A$are hierarchy of classes, data types, relations and attributes, and σ_a, σ_R are function that provide the signature for each $\sigma_a: A \rightarrow C \times T$ attribute and $\sigma_R: R \rightarrow C \times C$ relation.

Definition 2
Knowledge Base:

$$\mathcal{K} = (C_{KB}, T_{KB}, R_{KB}, A_{KB}, I, V, \iota_C, \iota_T, \iota_R, \iota_A)$$

Wherein $C_{KB}, T_{KB}, R_{KB}, A_{KB}, I, V$ are disjoint sets of concepts, data types, relation attributes, instances and data values. ι_C is the classes instantiation function $\iota_C: C_{KB} \rightarrow 2^I$. ι_T is the data type instantiation function $\iota_T: T_{KB} \rightarrow 2^V$. ι_R is the relation instantiation function $\iota_R: R_{KB} \rightarrow 2^{I \times I}$. ι_A is the attribute instantiation function $\iota_A: A_{KB} \rightarrow 2^{I \times V}$.

3.2 Indexing

To archive 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 used has an index. Articles and profiles are represented by instances in our knowledge base. Some relations in the system ontologies are used to model of article content, and users' interests.

Articles Indexing
The ambition is to create a machine understandable representation of the content of each article, so as to compare with profiles. The unstructured information contained in articles is analyzed. Two kinds of information can be distinguished: explicit pieces of information (e.g. places, persons, organizations and so on) and implicit pieces of information (e.g. the theme of each article). In the system, the theme is one criterion, corresponding to the main event related by the article. Company experts have predefined a hierarchical list of themes wherein each article must be classified. The tasks of information extraction, annotation, indexing are done with the help of the GATE platform [Cunningham 2002]. A web interface was developed. It enables the writers to create articles. Results of the analysis are presented in it. They can be validated/corrected by the writer.

Analysis
Some post processes are applied into articles, such as tokenizers, sentence splitters, POS taggers, gazetteers (which use the knowledge base lexical resources as a dictionary) before JAPE patterns matching engine. In the first prototype, we used handmade lexico-semantic patterns. The aim is to extract important entities (explicit pieces of information), Persons, Organizations, places, dates, etc.… Results of analysis are hand-checked, corrected and validated. Implicit pieces of information are specifically handmade.

Population
For each article analyzed, an instance of the concept article is created in the knowledge base, so as to represent the article. Automatic analysis, correction done by hand and specifications of not automatically funded criteria are used to characterize the content of each article. Relations are created in the knowledge base between the article's instance, and criterion' instances (result of the analysis). Instances of criteria and relations with the instances of articles permit to index the article and create a semantic representation understandable by the machine.

Profiles Indexing
In the company, sellers are in charge of understanding the needs of each customer. Several phone calls are necessary so as to acquire future customers. During the phone call the seller proposes a free trial period. This helps to create a first handmade profile for each customer by an expert, and avoids the problem of a cold start, common to all content-based recommender systems.

The profile indexing process is the same as the articles. A profile instance is created in the knowledge base. Relations are created between the profile instance and criteria instances. A web interface was developed to enable sellers to define / change the user profile. The choices are reflected in the Knowledge base that permits to index the profile, and create a semantic representation understandable by the machine.

3.3 Recommendation

The recommendation task is mainly based on the comparison between the profile and available items. The knowledge base is used by the system like an index, and profiles and articles are presented by 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 relation "isAbout" is created to link the article with its criteria. For each profile made by the sellers, the 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 instanced. We present the comparison method used in the following section.

4 Comparison

The comparison task enables to evaluate the relevance of an article for a profile, via some similarity measures between them. Classical methods directly used the similarity as a measure of relevance. The profile can be seen like the ideal item, so, more an item is similar to the ideal item (profile) more it is relevant to the user interests.

Definition 3
Similarity: $\mathrm{SIM}(x, y)\colon I \times I \to [0,1]$ is a function to measure the degree of similarity between x and y . The similarity function can satisfy some properties:
$\forall x, y \in I \ \mathrm{SIM}(x, y) \geq 0$ Positiveness
$\forall x, y \in I \ \mathrm{SIM}(x, x) = 1$ Reflexivity
The last one is the symmetry:
$\forall x, y \in I \ \mathrm{SIM}(x, y) = \mathrm{SIM}(y, x)$. But this axoim is open for debate in the community.
To extricate ourselves from the debate on symmetric or unsymmetric property of similarity, we chose to remain the classical analogy with the distance and so, to define the similarity with the symmetry axiom. However our final measure, the relevant measure, is an unsymmetric function. In our context, we need an asymmetric function for the relevance, because we consider that comparing a profile and an article is not the same as comparing two articles.

4.1 VSM

An approach based on the vector space model [Salton 1970] was used in the prototype. Articles and profiles are represented by vectors on a space wherein each dimension is a potential instance of criteria. Several methods can be used to compare vectors; the most common is the cosine similarity.

$$\mathrm{SIM}(\vec{a}, \vec{p}) = \cos\theta = \frac{\vec{a}.\vec{p}}{|\vec{a}| \times |\vec{p}|}$$

$$\mathrm{SIM}(\vec{a},\vec{p}) = \frac{\sum_{x=1}^{t} i_{a,x} * i_{p,x}}{\sqrt{\sum_{x=1}^{t} i_{a,x}^2} + \sqrt{\sum_{x=1}^{t} i_{p,x}^2}}$$

An article can be defined like a vector of instances of Entities and criteria. For the recommendation task in the prototype only instances of criteria are used.

$$\vec{a} = \{i_1, i_2, \dots i_t\}$$

Wherein, an article is represented by a vector $\vec{a}$ composed by a set of instances i_x.

$$i_x \in I'$$

Instances i_x are instances of concepts belonging to the set of concepts C'defined by indexing criteria. The set I' contains all instances of all concepts in the set C'.

$$I' \subseteq I$$

The set I' is a subset of the set of all instances I of the Knowledge Base $\mathcal{K}$.
A profile can be defined as a vector of instances of criteria.

$$\vec{p} = \{i_1, i_2, \dots i_t\}$$

Where, a profile is represented by a vector $\vec{p}$ composed of a set of instances i_x.
In our implementation, one vector for each criterion is used. This enables to weigh or use different similarity measures (e.g. cosine, jaccard, Euclidian) for each criterion. For example, location, theme and sectors are much more important than project and organization and so highly weighed.

$$\mathrm{SIMF}(\vec{a},\vec{p}) = \frac{\sum \mathcal{W}_c \mathrm{SIM}_c(\overrightarrow{a_c},\overrightarrow{p_c})}{\sum \mathcal{W}_c}$$

$\mathrm{SIM}_c(\overrightarrow{a_c},\overrightarrow{p_c})$ Is the similarity between profile $\overrightarrow{p_x}$ and article $\overrightarrow{a_x}$ and $\mathcal{W}_x$ the coefficient for a specific criterion c. $\forall i_{x,c} \in I'_c$ $\overrightarrow{p_c} = \{i_{1,c}, i_{2,c}, \dots i_{t,c}\}$ And $\overrightarrow{a_c} = \{i_{1,c}, i_{2,c}, \dots i_{t,c}\}$. One or more concepts are defined for each criterion.

Methods from the information retrieval fields can be used to enhance the recommendation. One of the first systems using external knowledge to improve the understanding of user needs is [Voorhees 1994]. The Voorhees approach used WordNet [Fellbaum 1998] to provide a query expansion. We can translate this kind of method to recommender systems, instead of query expansion; we can name this method 'profile expansion'. Middleton [Middleton et al. 2004] uses this method without naming it. Ijntema [Ijntema et al. 2010] also uses it, but unlike Middleton, he uses more powerful ontology relations (not just is_a) to expand the user profile. In our system, the profile expansion takes the form of adding instances, e.g. if the user U' profile shows an interest for company Co and in the knowledge base a symmetric relation like is_aSubsidiaryOf is instanced with another company Co', it is possible to add the other company to his profile. The Expanded VSM is developed in the following section.

4.2 Expanded Vectors

The 4.1 section presents an implementation of the similarity measure between two instances, by the creation of vectors of related instances. It was explained by Voorhees [Voorhees 1994] that all dimensions are orthogonal in the VSM and so, all elements of each vector are considered as independent. That is not really the case for the lexical used by Voorhees, and instances used in our system. In Voorhees's method, the meronomic and synonymic relations defined in WorldNet are used to add related lexical to the vector. In our case, relations between individuals exist, because we used a Knowledge base to manage the domain knowledge, and our criteria are hierarchically defined in it.

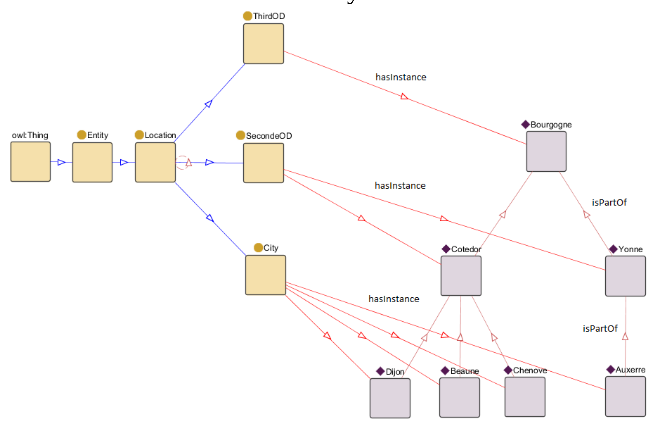

Figure 1. Example of location hierarchy.

Meronomic relations exist between instances of Location. With Voorhees's method, it seems logical to expand the profile (the query for Voorhee) with all sub instances enclosed by the most general instance. For example (fig 1), if the profile is interested in Bourgogne (Region, biggest Administrative division), it seems logical to add Cote d'Or and Yonne (Departement, sub administration division of Region) which are two departments within the Bourgogne region, and Dijon, Beaune, Chenove, Auxerre, which are cities within these departments. But, in the real case with this method, it is necessary to add four departments and its 2047 cities to the vector. The similarity between a profile interested in Bourgogne and an article about Dijon will be very low with this method.
So our method is to expand profile and articles, not only the profile and limiting the size of vector, instances added are including instances, not included (by meronomic relations). So our method is analogue to graph-based methods which search the common ancestor.
With this method the first drawback is solved, but the second still is not. Our Relevant function is symmetric and equivalent to the Similarity function. The precision of an article must not have the same consequences for the Rele-

vance as the precision of a profile. This problem is the subject of the following section.

5 Relevance

The previous section looked at how to take into account the relations between instances in the VSM. In this section in contrast with the section 4 the relevance is no longer defined as an equivalent of the similarity. This section is about the managing of the difference of precision between profiles and articles and its influence in the relevance e.g. if an article is about Dijon and a Profile is interested in Bourgogne, the relevance must be higher than the case of an article about Bourgogne and Profile is interested in Dijon, because of the loss of precision (Dijon is a city located in Bourgogne). The problem is the same if the profile is interested in a large variety of distinct locations, and the article is about at least one of these. The similarity for this criterion must be high. To solve this problem we used an intermediate vector for each criterion, a sub vector $\overrightarrow{s_c}$ composed of common instances of the article $\overrightarrow{a_c}$ and profile $\overrightarrow{p_c}$ vectors for the criteria.

Definition 4
Relevance: $\mathrm{Relevance}(\mathrm{x}, \mathrm{y}): \mathrm{I} \times \mathrm{I} \rightarrow [0,1]$ is a function to measure the degree of relevance of an article x relative to a profile y. The similarity function can satisfy some properties:
$\forall \mathrm{x}, \mathrm{y} \in \mathrm{I} \;\; \mathrm{Relevance}(\mathrm{x}, \mathrm{y}) \geq 0$ Positiveness
$\forall \mathrm{x}, \mathrm{y} \in \mathrm{I} \;\; \mathrm{Relevance}(\mathrm{x}, \mathrm{x}) = 1$ Reflexivity
The relevance is a concept from information science largely used in the fields of information retrieval and recommender systems. In our case the Relevance is not binary; an article can more or less meets the information need of the user.

Definition 5
Precision: In the hierarchy of concepts, more generals concepts enclosed more specifics ones. In the hierarchy of instances it is the same. Instances from the top of the hierarchy are less specific than instances from the bottom.
If the Article is about an instance form the bottom and the Profile is interested in top instance (of the same branch), the relevant must be higher than if the Article is about a top instance and the Profile is inserted in an instance from the bottom of the hierarchy because there is a loss of precision if the profile is more specific than the article, so the article is less relevant.

$$S_c = I'_{p,c} \cap I'_{a,c}$$

S_c is the sub set of common elements of the set of instances related to the profile $I'_{p,c}$ and the article $I'_{a,c}$.

$$\forall i_{x,c} \in \mathrm{S}_c \quad \overrightarrow{s_c} = \{i_{1,c}, i_{2,c}, \dots i_{t,c}\}$$

The vector $\overrightarrow{s_c}$ is composed by elements of the set S_c.

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

It is possible to weigh differently the precision of the profile and the precision of the article with this method. In our implementation we used $\mathcal{W}'_{1,c} = 1$ and $\mathcal{W}'_{2,c} = 4$, because we consider that the difference of precision of the profile must not influence the final note over 20%. However, it is possible to change the value, and it is also possible to use different values according to the criterion.

$$\mathrm{RelevanceF}(\vec{a}, \vec{p}) = \frac{\sum \mathcal{W}_c \times simGlob_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \mathcal{W}_c}$$

The final Relevance $\mathrm{RelevanceF}(\vec{a}, \vec{p})$ value is the sum of the Relevance measure for each criterion eventually weighted. This measure is used for the ranking of articles proposed to the user according to his profile.
With this method the Relevance value for the case A.2 (cf. Figure 2.) is higher than the case A.1, because in the case A.2, a and p have more common ancestors than in the case A.2. Moreover, the cases B.1 and B.2 figure the problem of precision. With our asymmetric method the value of similarity between a and p in the case B.1 is higher than in the case B.2 because the user needs are specific and the article information (about this criterion) very general relative to the user needs. So the article is less relevant for the user. The following section presents the conclusion and future work.

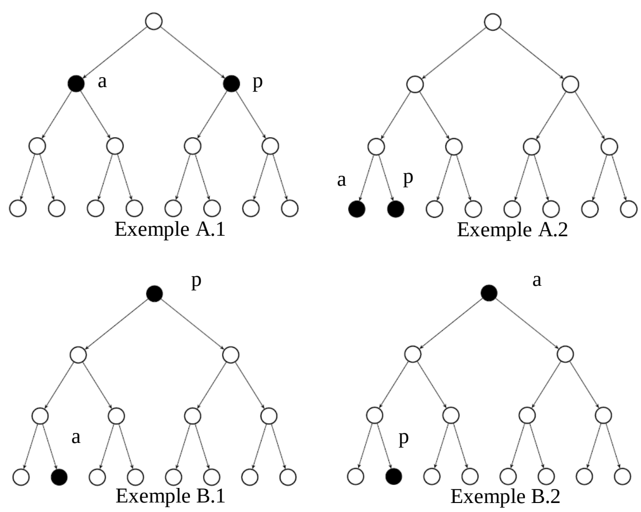

Figure 2. Examples of profiles and articles for one criterion.

6 Evaluation

The evaluation of recommender systems is a well-known problem. The comparison and evaluation of different approaches are difficult. Our System is specific for the type of items we try to recommend, we developed an indexing method based on some criterion extracted and/or deducted from the content of articles. In our case, we want to compare the method of creation of vectors and its influence on the relevance results. The measure of relevant is made by the comparison between the vectors that represent the user's needs (i.e. the profile) and the vectors that represent the content of each item (i.e. the articles). For the comparison between vectors, we used classical methods for VSM (i.e. Cosine Similarity, Jaccard Similarity and Euclidean Distance).

We compared three recommender algorithms, in two different ways (i.e. binary evaluation and graded evaluation). The first algorithm is the classical VSM model (cf. 4.1). It will be named as A in the following. The second is a modified VSM with an upwardly expansion of the two vectors (i.e. article and profile) (cf. 4.2). It will be named as B in the following. And the third is a new method, based on VSM model (cf. 5.). It will be named as C in the following. This method used a common sub vector and allows managing the difference of precision between profiles and articles. With these three methods, we used Jaccard, Euclid and Cosine algorithms for the comparison of vectors (i.e. profile and article).

6.1 Binary Evaluation

In the first way, we evaluated the binary recommendation. A set of 10 profiles and 70 articles was selected. A handcraft selection of perceived as relevant articles for each profile was performed by experts. Then the handmade selection was compared with results of recommender algorithms. To perform a binary evaluation, binary results are needed from recommender algorithms, but they provide ranked articles with a score between 0 and 1 for each. So, we defined a threshold beyond which an item is recommended and below which it is not. The threshold of 0.5 was chosen for the binary relevant evaluation. The classic binary evaluation measure is the compute of precision and recall.

Definition 6

Precision Recall and F-measure: are functions that produce scores ranging from 0 to 1.

$$Precision = vp / (vp + fp)$$

$$Recall = vp / (vp + fn)$$

With: true positive vp (i.e. item correctly recommended), false positive fp (i.e. item recommended but it should not be, so loss of precision), false negative fn (i.e. item not recommended but it should be, loss of recall) and, true negative vn (i.e. item correctly not recommended)

Optimization of precision and recall are central problematic in recommender systems. To consider both of them Lewis and Gale [Lewis, Gale, 1994] proposed a single measure the F-measure. The F-measure is a weighted combination of precision and recall that produce scores ranging from 0 to 1.

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times Precision \times Recall}$$

The β can be used to increase the importance of the precision compared to the recall and vice versa.

We used the classical F1-measure because we chose to assign equal importance to the precision and the recall. Generally, the precision is preferred to the recall in the cases of very large quantity of information, like the web, because of the redundancy of information. In our case, there is no redundancy, so we used β=1.

Algorithms	Precision	Recall	F1-measure
COSINE C	0.85609674	0.9714286	**0.91012347**
COSINE B	0.9166666	0.453721	0.6069975
COSINE A	0.8833334	0.18142858	0.30102864
JACCARD C	**0.9285715**	0.58809525	0.7201167
JACCARD B	0.8833334	0.15017858	0.25671256
JACCARD A	0.8833334	0.15017858	0.25671256
EUCLID C	0.5663902	0.9714286	0.71556884
EUCLID B	0.39675242	**0.9857143**	0.56577784
EUCLID A	0.54957575	0.49571428	0.52125734

Table1. Precision, Recall and F1-measure of each algorithm.

The results of the binary evaluation show that for each comparison method (i.e. Jaccard, Cosine, Euclid) when the vectors are upwardly extended (B) with our method (cf. 4.1), results are at least as good as classical (A) vectors for the F1-measure. It is the same observation for the comparison between expanded vectors (B) and our method with a common sub-vector (C) (cf. 5). The method with a common sub-vector (C) has at least as good results as expanded vectors (B) for the F1-measure. Our Method with a common sub-vector (C) and the cosine comparison of vector give the best results, with a good precision and Recall, and the best balance between them. We can observe a loss of precision between classic vectors (A) and expanded vectors (B) when the comparison method used is Jaccard. This problem of loss of precision has already been explained by Voorhees with his own method of expansion of vector (i.e. query expansion) [Voorhees 1994]. Indeed, the expansion of vectors aims the improvement of the recall as shown the results, which may have a cost in precision.

6.2 Graded Evaluation

In the second way, we evaluated the graded relevance. A set of 10 profiles and 70 articles was selected. A handcraft ranking in term of perceived as relevant articles for each profile was performed by experts. Then the handmade ranking was compared with results of recommender algorithms. To evaluate the graded relevance, we used popular measure of rank correlation, the Spearman rho and the Kendall's tau.

Definition 7

Spearman's rho and Kendall's tau: are metrics, which are used to measure the rank correlation, i.e. the similarity of results of the different ranking process. They produce scores ranging from -1 to 1. 0 is the absence of similarity, 1 the complete similarity and -1 the opposite.

Algorithms	Kendall's tau	Spearman's rho
COSINE C	**0.8367523906880769**	**0.8988165004887531**

COSINE B	0.8301725748175933	0.8940660193684133
COSINE A	0.7132217110093708	0.6937291126970984
JACCARD C	0.8362910125299354	0.8963642702840753
JACCARD B	0.8192108979830239	0.8863520480293786
JACCARD A	0.7126200638359759	0.6934511938708817
EUCLID C	0.7280499691108524	0.8169690973677465
EUCLID B	0.6493379299699591	0.7342625550561453
EUCLID A	0.548795410154673	0.6155680923763756

Table 2. Correlation measure, Kendall's tau and Spearman's rho of each algorithm.

The graded evaluation shows equivalent results to the binary evaluation. Expanded vectors (B) give better results than classical vectors (A) and our method with a sub-vector (C) gives the best results and that with all method of comparison of vectors (i.e. Jaccard, Cosine, Euclid). With the results of the two evaluation methods, we can conclude than our second method (cf. section 5) improve the F1-measure and give the best ranking of items. This method of relevance computation provides the best recommendation results when it is used with the cosine comparison of vectors.

7 Discussion

In the previous section, our method to compute the relevancy has been explained. In order to illustrate this point the figure 2 shows some examples which are composed of profiles and articles single labeled for only one of the criteria. This is a simplification of the problem which is in fact, for each criterion, multiple values. This section deals with multi labeled items and profiles, but all examples will be based on single criterion to ease the understanding.
For the perspective of a user, the perceived relevance is high when at least one of the profile's values matches with one of article's values for the same criterion. However, even if our method has good results, it remains in some cases side effects. Large profiles (e.g. with multiple label) are managed as deep profiles (e.g. specific). The figure 3 shows examples of side effect profiles. The Case C.1 is a large profile composed of three labels. The case C.2 is a deep profile. Unfortunately, in both cases the relevance of the article "a" for the profile "p" is equivalent with our algorithm.

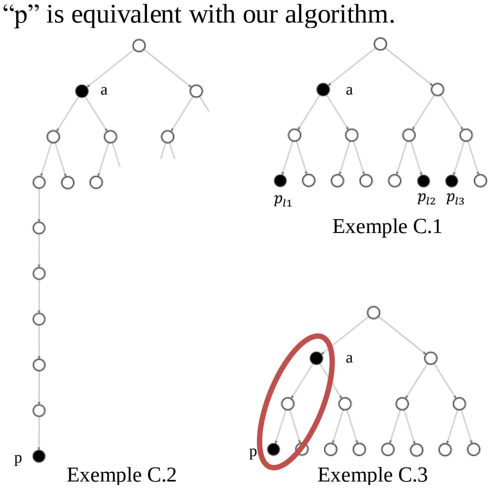

Figure 3. Examples for multilabeled profiles.

Actually, the perceived relevance for the user should be equivalent to the example C.3. A solution consists in comparing only close values. For instance, a relevance matrix should be used to compare pairwise values (Tab. 3).

	p_{l1}	p_{l2}	p_{l2}
a	0.8	0.1	0.1

Table 3. a relevance matrix pairwise.

This method permits to manage the difference of precision between multi-labeled profiles and articles, without confusing the width and depth. However, this process is time consuming. In addition, the number of relevance computation is exponential with regards the number of selected values for each profiles and articles. As a drawback, the advantages of vector space model computation celerity are lost. In the evaluation section, profiles and articles possess no more than two values for each criterion. This is why the relevance measured by the algorithm and the relevance felt by the user remain close. Thus, with the inclusion of a method similar to the relevance matrix pairwise, results have probably been better.

8 Conclusion and futur work

In this paper we have presented 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 first presented the context, our goal, and the existing approaches, then we explained the architecture in detail. Secondly we explained the specific task of comparison that we adapted to our case. Finally we evaluated our algorithms using both binary and graded evaluation and showed that the results are improved.
Some things must be taken into account. Firstly, we used a small set of articles and profiles for the evaluation, our goal in the future work is to lunch a new evaluation, with a bigger set and to taking into account user feedbacks about their user experience. Secondly, the Knowledge based used for the indexing relatively poor, but a new one is already under development; and finally, the performances, time consuming of each algorithm have not been evaluated. It is an important point in our system because it must be able to work with a large number of users.
As future work, we plan to improve our recommendation with a graph-based semantic similarity measure between profiles and articles. The idea is that our ontology is already structured information, so to restructure it to a vector space model for the comparison task must be useless. Profiles and articles can be seen as sub-ontologies. Moreover, side effects (cf. 7) must be more efficiently bearing out of the vector space model. Some approaches to compare ontology instances already exist [Albertoni, De Martino. 2006] [Ehrig et al. 2004].

Acknowledgement

This project is founded by the company Actualis SARL and the financing CIFRE research grant from the French agency ANRT. Special thank for the “Conseil Régional de Bourgogne” which fund partially this work.

References

[Albertoni, De Martino. 2006] Albertoni, R.—De Martino, M.: Semantic Similarity of Ontology Instances Tailored on the Application Context. On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, Vol. 4275, Lecture Notes in Computer Science, 2006, pp. 1020–1038.

[Aone, Ramos-Santacruz. 2000] Aone, C., Ramos-Santacruz, M.: REES: A Large-Scale Relation and Event Extraction System. In: 6th Applied Natural Language Processing Conference (ANLP 2000). pp. 76-83. Association for Computational Linguistics (2000)

[Billsus, Pazzani. 1999] Billsus, D., Pazzani, M.J.: A Personal News Agent that Talks, Learns and Explains. In: The Third Annual Conference on Autonomous Agents, ACM (1999) 268–275

[Brusilosky et al. 2007] Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open User Profiles for Adaptive News Systems: Help or Harm? In: 16th international conference on World Wide Web, ACM (2007) 11–20

[Cunningham 2002] Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities 36 (2002) 223–254

[Ehrig et al. 2004] Ehrig, M., Haase, P., Stojanovic, N., and Hefke, M.: Similarity for Ontologies - A Comprehensive Framework. ECIS 2005. Regensburg, Germany (2005)

[Fellbaum 1998] Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)

[Gabrilovich et al. 2004] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In WWW 2004, pages 482–490.

[Getahun et al. 2009] Getahun, F., Tekli, J., Richard, C., Viviani, M., Yetongnon, K.: Relating RSS News/Items. In: 9th International Conference on Web Engineering, Springer (2009) 442–452

[Ijntema et al. 2010] IJntema, W., Goossen, F., Frasincar, F., Hogenboom, F.: Ontology-Based News Recommendation. In: International Workshop on Business intelligence and the Web (BEWEB 2010), EDBT/ICDT Workshops. pp. 16:1{16:6. ACM, New York, USA (2010)

[Lewis, Gale, 1994] Lewis, D. and Gale, W.A. A sequential algorithm for training text classifiers. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1994, London, Springer Verlag.

[Li et al. 2002] Li, F., Sheng, H., Zhang, D.: Event Pattern Discovery from the Stock Market Bulletin. In: 5th International Conference on Discovery Science (DS 2002). Lecture Notes in Computer Science, vol. 2534, pp. 35-49. Springer-Verlag Berlin Heidelberg (2002)

[Li et al. 2011] Li L, Wang D, Li T, Knox D, Padmanabhan B. SCENE: A scalable two-stage personalized news recommendation system. In Proc. the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, Jul. 25-29, 2011, pp.124-134.

[Liu et al. 2010] J. Liu, P. Dolan, and E.R. Pedersen. Personalized news recommendation based on click behavior. In Proceedings of the 15th International Conference on Intelligent User Interfaces, pages 31–40. ACM, 2010.

[Middleton et al. 2004] Middleton, S.E., Shadbolt, N.R., Roure, D.C.D.: Ontological User Profiling in Recommender Systems. ACM Transactions on Information Systems 22(1) (2004) 54–88

[Nageswara 2008] Nageswara Rao, K., and V.G. Talwar, Application Domain and Functional Classification of Recommender Systems—A Survey‖, Journal of Library & Information Technology, Vol. 28, No. 3: 17-35, 2008.

[Piskorski, et al. 2007] Piskorski, J., Tanev, H., Wennerberg, P.O.: Extracting Violent Events From On-Line News for Ontology Population. In: 10th International Conference on Business Information Systems (BIS 2007). Lecture Notes in Computer Science, vol. 4439, pp. 287-300. Springer-Verlag Berlin Heidelberg (2007)

[Popov et al. 2003] Popov B., Kiryakov A., Kirilov A., Manov D., Ognyanoff D., Goranov M., KIM- Semantic Annotation Platform 2003

[Resnick et al. 1994] 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, Chapel Hill, North Carolina, United States. ACM Press, New York, 1994. pp. 175-186.

[Salton 1970] G. Salton. The SMART retrieval system : Experiments in automatic document processing. Prentice Hall, 1970.

[Stumme et al.] Stumme, G., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Sure, Y., Volz, R., Zacharias, V.: The Karlsruhe view on ontologies. Technical report, University of Karlsruhe, Institute AIFB (2003)

[Tanev et al. 2008] Tanev, H., Piskorski, J., Atkinson, M.: Real-Time News Event Extraction for Global Crisis Monitoring. In: 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems (NLDB 2008). Lecture Notes in Computer Science, vol. 5039, pp. 207-218. Springer-Verlag Berlin Heidelberg (2008)

[Voorhees 1994] Voorhees E., Query Expansion using Lexical-Semantic Relations. 1994

[Werner et al. 2012] Werner D.; Christophe Cruz C.; Christophe Nicolle C.; Ontology-based Recommender system of economic articles; In Proceedings of the 2012 Webist Conference, Porto.