Knowledge-Based Systems 22 (2009) 421–429

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Discovering unexpected documents in corpora

François Jacquenet *, Christine Largeron

University of Lyon, University of Saint-Etienne, Laboratoire Hubert Curien, UMR CNRS 5516, 18 rue Benoit Lauras, F42000 Saint-Etienne, France

ARTICLE INFO

Article history:
Received 12 April 2008
Accepted 25 May 2009
Available online 9 June 2009

Keywords:
Text mining
Unexpected patterns
Information retrieval

ABSTRACT

Text mining is widely used to discover frequent patterns in large corpora of documents. Hence, many classical data mining techniques, that have been proven fruitful in the context of data stored in relational databases, are now successfully used in the context of textual data. Nevertheless, there are many situations where it is more valuable to discover unexpected information rather than frequent ones. In the context of technology watch for example, we may want to discover new trends in specific markets, or discover what competitors are planning in the near future, etc. This paper is related to that context of research. We have proposed several unexpectedness measures and implemented them in a prototype, called *UnexpectedMiner*, that can be used by watchers, in order to discover unexpected documents in large corpora of documents (patents, datasheets, advertisements, scientific papers, etc.). *UnexpectedMiner* is able to take into account the structure of documents during the discovery of unexpected information. Many experiments have been performed in order to validate our measures and show the interest of our system.

1. Introduction

Exceptions, outliers, rare events, hot topics, emerging patterns, etc., have long been ignored or considered as noise. Nevertheless, since the mid 90's this kind of data has attracted much attention. Indeed, with the rapid development of business intelligence techniques, such kind of information has revealed to represent potentially important information for strategic managers of companies. Business intelligence [13,14,24] has been defined as the set of actions involved in retrieving, processing, disseminating and protecting legally obtained information that is useful to economic players. When the data analyzed are scientific and technical, the more specific term used is technology watch, meaning the monitoring of patents and scientific literature (articles, theses, etc.). In fact a watch process can be broken down into four main stages: needs audit, data collection, processing of the data collected, integration and dissemination of the results. It is easy to understand that the most sensible step is the third one which is also the main concern of this article. For the purpose of automatically processing the data collected, data mining techniques are attractive and seem to be particularly suited considering that most of the data are available in digital format. Many of the data available for technology watch are in the form of textual data, thus we talk about text mining, following Feldman that introduces this term in 1995 [11]. Text mining has been defined by Sebastiani [32] as the set of tasks designed to extract the potentially useful information, by analysis of large quantities of texts and detection of frequent patterns. In fact text mining is now a wide area of research that provides useful techniques that can be used in the context of technology watch. Losiewicz et al. [22] for example show that clustering techniques, automatic summaries, information extraction can be of great help for business leaders. Zhu and Porter [47,46] show how bibliometrics can be used to detect technology opportunities from competitors information found in electronic documents. Another use of text mining techniques for technology watch has been proposed in [17] where the authors tried to find new trends from an IBM patent database using sequential pattern mining algorithms [1]. The idea was to observe over time, sequences of words that were not frequent in patents at a particular period and that became frequent later. In a similar way, but in the Topic Detection Tracking[1] framework, Rajaraman and Tan [30] proposed to discover trends from a stream of text documents using neural networks.

In fact we can see that in these works, text mining techniques have been mainly used to help managers dealing with large amount of data in order to find out frequent useful information or discover some related works linked with their main concerns. Nevertheless, one important goal of technology watch and more generally business intelligence is to detect new, rare, unexpected and hence generally infrequent information. Thus, the algorithms for extracting frequent patterns that are commonly used for data mining purposes are inappropriate to this area. Indeed, these tools are tailored for information that occurs frequently in a database.

* Corresponding author.
E-mail addresses: Francois.Jacquenet@univ-st-etienne.fr (F. Jacquenet), Christine.Largeron@univ-st-etienne.fr (C. Largeron).

[1] http://www.nist.gov/speech/tests/tdt.

0950-7051/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2009.05.009

This is no doubt one of the main reasons why the software packages marketed so far fail to fulfill knowledge managers needs adequately.

Thus, in the statistical and machine learning communities and then in the data mining field, discovering unexpected information has become a great challenge for many researchers. *Unexpectedness* [33,34,26] and *novelty* [7] have been introduced as evaluation functions for subjective interestingness of discovered patterns. Therefore, an increasing number of researches are now devoted to the automatic detection of what is called, depending on the authors, *emerging patterns*, *rare events*, *emergent topics*, *novelty* or *new information*. For example, *emerging patterns* [10,45] or *strong emerging patterns* [36] are patterns whose supports strongly varies from one dataset to another. Thanks to their capacity to characterize the classes, emerging patterns enable to build classifiers. The notion of *surprise* (or *surprising pattern*) introduced in [5] is based on variation of inter-item correlations over time. Other studies [39,38] focused on *exception rules*. In exception rule discovery, an exception represents a rule that deviate from the common sense rules of high generality and accuracy. Several methods have been proposed to discover exception rules, either directly [21,28,34], or indirectly [19,37] when common sense rules are not given. By extension, in exception structured-rules discovery [38], an exception represents a set of rules related to each other. In the same way, Tuzhilin et al. [28,27] proposed a method for discovering *unexpected rules* or minimal set of *unexpected patterns* which can be considered as a kind of exception rules where common sense rules are defined as a set of beliefs. These beliefs are provided by experts or learned from data and then selected by the experts. Like in our approach, this prior knowledge of decision makers is taken into account to seed the search for the data that are unexpected relative to these beliefs.

In the works cited above, the data that were considered were classical structured data stored into tables of relational databases where each line (or transaction) is described by features (or items) and the patterns extracted are either itemsets (emerging patterns, strong emerging patterns) or rules (exception rules, unexpected rules). In the case of more complex data such as time-series, video streams or texts, other approaches are required to handle them and discover interesting knowledge. In this paper we focus on text data that is the most frequent form of data available in digital libraries, or on the Internet, for watchers. The objective is then to discover unexpected information in the sense that they were previously unknown from the user.

Among the first works that focused on that subject, we may cite the *TDT* initiative (*Topic Detection and Tracking*[2]) launched by the *DARPA* in 1996 and that aims to identify new events in a stream of news [2,43]. The main approaches proposed in *TDT* rely on incremental classification algorithms, nearest neighbor and probabilistic models. More recently a challenge has been organized on the theme of *Novelty detection* in the framework of the *TREC* conference.[3] In this context, novelty or new information has been defined as "answers to the potential questions representing a user's request or information need" [18]. Nevertheless, the dataset proposed for *TREC 2003* is made up of sentences and not full texts [35]. Moreover, the list of documents provided, that is sentences for *TREC* or news for *TDT*, is sorted in a chronological order. Thus, the problem to be solved rather consists in searching for new documents over time such as in [6]. In such conditions, most systems propose to identify, using a similarity measure, a relevant document by comparing it with the preceding and consecutive ones in the dataset. Nevertheless, in various applications, such an approach is not relevant because the dataset is not chronologically sorted. This is the case in the context of technology watch where watchers have to identify texts describing technological innovations in scientific text corpora but also Web pages or datasheets for which they do not get any information about the release date. Other works have focused on the detection of emergent topics. Thus, Bun et al. [3,4] have proposed a system that may observe the changes that appear in a set of Web sites and that point out the emergent themes from words contained in those modified pages. Nevertheless, with such a strategy, the system is not able to find an unexpected information on a Web site during its first visit. Matsumura et al. [23] have also designed a system for emergent topics discovery based on Web communities. After having classified communities made up of members sharing common interests, the system analyze and visualize co-references between pages using the *KeyGraph* algorithm [25]. In that context, the emergent topics are Web pages that are interesting for several communities. The main weakness of this system is that it assumes such communities could be defined and that Web pages could be assigned to them.

WebCompare designed by Liu et al. [20] is probably the system which is the most related to our interest. It is intended to provide some technology watch facilities. The user has to specify the *URLs* from its competitors' Web sites. Then *WebCompare* is able to discover the pages that contain unexpected information with respect to the user's Web site. The unexpectedness of a Web page is calculated using a measure based on the $TF \times IDF$ paradigm, as the measures proposed for the *UnexpectedMiner* system presented in [15]. This latter system we designed previously aims to extract, from text corpora, documents that are relevant to the user because they contain previously unknown information that may interest him.

Nevertheless, one important feature of documents that is not taken into account in these works is their structure. Recent studies in information retrieval, classification and clustering, and more generally information mining [12,41,29,8,9,44,42] have shown the interest of taking the structure of documents into account. In particular, the continuous growth of XML data repositories has led to the development of content oriented XML retrieval which focuses on exploiting the available structural information of documents to implement more efficient information retrieval systems. Hence, the Initiative for Evaluation of XML Retrieval (INEX[4]), provides since 2002 large test collections and appropriate scoring methods for evaluating such systems. In the framework of new information discovery, taking the structure into account seems to be also justified because one can see that each part of a document does not have the same impact (weight) and we can expect it to be the same for their terms. In fact, in the domain of technology watch, the documents we have to analyzed are most of the time strongly structured. Indeed, they are scientific papers, or abstracts of Ph.D. thesis made up of clearly identified parts such as title, authors, keywords, abstract, sections, etc. It is the same with patents or information spread on the Web in the form of structured XML files. One important feature of our system is to be able to take into account the structure of documents that are processed.

Next section presents the global architecture of the system we designed, and called *UnexpectedMiner*, to automate the discovery of unexpected information in text corpora. Section 3 presents two ways of representing documents depending on if we take the structure into account or not. Section 4 shows how our system is able to take into account the structure of the documents in order to discover the unexpected information more accurately. Section 5 presents the various measures we proposed to mine unexpected information in texts. Section 6 presents some experiments we

[2] http://www.nist.gov/speech/tests/tdt.

[3] The "novelty detection" challenge was held for the first time at TREC 2002. The papers presented to this conference and next ones are available at the URL: http://trec.nist.gov.

[4] http://inex.is.informatik.uni-duisburg.de/2007/.

made to show the efficiency of each measure with respect to each other before we conclude with some future works.

2. Architecture of the unexpectedminer system

We have developed the *UnexpectedMiner* system that aims to extract documents that are relevant to a watcher from a corpus of documents inasmuch as they deal with topics that were unexpected and previously unknown to the manager. In addition, the system must specifically treat the watcher's request without relying to any large extent on her or his participation. Finally, an important feature we wanted to build into our system is genericity, i.e. the system must not be dedicated to a particular field or topic.

Keeping in mind those objectives, we proposed a system made up of several modules as illustrated in Fig. 1.

In the first stage, the watcher specifies his needs by producing some reference documents. In the remainder of this article, this set of documents shall be designated by *R*. In practice, between ten and twenty documents should be enough to target the scope of interest for the technology watch. The system must then review new documents from various corpora made available and retrieve unexpected information that may be innovative. This set of new documents shall be referred to below as *N*.

Sets *R* and *N* undergo a pre-processing stage. The module designed for that purpose includes a number of conventional processing steps such as removing irrelevant elements and stop words from the documents (logos, url, tags, etc.) and carrying out a stemming process on the words of all the sentences. Finally, each document is classically represented in vectorial form (see next section). Some similarity measures are then used in order to retrieve documents of *N* that are similar enough to documents of *R*, what builds the set *S*.

The documents of *S* are then processed in order to discover unexpected documents among them. In fact this task constitutes the core of the *UnexpectedMiner* system. The purpose of the unexpectedness module is to find the documents of *S* that contain unexpected information with respect to that contained not only in the reference documents *R* but also in the document subset *S* extracted in the previous stage. Indeed, a document is highly unexpected when it deals with topics that are found neither in any other document of *S* nor in any document of *R*. This module is described in detail in Section 5.

At the end of the whole process, the system returns a list made up of all the documents of *S* ordered by decreasing unexpectedness order. The first elements of that list are the most unexpected documents of *S* according to the system.

3. Document representation and similarity detection

3.1. Standard representation

One of the most widely used document representation in information retrieval is the vectorial form, first introduced by Salton and McGill [31].

In this model, an index lists all the terms $t_1, t_2, \ldots, t_m$ contained in the set of documents. Each document d_j is then represented by a vector of weights $\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{m,j})$ where $w_{i,j}$ represents the weight of the term t_i in the document d_j. If the term t_i does not appear in the document d_j then $w_{i,j} = 0$.

To calculate the weight of a term in a document we usually use the $TF \times IDF$ formula. *TF* (Term Frequency) measures the relative frequency of a term t_i in a document d_j and is defined by:

$$tf_{i,j} = \frac{f_{i,j}}{max_h f_{h,j}}$$

where $f_{i,j}$ is the frequency of the term t_i in the document d_j. The more frequent the term t_i in the document d_j, the higher $tf_{i,j}$.

IDF (Inverse Document Frequency) measures the discriminatory power of a term t_i and is defined by:

$$idf_i = \log_2 \frac{C}{n_i} + 1$$

where *C* is the size of the set of documents and n_i is the number of documents that contain the term t_i. The more rare the term t_i in the set of documents, the higher idf_i. Practically, the inverse document frequency of a term t_i is more simply calculated by:

$$idf_i = \log \frac{C}{n_i}$$

The weight $w_{i,j}$ of a term t_i in a document d_j is then obtained by combining the two previous criteria:

$$w_{i,j} = tf_{i,j} \times idf_i$$

This weight is large when the term t_i is frequent in the whole document d_j and rare in the others.

3.2. Taking the structure into account

Our system integrates this representation technique by adapting it in order to take into account the structure of the documents. Indeed, we propose to calculate the number of occurrences of a term in each part of the document instead of the whole document. Then, doing a weighted sum of these numbers of occurrences we

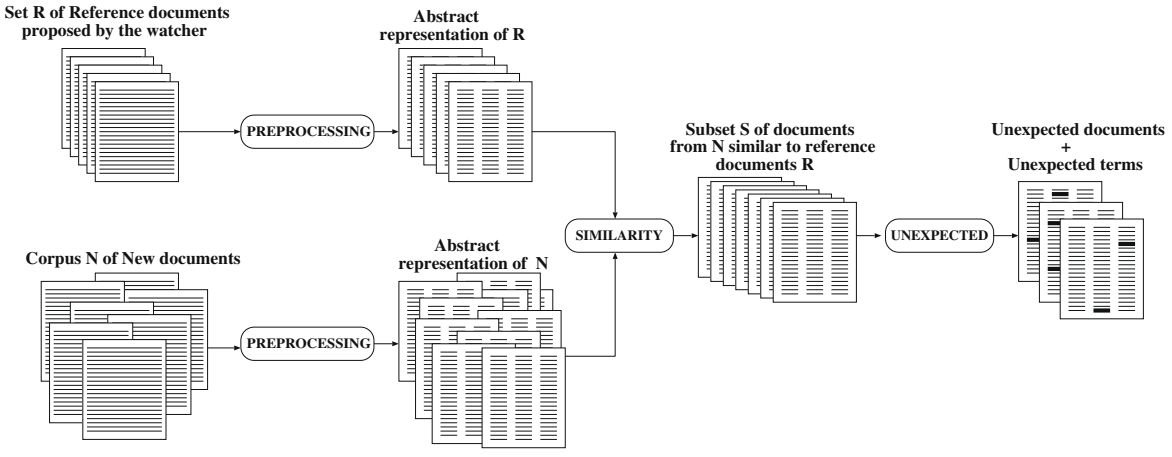

Fig. 1. Unexpected miner system architecture.

are able to calculate the global weight of each term in the document.

Thus, if in the documents processed by the system, we distinguish k different parts, the importance of each part l is given by a coefficient sc_l, that we called *structuring coefficient* such that:

$$\sum_{l=1}^{k} sc_l = 1$$

The weight of a term t_i in a document d_j is then calculated in that way:

$$w_{i,j} = k \sum_{l=1}^{k} sc_l \times tf_{i,j}^{l} \times idf_i$$

with

$$tf_{i,j}^{l} = \frac{f_{i,j}^{l}}{max_h f_{h,j}}$$

where $f_{i,j}^{l}$ is the frequency of the term t_i in the part l of the document d_j

We may note that assigning the same value $\frac{1}{k}$ to each coefficient leads to the vectorial representation of Salton. Then we have to determine the specific values of the structuring coefficients that have to be assigned to each part of the documents in order to effectively take the structure into account. Those values have to be linked to the relative importance of each part. For example, for documents splitted into three parts (title, keywords and body), one could decide for one dataset to assign the first part a coefficient twice higher than the other parts (that is we would define a weighting vector such as (0.5, 0.25, 0.25)) whereas for another dataset where keywords are more important, we could define a weighting vector such as: (0.25, 0.5, 0.25). In fact, even within the help of an expert, defining the structuring coefficients is a difficult task. It is the reason why we decided to make the system learn them automatically from a subset of the considered textual database to be processed.

3.3. Retrieval of similar documents

The goal of the second module of the *UnexpectedMiner* system is to extract from the database N new documents which are the most similar to the reference documents R provided by the watcher. The similarity s_{jk} between a new document $d_j \in N$ and a reference document $d_k \in R$ is equal to the cosine measure, commonly used in information retrieval systems. It is equal to the cosine of the angle between the vectors that represent these documents:

$$s_{jk} = \frac{\vec{d}_j \cdot \vec{d}_k}{|\vec{j}| \times |\vec{k}|}$$

where

$$\vec{d}_j \cdot \vec{d}_k = \sum_i w_{i,j} \times w_{i,k}$$

$$|\vec{j}| = \sqrt{\sum_{i=1,m} w_{i,j}^2}$$

The mean similarity s_j of a new document $d_j \in N$ with the set of reference documents R is equal to:

$$s_j = \frac{1}{|R|} \sum_{k=1}^{|R|} s_{jk}$$

After classifying the mean similarity of new documents in the descending order, a subset S is extracted from N. This is the set of the new documents that are the most similar to those supplied as reference documents by the watcher.

4. Choosing the structuring coefficients

We have seen in the previous section that in order to take the structure of documents into account while discovering unexpected information, we introduced some structuring coefficients in the calculus of each weight of terms in documents. Those coefficients have to be determined and the purpose of this section is to explain the process we implemented in the *UnexpectedMiner* system. Our aim is to make the system automatically determine the best structuring coefficients.

Hence, the objective is to search, using a sample set for which we know some of its documents contain unexpected information, the weighting vector that allow the system to discover these unexpected documents in the best way. The problem may be laid down in term of optimization techniques. To do so we have to specify an objective function, used to compare the various weighting vectors, that we have to maximize.

To build such a function, we may observe that, for each tested configuration (*e.g.* weighting vector), when the *UnexpectedMiner* system has finished to process the sample set, we get a list of documents sorted by decreasing unexpectedness order. Nevertheless, this ranking may unfortunately contain some documents that are not really unexpected. That means the system made a mistake each time such a document appears in the list. Nevertheless, the error is more important if it concerns a document at the beginning of the list rather than at the end. The objective function we have defined to quantify these ranking errors takes into account the position of the errors in the ordered list of unexpected documents. If the system extracts m documents, then the objective function for a configuration C_i is defined by:

$$f_{C_i} = \sum_{j=1}^{m} \left(\frac{1}{r} \times [j] \right)$$

where $[j]$ is equal to 1 if the document d_j, that appears at the r^{th} position, is not unexpected and 0 elsewhere. Thus, an error at the first position costs 1 whereas an error at the tenth position only costs 0.1. The weaker the value of the function, the better the ranking of the documents.

Let us suppose for example that we require the system to extract ten unexpected documents and that it makes three mistakes, the Table 1 gives some examples of values for the ranking function depending on the position of the errors made by the system while ranking the unexpectedness of these ten documents.

Finding the good configuration for the structuring coefficients that leads to the best ranking for the unexpected documents is equivalent to minimize the ranking function. To do so, we could calculate the value of this function for all the possible configurations and keep the one that leads to the lowest value of the ranking function. Such a process in nevertheless not realistic because the coefficients are real numbers and then it exists an infinite number of configurations to be tested. Moreover if we wanted to restrict the number of configurations to be tested by fixing a step between two possible values for a coefficient, the number of configurations would still remain large even when the number of parts of the documents is not large.

To solve this optimization problem, we have chosen to use a simulated annealing method [16]. The main idea of this method

Table 1
Examples of values for the ranking function.

Position of errors	Value of the ranking function
8, 9, 10	0.3361
3, 4, 7	0.7262
1, 6, 10	1.2667
1, 2, 3	1.8333

is, starting from an initial configuration, to make it go through some small changes step by step. If a change increases the value of the objective function, then it is kept. If it decreases the value of the objective function, then this configuration is kept with a probability p that depends on a combination of that decrease and the step of the process. In that way at the beginning of the process it is easier to accept an important change whereas at the end, when we are near the best solution, it becomes harder to accept even a little change. This probability is calculated using a Gibbs–Boltzmann distribution. To skip from a configuration C_i to a configuration C_j, we calculate the variation $\Delta f_{ij} = f_{Cj} - f_{Ci}$ of the objective function and the probability $P(i,j)$ to accept this transformation is then defined by:

$$\begin{cases} P(i,j) = 1 & \text{if } \Delta f_{ij} \leqslant 0 \\ P(i,j) = \exp^{-\frac{\Delta f_{ij}}{t}} & \text{if } \Delta f_{ij} > 0 \end{cases}$$

t is a control parameter equal to kT where T is the temperature and k is the Boltzmann's constant which is equal to 6.18E-23. This definition for the probability perfectly fit the requirements that is, if the temperature T is high, then a large number of transformations will be allowed and if it is low, only the transformations that improve the solution will be accepted. To determine if a new configuration is accepted, a random number is chosen between 0 and 1, and then compared to $P(i,j)$. If $P(i,j)$ is smaller, the solution is accepted, else it is rejected. Then the process iterates either with the new configuration or with the last one. All along the process, the best configuration is stored in memory.

Decreasing the temperature is done step by step and for each step, a given number of transformations are tried. Thus, each tested configuration belongs to the neighborhood of the current one and the configuration space is traversed without having all the configurations tested.

In this process, the neighborhood of a configuration and the decreasing value have to be fixed by the user. The neighborhood is defined using a step between two consecutive possible values for a coefficient; which is equivalent to state a valid interval for the coefficient of each part of the documents. Each coefficient can vary in that interval during a transformation. The only condition we have to assume is that the sum of the coefficients remains equal to 1. For example, if the current configuration is (0.05, 0.37, 0.58) for a document containing three parts and we allow a change of ± 0.02 for each weight, then a configuration in its neighborhood could be (0.06, 0.38, 0.56) or (0.03, 0.38, 0.59).

The initial temperature must be high enough at the beginning of the process to allow a large number of transformations to be accepted. Kirkpatrick et al. [16] proposed to choose a temperature t and then try some transformations and calculate the ratio of accepted transformations. If it is at least equal to 80% we can keep t as initial value, either, we multiply its value by two and try the initialization again.

The decreasing function g for the temperature is usually a geometric function $g = \mu t$ with $0 < \mu < 1$. If we choose a value far from 1, we may make the temperature decrease too rapidly. A good choice, according to Kirkpatrick is between 0.85 and 0.95.

The number of steps for the change of the temperature depends on the decreasing function. It must be chosen in such a way that, at the end of the process, the temperature is low enough for that no other transformation to be accepted. The value of t must be approximately equal to 5% of its initial value according to Kirkpatrick.

5. Unexpectedness measures

We have proposed four measures for assessing the unexpectedness of a document.

5.1. Measure 1

The first measure is derived directly from the criterion proposed by Liu et al. [20] for discovering unexpected pages on a Web site. It is defined by:

$$M1(d_j) = \frac{\sum_{i=1}^{m} U^1_{i,j,c}}{m}$$

with

$$U^1_{i,j,c} = \begin{cases} 1 - \frac{tf_{i,c}}{tf_{i,j}} & \text{if } tf_{i,c}/tf_{i,j} \leqslant 1 \\ 0 & \text{else} \end{cases}$$

where d_j is a document in S and $D_c = R \cup S - \{d_j\}$ is the document obtained by combining all the reference documents in R with the similar documents except d_j.

The main drawback of this measure is that it gives the same value for both terms t_i and $t_{i'}$ that occur with different frequencies in a new document $d_j \in S$ once these terms do not occur in D_c (in other words, in the other documents in $R \cup S - \{d_j\}$). Now it would be desirable to get an unexpectedness value $U^1_{i,j,c}$ for t_i greater than the value $U^1_{i',j,c}$ found for $t_{i'}$ when t_i is more frequent than $t_{i'}$ in d_j. This is particularly the case when t_i pertains to a word that has never been encountered before whereas $t_{i'}$ is a misspelled word. This consideration led us to propose and experiment other measures for assessing the unexpectedness of a document.

5.2. Measure based on term frequency (M2)

With the second measure, the unexpectedness of a term t_i in a document $d_j \in S$ with respect to all the other documents D_c is defined by:

$$U^2_{i,j,c} = \begin{cases} tf_{i,j} - tf_{i,c} & \text{if } tf_{i,j} - tf_{i,c} \geqslant 0 \\ 0 & \text{else} \end{cases}$$

Just as in $M1$, the unexpectedness of a document d_j is equal to the mean of the unexpectedness values associated with the terms representing d_j:

$$M2(d_j) = \frac{\sum_{i=1}^{m} U^2_{i,j,c}}{m}$$

This second measure gets rid of the drawback in the first one. Indeed, if we consider the previous example, if the term t_i occurs more frequently than $t_{i'}$ in document d_j and that neither appear in D_c, then:

$$U^2_{i,j,c} > U^2_{i',j,c}$$

However, none of these two measures takes into account the discriminatory power of a term as expressed by IDF. This inadequacy can partially be overcome by combining all the documents. Nonetheless, it seemed to us valuable to design unexpectedness measures that make direct use of this information, as with the two methods described below.

5.3. Measure based on discriminatory power (M3)

The third measure makes direct use of the discriminatory power idf_i, of the term t_i by evaluating the unexpectedness of a document d_j through the sum of the weights $w_{i,j}$ of the terms t_i that represent it (remember $w_{i,j} = tf_{i,j} \times idf_i$):

$$M3(d_j) = \sum_{i=1}^{m} w_{i,j}$$

With this measure, two documents d_j and d'_j may nonetheless have same unexpectedness value in spite of the fact that the weights of

the terms representing the first document are equal while those for the second document are very different.

5.4. Measure based on the highest weight (M4)

To overcome the limitation of $M3$, the fourth measure we proposed assigns the highest weight in a document's vector of representation as that document's unexpectedness value:

$$M4(d_j) = \max_l w_{l,j}$$

Tests have been performed to evaluate this system and compare the various measures, these are described in the next section.

6. Experiments

As we are mainly interested in this article in the discovery of unexpected information, we focus in this section on the efficiency of the unexpectedness module. We tested it depending on it takes the structure of the documents into account or not. In the first case, the documents were represented by the model introduced in Section 3.2 while in the second case they were described by the standard vectorial model. The initialization of the simulated annealing algorithm were done with random values assigned to the structuring coefficients.

6.1. Corpora and evaluation techniques

We did a set of experiments on five corpora we built in various domains using our own expertise and the one of a professional watcher. For each experiment, the set R of reference documents were made up of 20 English scientific papers dedicated to a main domain of interest. The set N contained 2000 new documents from which 200 documents (the ideal set S of similar documents to documents of N) were related to that main domain of interest. From S, 18 documents contained unexpected information according to the watcher. The 1800 remaining documents of N, not related to the main domain of interest, were papers about various other domains.

As we wanted to focus our experiments on the evaluation of the unexpectedness module, we therefore considered the set S made up of the 200 documents related to the main domain of interest as the input of the unexpectedness module. Let us recall that given R and S the unexpectedness module returns a list made up of all the documents of S ordered by decreasing unexpectedness order. The first elements of that list are the most unexpected documents of S according to the system. The criteria we used to evaluate the unexpectedness module are those commonly used in the information retrieval community, that is precision and recall [40]. In the context of our system, precision gives the percentage of documents extracted by the system that are truly unexpected. Recall measures the percentage of truly unexpected documents discovered by the system. The experiments we made on our five corpora provided some similar results, thus the tables and charts given in the next section have been calculated as the mean of the results obtained on each experiment made on each one of the five corpora. As we knew S contained at most 18 truly unexpected documents in each one of the five corpora, we drew the charts for precision and recall criteria in the following way. For all n between 1 and 18 we considered the n first documents of the ordered list of unexpected documents returned by the system and counted the number n_{truly} of truly unexpected documents from those n documents. Precision were then given by $\frac{n_{truly}\times 100}{n}$ and recall by $\frac{n_{truly}\times 100}{18}$. Fig. 2 illustrates those criteria. N is the set of 2000 new documents provided by the watcher, S is the set of 200 documents from N that are similar to the 20 reference documents R. Truly unexpected documents in S are noted *tu* while documents of S that are not unexpected are noted *nu* (we do not draw 200 documents in S in that figure due to space limitation of course). Now for example, if $n = 9$ and the 9 first documents of the list returned by the system are those in the dotted ellipse, we get *precision* $= \frac{6\times 100}{9} = 66\%$ and *recall* $= \frac{6\times 100}{18} = 33\%$.

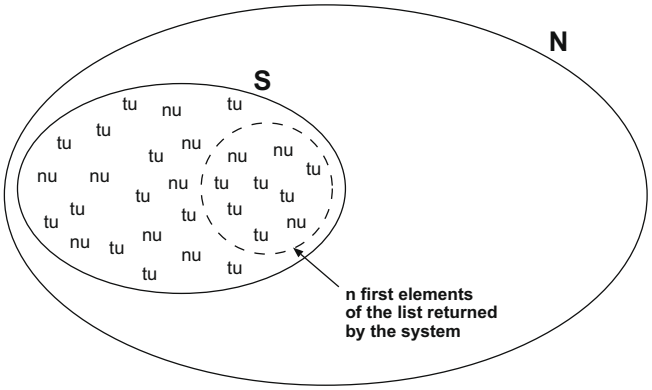

Fig. 2. An example of precision and recall evaluation of the system.

6.2. Evaluation of unexpectedness measures

The values of the ranking function obtained for each measure are given in Table 2 (without the structure) and Table 3 (with the structure).

The value of the ranking function being higher when errors were made at the beginning of the sorted list, it seems that measure $M4$, based on the highest weight, and measure $M3$, based on the weights of terms, really return as first documents of the ordered list, some truly unexpected documents. Measures $M1$ and $M2$ seem to generate some errors at the first positions of the sorted list returned by the system. Moreover, we can see that taking the structure of documents into account improves the efficiency of the unexpectedness module. Indeed, we may see in that case the value of the ranking function decreases significantly for measures $M1, M3$ and $M4$.

Measure $M4$, based on the highest weight, gives the best results while measure $M1$ generates the worst, as well as taking the structure into account or not.

Figs. 4, 6, 8, 10 on one hand, and Figs. 3, 5, 7, 9 on the other hand compare the efficiency of each measure taking the structure into account or not.

Table 2
Values of the ranking function for the unexpectedness module, not taking the structure into account.

Measure	Value of the ranking function
M1	2.07
M2	1.11
M3	1.31
M4	0.75

Table 3
Value of the ranking function for the unexpectedness module, taking the structure into account.

Measure	Values of the ranking function
M1	2.05
M2	1.11
M3	0.89
M4	0.59

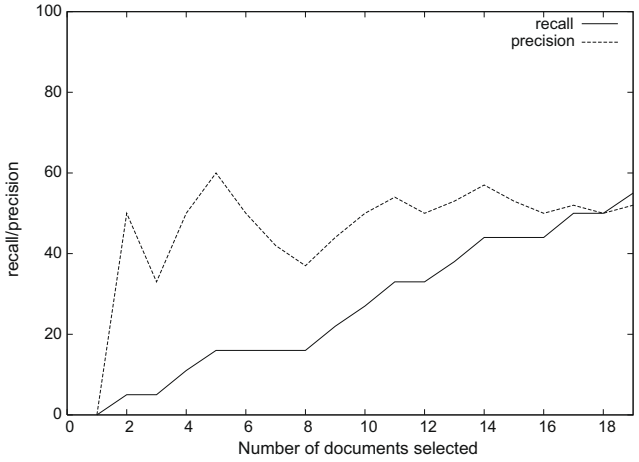

Fig. 3. Measure $M1$ without structure.

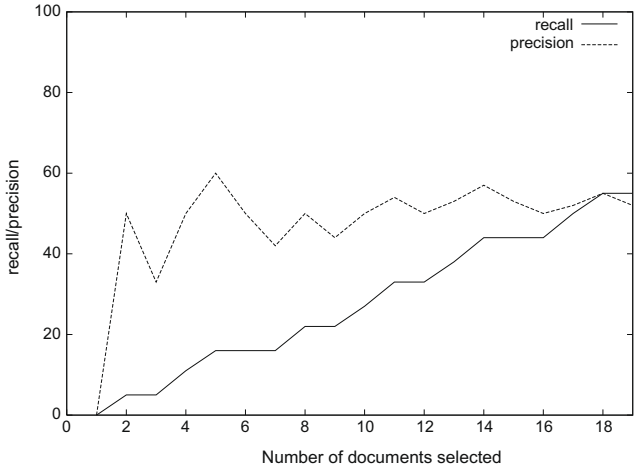

Fig. 4. Measure $M1$ with structure.

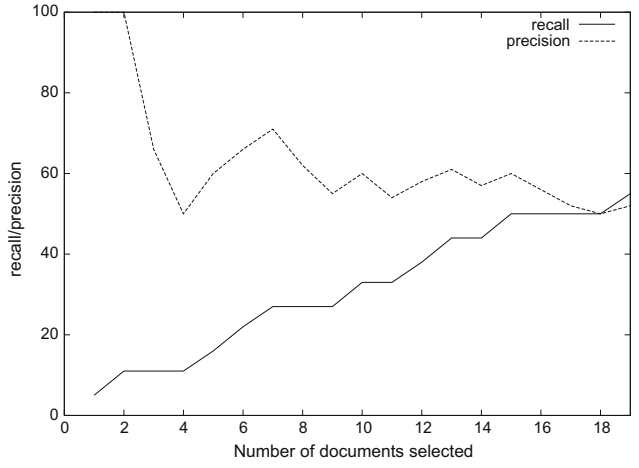

Fig. 5. Measure $M2$ without structure.

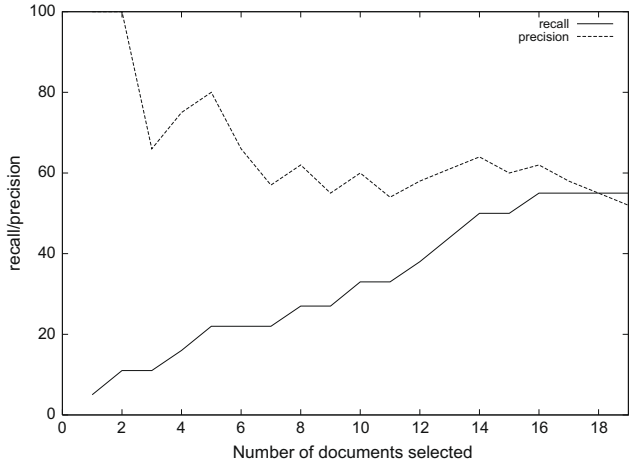

Fig. 6. Measure $M2$ with structure.

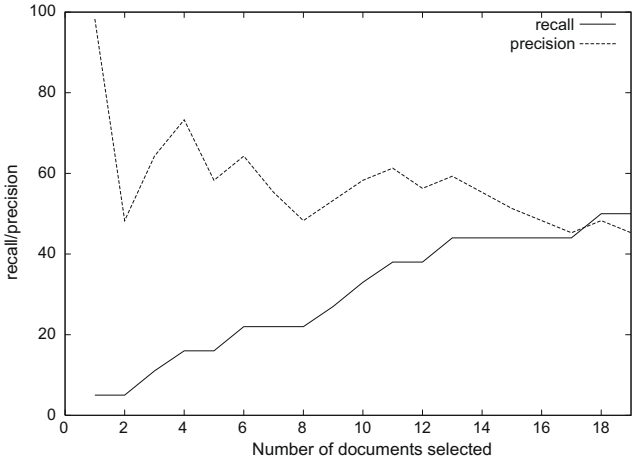

Fig. 7. Measure $M3$ without structure.

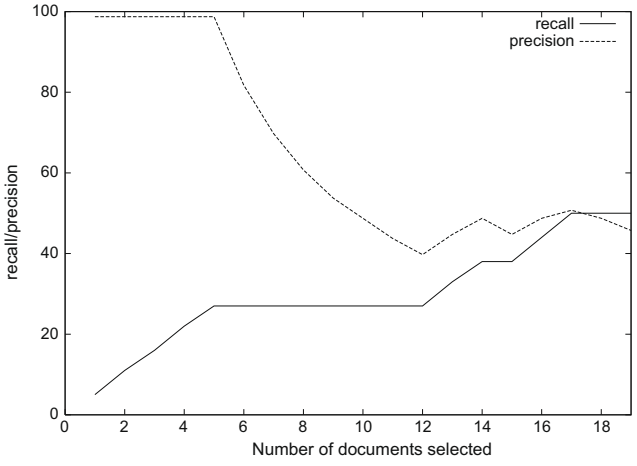

Fig. 8. Measure $M3$ with structure.

Without taking the structure into account, only the measure $M1$ makes a mistake, since the precision value is equal to 0%, considering the first document of the ordered list returned by the system (Fig. 3) while the value is equal to 100% for the other measures (Figs. 5, 7, 9). The results achieved while taking the structure of documents into account are more satisfactory. Indeed, if they are relatively unchanged for measures $M1$ and $M2$ (Figs. 4 and 6), they are really better for $M3$ and $M4$ since the system only makes its

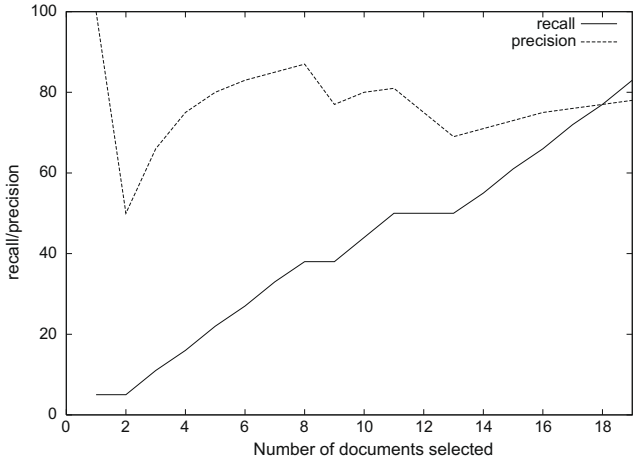

Fig. 9. Measure *M*4 without structure.

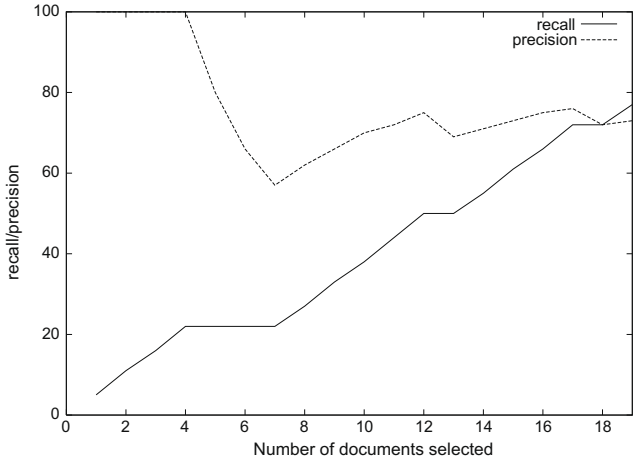

Fig. 10. Measure *M*4 with structure.

first mistake while returning the fifth or the sixth supposed unexpected document (Figs. 8 and 10). Measure *M*4 remains the best one whatever the number of unexpected documents requested to the system is, with or without taking the structure into account. Indeed, using *M*4 we may see that in the ordered list of documents returned by the system, among the 18 first documents, 15 documents are truly unexpected. Remember that from the 200 documents processed by the unexpectedness module we know that only 18 are truly unexpected ones. Using *M*4 and the structure of the documents, we can also see that the four first documents in the ordered list of documents returned by the system are truly unexpected ones. That means watchers may feel confident about the results obtained using *UnexpectedMiner* in that configuration.

7. Conclusion

In this paper we have presented a system, called *UnexpectedMiner* that we designed in order to discover unexpected documents from text corpora. We have shown how it was possible to take into account the structure of documents to improve its efficiency. Then we provided four measures to characterize the unexpectedness of a document with respect to a set of other documents.

Our experiments brought to the fore that taking the structure of the documents into account may significantly improve the efficiency of the *UnexpectedMiner* system to discover several unexpected documents, without any mistake, from a set of documents related to various domains.

In the future we want to mainly focus on improvements of the unexpectedness module which is the core of the system. In that way, increasing the efficiency of the automatic inference of structuring coefficients is an interesting domain of research that could significantly improve the overall results of the *UnexpectedMiner* system.

Acknowledgement

This work has been supported in part by the french national research agency under the Bingo project and by the ISLE cluster of the "région Rhône-Alpes" under the Web Intelligence project.

References

[1] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the 11th International Conference on Data Engineering, IEEE, 1995, pp. 3–14.
[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, Y. Yang, Topic detection and tracking pilot study: final report, in: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1998, pp. 194–218.
[3] K.K. Bun, M. Ishizuka, Emerging topic tracking system, in: Proceedings of the 1st International Conference on Web Intelligence, LNCS, vol. 2198, 2001, pp. 125–130.
[4] K.K. Bun, M. Ishizuka, Emerging topic tracking system in WWW, Knowledge-Based Systems 19 (3) (2006) 164–171.
[5] S. Chakrabarti, S. Sarawagi, B. Dom, Mining surprising patterns using temporal description length, in: Proceedings of the 24th International Conference on Very Large Databases, Morgan Kaufmann, 1998, pp. 606–617.
[6] K.Y. Chen, L. Luesukprasert, S.T. Chou, Hot topic extraction based on timeline analysis and multidimensional sentence modeling, IEEE Transactions on Knowledge and Data Engineering 19 (8) (2007) 1016–1025.
[7] H. Cherfi, A. Napoli, Y. Toussaint, Towards a text mining methodology using association rule extraction, Soft Computing 10 (5) (2006) 431–441.
[8] G. Costa, G. Manco, R. Ortale, A. Tagarelli, A tree-based approach to clustering xml documents by structure, in: Proceedings of the Eighth European Conference on Principles and Practice of Knowledge Discovery in Databases, LNCS, vol. 3202, 2004, pp. 137–148.
[9] T. Dalamagas, T. Cheng, K.J. Winkel, T.K. Sellis, Clustering xml documents using structural summaries, in: Proceedings of the Extended DataBases Technologies Workshops, 2004, pp. 547–556.
[10] G. Dong, J. Li, Efficient mining of emerging patterns: Discovering trends and differences, in: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, 1999, pp. 43–52.
[11] R. Feldman, Ido Dagan, Knowledge discovery from textual databases. in: Proceedings of the First International Conference on Knowledge Discovery from DataBases, 1995, pp. 112–117.
[12] F. Fourel, Modelling, Indexing and Retrieval of Structured documents, Ph.D. Thesis, University of Grenoble I, France, 1998.
[13] B. Gilad, J. Herring, The Art and Science of Business Intelligence Analysis, JAI Press, 1996.
[14] C. Halliman, Business Intelligence Using Smart Techniques: Environmental Scanning Using Text Mining and Competitor Analysis Using Scenarios and Manual Simulation, Information Uncover, 2001.
[15] F. Jacquenet, C. Largeron, Discovering unexpected information for technology watch, in: Proceedings of the Eighth European Conference on Principles and Practice of Knowledge Discovery in Databases, 2004, pp. 219–230.
[16] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[17] B. Lent, R. Agrawal, R. Srikant, Discovering trends in text databases, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1997, pp. 227–230.
[18] X. Li, W.B. Croft, Improving novelty detection for general topics using sentence level information patterns, in: Proceedings of the Fifth ACM Conference on Information and Knowledge Management, ACM Press, 2006, pp. 238–247.
[19] B. Liu, W. Hsu, Y. Ma, Pruning and summarizing the discovered associations, in: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, 1999, pp. 125–134.
[20] B. Liu, Y. Ma, P.S. Yu, Discovering unexpected information from your competitors' web sites, in: Proceedings of the Seventh international Conference on Knowledge Discovery and Data mining, 2001, pp 144–153.
[21] Bing Liu, Wynne Hsu, Lai-Fun Mun, Hing-Yan Lee, Finding interesting patterns using user expectations, IEEE Transactions on Knowledge and Data Engineering 11 (6) (1999) 817–832.
[22] P. Losiewicz, D.W. Oard, R. Kostoff, Textual data mining to support science and technology management, Journal of Intelligent Information Systems 15 (2000) 99–119.

[23] N. Matsumura, Y. Ohsawa, M. Ishizuka, Discovery of emerging topics between communities on WWW, in: Proceedings of the First International Conference on Web Intelligence, LNCS, vol. 2198, 2001, pp. 473–482.
[24] L.T. Moss, S. Atre, Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications, Addison-Wesley, 2003.
[25] Y. Ohsawa, N.E. Benson, M. Yachida, Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor, in: Proceedings of the Advances in Digital Libraries Conference, 1998, pp. 12–18.
[26] B. Padmanabhan, A. Tuzhilin, Unexpectedness as a measure of interestingness in knowledge discovery, Decision Support Systems 27 (3) (1999).
[27] B. Padmanabhan, A. Tuzhilin, On characterization and discovery of minimal unexpected patterns in rule discovery, IEEE Transactions on Knowledge and Data Engineering 18 (2) (2006) 202–216.
[28] Balaji Padmanabhan, Alexander Tuzhilin, A belief-driven method for discovering unexpected patterns, in: Knowledge Discovery and Data Mining, 1998, pp. 94–100.
[29] B. Piwowarski, Machine Learning for Processing Structured Information: Application to Information Retrieval, Ph.D. Thesis, University Paris VI, France, 2003.
[30] K. Rajaraman, A.H. Tan, Topic detection, tracking, and trend analysis using self-organizing neural networks, in: Proceedings of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNCS, vol. 2035, 2001, pp. 102–107.
[31] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.
[32] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (1) (2002) 1–47.
[33] A. Silberschatz, A. Tuzhilin, On subjective measures of interestingness in knowledge discovery, in: Proceedings of the First International Conference on Knowledge Discovery and Data Mining, 1995, pp. 275–281.
[34] A. Silberschatz, A. Tuzhilin, What makes patterns interesting in knowledge discovery systems, IEEE Transactions on Knowledge And Data Engineering 8 (1996) 970–974.
[35] I. Soboroff, D. Harman, Overview of the trec 2003 novelty track, in: NIST Special Publication: SP 500-255, The 12th Text Retrieval Conference, 2003, pp. 38–53.
[36] A. Soulet, B. Crémilleux, F. Rioult, Condensed representation of eps and patterns quantified by frequency measures, in: Lectures Notes in Computer Science, Knowledge Discovery in Inductive Databases, vol. 3377/2005, Springer Verlag, 2005, pp. 173–189.
[37] E. Suzuki, Undirected discovery of interesting exception rules, International Journal of Pattern Recognition and Artificial Intelligence 16 (8) (2002) 1065–1086.
[38] E. Suzuki, Data mining methods for discovering interesting exceptions from an unsupervised table, Journal of Universal Computer Science 12 (6) (2006) 627–653.
[39] E. Suzuki, J.M. Zytkow, Unified algorithm for undirected discovery of exception rules, International Journal of Intelligent Systems 20 (7) (2005) 673–691.
[40] J.A. Swets, Information retrieval systems, Science 141 (1963) 245–250.
[41] A. Termier, M.C. Rousset, M. Sebag, Treefinder: a first step towards xml data mining, in: Proceedings of the IEEE International Conference on Data Mining, 2002, pp. 450–457.
[42] A.M. Vercoustre, M. Fegas, S. Gul, Y. Lechevallier, A flexible structured-based representation for xml document mining, in: Proceedings of the Fourth International Workshop of the Initiative for the Evaluation of XML Retrieval, 2006, pp. 443–457.
[43] C. Wayne, Topic Detection and Tracking (TDT) Overview and Perspective, 1998.
[44] Guillaume Wisniewski, Francis Maes, Ludovic Denoyer, Patrick Gallinari, Probabilistic model for structured document mapping, in: Proceedings of the Fifth International Conference Machine Learning and Data Mining in Pattern Recognition, LNCS 4571, 2007, pp. 854–867.
[45] X. Zhang, G. Dong, K. Ramamohanarao, Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets, in: Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining, 2000, pp. 310–314.
[46] D. Zhu, A.L. Porter, Automated extraction and visualization of information for technological intelligence and forecasting, Technological Forecasting and Social Change 69 (2002) 495–506.
[47] D. Zhu, A.L. Porter, S. Cunningham, J. Carlisie, A. Nayak, A process for mining science and technology documents databases, illustrated for the case of "knowledge discovery and data mining", Ciencia da Informação 28 (1) (1999) 7–14.