Sequential Patterns for Maintaining Ontologies Over Time

Lisa Di Jorio, Sandra Bringay, Céline Fiot, Anne Laurent, Maguelonne Teisseire

Sequential Patterns for Maintaining Ontologies over Time

Lisa Di-Jorio, Sandra Bringay, Céline Fiot, Anne Laurent, and Maguelonne Teisseire

LIRMM – Université de Montpellier 2 – CNRS
161 rue Ada, 34392 Montpellier – FRANCE

Abstract. Ontologies are known as a quality and functional model, allowing meta data representation and reasoning. However, their maintenance plays a crucial role as ontologies may be misleading if they are not up to date. Actually, this work is done manually, and raises the problem of expert subjectivity. Therefore, some works have developed maintenance tools but none has allowed a precise identification of the relations that could link concepts. In this paper, we propose a new fully generic approach combining sequential patterns extraction and equivalence classes. Our method allows to identify terms from textual documents and to define labelized association rules from sequential patterns according to relevance and neighborhood measures. Moreover, this process proposes the placement of the found elements refined by the use of equivalence classes. Results of various experiments on real data highlight the relevance of our proposal.

1 Introduction

Ontologies offer a generic model for knowledge representation. These special structures are widely used for capturing knowledge of a particular domain of interest, and for easing data manipulation and exchange. As this knowledge is constantly evolving, ontologies must be updated, by adding, deleting or replacing knowledge. Often, new knowledge is reported on textual documents. Ontology updating implies selection of interesting terms from various documents in a first time, and then a placement of these terms, either as new concepts, or as a new relation labels. For a human expert, regarding the quantity of data to mine, this is a difficult, time consuming and tedious work. Moreover, it differs from one expert to another one depending on their point of view.

One way to overcome expert subjectivity is the use of automatic tools. Mostly based on statistical or syntactic analysis, these tools focus on finding and adding new concepts. However, as far as we know, none of them allows detection of one important specific knowledge: the links between concepts, and the terms labeling them. These elements are the specificity of ontologies as they model semantic knowledge.

Expanding an ontology can be done through a feedback loop combining web mining and formal semantic [1]: data mining extracts new terms to add, and the placement is done using some semantic logic rules. With these new elements, the process is repeated as many times as possible. Most of the time, the user is involved into the loop, either to manually place a new discovered element, or to label a relation, or to validate an adding... However, applied data mining techniques often result in a large quantity of elements, making such tools inefficient. Furthermore, no tool automatically fills the

entire process: that is, look for new concepts/relations and place them into the ontology at the right level of abstraction. We propose in this paper a fully automatic process to expand a given ontology, based on data mining techniques and on equivalence classes. Our contribution is twofold. On the one hand we propose a method to select new terms through sequential patterns mining and to categorize them as concepts or as relation labels. On the other hand we define an equivalence class based approach to allow the most precise term placement, and to process a large quantity of discovered data. Uncovered elements are grouped according to concepts and are added at the right place through labeled relations.

The rest of this paper is organized as follows. First, we give an overview of the problem statement, presenting current work (Section 2). Then, we detail our contribution, explaining each step of the ontology expanding process (Section 3). Finally, experiments described in Section 4 highlight the relevance of our proposal.

2 Related Work

2.1 Ontologies and Maintenance

It is a challenging issue to define precisely what is an ontology, since this term is used in many areas, from philosophy or linguistic to artificial intelligence. According to [2], an ontology can be viewed as "*an explicit specification of a conceptualization*". They allow both data exchange and human / machine readability. Often, ontologies describe objects of the real world as concepts, and formalize relations linking them either as hierarchical relations, or as semantical relations. Most of the time, these relations are labeled by a word or an expression. These relations stress the difference between ontologies and other structures giving a semantic description of concepts interaction.

Example 1 *Let us consider the environmental area. Living beings drink water.* ***Air*** *and* ***Water*** *are specific to the* ***Environment concept****. Figure 2 illustrates part of this ontology.*

All the approaches presented in this section follow two generic steps. First, documents are preprocessed, i.e. words are replaced by their lemma, that is the generic form of a word. For example, the word "*is*" will be replaced by "*to be*", allowing to consider words regardless of their declension. At this stage, lemmatized words are called *terms*. Among them are new potential candidates for the maintenance, such as new concepts or new relation labels. Then, enriching ontologies consist of selecting these terms as a first step. Lately, we have noticed two major tendencies for terms selection: statistical based methods, and syntactic based methods. We describe these methods in the following section.

2.2 Statistical Based Methods

Statistical methods consist of counting the number of occurrences of a given term in the corpus. The more frequent a term is (according to a measure), the more it will be considered as a candidate. In a statistical context, many measures have been proposed,

mostly based on term distribution in the corpus. The easiest way is to count the number of apparitions of each term among the entire corpus [3,4,5]. More complex measures have also been defined. For instance, [6,7] successively test mutual information, Tf.Idf, T-test or statistic distribution laws, resulting in a good terms selection. However, these measures never take the domain into account, nor terms appearing alone. To overcome the domain representation problem, [8] proposes a new definition of mutual information. Whereas experiments show that representative terms are extracted, some relevant terms remain uncovered. Moreover, as these methods only select terms without any external information, it is impossible to distinguish concept terms from relation labels. So, all the extracted terms are considered as potential new concepts.

Then, discovered elements shall be placed into the ontology. Because of the quantity of extracted terms, this step cannot be manually done. To avoid a manual placement of a huge quantity of terms, [4] proposes to use term co-occurrence implying one or more existing concepts of the ontology. This involves knowing where a new concept should be added. However, they only bring the closest new concepts, and never add them at a precise level with a precise relation (ie, no relation label found). We thus argue that this method will not generate semantic knowledge.

Other approaches use data mining techniques, such as classification (grouping items in an already known class) or clustering (grouping items to, but in a class found during the process). [9] and [7] bring new terms close to existing ones in the ontology thanks to a classification method. Similarly, [3] and [5] group extracted terms using a clustering method. Each cluster shows a possible relation between grouped terms. However, it is not possible to define what kind of relation it is, or to label it.

Therefore, statistical methods allow new term selection and correlation detection with existing terms, but do not place new terms directly into the ontology. Moreover, statistical methods ignore the linguistic structure of the analyzed sentences, which can give information about relations linking concerned concepts. Using syntactic methods can overcome the problem.

2.3 Syntactic Based Methods

Methods based on syntax consist of a grammatical sentence analysis, most of the time preceded by a part of speech tagging (POS) process. Syntactic methods suppose that grammatical dependencies reflect semantic dependencies [10,11]. Thus, two syntagms are considered as concepts, and the gramatical relation linking them as a semantic relation. These considerations allow adding extracted terms as new concepts at the right place, linking them to the right existing concept.

However, these approaches suffer from the same problems as statistical ones: a huge quantity of related terms are extracted, as there exists more than one grammatical dependency into a sentence. Therefore, data mining techniques are also applied in some approaches. [10,12,1] extract association rules from the syntactic dependencies. Association rules have been proposed by [13] and allow strong correlation detection like "*when the term A is employed, the term B is employed too*". These correlations highlight frequent grammatical dependencies and thus are a good way to prune many insignificant dependancies. Some syntactic based work defines regular expressions in order to

find terms corresponding to one and only one kind of relation. For instance [14] looks for hyponyms form large text corpora.

Even though syntactic based approaches automatically put new terms into the existing ontology, they do not label new relations. Once again, the extraction of a common semantic structure from a large text corpus is missed. Relation labeling has to be done by the user. Moreover, data mining techniques are always used as a second step of the generic enriching process whereas these techniques can be directly used to extract new terms. Indeed, efficient techniques have been developed, allowing among other options to restrict term selections by semantic or time constraints [15,16].

However no work uses sequential patterns in order to detect or add new elements, eventhough these structures have been proved to be efficient [17] for large text databases mining. Sequential patterns lead to a finest text analysis as they store word apparition order and frequently co-occurring words. Moreover, sequential patterns keep a track of the document structure without requiring any external knowledge. We present in this paper a fully automatic approach based on sequential patterns extraction. Indeed, we propose to use them to discover new concepts and relations labels which link them to other concepts.

3 SPONTEX : Sequential Patterns For Ontology Expansion

3.1 Overview

In this section, we introduce SPONTEX, a new algorithm for expanding an ontology, by mean of sequential patterns. Our general process, described by Figure 1, starts from these patterns. However, taken as an ordered list of words, sequential patterns need to be transformed and refined to raise a new kind of knowledge: concepts and semantic relations. We called this **labeled association rules**.

Labeled association rules are generated through two steps: (1) words that may share semantic with an existing concepts are first selected and organized around this concept. These candidates constitute a **neighbor set**. In order to refine the search space, we define a new measure, called **closeness**. (2) Among these words are concept terms and relation labels. The generation of labeled association rules disambiguates the role of terms. At the end of the second step, the process is close to be finished: we know new concepts terms, the existing concept they can be attached to, and the label of the relation linking them. However, we have not organized our new terms around concepts so far. This is the aim of the third step. We propose to use **equivalence classes**. Every details and formalization are provided at section 3.6. Finally, we add mined elements to the existing ontology during step 4. In order to respect our formal ontology definition (following section), we check that we do not add a same word as a term concept and a label relation.

3.2 Sequential Patterns

Sequential patterns were originally introduced by [18]. They model an ordered list of itemsets usually associated to a given time period.

Let $\mathcal{O}$ be an object set and $\mathcal{I}$ be a set of **items** stored in a database $\mathbf{DB}$. Each record E is a triplet (*id-object*, *id-date*, *itemset*) as illustrated by table 1 .An **itemset** is a non

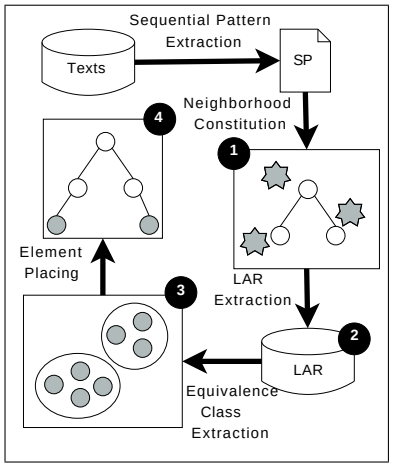

Fig. 1. General Process

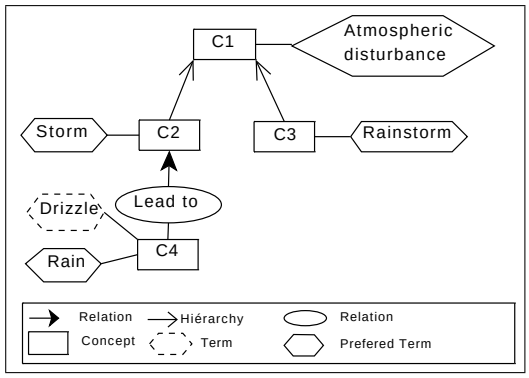

Fig. 2. Ontology Example

empty set of items from $\mathcal{I}$ represented by $(i_1, i_2, ..., i_n)$. A **sequence** S is defined as an ordered and non empty list of itemset $<s_1 s_2 ... s_n>$. A n-sequence is a sequence of length n (containing exactly n items).
Thus, $\mathbf{DB}$ associates a list of items to the object *id-obj* at the date *id-date* and can be represented in a *object-sequence* manner, as shown in table 2. In this table, the sequence S=<(a)(b c)> associated to object 1 means that the item a has been recorded, then b and c together. S is a 3-sequence.

id-object	*id-date*	*itemset*
1	1	a
1	3	b c
2	2	d e
2	3	d

Table 1. A transaction database example

id-object	*sequence*
1	<(a)(b c)>
2	<(d e)(d)>

Table 2. A sequence database example

Given a sequence $S' =< s'_1 s'_2 ... s'_n >$ and a sequence $S =< s_1 s_2 ... s_m >$, S' is included into S if and only if there exist integers $a_1 < a_2 < ... < a_n$ such that $s'_1 \subseteq s_{a_1}, s'_2 \subseteq s_{a_2}$,... $s'_n \subseteq s_{a_n}$. S' is then called a *subsequence* of S and S is a *supersequence* of S'.
For example, S' = <(a)(b)> is a subsequence of S because $(a) \subseteq (a)$ and $(b) \subseteq (b\ c)$. On the other hand, <(b)(c)> isn't a subsequence of <(b c)>.

An object *o* supports a sequence S if and only if S is included into the data sequence of this object. The *sequence support* (also called *frequency*) $Freq()$ is defined as the number of objects of the database $\mathbf{DB}$ supporting S. Given a threshold $minSupp$, a sequence S is frequent if $Freq(S) \geq minSupp$.
Extracting sequential patterns from a database like $\mathbf{DB}$ means finding all the maximal

sequences (not included in others) which support $Freq()$ is at least equal to $minSup$. Each maximal sequence is a **sequential pattern**.

During this last decade, efficient algorithms for sequential patterns extraction have been proposed [18,19,20,21].

As sequential patterns were initially introduced to deal with market data, we need to transpose the context in order to apply extraction from a text documents database. Here, a date is represented by one or more sentences, and an item corresponds to a lemmatized word. Considering that a sentence is a unit of time, extracting the sequence <(living)(environment lake)(flood)> means that among all the documents, the word "living" frequently occurs in a sentence, followed by the co-occurrence of the words "environment" and "lake".

3.3 Formal Definition of an Ontology

We need to formally state what ontologies are, in order to properly use them during the adding process. In agreement with a common point of view, an ontology is constituted by concepts organized into a hierarchy. A concept is an object with associated terms describing their semantic. Computers infer knowledge starting from these concepts, and human understand their sense when reading the words associated to these ones. Moreover, our aim is to represent possible interactions between these concepts. This is done thanks to the definition of semantic relations. We use the following definition, initially proposed by [22]:

Definition 1. *Let $\mathcal{C}$ be a concept set, $\mathcal{T}$ a term set, $\mathcal{R}_c$ a relation (between concepts) set, $\mathcal{R}_t$ a relation (between terms) set and $\mathcal{L}$ a relation's label set (semantic name of a relation). An ontology* **O** *is defined by:*

$$\mathbf{O} = \{\mathcal{C}, \mathcal{T}, \mathcal{R}_c, \mathcal{R}_t, \mathcal{L}, <_c, f_{tc}, f_{rc}\}$$

with :
- $<_c$: $\mathcal{C} \times \mathcal{C}$ is a partial order relation on $\mathcal{C}$ defining concept hierarchy,
$<_c (c_1, c_2)$ means that c_1 is more generic than c_2
- f_{tc} : $\mathcal{C} \rightarrow \mathcal{T}$ is the association function between a prefered term and a concept
- f_{rc} : $\mathcal{R}_c \rightarrow \mathcal{C} \times \mathcal{C}$ is an associative function between concepts

To avoid any confusion, we consider that an ontology concept is designed by one of its associated *term*. This term is said to be the **preferred term** of the concept. Talking about semantic relation amounts to use its **relation label**.

Example 2 *Figure 2 shows a part of an ontology about atmospheric disturbances. Rectangles represent concepts, diamonds represent terms and ellipsis show relations.*

The concept set $\mathcal{C}$ group {C1, C2, C3, C4}, the term set is $\mathcal{T}$={Atmospheric disturbance, Storm, Rainstorm, Rain, Drizzle}, and the relation set $\mathcal{R}_c$ is formed by only one relation, which label is Lead to. "Atmospheric disturbance" is the preferred term of C_1 concept: when we talk about C_1, we talk about all the atmospheric disturbances phenomema. $f_{rc}(Lead\ to) = (C_2, C_4)$ is a relation meaning that storm leads to inundation.
Concepts hierarchy $<_c$ is indicated by simple arrows and means that, for example, C1 is more specific than C2.

By keeping track of frequent words co-occurring and their order, sequential patterns are an efficient tool for data extraction. However, taken as an ordered list of words, they do not allow to directly infer semantic knowledge. We thus raise the following questions: *how can we use them for an ontology updating process? How can we distinguish a word associated to a concept from a word associated to a relation?*
Among extracted sequential patterns are some already known terms, as they are already referenced by the ontology. We propose to refine the search space by only considering neighbors of these concepts. A neighbor of a given concept c_o is a term that can be accessed by using one link (hierarchical or semantic) from c_o:

Definition 2. *Let c_o be a concept, the neighbor set $\mathcal{V}_{c_o}$ of c_o is defined as the concepts c and relations r set:*
$\forall c \in \mathcal{V}_{c_o}, \exists r \subseteq \mathcal{R} \mid f_{rc}(r) = (c_o, c) \ \vee f_{rc}(r) = (c, c_o) \vee <_c (c_o, c) \vee <_c (c, c_o)$

This notion allows the association of new terms extracted by sequential patterns with existing concepts. In the rest of this paper, a term candidate appearing in a sequential patterns is called an **item**.

Example 3 *The neighbor set associated to the "Storm" concept is $\mathcal{V}_{storm}$ ={"Rain", "Atmospheric disturbance", "drag"}, because f_{rc}(Lead to)=("Storm", "Rain"), and $<_c$("Atmospheric disturbance", "Storm").*
The term "Rain" represents one of the ontology's concept of the picture 2 and "Lead to" is a label of a relation of the ontology, whereas "Cause" or "Inundation" are items of the sequential pattern <(Rain)(Cause Inundation)>.

3.4 Constructing The Neighbor Set

Each sequential pattern containing a concept term may participate to the neighbor set construction. Obviously, we cannot consider that every item of such a sequence is a neighbor, as the set cardinality will quickly explode. Therefore we propose to use a measure based on sequence support in order to add an item as a neighbor of a known term. This measure, called **closeness**, indicates the neighborhood degree between a term and is defined as follow :

Definition 3. *Let S be a sequential pattern, i and c_o two different items from this sequence such as $c_o \in \mathcal{T}$. The **Closeness measure** of the item i as a term or a relation label of the c_o neighbor set is defined by :*

$$Closeness(c_o, i) = \max \begin{pmatrix} \max(\frac{Freq([(i\ c_o)])}{Freq([(c_o)])}, \frac{Freq([(i\ c_o)])}{Freq([(i)])}), \\ \max(\frac{Freq([(i)(c_o)])}{Freq([(i)])}, \frac{Freq([(i)(c_o)])}{Freq([(c_o)])}), \\ \max(\frac{Freq([(c_o)(i)])}{Freq([(c_o)])}, \frac{Freq([(c_o)(i)])}{Freq([(i)])}) \end{pmatrix}$$

Here, word order does not infer on the neighbor set. Three configurations are possible : (1) the word frequently appears in the same sentence than the term, (2) the word frequently appears before the term and (3) the word frequently appears after the term. By keeping the best apparition proportion of a configuration order relative to only one item, we consider the influence of each item on another. Moreover, as we are only interested in the best configuration, we keep the maximum proportion rate among the tree possible configurations. Example 4 illustrates this idea:

Example 4 *Table 3 shows extracted sequences from a document set.*

Sequence	$Freq$
[**(rain inundation cause)**]	0.4
[**(rain inundation)(cause)**]	0.3
[**(rain)(inundation cause)**]	0.3
[**(rain)(inundation)(cause)**]	0.2
[**(rain cause)(inundation)**]	0.5
[**(rain)(cause)(inundation)**]	0.3
[**(inundation)(rain)(cause)**]	0.5
[**(cause)(rain)(inundation)**]	0.3
[**(inundation)(cause)(rain)**]	0.3
[**(inundation cause)(rain)**]	0.3

Sequence	$Freq$
[**(inundation)(cause rain)**]	0.2
[(rain inundation)]	0.5
[(rain)(inundation)]	0.5
[(inundation)(rain)]	0.6
[(rain cause)]	0.5
[(rain)(cause)]	0.6
[(cause)(rain)]	0.5
[(rain)]	1
[(inundation)]	0.7
[(cause)]	0.7

Table 3. Extracted sequences

The item "rain" already belongs to the ontology shown on the Figure 2 as a concept term. Let us compute "rain" and "inundation" closeness rate.

$$Closeness(\text{"}rain\text{"},\text{"}inundation\text{"}) = \\ max\left(\begin{array}{l} max(\ \frac{Freq([(inundationrain)])}{Freq([(rain)])},\ \frac{Freq([(inundationrain)])}{Freq([(inundation)])}) \\ max(\ \frac{Freq([(inundation)(rain)])}{Freq([(rain)])},\ \frac{Freq([(inundation)(rain)])}{Freq([(inundation)])}) \\ max(\ \frac{Freq([(rain)(inundation)])}{Freq([(rain)])},\ \frac{Freq([(rain)(inundation)])}{Freq([(inundation)])}) \end{array}\right) \quad (1)$$

$= max(max(\frac{0.5}{1}, \frac{0.5}{0.7}), max(\frac{0.6}{1}, \frac{0.6}{0.7}), max(\frac{0.5}{1}, \frac{0.5}{0.7}))$
$= max(0.71, 0.86, 0.71) = 0.86$

The building of the neighbor set is done through the algorithm 1, called *Neighbor-Generation*. Given a set of sequential patterns, an already known concepts set and a minimal closeness threshold fixed by the user, the algorithm returns the set $\mathcal{V}$ of all the neighbor sets V_{C_i} of the ontology concepts. More formally:
$\forall V_{C_i} \in V,\ V_{C_i} = \{(item\ j_1, closeness(C_i, j_1)), ..., \\ (item\ j_n, closeness(C_i, j_n))\}$
This allows to store the closeness rate between a concept C_i and an item j. Example 5 shows such a set $\mathcal{V}$.

Example 5 *Bold sequences from table 3 are sequential patterns. Algorithm 1 will successively test the following closeness threshold :*

Algorithm 1: NeighborGeneration

Data: Sequential patterns set $\mathcal{S}$,
The pattern prefixed tree $\mathbf{PSP}$,
The ontology $\mathbf{O}$
$minProx$ the minimal closeness
threshold fixed by the user
Result: $\mathcal{V}$, the set of all closeness relations

$\mathcal{V} \leftarrow \varnothing$
foreach $s \in S$ **do**
  foreach $c_o \in \mathcal{C}$ *such as* $c_o \in s$ **do**
    foreach $i \in s$ *such as* $i \neq c_o$ **do**
      if $Prox(c_o, i) \geq minProx$ **then**
        $\mathcal{V}_{c_o} \longleftarrow i$
      end
    end
    $\mathcal{V} \longleftarrow \mathcal{V}_{c_o}$
  end
end
return $\mathcal{V}$

- $Closeness(Rain, Inundation) = 0.86$
- $Closeness(Rain, Cause) = 0.71$

With a minimal closeness threshold of 0.5, algorithm 1 will return the set $\mathcal{V} = \{\mathcal{V}_{C_4}\}$, *with* $\mathcal{V}_{C_4} = \{(Inundation, 0.86),$
$(Cause, 0.71)\}$.

Note that building this neighbor set is only selecting new items to add to the ontology. However, we ignore if these items are concept terms or relation labels. For example, we do not know if the words "Inundations" or "Cause" selected in example 5 are new concept terms, or new relation labels. This will be determined through the extraction of labelled association rules, as it is explained in the following section.

3.5 Extracting Labeled Relations

We notice that when a document address two concepts linked by a relation, *the relation label is frequently employed in the same sentence than one of the two concepts*. Therefore, we need to determine which item is frequently used in the same sentence than a known concept term. Sequential patterns provide this information, so we propose to exploit it with the following relationship measure:

Definition 4. *Let* c_o *be a term such that* $\mathcal{V}_{c_o} \in \mathcal{V}$, i *and* j *two items from* $\mathcal{V}_{c_o}$ *such as* $i \neq j$. *Then, the* ***relationship level*** *of the item* i *as a relation label between* c_o *and* j *and is defined by :*

$$RL_i(c_o, j) = max \left(\begin{array}{l} \frac{Freq([(i\ j\ c_o)])}{Freq([(j\ c_o)])}, \frac{Freq([(c_o)(i\ j)])}{Freq([(c_o)(j)])}, \\ \\ \frac{Freq([(c_o\ i)(j)])}{Freq([(c_o)(j)])}, \frac{Freq([(j)(i\ c_o)])}{Freq([(j)(c_o)])}, \\ \\ \frac{Freq([(j\ i)(c_o)])}{Freq([(j)(c_o)])} \end{array} \right)$$

The relationship level represents the proportion of documents which employed terms c_o and i in the same sentence, or terms c_o and j in the same sentence. This proportion could be considered as a kind of confidence, as it represents the maximal probability that i co-occurs with c_o knowing j or that i co-occurs with j knowing c_o.

The relationship level is not bounded to confidence rating. As a relation between two concepts is frequently co-occuring with one of these concepts, RL permits to distinguish a word role as a concept or as a relation. If $RL_i(c_o, j) > RL_j(c_o, i)$, this means that $(i\text{-}c_o)$ and $(i\text{-}j)$ co-occurs more frequently than $(j\text{-}c_o)$ and $(j\text{-}i)$. In that case, i is more likely to be a relationship according to our observations. This is why an item role is determined by the highter confidence, i.e, the higther RL.

Example 6 *From the pattern set shown in Figure 3, two relationship levels can be computed:* $RL_{cause}(rain, inundation)$ *and* $RL_{inundation}(rain, cause)$.
$RL_{cause}(rain, inundation)$ *represents the relationship level of "cause" as being a relation linking the concepts "rain" and "inundation"; and* $RL_{inundation}(rain, cause)$ *represents the relationship level of "inundation" as being a relation linking "rain" and "cause" concepts.*
Here, we only detail the $RL_{cause}(rain, inundation)$ *calculation:*

$$RL_{cause}(rain, inundation)$$
$$= max \left(\begin{array}{l} \frac{Freq([(cause\ inundation\ rain)])}{Freq([(inundation\ rain)])}, \frac{Freq([(rain\ cause)(inundation)])}{Freq([(rain)(inundation)])}, \\ \frac{Freq([(rain)(cause\ inundation)])}{Freq([(rain)(inundation)}, \frac{Freq([(inundation\ cause)(rain)])}{Freq([(inundation)(rain)}, \\ \frac{Freq([(inundation)(cause\ rain)])}{Freq([(inundation)(rain)} \end{array} \right)$$
$$= max(max(\tfrac{0.4}{0.5}, \tfrac{0.4}{0.5}), \tfrac{0.5}{0.5}, \tfrac{0.3}{0.5}, \tfrac{0.3}{0.5}, \tfrac{0.2}{0.5})$$
$$= max(0.8, 1, 0.6, 0.5, 0.33) = 1$$

In the same fashion, we find that $RL_{inundation}(rain, cause) = 0.8$.

As $RL_{inundation}(rain, cause) < RL_{cause}(rain, inundation)$*, we can consider that "inundation" is a new concept, and "cause" is a relation linking it to the existing concept "rain".*

Summarizing, at this stage, we have started from sequential patterns to select a set of potential new knowledge that can be linked to existing concepts. By building the neighbor set, we keept only interesting words. Thanks to the relationship level, we have decided if these new elements will be considered as new concept terms, or as new relation labels. We can now formelize a new kind of association rules integrating semantic knowledge, called **labeled association rules**:

Definition 5. *A* ***labeled association rule*** *or* LAR*, denoted by* $i \stackrel{r}{\Longrightarrow} j$*, define an item* j *implication by an item* i *according to the relation* r.

Existence of such a rule between an item and a concept from the ontology indicates the existence into the ontology of a relation linking this concept to this item.

Definition 6. *The left part of a labeled association rule is the **acting** concept of the relation, and the right part is the **receiving** concept of the labeled relation.*

A labeled association rule characterizes a relationship level as well as the relation direction. So, for each association of three items i, j and k with k corresponding to a concept c_o of the ontology, we can determinate by the relationship level calculation if one of the others items i or j define a relation between c_o and the third item.

The relation direction has been calculated during the relationship level determination. We define the **implication rate** of a labeled association rule. It represent the document proportion from the textual base for which having items c_o and j imply having item i.:

Definition 7. *Let i, j, c_o be a triplet such that i is the label of a labeled association rule between j and c_o. The **implication rate** of a labeled association rule $c_o \overset{i}{\Longrightarrow} j$ is given by:*

$$IR(c_o \overset{i}{\Longrightarrow} j) = max\left(\frac{Freq([(c_o)(i\ j)])}{Freq([(c_o)(j)])}, \frac{Freq([(c_o\ i)(j)]}{Freq([(c_o)(j)])} \right)$$

A high implication rate confirms that the relation i is a link between c_o and j. Therefore, from a triplet formed by a candidate concept j, an item i and a concept c_o, we compute the implication rates of the rules ($j \overset{i}{\Longrightarrow} c_o$) and ($c_o \overset{i}{\Longrightarrow} j$). The rule having the best implication rate is kept while the other one is left.

In order to optimize the iteration number on sequential patterns and subsequences and consequently to reduce the execution time, we conceived Algorithm 2, named *LARGeneration*. This one computes through a single iteration the relationship level and the direction of the labeled association rules together.

Algorithm *LARGeneration* determines, from ontology concept neighborhood, the items which label relations or which are new terms.

This algorithm takes as input a prefixed tree called **PSP**. Proposed by [19], the prefixed tree is an efficient way to store sequential patterns. Each node is associated with an item. A PSP tree contains two kinds of edges: dashed edge representing the notion "*and after*" and plain edge representing the notion "*in the same time*". Leaves of a PSP are sequential patterns that can be read from the root to the leaves. This structure is output from our sequential pattern mining algorithm, and we choose to exploit it directly during the LAR mining process.

Given a set of all the neighbors and the PSP tree obtained from the sequential patterns extraction process, Algorithm 2 generate all possible labeled association rules. We iteratively compute the implication level $RL_j(c_o, i)$ and $RL_i(c_o, j)$ by combinating each item couples i, j of the neighbor set $\mathcal{V}_{c_o}$ and a concept c_o (lines 1-5). Once the item role is determined, we apply the implication rate calculation in order to fix the new LAR sense (line 6) and finally add it to the LAR set (line 7).

Algorithm 2: LARGeneration

Data: The neighborhood set $\mathcal{V}$,
The pattern prefixed tree $\mathbf{PSP}$
Result: The labeled association rules set RAL

$RAL \longleftarrow \varnothing$
foreach $\mathcal{V}_{c_o} \in \mathcal{V}$ **do**
  foreach $j \in \mathcal{V}_{c_o}$ **do**
    foreach $k \in \mathcal{V}_{c_o}$ *such as* $k > j$ **do**
      $ral =$Max$(RL_j(c_o, k), RL_k(c_o, j))$
      $ImplicationRate(ral)$
      $RAL \longleftarrow ral$
    end
  end
  return RAL
end

3.6 Grouping LAR

At this stage, we have obtained a large quantity of new concepts and labels of relations which link them to existing concepts. It is now possible to add these new elements to the ontology. However, this means creating a concept for each selected term, and thus associating only one term by concept. We would overload the existing ontology with too many new concepts, missing benefit from concept philosophy (a concept allows for grouping equivalent words) and biasing user navigation during the validation step. Therefore, new terms need to be effectively grouped around common concepts.

Terms linked to a same concept through the same relation label share a common semantic. It is thus possible to exploit this property in order to group them by the use of equivalence classes. That way, homograph terms will be distinguished.

In the Set Theory, an equivalence class is defined on a set and through the definition of an equivalence relation. In our context, we consider the labeled association rules set E_{RAL}, on which we define two binary relations.

Definition 8. *Let $\mathcal{R}$ (resp. $\mathcal{S}$) be a binary relation on the labeled association rules set $\mathcal{E}$ such as:*

$$\mathcal{R} = \{(a, b) \mid a, b \in \mathcal{E} \wedge a.rec = b.rec \wedge a.label = b.label\}$$
$$\mathcal{S} = \{(a, b) \mid a, b \in \mathcal{E} \wedge a.act = b.act \wedge a.label = b.label\}$$

where RAL.act *is the acting concept of the rule* RAL, RAL.$label$ *is the rule label and* RAL.rec *is the rule recipient.*

The relation $\mathcal{R}$ (resp. $\mathcal{S}$) allows to select rules having the same label and the same recipient concept (resp. acting concept).

Proposition 1. *Any relation $\mathcal{R}$ (resp. $\mathcal{S}$) as defined in definition 8 is an equivalence relation, as $\mathcal{R}$ (resp. $\mathcal{S}$) is reflexive, symmetric and transitive.*

Proof. The proof is omitted as it is easy to see that all properties hold.

These equivalence relations allow building equivalence classes from labeled association rules. These classes are then added as a concept into the existing ontology.

Example 7 *Given an extracted LAR set* $\mathcal{RL} = \{Rain \overset{cause}{\Longrightarrow} Inundation, Rain \overset{cause}{\Longrightarrow} Landslide, Rain \overset{cause}{\Longrightarrow} Submersion\}$*, then we can build an equivalence class based on the label "cause" and on the concept which prefered term is "Rain":*

$$[\overset{cause}{\Longrightarrow}]_{RAIN} = \mathcal{RL}$$

This allows us to group terms $Inundation$*,* $Landslide$*, and* $Submersion$ *under the same concept.*

3.7 Expanding the Ontology

Adding new elements is the final step of our method, before validation by a human expert. We create new concepts around which are associated equivalent terms. We add then our new concepts linking them to existing ones by the mean of labeled relations.

Some classes give a different role to a same item, either as a term concept, or as a relation label. As in works using syntactic based methods, we consider that an item can only have one role into the same ontology. So, before adding a new equivalence class $\mathbf{ce}$, the algorithm checks if the label relation linking $\mathbf{ce}$ to the ontology is not already considered as a concept term, and rejects it otherwise.

Example 8 *Figure 3 shows the evolution of the ontology presented in figure 2. We notice that a new concept which prefered term is "Inundation" have been added, linked to the existing concept "Rain" using the relation label "Cause".*

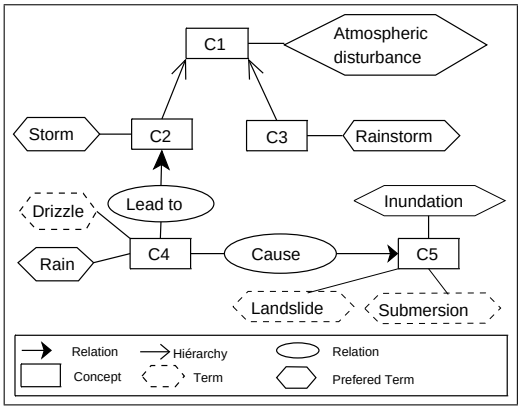

Fig. 3. Expanded Ontology

4 Experiments

4.1 Data

Our method has been tested on the EMWIS ontology[1], Euro-Mediterranean System of Water Domain Information. EMWIS is an European project concentrating his efforts on developing an ontology about water domain. The major goal is to improve communication between all the water area protagonists. This ontology is formed by 1006 concepts organized around three hierarchy levels and 29 non-labeled relations.

Concepts have been grouped around 25 themes in order to ease navigation. We decided to use them during our experiments. We built for each concept term a query which have been ran on a search engine. We downloaded the first 20 retrieved documents, obtaining thus a thematic corpus.

Experiments presented here have been realized on the "*Water Needs*" thematic, composed by 136 concepts. This theme was chosen for its ontology representativity: it was the one grouping the highest number of concepts. Then, the textual corpus built from the Web have 2720 documents.

4.2 Preprocess and Sequential Patterns Extraction

The relevance of the extracted patterns depends on document preprocessing, which is done according to three steps: (1) content extraction, (2) lemmatization, and (3) items selection.

Content extraction erases noise in documents (advertising, images, hypertext links...) and only keeps the main page text. In our experiments, lemmatization have been realized with TreeTagger [23], which is able to process french and english texts. After this step, all words are in their generic form: they are now considered as *terms*.

In order to keep only relevant words, we used tf.idf measure [24], allowing us to evaluate how important a word is to a document in a collection or corpus. At the end of this process, we obtain the most important terms according to tf.idf, and we keep terms already present into the ontology.

Sequential patterns have been extracted using a JAVA implementation of the algorithm VPSP [21], which combines prefixed tree projection of PSP [19] and memory projection of SPADE [20]. VPSP is a generate-prune algorithm which uses $(k\text{-}1)$-sequences to generate k-sequences, and after a frequency computation prunes sequences under minimal support fixed by the user. We adapted VPSP to keep in memory sequences of length 1, 2 and 3, necessary for the RL measure computation. We optimized this step in order to minimize required memory space, gaining the time which would have been used to compute again these information. For our dataset about *Water Needs*, we efficiently generated and stored about 14,000,000 of 3-sequences.

4.3 Results

Results obtained with various closeness clue highlight our proposal relevance. Indeed, we noticed that our method allows detection of a high number of new concepts, and proposes a correct placement of them into the existing ontology.

[1] http://www.semide.net/portal_thesaurus

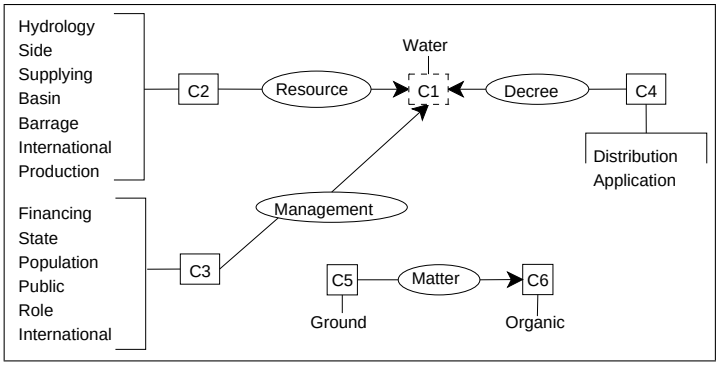

Fig. 4. Expanding Ontology Results

We studied closeness clue value impact on the enrichment process, as it is a determining factor concerning the items selection. We noticed that the lower the closeness clue value, the larger the neighborhood set cardinality is. Consequently the higher neighbors we obtain, the higher is the combination possibilities to generate labeled association rules. This is why we decided to use the relation level to prune rules, whereas closeness clue is very low.

We noticed that closeness clue has a strong influence on the number of LAR generated: when it is low we obtain more rules with a 100% confidence. Actually, if occurrences of the triplet (concept relation item) are lower, then these element are more often found together, which explain these observations.

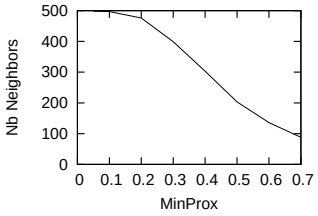

Fig. 5. Neighbor/minP

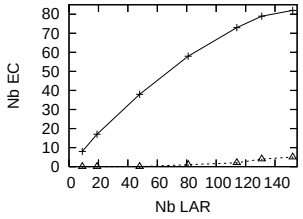

Fig. 6. Eq Class/LAR

We tested results obtained without using equivalence classes: this led to a new concept creation for each added terms, and mostly using a common relation label. Equivalence classes empirically prove to significantly improve results. First, we noticed that there is no loss of information: each selected term is placed into the existing ontology. Equivalence classes group terms sharing a common lexical field. As an example, we can see on the figure that terms "*Basin*", "*Hydrology*" or "*River*" are grouped under the same concept and linked as "*Resource*" to the "*Water*" concept. Secondly, adding elements driven by equivalence classes eases the navigation of the expanded ontology.

Furthermore, this method detects homographic terms. Homographics are words sharing the same spelling, but having different meanings. For example "*shift*" can be used as "*a change*" or as "*a period at work*". In our experiments, we found the term "*Source*" was associated to the financial area, whereas it was already associated into the ontology to the "*Water*" concept, meaning "*a river origin*". In such a case, equivalence classes lead to the placement of two distincts terms "Source", each one associated to a different concept.

Figure 6 shows the number of equivalence classes found according to the number of LAR. Curve with cross points represents classes based on receiving concept, whereas curve with triangular points represents acting based ones. Notice that our real dataset sample led to a high number of received based classes. The number of equivalence classes grows as the number of LAR grows, but the number of new added concepts is significantly reduded: as an example, we can see that 140 LAR led to the creation of half less new concepts (i.e. 80 concepts).

These results, i.e. high number of labeled association, let us choose a low and therefore less selective minimal closeness clue. To obtain great quality results, we preferred to put a restriction on the implication level of our labeled association rules, keeping the more interesting ones.

Figure 4 presents some obtained results, realized with a minimal closeness clue fixed to 40%, and keeping labeled association rules having an implication level of at least 50%. Indeed, the chosen thematic - Water Needs - is large and then includes distant concepts. We fixed a large closeness value in order to catch less evident relations. Once labeled association rules have been generated, we noticed the high number of rules having an implication level superior to 80%, confirming that a threshold of 50% was sufficient for the extraction of most of the necessary rules to obtain relevant process result.

Figure 4 displays existing concepts, presented with dashed lines, and added element (concepts, relations and labels) by plain lines. A 40% minimal closeness value allowed to retrieve 303 of concepts/items pairs (or neighbor couples) potentially candidates.

From these pairs, 498 labeled association rules have been generated. We retained those having an implication level of at least 50% to build equivalence classes. The obtained results are coherent, as most of them have been correctly placed into the ontology. Elements added are rather general terms, which is a normal phenomenon considering that the built corpus cover a large part of the ontology. We can observe the high number of concepts added around the water concept, which seems normal as we have an ontology constructed for the water domain.

Finally, EMWIS ontology has 29 non-labeled relations. We observe that our method allowed to name one of them: the one linking "*Hydrological basin*" and "*Watercourse*". Whereas it seems to be limited, this result goes ahead compared to existing approaches. Moreover, this weak number of existing labelization can be explained by two reasons: firstly, the specific character of these relations, as they indeed concern only 0.02% of the total ontology and secondly, these relations link subconcepts at a very specific level of the hierarchy. Last, we noticed that these relations mainly concern other themes, such as politic or agriculture.

5 Conclusion

Ontologies are powerful structures, allowing knowledge representation and sharing. However, their relevance directly depends on their maintenance. Ontology updating mainly consists of adding concepts and/or relations and is mostly manually done.

Some works propose tools for automatically updating ontologies. However, none of them have obtained complete results, because of their lack of automatism, i.e. the user is involved in the enrichment loop, or only one type of knowledge (mostly concepts) is considered; none allows automatic relation label detection.

In this paper, we proposed the use of sequential patterns, allowing term research and element extraction (concept or relation). Our proposal is fully automatic, as it places the user at the end of the process in order to validate obtained results. Our main contribution is the automatic discovery of relation label and new concepts, and a finest terms analysis by the mean of equivalence classes. We have described in details all the process steps. From sequential patterns, we determined items which can be used to expand the ontology through the building of neighbor sets. From these sets we construct a new kind of knowledge named Labeled Association Rules, allowing by their implication level to distinguish between concept and label. Finally, an equivalent classes based approach refined the results and drove to a coherent and readable adding to the ontology.

We noticed during our experiments the real quality of added elements: not only chosen elements are relevant but element grouping and placement is totally coherent too. This proves that sequential patterns allow to highlight semantically correlated terms, and that an equivalence based method correctly groups elements sharing a common lexical field. This results in the detection of homograph terms, and improves the ontology readability. Moreover, this opens some interesting perspectives. First of all, refined method for sequential patterns extraction have been proposed [13,15], such as the use of a minimal or maximum windowsize in order to restrict itemset selection to close or distant one. This work can be adapted in our context in order to select more closed patterns. Moreover, we could use an ontology driven extraction method to integrate the concept hierarchy during the mining process.

6 References

1. Stumme, G., Hotho, A., Berendt, B.: Semantic web mining: State of the art and future directions. Web Semantics: Science, Services and Agents on the World Wide Web **4**(2) (June 2006) 124–143
2. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (1993) 199–220
3. Agirre, E., Ansa, O., Hovy, E., Martinez, D.: Enriching very large ontologies using the WWW. In: ECAI 2000 workshop on Ontology Learning. (2000)
4. Faatz, A., Steinmetz, R.: Ontology enrichment with texts from the WWW. In: the Semantic Web Mining Conference (WS'02). (2002)
5. Parekh, V., Gwo, J.P., Finin, T.: Mining Domain Specific Texts and Glossaries to Evaluate and Enrich Domain Ontologies. In: International Conference of Information and Knowledge Engineering. (2004)

6. Xu, F., Kurz, D., Piskorski, J., Schmeier, S.: A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In: the 3rd international conference on language resources and evaluation. (2002)
7. Neshatian, K., Hejazi, M.R.: Text categorization and classification in terms of multi-attribute concepts for enriching existing ontologies. (2004) 43–48 In 2nd Workshop on Information Technology and its Disciplines.
8. Velardi, P., Missikoff, M., Fabriani, P.: Using text processing techniques to automatically enrich a domain ontology. In: Proceedings of ACM- FOIS. (2001)
9. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: The 4th European Conference of Principles of Data Mining and Knowledge Discovery. (2000) 424–431
10. Bendaoud, R.: Construction et enrichissement d'une ontologie à partir d'un corpus de textes. RJCRI'06 (March 2006) 353–358
11. Roux, C., Proux, D., Rechermann, F., Julliard, L.: An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions (2000)
12. Maedche, A., Staab, S.: Mining ontologies from text. Volume 1937., Springer-Verlag (2000) Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management.
13. Srikant, R., Agrawal, R.: Mining generalized association rules. Future Generation Computer Systems **13**(2–3) (1997) 161–180
14. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. Technical Report S2K-92-09 (1992)
15. Fiot, C., Laurent, A., Teisseire, M.: Extended time constraints for sequence mining. In: 14th International Symposium on Temporal Representation and Reasoning. (2007)
16. Masseglia, F., Poncelet, P., Teisseire, M.: Pre-processing time constraints for efficiently mining generalized sequential patterns. In: 11th International Symposium on Temporal Representation and Reasoning. (2004) 87–95
17. Jaillet, S., Laurent, A., Teisseire, M.: Sequential patterns for text categorization. Intelligent Data Analalysis **10**(3) (2006) 199–214
18. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: the 11th IEEE International Conference on Data Engineering. (1995) 3–14
19. Masseglia, F., Cathala, F., Poncelet, P.: The PSP approach for mining sequential patterns. In: the Second European Conference on Principles of Data Mining and Knowledge Discovery. (1998) 176–184
20. Zaki, M.J.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning **42**(1/2) (2001) 31–60
21. Di-Jorio, L., Jouve, D., Kraemer, D., Serra, A., Raissi, C., Laurent, A., Teisseire, M., Poncelet, P.: VPSP : extraction de motifs séquentiels dans weka. In: Démonstrations dans les 22èmes journées "Bases de Données Avancées" (BDA'06). (2006)
22. Di-Jorio, L., Fiot, C., Abrouk, L., Hérin, D., Teisseire, M.: Enrichissiment d'ontologie : Quand les motifs séquentiels labellisent des relations. In: 23 ème journées Bases de Données Avancées. (2007)
23. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: International Conference on New Methods in Language Processing, Manchester, UK, unknown (1994)
24. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. (1988) 143–160