Mining Patterns with Domain Knowledge: a case study on multi-language data

Cláudia Antunes, Tiago Bebiano
Department of Computer Science and Engineering
Instituto Superior Técnico
Lisbon, Portugal
{claudia.antunes, tiago.bebiano}@ist.utl.pt

***Abstract*— Multi-language data impairs the application of mining techniques in a generalized form, since language remains an impenetrable barrier. The advances on domain driven data mining and the study of its semantic aspects open a first window over it, in particular the *D^2PM* framework [1]. This paper proposes a new method for mining patterns over multi-language data, through the use of the *D^2FP-Growth* algorithm and a language constraint, both defined in the context of the referred framework. The new constraint allows for interpreting a word by its meaning and consequently to overcome language differences.**

Keywords: Domain Driven Pattern Mining, Multi-language, Ontology

I. INTRODUCTION

The deployment of data mining processes on several distinct business contexts and their relative success, have contributed to enlarge the interest on data mining, bringing a set of new challenges into the arena, like dealing with complex data, such as multi-language.

Recent advances in the area of domain driven data mining and on the study of its semantic aspects bring new clues on how to use domain knowledge to focus the mining process. In particular, the *D^2PM (Domain Driven Pattern Mining*) framework [1] introduces domain knowledge into the mining process through an ontology and makes use of constraints to focus the discovery according to users expectations, filtering out undesirable patterns.

In this paper we propose a method for mining patterns over multi-language data in the context of the *D^2PM* framework, where mining is done through a new algorithm – the *D^2FP-growth* and a new constraint. This constraint aims for abstracting the language in which each word is presented, allowing for considering it only by its meaning. In this manner, the language constraint considers that two words correspond to the same item if they instantiate the same concept, which is true if they have the same meaning, despite their original language.

The rest of the paper is organized as follows: next related work is overviewed, with particular attention to the *D^2PM* framework. In section 3, the new algorithm is described, and in section 4, the method for dealing with multi-language data is detailed. In section 5, the results of applying the proposed method to a real set of documents in the area of medicine are presented. Section 6 concludes the paper by pointing some guidelines for future research.

II. BACKGROUND

The process of knowledge discovery, usually known as data mining, has been defined as "the nontrivial extraction of implicit, previously unknown and potential useful information from data" [2]. However, despite the deep advances in the area there are a few open issues that deserve attention. How to use domain knowledge to guide the discovery process is one of such topics, but dealing with multi-language data also poses new challenges.

While the first area has been recently addressed by the research on *domain driven data mining* (*D3M* for short) [3], to our knowledge there is no result on data mining to deal with the second issue. Indeed, it is necessary to turn our search into information retrieval to find significant results.

The field of c*ross-language information retrieval* [4] aims for finding relevant information from a query written in a different language. Its key problem is translation, which is accomplished by one of three ways: by translating the query into the document language; by translating the documents into the query language; and by translating both queries and documents to a third language. Among the different approaches to translation emerged in the last years, is possible to identify the ones that use *machine-translation systems* [5], dictionaries [6], parallel or comparable corpora [7] and the Wikipedia [8].

A. Pattern Mining and D^3M

Pattern mining is a subtask of mining association rules, first formulated in 1993 in the context of basket analysis [9] and further developed from then. The goal of pattern mining is to find all frequent sets of items in a dataset, i.e. all sets of items that co-occur at least a minimum number of times in the dataset. In a more generic formulation, items correspond to propositions, pairs attribute / value, most of the times representing only one attribute for some entity, instead of an entire entity as in basket analysis.

The vast application of pattern mining in real applications, in a large set of domains, has shown that pattern mining tend to be un-useful due to the enormous amount of discovered patterns. Indeed, the user is the unique responsible for determining the frequency threshold and the right abstraction level for representing items, a task only executable by experimented users. However, traditional algorithms are not able to focus the discovery in accordance to user expectations, and only constraints have been shown to be useful to accomplish this goal. In fact, constraints are the most effective technique to reduce the number of discovered patterns [10].

The advances in the area of knowledge management, verified in the last years, and in particular the development of the Semantic Web, contributed considerably to advance the area of ontologies, and now they are commonly accepted as a mean to represent and share existing knowledge in information systems, and more recently as a way to incorporate richer constraints in mining problems. The *D^2PM* framework [1] is an example, which provides the means to define constraints that may reduce the number of discovered patterns, and simultaneously, improving processing times.

An *ontology* is a specification of an abstract, simplified view of a domain [11]. Formally, it corresponds to a triple $O=\{C, R, A^O\}$, where C is a set of concepts, which represent the entities in the domain; R is a set of attributes that characterize the concepts and A^O a set of axioms that describe constraints on the ontology, making explicit implicit facts. Among R elements, there are relations, a particular case of attributes, whose range are concepts.

In the counterpart of ontologies are knowledge bases, which specify the known instances of each concept in a particular ontology. A *knowledge base* is a tuple $KB=\{O, I, inst^{-1}\}$, where O is an ontology as defined above; I a set of instances and $inst^{-1}$a function from I to C in ontology O, that identifies the relations among instances and concepts.

B. The D²PM framework

The *D^2PM* framework aims for addressing the problem of pattern mining in the presence of domain knowledge. It is an extension of the *Onto4AR* framework [12] previously proposed, maintaining ontologies and constraints in its core. In this new context, a new formulation of the problem is assumed, where the meaning of an item is clarified and constraints are defined from the combination of several functions, defined in the context of the ontology.

Let $KB=\{O, I, inst^{-1}\}$ be a knowledge base and $O=\{C, R, A^O\}$ a domain ontology as defined above. Consider the subset of R that exclude the relations, this is, the set of attributes whose range is not a concept. Let F be the set of *features* that includes all possible values for attributes.

As in the original formulation of transactional pattern mining, let $L_D=\{i_1, i_2, \dots i_m\}$ be a set of items that appear in D, the dataset containing a set of transactions.

In order to address items in the context of the domain knowledge, available through the knowledge base KB, consider a *d^2item* to be a link to a feature from F, and L to be the set of all *d^2items*. In this new context, an *item* is just a *d^2item* that occurs in a dataset D, and L_D is just a subset of L ($L_D \subseteq L$). Similarly, consider a *d^2itemset* as a set of *d^2items*, and an *itemset* as a *d^2itemset* whose elements are all items, which means that all of its items occur in the dataset.

Now let C_O be a *constraint* defined in the context of the ontology O, as a tuple $C_O=(\sigma, \mu, \psi, \varphi)$, where σ is the minimum support threshold; μ *is* the *mapping function* that maps items to features; ψ is the *equivalence function*, a predicate among *d^2items*; and φ, the *acceptance function,* a predicate over *d^2itemsets* as defined in the traditional formulation. It is said that a *d^2itemset X* occurs in D under C_O, if and only if some transaction T in D contains X under $C_{O,}$ by its hand, T is said to *contain* X under C_O if for all *d^2item x* belonging to X there is an item t in T such t is equivalent to x in the context of C_O.

Given an ontology O, a dataset D and a constraint C_O, the problem of mining all patterns in D under C_O, corresponds to the discovery of the set of all *d^2itemsets* that occur in D under $C_{O,}$ known as *d^2patterns*.

III. The D²FP-Growth Algorithm

To our knowledge, the only algorithm able to perform transactional pattern mining in the presence of domain knowledge is the *D2Apriori* [13]. However, it suffers from a large consumption of memory resources, due to the inexistence of a right way to deal with items at the best level of abstraction.

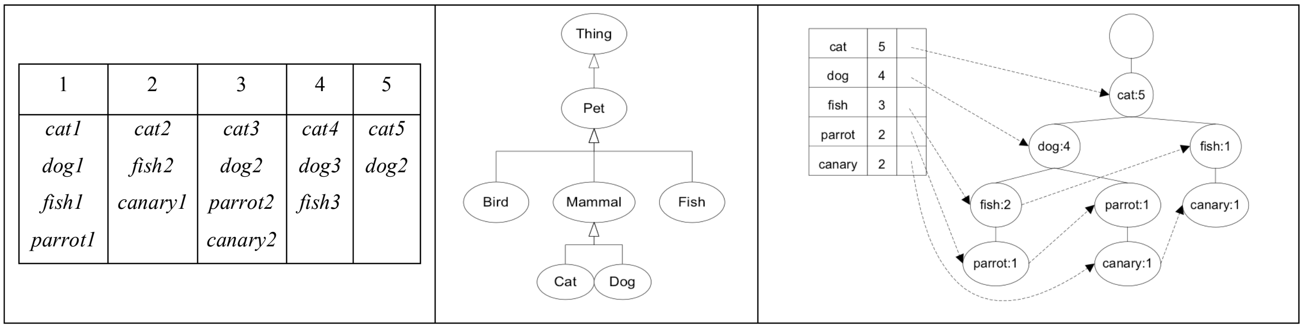
Domain Driven FP-Growth, or just *D^2FP-Growth*, is a generalization of *FP-Growth* [14] to work in the context of the *D^2PM* framework that aims for mining transactional data using domain knowledge. This knowledge acts as a guide for the mining process, more precisely, its goal is to find patterns on the context defined by an ontology and that conform to some constraint as described previously.

As usual in transactional pattern mining, the algorithm receives a dataset composed by a set of items, but also a domain ontology (O) and a constraint (C_O). As output, it returns a set of patterns accepted by the constraint.

With this input, the algorithm begins by reading the ontology and filling the knowledge base (KB) with already known instances, if they are present in advance. Having O and KB as the context the algorithm starts reading each transaction in the dataset, creating its corresponding *d^2itemset*. This is rather a complex process, since, from transaction to transaction is necessary to create the *d^2itemsets* with domain information including corresponding instances and features. The knowledge necessary to instantiate items to the right concepts is given by the mapping function in the constraint in use ($C_O.\mu$): for each item present in each transaction, it is necessary to create a *d^2item*, which includes a link to a feature.

Once all transactions have been read, the FP-Tree must be created. However, this structure only contains *d^2items* accepted by the constraint in use, *i.e.*, the *d^2items* equivalent to the items in the dataset that are accepted by $C_O.\varphi$ and whose support is at least equal to $C_O.\sigma$, instead of the items frequent in the dataset. In particular, it is possible to insert a *d^2item* that does not occur in the database, which is called an *abstract item*, which corresponds to the *d^2item* accepted by the constraint equivalent to one or more items that occur in the database, according to the equivalence function $C_O.\psi$.

With only accepted *d^2items* in the tree is then easy to discover patterns, just following the original *FP-Growth* and testing each created pattern against the constraint, validating its acceptance through the minimum support $C_O.\sigma$ and the acceptance function $C_O.\varphi$. The discovered patterns are then *d^2itemsets*, sets of d^2items accepted by the constraint, not necessarily effective itemsets.



1	2	3	4	5
cat1 *dog1* *fish1* *parrot1*	*cat2* *fish2* *canary1*	*cat3* *dog2* *parrot2* *canary2*	*cat4* *dog3* *fish3*	*cat5* *dog2*

Figure 1. Dataset for pets (left), ontology (middle) and corresponding FP-Tree

A. *Illustration*

Consider a dataset of pets with corresponding owners and the goal of finding the frequent sets of kinds of pets adopted together. With this purpose, applying *D^2FP-Growth* only requires the dataset, an ontology describing the relevant characteristics of pets and a constraint *C* as described next.

Let *O* be the ontology represented in Figure 1 (middle), with only one attribute *species* defined for concept *Pet*, and *C* be a constraint as defined above, for describing which animals should be considered. More precisely C=(σ, μ, ψ, φ), with σ=20%, μ establishes that each pet should be mapped to its species, which means that for example *cat1* is mapped to the feature *species* and value *cat*; ψ determines that two pets are equivalent if they belong to the same species; and φ doesn't impose any other constraint. From the dataset in Figure 1 (left), the ontology (in the middle), and the described constraint, *D^2FP-Growth* will find the FP-Tree in Figure 1 (right). Note that, in this example, the tree only contains items that do not occur in the dataset, since none of them correspond to a particular item. Against these results the traditional *FP-Growth* algorithm would design an empty FP-tree, since in our example each pet has only one owner. In order to reach the same results it would be required to pre-process the dataset creating a new one, where each pet would be represented by its species. This could seem an easy way to solve the problem. In this case where the constraint is simple, that will work, but wouldn't allow for changing the abstraction level considered by the constraint during the mining process without re-computing the entire dataset.

IV. Multi-language Pattern Mining

The goal of multi-language pattern mining is to discover the set of terms that co-occur a significant number of times in a dataset composed by transactions in different languages. The language of the data is a common barrier to the mining process; so, the challenge is to properly abstract the language. In other words, the point is to consider a word by its meaning without taking the language into account.

Our proposal to deal with multi-language problem is to use the *D^2PM* framework in order to establish a context where several words may count for the support of a single one, overcoming the differences among languages.

Multi-language data may appear in a set of different contexts, with documents one of its major expressions. In order to deal with them, it is necessary to define a method, based on the proposed framework. It comprises four steps: data pre-processing, framework setup, mining process and evaluation (Figure 2). Next sections describe each step in detail.

A. *Data Preprocessing*

Data pre-processing of documents usually involves a chain of steps that goes from representing each document into a manageable form, to processing each word in order to reduce the number of different ones per document.

Considering a set of documents as input, first, it is necessary to transform data into a readable format for the tools used in the next steps. The most used has been the array of words, where each document is characterized by a list of the words contained.

After this transformation, follows the removal of stop words, words that have little information per se (like conjunctions and articles). For instance, the sentence "The President decided to accept the petition" might be converted to "President decided accept petition" if the stop list includes the word "the" and "to". In this step we also include the removal of punctuation and irrelevant terms, for example, dates and mathematical expressions.

Lemmatization is the next step, in which a valid word of the language (the *lemma*) is used as a representative for all the lexical variations that may apply. It is the headword that appears in a dictionary definition (e.g.: "Eating" is lemmatized to "Eat"). This is a fundamental step since it collapses a lot of words into a single one. It is important to understand that reducing the amount of words has a direct impact on the number of discovered patterns. The common way to find the lemma of a word is using a part-of-speech tagging tool [15]. The previous sentence would be reduced to "President decide accept petition".

After lemmatization, documents are then just lists of possible relevant words that occur in it, able to serve as input for the mining process.

B. *Framework Setup*

After preparing the data is time to prepare the framework, by defining both the ontology and the knowledge base that represent the available domain knowledge, as well as by defining the constraint to guide the mining process.

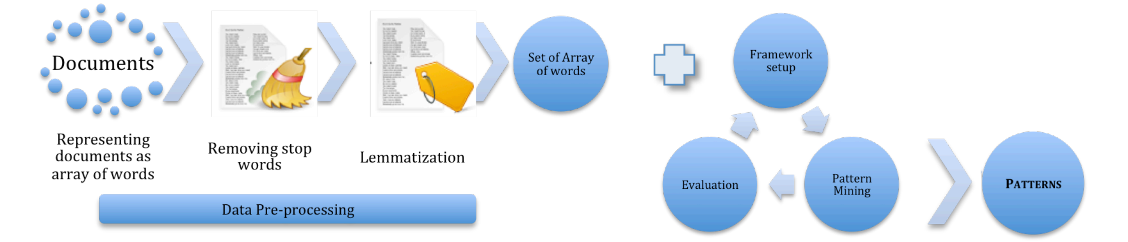


Figure 2. Method for multi-language pattern mining

The first choice is naturally the domain ontology of interest. It can be created by scratch or just by reusing an existing one. However, and since the goal is to find multi-language patterns, the selection of the ontology language directly affects the volume of available resources and the precision of the translations. Being English considered the universal language, the ontology may be designed in English, increasing the probability of a known translation for the majority of languages.

Another important issue is that the ontology should have a top concept with at least one attribute. This concept may be called *Word*, and its attribute denoted *label*. The idea of this concept is to facilitate the definition of the constraint to use, warranting a common basis for all other words in the ontology.

The next step is optional and refers to the construction of the corresponding knowledge base. In particular, the set of instances can be filled with words that are synonyms of the concepts in the ontology. This means that for each concept in the ontology, we may introduce the corresponding word as an instance in the knowledge base, but also all known synonyms for it.

The last thing to do is to define the constraint to use, being an instantiation of the language constraint defined as follows.

A *language constraint* is a constraint defined in the context of D^2PM, denoted $C_L=(\sigma, \mu, \psi, \varphi)$, where the minimum threshold σ remains a user-defined variable; the mapping function μ works as follows:

Case 1: If the word x is equal to the label of concept X, then x is an instantiation of X

Case 2: If there is a translation of the word x that is equal to the label of concept X, then x is an instantiation of X

Case 3: If the word y is a synonym of the word x that is equal to the label of concept X, then y is an instantiation of X

Case 4: If there is a translation of the word z *that* is equal to the word y, which in turn is a synonym of the word x, equal to the label of concept X, then z is an instantiation of X

Case 5: If the word x doesn't match any of the previous cases, then x is instance of concept *Word.*

The equivalence function ψ establishes that two instances are equivalent if they instantiate the same concept.

At last, the acceptance function φ accepts all instances that instantiate some concept other than the *Word* concept. Naturally, this function may be more restrictive and may be redefined by the user, keeping this minimum structure.

With the right framework, it is time to find patterns through the use of the D^2*FP-Growth*.

It is important to note that with the proposed ontology there is a univocal correspondence between instances and features, since each concept has the unique attribute *label*; which in turn imposes that there is a univocal correspondence between words (items) and instances. In this manner, the patterns found are just sets of words that co-occur in the same document a frequent number of times, independently of their language.

The evaluation of results is done as usual in pattern mining, by analyzing the quantity of discovered patterns but also the quality of each pattern per se.

From the perception of the quality of the achieved results, it may be necessary to re-run the mining process, by redefining the constraint in use. This may be done just by adjusting the minimum support threshold or by imposing more restrictive constraints in the acceptance function.

C. Example

For illustration purposes, consider a set of three small documents, named *A, B* and *C*, written in English, Portuguese and Spanish (Figure 3 – left). After accomplishing data pre-processing documents are just sets of words in a given language (Figure 3 – right). Notice that meaningless words were excluded and the words were reduced to their lemma.

Now consider the ontology illustrated in Figure 4 (left) that illustrates an ontology in the domain of medicine adapted from BioPortal (http://bioportal.bioontology.org/). The constraint should be just the language constraint defined above, with a minimum support threshold of 60%, which means that a pattern is considered frequent if it occurs at least in two documents.

In the mining step, by applying the D^2*FP-Growth*, the first task is to read the dataset. The word "lung" originates a d^2*item* linked to the concept *Lung*, so as the word "cancer" (linked to the disease *Cancer*) and similar for "asthma". Note that these words match the first rule in the mapping function. The word "car" originates a d^2*item* linked to the concept *Word*, matching the last rule of the mapping function, which will be rejected by the acceptance function. For the word "adulto" it is created an abstract item since its translation corresponds to the concept "adult", following the second rule of the mapping function. The remaining words "pulmón" and "asma" will be linked to the concepts *Lung* and *Asthma*, respectively, matching the second rule on the mapping

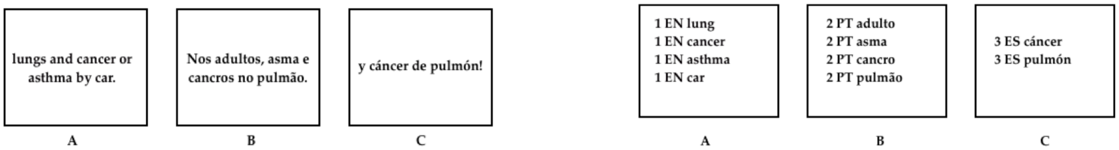


Figure 3. Sample documents in English, Portuguese and Spanish before (left) and after pre-processing (right)

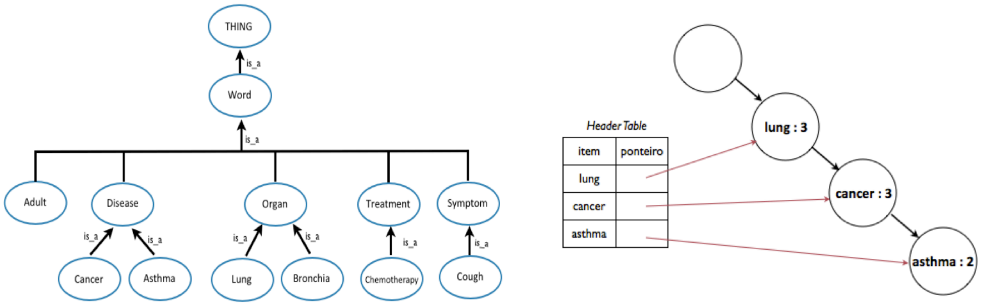


Figure 4 . Domain ontology for medicine (left) and FP-Tree created by D^{2}FP-Growth for a support of 40% (right)

function.

By this time the method converges to the support counting for each *d^{2}item*. It is important to remember that the counting process isn't equal to the one proposed in the FP-Growth algorithm. In particular, items like "pulmão" and "lung" will count for the support of the same item, following the equivalences defined by the equivalence function in the constraint. Then, the support for each *d^{2}item* is: *lung*:3, *adult*:1, *cancer*:3 and *asthma*:2. For a minimum support of 60% only *adult* isn't frequent. And the resulting FP-Tree only contains one path, as illustrated in Figure 4 (right).

From this tree, the patterns are just the combinations of items along the path. Representing each pattern as *itemset* : *support*, the patterns discovered are: *lung*:3, *cancer:*3, *asthma*:2, (*lung*, *cancer*):3, (*lung*, *asthma*):2, (*cancer*, *asthma*):2 and (*lung*, *cancer*, *asthma*):2.

V. Experimental Results

This case study aims for demonstrating that is possible to mine multi-language data, using domain knowledge. To validate the statements made along the document, the following charts show the result of applying the method to a set of documents about pulmonology written in four different languages – Portuguese, Spanish, English and French. All translations needed are based on Bing Translator API (http://www.microsofttranslator.com/).

The domain ontology used has about 250 concepts related to the subject of the documents. The dataset is composed by 92 documents (approximately 7000 word), 46 in English and the rest uniformly distributed by the other three languages. English documents are just translations of the other documents made by their own authors. The performance of the algorithm is evaluated by its efficiency (time and memory consumption) and number of patterns found.

Figure 5 (left) shows the number of patterns found. It is clear that FP-Growth finds more patterns, but only for supports below 30%, since for larger supports there aren't enough documents in each individual language for supporting them. For a support of 5%, FP-Growth finds five times more patterns than *D^{2}FP-Growth*, indeed it finds every set of co-occurrences in any language, including terms that are not related to the domain in consideration. This explosion is usual in pattern mining, and is avoided in the *D^{2}PM* framework, finding less than 500 patterns.

Figure 5 also shows the memory (middle) and time (right) consumption. As it is easily understood, as the number of discovered patterns increase, the memory and time consumption increases. *D^{2}FP-Growth* consumes about 40% of the time spent by FP-Growth on mining multi-language data. And memory consumptions show that the memory needed in the new algorithm is approximately constant and mostly wasted on storing the original data and its context (instances in the knowledge base).

Figure 6 reflects the inefficiency of FP-Growth in multi-language case. With higher supports the algorithm can't find any patterns and consequently the time and memory spent by pattern is too high. With lower supports the time and memory consumption by pattern decreases. The same occurs with *D^{2}FP-Growth* but with softer variations.

VI. Conclusions

Mining multi-language data is a challengeable issue in the knowledge discovery process, mostly due to the range of existing languages and the complexity associated with, for example, words morphology and adequate translations.

Ontologies begun to be used worldwide, and they can be both used by humans and by information systems. In the area of information systems, it is now generally accepted that these formalisms are a key to share and reuse the existing domain knowledge, and in the last years they become to be considered in the mining process. The *D^{2}PM* framework is one of the mining approaches that try to incorporate available domain knowledge into the pattern mining process, through the use of ontologies.

In this document, we have explored this framework to solve another unsolved issue in the area of pattern mining – the discovery of information in data stored in more than one

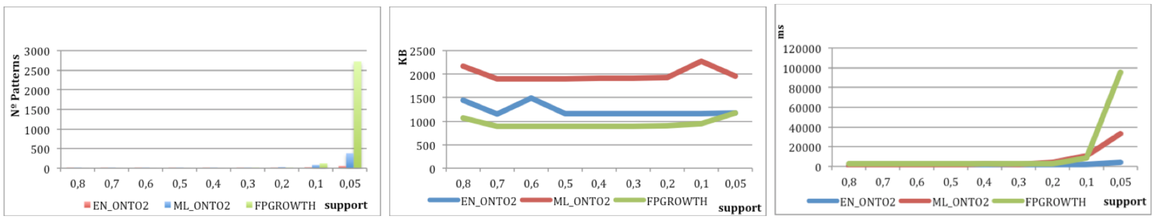


Figure 5. Number of patterns, memory (left) and time (right) spent by D2FP-Growth and FP-Growth.

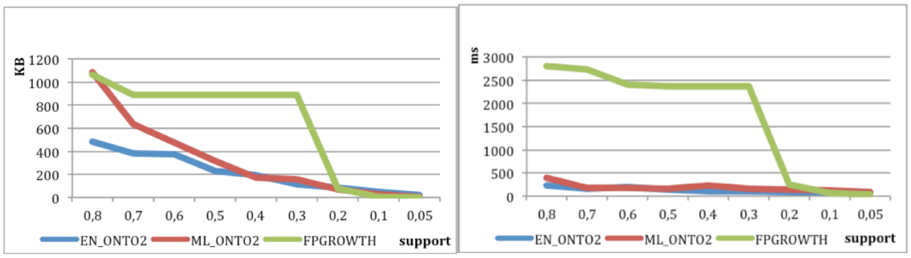


Figure 6. Average memory (left) and time (right) spent to find a pattern by D2FP-Growth considering English transactions, multi-language transactions contrasting with FP-Growth.

language. This is achieved, by making use of a language constraint that allows for seeing a word by its meaning, and the results overcome the ones achievable by traditional pattern mining approaches.

Despite the proposed method fulfill our goal, it is clear that can achieve better results. The matching between data and ontology concepts directly influences the number of patterns found; so, it would be important to associate synonyms to a concept in order to increase the correspondence (a word can have various translations that have the same meaning and without synonyms they aren't considered in our experiments). This can be done by introducing the synonyms as instances in the knowledge based as described previously.

Another important issue is the quality of the translation. Indeed cross-language information retrieval favors approaches where context can be used. In our case, this is only possible if the translation of each word would be done during the pre-processing, before the removal of stop words.

ACKNOWLEDGMENT

This work is partially supported by Fundação para a Ciência e Tecnologia under research project *D2PM* (PTDC/EIA-EIA/110074/2009).

REFERENCES

[1]. Antunes, C.: D2PM: a framework for mining generic patterns. Technical, Instituto Superior Técnico, Lisbon (2011)

[2]. Frawley, W., Piatetsky-Shapiro, G., Matheus, C.: Knowledge discovery in databases: an overview. AI Magazine 13(3), 57-70 (1992)

[3]. Cao, L.: Domain-Driven Data Mining: Challenges and Prospects. IEEE Transactions on Knowledge and Data Engineering 22(6), 755-769 (June 2010)

[4]. Nie, J.: Cross-language Information Retrieval. Synthesis Lectures on Human Technologies 3(1), 1-125 (2010)

[5]. Zhang, T., Zhang, Y.: Research on chinese-english information retrieval. In : 7th International Conference on Machine Learning and Cybernetics (2008)

[6]. Ballesteros, L., Croft, B.: Dictionary-based methods for cross-lingual infomation retrieval. In : 7th International DEXA Conference on Database and Expert Systems Applications, pp.791-801 (1996)

[7]. Chew, P. A., Bader, B. W., Kolda, T. G., Abdelali, A.: Cross Language Information Retrieval Using PARAFAC2. In : International ACM SIGKDD Conference on Knowledge Discovery in Databases (2007)

[8]. Nguyen, D., Overwijk, A., Hauff, C., Trieschnigg, R. B., Hiemstra, D., de Jong, F.: WikiTranslate: Query Translation for Cross-lingual Information Retrieval using only Wikipedia. (2009)

[9]. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In Buneman, P., Jajodia, S., eds. : SIGMOD Conference, Washington, D.C., pp.207-216 (1993)

[10]. Bayardo, R.: The Many Roles of Constraints in Data Mining. SIGKDD Explorations 4(1), i-ii (2002)

[11]. Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 21–66 (1998)

[12]. Antunes, C.: An Ontology-based Framework for Mining Patterns in the Presence of Background Knowledge. In : Int'l Conf. on Advanced Intelligence, Beijing, China, pp.163-168 (October 2008)

[13]. Antunes, C.: Pattern Mining over Star Schemas in the Onto4AR Framework. In Saygin, Y., Yu, J., Kargupta, H., Wang, W., Ranka, S., Yu, P., Wu, X., eds. : 2nd Int'l Workshop on Semantic Aspects in Data Mining in the Int'l Conf on Data Mining, Miami, pp.453-458 (December 2009)

[14]. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In : Int'l Conf. on Management of Data, Dallas, pp.1-12 (May 2000)

[15]. Martinez, A. R.: Part-of-speech tagging. Wiley Interdisciplinary Reviews: 11 Computational Statistics(2011)