Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 21 (2013) 75 – 82

Procedia Computer Science

The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2013)

A new Data Mining System for ontology learning

Using Dynamic Time Warping alignment as a case

Choukri Djellali

LATECE UQAM, 201, PK 4470, Président Kennedy Montréal (Québec) H2X 3Y7, Canada

Abstract

In recent years, several approaches have been proposed to solve the problem of ontology learning. In most approaches, the text representation is only based on the information contained in term weighting and does therefore not process the semantic contained in the sequence in which the words appear. Moreover, the use of many dimensions adds unnecessary noise in the generated model and affects the quality of learning (generalization). Hence, in the present study, we propose a semi-automatic approach that uses the variables selection and clustering to find the candidate changes. In order to identify the correspondence between the ontological artifacts and candidate changes, we used an alignment process. Our approach exploits natural language processing, indexation and machine learning techniques to increase the productivity of ontology engineering task during the enrichment of conceptual model. Good experimental studies demonstrate the multidisciplinary applications of our approach.

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: Learning, Data Mining, NLP, Ontology, Semantic Web, Alignment, Indexation, Variables Selection, TSVD.

1. Introduction

Ontology is used to identify and overcome the knowledge sharing barriers. Ontology provides a shared vocabulary, which can be used to model a domain and support reasoning about concepts. The literature contains many definitions of ontology; many of these contradict one another. However, the best known and most cited is (Gruber, T R, see [10]), which is also the definition we adopt in our paper: «*Ontology is an explicit specification of a conceptualization*». Ontology contains information like any other structure. In many cases, we can change the perspective of the domain, discover a problem in the original conceptualization or incorporate additional functionality by changing the user needs. In order to solve the problem of knowledge evolution, we propose a semi-automatic approach that uses natural language

djellali.choukri@courrier.uqam.ca

Selection and peer-review under responsibility of Elhadi M. Shakshuki

doi:10.1016/j.procs.2013.09.012

processing, machine learning techniques and an alignment process that controls the syntactic neighborhood between the compared artifacts. Structural rules are applied to improve the degree of similarity between ontological entities. This hybrid process creates an alignment rules that define how to transform the inputs by defining all types of possible associations between ontological entities.

The paper is organized as follows: In Section 2, we give the current state of the art, our research questions and the problematic of ontology learning. The conceptual architecture of our approach is given in Section 3. Before we conclude, we give in Section 4 a short evaluation with benchmarking models for our conceptual model. Then, a conclusion (Section 5) ends the paper.

2. State of the art, Problem and Research Questions

The use of ontology is a possible approach to overcome the problem of semantic heterogeneity. Ontology is proposed as a way to overcome the obstacles of knowledge integration. It is used to unify Databases, Data Warehouses, knowledge bases vocabularies and even to maintain consistency in updating Corporate Memories used in knowledge management in modern enterprises [23]. As indicated above, an ontology is an abstract view of a particular domain of interest. These abstract views cannot be considered as static because there are several occasions that can make it necessary to change the ontology (conceptualization, user needs, explicit feedback, changes in the field, the adaptations to the different tasks, etc.). In recent years, several approaches have been proposed to solve the problem of ontology learning. These approaches include: ontology pruning [19], conceptual grouping [17], formal concept analysis (FCA) [15], association rules [16], pattern extraction [4] and conceptual learning [9]. However, these approaches do not consider all available information to make a realistic decision. They are often focused on limited types and neglect others.

Firstly, the data representation generates a highly dimensional space, where the number of attributes is larger than the number of documents (Bellman's curse of dimensionality). The curse of dimensionality is a crucial challenge for several learning tasks (linear inseparability, big data, rapid response systems, sampling with limited samples, real-time systems, representativeness, noisy variables, etc.). On one hand, the model cannot explain the intrinsic relations in the text with a very small projection space. On the other hand, the use of many dimensions adds unnecessary noise in the generated model and affects the quality of learning (generalization).

Secondly, the text representation does not capture the semantic contained in the sequence of words and therefore it provides an abstraction of syntactic relations between different linguistic units. It is less discriminatory if the term appears in several documents and the distribution of categories is related to the distribution of documents containing a specific term.

Most previous approaches provide limited support for all activities of the engineering process, in particular, the phase of evolution. In these approaches, there are no built-in methods or tools that combine different techniques and heterogeneous sources of knowledge with existing knowledge to accelerate the evolution process. In order to overcome the obstacles mentioned above, our approach exploits natural language processing, indexation, variables selection and machine learning techniques to find the relevant patterns. The uncovering of hidden patterns is performed by pre-processing, indexation and Truncated Singular Value Decomposition (TSVD) techniques.

3. The architecture of our Data Mining system

In this section, we introduce the architecture adopted in our approach. The learning process illustrated in Figure 1 starts with capturing terms from available documents. In pre-treatment, negative dictionary and stemming are used to filter out the words having no informative value in describing the document contents. In order to represent the textual document, we used the vector space model (VSM) (also known as bag of words). Each document is indexed by its terms in a vector and the weight of each term is

calculated by TF-IDF method (Term Frequency Inverse Document Frequency) [2]. This vector representation generates a highly dimensional space, where the number of variables is larger than the number of documents available for learning. We used variables selection Wrapper Model [21] to overcome the obstacles mentioned above. This method assumes that there is a hidden latent structure behind data. The uncovering of hidden structures is performed by Truncated Singular Value Decomposition process (or truncated SVD) [6]. The Wrapper Model derives an optimal representation of the original data in a lower dimensional space. The clustering process identifies the relevant patterns in the process of knowledge acquisition. In most clustering algorithms, the initial number of clusters is provided by the user. However, this knowledge is usually not known in advance. It is desirable to automatically identify the number of clusters to discover the intrinsic structure across documents. For these reasons, we used the neural network Fuzzy Adaptative Resonance Theory (also known as Fuzzy ART) to organize documents into thematic subsets according to their semantic [12]. This model does not depend on the order of on-line presentation (plasticity-elasticity). All clusters are described by keywords (labels) representing their contents. Labels and ontological artefacts are compared using an alignment process. The overall objective is to achieve an alignment of strings in an ontological model where the alignment has some kind of undesirable error.

In order to identify the alignment rules, the process of extracting similarities apply syntactic rules to produce a similarity matrix reflecting similarities between compared artifacts. It creates alignment rules that define how to transform the entities by defining all types of possible associations between the ontological artefacts and labels.

The enrichment process uses the identified alignment rules to provide the necessary update. Before using the module provided by the Data Mining system, the user must first use the administration module to create an index of documents. This is necessary to update the indexing model used in the retrieval and learning modules. When the index is created the system should maintain a temporal version to check if the document collection was modified after the creation of indexes. Once this step is completed, the user can use the Data Mining module to enrich the ontology. The CRISP-DM-OWL[1] ontology used in this project is integrated into a hybrid system DM [18], describing the artefacts and the basic rules to improve the intelligence level of the system. The ontology acts as a source of additional knowledge in the system.

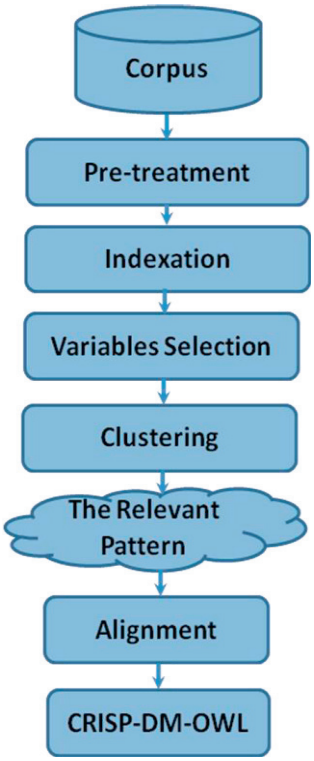

Fig. 1. The conceptual model

4. Experimentation

4.1. Configuration

The training corpus consists of a set of IEEE abstracts divided in several categories. The average length of the document in terms of words is 182.53 in the training set and 178.14 in the test set. The number of documents in each category is highly unbalanced. Thirty percent of the data are selected to test the model (no theoretical justification for this percentage). Table (1) shows in detail the statistical

(1) *http://www.elmanahel.ca/ontology/crisp-dm-owl.owl*

distribution of words in the Data sets ($\overline{L}$ is the average length of the document and σ_L the standard deviation of document length).

Table 1. The Statistical Distribution of Words.

Data set	$\overline{L}$	σ_L
Learning	182.53	60.65
Test	178.14	60.55

4.2. Pre-treatment

The removal of punctuation, negative dictionary and stemming are the most frequently used pre-treatment techniques to remove noise. In negative dictionary, we used the Glasgow list [20] as a stop words list in our experiments. This list is widely used as English standard stop word; it covers a large number (351 stop words). Among several implementations of stemming algorithms, we choose the version that was published by Martin Porter [13]. This version has the advantage of a clear separation between the substitution rules and procedures that test the conditions attached to a particular lexeme.

4.3. The variables selection

The Truncated Singular Value Decomposition of an $m \times n$ real [document,term] $\hat{D}$ is a factorization of the form defined by formula (1). $\hat{D} = U_k \Sigma_k V_k^T$ (1)

U_k : the left-singular vectors of $\hat{D}$ with orthonormal columns; Σ_k : diagonal singular values matrix.

V_k^T :the right-singular vectors of $\hat{D}$ with orthonormal columns.

Figure (2) (a) shows the rank approximation of the term-document matrix $\hat{D}$. The rank of $\hat{D}(r = rank \mid \hat{D} \mid = 1100)$ is given by the number of singular values σ_i those are non-zero. Singular values (also known as canonical multipliers) are positive real numbers and by convention they are sorted in descending order along the diagonal of singular values matrix Σ_{1100} .

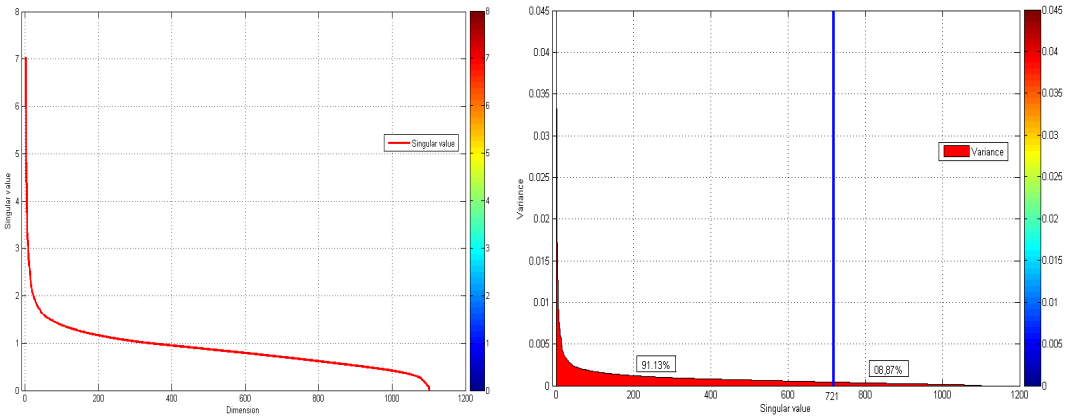

Fig. 2. (a) The Rank Approximation; (b) Variance vs. Singular Value

We used the additional variance algorithm [1] as a criterion to find the number of relevant variables. The i^{th} singular value σ_i is proportional to the further variance var_i described by the following equation:

$$\text{var}_i = \frac{\sigma_i^2}{\sum_{j=1}^{j=r} \sigma_j^2} \quad (2)$$

In our case, the bias-variance trade-off is measured between the (k) relevant variables and the cumulative variance. Hence, the TSVD Wrapper framework to select the most relevant variable subset $S_i = \Phi$ given the set of singular values $S = \{diag\ (\Sigma_r) = \{\sigma_1, \sigma_2, ...\sigma_r\}$ is defined by the following formula:

$$\psi^{k+1} = ArgMax_{s_i \subset s} = \sum_{i=1}^{k} \frac{\sigma_i^2}{\sum_{j=r}^{\sum_{j=1} \sigma_j^2}} \quad (3)$$

This monotone criterion ensure the optimality of search for a subset of variables, that is, any change in the size of the subset is positively correlated with the value of the selection criterion function.

Figure (2) (b) shows the explained variance using all singular values. More than 91.13% of the variance in the data was explained by the first 721 singular values and a little further explanation of the variance is found in the range [722.1100]. Thus, the vertical line maximizes the cumulative variance of projected points cloud in the selected space d^{721}.

The first 721 singular values are much greater than the last singular values (Figure 2 (a)) and the cumulative effect of the variance of the last 379 singular values does not exceed the contribution of the first singular values (Figure 2 (b)). In addition, the variables related to the small singular values are almost irrelevant and do not affect the measures of similarity between documents, i.e., their inclusion would reduce the accuracy of judgment. As a result, we generated reduced projection space by keeping only the first 721 singular values in the matrix Σ_{1100} .

4.4. Clustering

In order to use the Fuzzy Adaptive Resonance Theory, many problems must be solved. Firstly, the neural network creates prototypes increasingly over time corresponding to the input patterns with high values. The prototypes with low values could never be accessed during the learning process. Therefore, the neural network prototypes are not accessible during the learning process, i.e., category proliferation [22]. Secondly, the random initialization reduces the convergence speed of clustering. Hence, the task of clustering with Fuzzy ART network requires a set of pre-processing operations before presenting the input vectors to the input layer.

• Category proliferation: in order to overcome this obstacle, we used the complement coding of the input patterns. The complement coding allows a complete preservation of any information stored in the vector length, i.e., maintaining the amplitude of the vector and generating redundancy to distinguish the noisy variables (symmetric coding theory).

• Typical initialization: this initialization scheme reduces the computation time and improves the convergence speed to achieve the neighbourhood vicinity of the response (global optimum). Table (2) shows the Fuzzy clustering architecture configuration. The ascending weights b_{ij} are initialized by low values and backward weights t_{ij} are initialized by the value 1. The resonance parameter controls the number of neurons in the output layer. When the resonance increases, the number of category in the output layer also increases. If $\rho = 1$, the neural network generates a new class for each input vector in the learning set (also known as base set). The parameter α (choice parameter) takes its values in the range $[0, \infty[$ (the typical value of α is 0.001). The parameter L (uncommitted choice parameter) takes values in the interval $[1, \infty[$. The learning rate β is independent of time; it is placed in the interval [0,1] (typical value of β is 0.9) [5], [8].

Table 2. Architecture Configuration

Parameter	Allowable value	Typical Value
L	L>1	1
ρ	$0<_{\rho}<=1$	0.9
b_{ij}	$0 \prec b_{ij}(0) \prec \frac{L}{L-1+\mid corpus \mid}$	0.0001
t_{ij}	$t(0)_{ij} = 1$	1
α	$[0,\infty[$	0.001
β	[0,1]	0.9

4.5. Alignment

To identify the correspondence between the ontological artifacts and descriptive labels, we used the method of Dynamic Time Warping similarity (DTW) (also known as Levenshtein) [7], [14].

$$Cof_{sim_D}(Label_k, C) = 1 - \frac{\lambda_D(Label_k, C)}{\max(|Label_k|, |C|)}, \quad C \in \Gamma^m, Label_k \in \Gamma_{OWL}^n$$

λ_D : Dynamic Time Warping distance.

Γ : the set of alphabets used to build chains of descriptive labels.

Γ_{OWL} : the set of alphabets representing the artefacts of the ontology CRISP-DM-OWL.

Table (3) shows the produced matrix when the Dynamic Time Warping distance is calculated between two strings « *clustering* » and « *ClusteringAlgorithm* ». The sequences of alignment operations can be easily recovered from the matrix by traversing the path: $\lambda_D^{|\text{clustering}|-1|\text{Clustering Algorithm}|-1} \rightarrow \lambda_D^{0,0}$.

Figure 3 shows the calculated similarity values between the generated descriptive labels and all CRISP-DM-OWL ontological artefacts.

Table 3. The DTW Distance between two strings *clustering* and *ClusteringAlgorithm*.

		c	l	u	s	t	e	r	i	n	g
	0	1	2	3	4	5	6	7	8	9	10
C	1	**0**	1	2	3	4	5	6	7	8	9
l	2	1	**0**	1	2	3	4	5	6	7	8
u	3	2	1	**0**	1	2	3	4	5	6	7
s	4	3	2	1	**0**	1	2	3	4	5	6
t	5	4	3	2	1	**0**	1	2	3	4	5
e	6	5	4	3	2	1	**0**	1	2	3	4
r	7	6	5	4	3	2	1	**0**	1	2	3
i	8	7	6	5	4	3	2	1	**0**	1	2
n	9	8	7	6	5	4	3	2	1	**0**	1
g	10	9	8	7	6	5	4	3	2	1	**0**
A	11	10	9	8	7	6	5	4	3	2	**1**
l	12	11	10	9	8	7	6	5	4	3	**2**
g	13	12	11	10	9	8	7	6	5	4	**3**
o	14	13	12	11	10	9	8	7	6	5	**4**
r	15	14	13	12	11	10	9	8	7	6	**5**
i	16	15	14	13	12	11	10	9	8	7	**6**
t	17	16	15	14	13	12	11	10	9	8	**7**
h	18	17	16	15	14	13	12	11	10	9	**8**
m	19	18	17	16	15	14	13	12	11	10	**9**

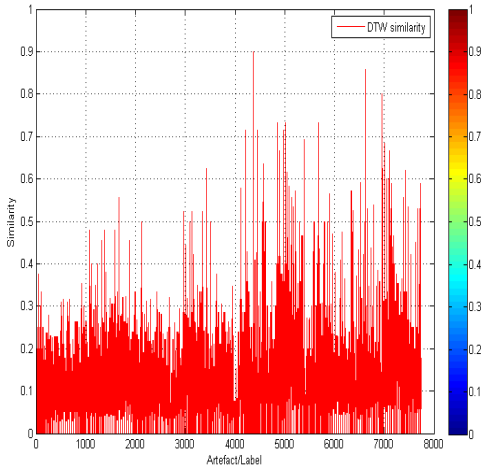

Fig. 3. The DTW Alignment

The Dynamic Time Warping distance similarity has received a lot of attention because the discriminative alignment is powerful for several applications. It adds the ability to take into account the insertion, deletion and substitution. The use of dynamic programming for distance calculation gives the best alignment, but leads to very slow execution time. For this problem we always use approximate methods called heuristics. Hence, the choice of an appropriate distance measure to meet the application needs is crucial task and an attention should be paid to the selection of an appropriate measure for an alignment process:

- The distance functions require extensive pre-processing tasks such as noise removal.
- The distance functions are sensitive to the transformations of patterns (translation, rotation, transposition, scaling, etc.).
- The sequence of alignment operations and the cost of processing are different.
- The algorithms vary depending on the type of search and the methods used to achieve the optimal transformation.
- The choice of the distance measure is closely related to the determination of an optimal alignment.

4.6. Update

The computable model is an explicit representation of the acquired conceptualization. It implies in particular:

- Choosing an ontology editor: we used the plug-in OWL-DL[(2)] (Protégé extension) to implement and update the computable model. This plug-in provides an interface for the reasoning tools based on descriptive logic.
- Choosing a representation language to encode the ontology: the updated ontology is saved as OWL-DL. This language takes advantage of the descriptive logic, including well-defined semantics and automated reasoning techniques.

4.7. Evaluation

Consistency and subsumption can accentuate the main features of the ontology scheme as well as its population. Description logic seems perfectly suited to this situation. It has a formal semantics based on logic and equipped by decision procedures that have been designed for several automated reasoning systems. As illustrated in Figure (4), the Descriptive Inference System used to evaluate the updated ontology completeness is based on the inference engine RacerPro[(3)] (Renamed A-Box and Concept Expression Reasoner). Thus, we can consider the reasoning tool as an expert system based on structure of facts and rules.

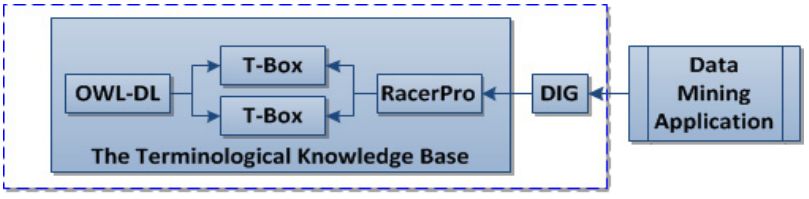

Fig. 4. The Descriptive Inference System

The updated ontology can be regarded as a T-Box/A-Box representation with a hierarchy of roles describing the domain in terms of classes (concepts) and properties (roles). The DIG protocol [3] is used to connect the Data Mining applications to the inference system. In this way, we can query the Terminological Knowledge Base to ensure that all facts can be inferred from the updated ontology. We chose RacerPro as a reasoning tool for our approach because it includes several optimization techniques to ensure good performance of search, in particular, the dependency-directed backtracking and DPLL-style semantic branching [11].

(2) http://protege.stanford.edu/overview/protege-owl.html

(3) http://www.racer-systems.com/products/racerpro

5. Conclusion

The Data Mining system uses the Wrapper Model based on Truncated Singular Value Decomposition to reduce the noise in the representation of the original matrix. The clustering model chosen in our system does not depend on the order of on-line presentation (plasticity-elasticity). Thus, it eliminates the laborious process of knowledge engineering involved in the process of knowledge acquisition. The convergence speed of our clustering model is based on typical initializations. This initialization scheme reduces the computation time and improves the convergence speed to achieve the neighbourhood vicinity of the response. We choose to use the method of on-line learning to avoid storing the complete data set. In order identify the correspondence between descriptive labels and ontology artefacts; we used an alignment process based on Dynamic Time Warping similarity. This alignment process gives the best alignment, but leads to very slow execution time. Hence, the choice of a distance measure is a very significant decision in an alignment application. To support the maximum expressiveness while retaining computational completeness and decidability, we used OWL-DL to encode the updated ontology.

References

[1] O Alter, P O Brown, and D Botstein. Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences, 97(18):10101-10106, 2000.
[2] M Baena-Garcia, J M Carmona-Cejudo, G Castillo,and R Morales-Bueno. TF-SIDF: Term frequency, sketched inverse document frequency. In Intelligent Systems Design and Applications (ISDA) 2011 pages 1044-1049.
[3] S Bechhofer, R Moller, and P Crowther. The DIG description logic interface: DIG/1.1. 2003.
[4] A Bellandi, S Nasoni, A Tommasi, and C Zavattari. Ontology-Driven Relation Extraction by Pattern Discovery. In Information, Process, and Knowledge Management, 2010. eKNOW '10. Second International Conference on, pages 1-6, 2010.
[5] Mounir Boukhadoum. Introduction to the information processing by neural networks, COURS, DIC UQAM, 2010.
[6] Kuo Chih-Hao and M Moghaddam. Scattering From Multilayer Rough Surfaces Based on the Extended Boundary Condition Method and Truncated Singular Value Decomposition. Antennas and Propagation, IEEE Trans. on, 54(10):2917-2929, 2006.
[7] Cheng Gang, Wang Fei, Lv Haiyang, and Zhang Yinling. A new matching algorithm for Chinese placenames. In Geoinformatics, 2011 19th International Conference on, pages 1-4, 2011.
[8] M Georgiopoulos, I Dagher, Properties of learning of a fuzzy ART variant. Neural networks, 12(6):837-850, 1999.
[9] A Gomez-Perez and D Manzano-Macho. A survey of ontology learning methods and techniques. OntoWeb Deli, 1:5, 2003.
[10] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. International journal of human computer studies, 43(5):907-928, 1995.
[11] V Haarslev, K Hidde, R M□oller, and M Wessel. The RacerPro knowledge representation and reasoning system. Semantic Web, 2011.
[12] H Isawa, H Matsushita and Y Nishio. Fuzzy Adaptive Resonance Theory Combining Overlapped Category in consideration of connections. In Neural Networks, 2008. IJCNN 2008. IEEE International Joint Conference on, pages 3595,3600, 2008.
[13] B Issac and W Jap. Implementing spam detection using Bayesian and Porter Stemmer keyword stripping approaches. In TENCON 2009 - 2009 IEEE Region 10 Conference, pages 1,5, 2009.
[14] W Jiannan, L Guoliang and F Jianhua. Fast-join: An efficient method for fuzzy token matching based string similarity join. In Data Engineering (ICDE), 2011 IEEE 27th International Conference on, pages 458-469, 2011.
[15] Ning Liu, Guanyu Li, and Li Sun. Using Formal Concept Analysis for Maritime Ontology Building. In Information Technology and Applications (IFITA), 2010 International Forum on, volume 2, pages 159-162, 2010.
[16] Tseng Ming-Cheng, Lin Wen-Yang, and Jeng Rong. Incrememtal Maintenance of Ontology-Exploiting Association Rules. In Machine Learning and Cybernetics, 2007 International Conference on, volume 4, pages 2280-2285, 2007.
[17] R R Starr and J M P de Oliveira. Conceptual Maps as the First Step in an Ontology Construction Method. In Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International, pages 199-206, 2010.
[18] Shen Yanfen. A formal ontology for Data Mining : principles, design and evolution: thesis presented at UQTR 2007.
[19] Lv Yanhui. An approach to ontologies integration. In Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on, volume 2, pages 1262-1266, 2011.
[20] A N K Zaman, P Matsakis, and C Brown. Evaluation of stop word lists in text retrieval using Latent Semantic Indexing. In Digital Information Management (ICDIM), 2011 Sixth International Conference on, pages 133-136, 2011.
[21] J Zhao, G Y Wang, Z F Wu, The study on technologies for feature selection. Volume 2, page 689-693 vol. 2. IEEE, 2002.
[22] E Gomez-Sanchez, Y A Dimitriadis, J M Cano-Izquierdo, and J Lopez-Coronado. Safe ARTMAP: a new solution for reducing category proliferation in Fuzzy ARTMAP. In Neural Networks, 2001. Proceedings. IJCNN'01 volume 2, p 1197-1202. IEEE, 2001.
[23] Choukri Djellali. A New Digital Conceptual Model Oriented Corporate Memory Constructing: Taking Data Mining Models as a Case. ANT/SEIT: 977-983, 2013.