2009 First International Conference on Advances in Databases, Knowledge, and Data Applications

Visualization and Integration of Databases using Self-Organizing Map

Farid Bourennani, Ken Q. Pu, Ying Zhu
University of Ontario Institute of Technology
{farid.bourennani, ken.pu, ying.zhu}@uoit.ca

***Abstract*—With the growing computer networks, accessible data is becoming increasingly distributed. Understanding and integrating remote and unfamiliar data sources are important data management issues. In this paper, we propose to utilize self-organizing maps (SOM) clustering to aid with the visualization of similar columns, and integration of relational database tables and attributes based on the content. In order to accommodate heterogeneous data types found in relational databases, we extended the TFIDF measure to handle, in addition to text, numerical attribute types for coincident meaning extraction. We present a SOM clustering based visualization algorithm allowing the user to browse the heterogeneously typed database attributes and discover semantically similar clusters. Additionally, we propose a new algorithm Common Item Based Classifier (CIBC) to smoothen the homogeneity of the clusters obtained by SOM. The discovered semantic clusters can significantly aid in manual or automated constructions of data integrity constraints in data cleaning or schema mappings in data integration.**

***Index Terms*—SOM, Common Item Based Classifier (CIBC), Data Integration, Information Retrieval (IR)**

I. Introduction

THE vast availability of computer networks has highlighted the many challenge of integration of distributed databases. Consider the following scenarios.

- For many modern data warehouses, the data is stored distributedly across multiple data centers and maintained by different database servers. In these cases, globally enforcing data integrity across different data centers can be challenging and possibly impossible. Identifying similar or semantically equivalent attributes over different databases is essential toward defining data integrity constraints (such as foreign-key constraints).
- Consolidation of previously distinct databases is a commonly occurring event. Semantically equivalent database attributes are often found in multiple databases. Due to different languages and business practices, these semantically equivalent database attributes may have been named differently. In order to maintain a high level of data integrity of the consolidated database, it is important to identify these equivalent attributes.
- By joining local data with remote databases (i.e. public DB), the database application can significantly increase the data richness. There are two issues with making use of remote data sources: (1) discovery of relevant data sources, and (2) performing the proper joins between the local data source and the relevant remote databases. Both can be solved if one can effectively identify semantically related attributes between the local data source and the available remote data sources.

It is evident that the identification of closely related database attributes is an important problem central to the integration of databases. Many automatic schema mapping techniques have been proposed by researchers [1], [2]. However, due to the fundamental nature of the problem, it is inevitable that mistakes will be made. For mission critical applications, mistakes in the schema mapping between databases may result in serious errors. In this paper, we are motivated to look at visualization techniques to aid the end-user to identify semantically similar data attributes in a semi-automated fashion. The goal is to present a visual presentation of the clusters of semantically similar attributes. The visualization should be intuitive and scalable to large databases.

In this paper we have developed a visualization technique based on self-organizing maps (SOM)[3]. SOM benefits from fast training algorithms (i.e. batch training [3], [4] with sampling), and is easy to visualize due to its inherit low dimensional regular grid layout. SOM has been applied in numerous work in visualization of text corpus [4]. Since our data is from general databases, in addition to text data, we would like also to handle numerical data.

Contribution of the paper

- We demonstrate that's possible using SOM to meet the challenge of extracting *Coincident Meaning* from *Heterogeneous* textual and numerical data types, by extending the traditional text analysis from information retrieval.
- We propose a new algorithm, named *Common Itemset Based Classifier (CIBC)*, that enhances the results obtained from SOM's trained map. It improves the precision and heterogeneity of clusters and helps differentiating visually between heterogenous and homogenous data clusters.
- We show that a SOM-based visualization technique can aid end-users to perform data integration and schema mapping in a semi-automated fashion. This allows the user to have a clear understanding of the relationship among attributes from different and unfamiliar databases.

Outline of the paper

The structure of the rest of the paper is as follows. Section II and III will discuss Pre-Processing and the Processing phase respectively. The section IV is devoted to experiments while section V concludes the present paper.

II. Preprocessing

In order to implement any classification technique, it is necessary to transform the input documents into an algebraic model so they can be processed.

978-0-7695-3550-0/09 $25.00 © 2009 Crown Copyright
DOI 10.1109/DBKDA.2009.30

The standard practice in information retrieval is the usage of the vector space model (VSM) to represent textual documents [8]. Documents are symbolized in t-dimensional Euclidean space where each dimension corresponds to word (term) of the vocabulary [14]. Despite its simplicity and efficiency, the VSM has the disadvantage of focusing only on textual data type, and, it's hard to obtain accurate information of semantic relatedness automatically from textual information only [4]. And, more complex is the extraction of coincident meaning from heterogenous data. [12]

Therefore, our approach proposes to pre-process other data type such as numerical data separately, then, combine textual with numerical mining by *unified vectorization* for better concurrent semantic clustering results as shown on fig. 1.

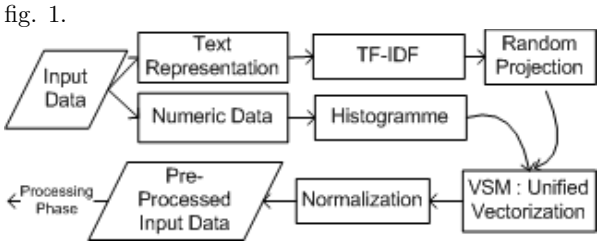

Fig. 1. Preprocessing Phase

A. Treatment of Textual Data

All textual portions of the columns x_i are transformed into a vector: $x_i = (w_{1j}, w_{2j}, ...w_{|T|j})$, where $|T|$ is the number of terms in whole set of terms T, and w_{kj} represents the the weight of the term t_k in the document d_j.

Several text representations are mentionned in the litterature, the most important one are: Bag of Words [7], N-gram[9], Stemming and lemmatization [7]. In this research "bag of words" and N-gram text representations are used. Firstly, the most common text representation within VSM framework is "bag of words" [7] [8]. Secondly, N-gram has been chosen because it offers several advantages; it's an easy and fast way to solve the syntax related issues such as misspeling, and, it finds common patterns between words with the same roots but different morphological forms (e.g., finance and financial), without treating them as equal, which happens with word stemming[9]. As a reminding, N-gram is a substring of n consecutive characters generated for a given document by mainly displacing a window of n characters along the text.

A.1 TF-IDF Weighting

Most Approaches[7][11] are centered on a vectorial representation of texts using Term Frequency - Inverse Term Frequency(TF-IDF) measure, defined as:

$$\text{TF-IDF}(t_k, d_j) = \frac{Freq(t_k, d_j)}{\sum_k Freq(t_k, d_j)} \times Log \frac{N_{\text{doc}}}{N_{\text{doc}}(t_k)}$$

where $Freq(t_k, d_j)$ denotes the number of times the term t_k occurs in the document (column) d_j, $\sum_k Freq(t_k, d_j)$ is the total number of all the term occurrences in the same document d_j, and N_{doc} is the total number of documents in the corpus, while $N_{\text{doc}}(t_k)$ is the number of documents in the corpus with therm t_k.

A.2 Dimensionality reduction

In textual application context, the high dimensionality is due to the large vocabulary that leads to burdensome computations and even restricts the choice of data processing methods. A statistical optimal of dimensionality reduction is to project the data onto a lower-dimensional orthogonal subspace that captures as much of the variation of the data as possible. The most widely used way to do this is Principal Component Analysis (PCA), however, it is computationally expensive and is not feasible on large, highdimensional data[15]. Therefore, another powerful technique that solves these problems is the Random Projection (RP) which is simple, offers clear computational advantages and preserves similarity[15]: Given a matrix X, the dimensionality of the data can be reduced by projecting it through the origin onto a lower-dimensional subspace, formed by a set of random vectors :

$$A_{[k \times n]} = R_{[k \times m]} \bullet X_{[m \times n]}$$

The k in the subscripts is the desired, reduced dimensionality.

Note, RP was successfully tested with SOM on text and image data type and it appeared to be a good alternative to traditional methods of dimensionality reduction.[15]

B. Treatment of Numerical Data:

Because of the different nature of data, the numerical data is preprocessed differently from the textual one. As an illustration let's have two numbers 1988 and 1991, representing years of birth or financial values, present in two columns (documents). Their proximity won't be detected by using traditional textual representations such as bag of words or n-gram because they don't possess enough textual similarities. That's why it is essential to preprocess the numeric input data differently so that their value reflects their semantic similarity.

Several techniques are mentioned in the literature to specify concept hierarchies for numerical attributes such as benning, histogram analysis, entropy-based discretization, Z2-merging, cluster analysis and discretization by intuitive partitioning[10].

In this research, Histogram analysis [10] is used to ease the SOM neural network's learning process and improve the quality of the map. More precisely Equal-Frequency (Equal-Depth) Histogram [10] was used because of its good scaling properties and simplicity to implement. The values were partitioned so that, ideally, each partition contains identical number of tuples. Another good reason for using histogram is that it reduces the dimensionality of the numerical portion of the VSM by s times, where s is the size of the bin. However, sometimes the size of the bins can be bigger than s in order to avoid cutting a cluster of the same value in the middle because of trying to respect the bin size. All the numerical data n_i of the document d_j are transformed into a vector: $n_i = (v_1 j, v_2 j, ...v_{|N|j})$

where N is the total number of histogram bins, and v_{lj} represents the number of observations that fall into various disjoint bin b_l as shown on fig. 2.

C. Combination of textual and numerical mining by Unified Vectorization

Now that the textual and numerical data have been vectorized and the dimensionality reduced, is proposed the unified vectorization of the numerical and the textual data in order to be processed simultaneously and meet the challenge of extracting coincident meaning out of this *heterogeneous* data. However, the unified VSM should be normalized in order to avoid any unjustified influence of one the two data type during the SOM training phase.

Docs	Terms				Bins			
	t_1	t_2	...	t_n	b_1	b_2	...	b_l
d_1	w_{11}	w_{12}	...	w_{1n}	v_{11}	v_{12}	...	v_{1k}
d_2	w_{21}	w_{22}	...	w_{2n}	v_{21}	v_{22}	...	v_{2k}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
d_m	w_{m1}	w_{m2}	...	w_{mn}	v_{m1}	v_{m2}	...	v_{mk}

Fig. 2 Unified Vectorization of Textual & Numerical Data

D. Normalization

The data does not necessarily have to be preprocessed at all before creating SOM and using it. However, in most real tasks preprocessing is important; perhaps even the most important part of the whole process[13]. All the values of the unified VSM matrix were normalized similarly in a range of [0,1] through a linear operation.

III. Processing

Unsupervised classification or "clustering" is one of the fundamental data mining techniques. Furthermore, Self Organizing Map (SOM) is an unsupervised learning neural network that produce a topologically clusters mapping on a plan (2D) more convenient to visualize.

A. Self Organizing Map

Self Organizing Map (SOM) of Kohonen is an unsupervised learning method which is based on the principle of competition according to an iterative process of updates[6]. SOM has two training modes that are mentioned in the literature: sequential and batch version. They differ basically in the method of updating weight vectors. The advantages of batch method over the sequential version are: a) it produces a map much faster and b) it does not need a learning rate to converge. More details can be found in [3][4].

B. SOM Visualization

The easiest way to visualize the clustered documents (columns) is to match every column d_j to its respective Best Matching Unit (BMU) node[3] on the trained SOM's map. This will result in having semantically similar documents clustered on the same Map's node, let's call that cluster Cl_i. Even if the results were satisfactory, it has been observed as shown in fig. 3 that: (1) some nodes were too big because of grouping together multiple *heterogenous* clusters, and (2) some documents were not matched to their best class. In other words, the map is unbalanced because of too many nodes are empty and paradoxically some other nodes are overloaded.

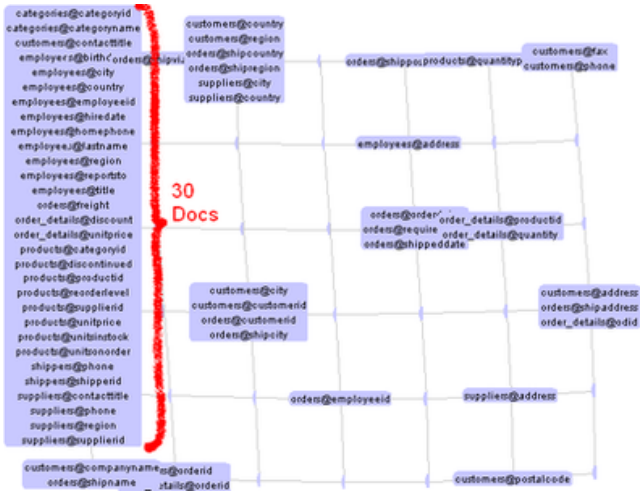

Fig. 3. Trained SOM map

C. Common Item Based Classification

As a solution to the two previous issues, a new algorithm called *Common Itemset Based Classifier (CIBC)* is proposed in order to *smoothen* the clusters obtained in the previous section and make the visual presentation clearer.

Firstly, CIBC refines the clusters by validating the homogeneity of every cluster Cl_i and re-cluster them into heterogenic sub-clusters, when necessary, by preserving their topological closeness. Secondly, the algorithm finds for every un-clustered column a possible matching cluster Cl_i. Finally, it distinguishes visually on the map the clusters with homogenous data from clusters with heterogenous data. More details can be found in [5].

To illustrate the CIBC algorithm, suppose that we have a normalized term-document input VSM matrix, similar to fig. 2, that has been preprocessed and then processed through the Batch version of SOM. The next step is to tune up the trained map using CIBC algorithm as follow:

Phase 1: Forming Itemsets

Let's assume we are given a set of document items D. Itemset $I_{t_k} \subseteq D$ is some subset of *similar* documents d_j (at least 2) based on the common term t_k:

$$I_{t_k} = \{d_j \in D \mid w_{jk} > 0 \wedge |I_{t_k}| \geq 2\} \quad (1)$$

At this point, some itemsets contain a high number of similar documents because of stop words like "the" for example or some other type of noise. Therefore in order to eliminate these insignificant Itemsets, the number of similar documents in every itemset $|I_{t_k}|$ should be smaller than a certain threshold called Max_I:

$$\frac{|I_{t_k}|}{|D|} < Max_I \text{ where } 0 < Max_I < 1 \quad (2)$$

PHASE 2: SOM'S CLUSTERS HOMOGENEITY VALIDATION
As shown in the fig. 3 some cluster Cl_i, obtained from SOM's visualization phase, are not heterogeneous and as a consequence some nodes are overloaded by them and many other ones are complectly empty. Therefore, the heteregoneity of every cluster Cl_i should be validated.

First, for every cluster Cl_i has to be found all the itemsets I_{t_k} having at least two documents in common. Then, Should be kept only the intersection of the two subsets $Cl_i \bigcap I_{t_k}$, named: I_{Cl_i,t_k} . Let's call all the identical itemset I_{Cl_i,t_k} : $I_{Cl_i,n}$ where $n \in \mathbb{N}$.

Secondly, among all the itemset $I_{Cl_i,n}$ should be found the one having the *biggest* number of documents and let's called it: Cl'_i. However, Cl'_i should respect the following condition in order to keep only the semantically related documents :

$$\frac{|T_{Cl'_k}|}{|T_{Max(d_j,Cl'_k)}|} > \alpha, \ (1)$$

Where $|T_{Cl'_k}|$ is the size of the vocabulary of the subcluster Cl'_k, $|T_{Max(d_j,Cl'_k)}|$ is the vocabulary size of the document from Cl'_k having the richest vocabulary and α (usually equal to 0.05) is threshold to keep only strongly related documents of the current sub-cluster Cl'_k.

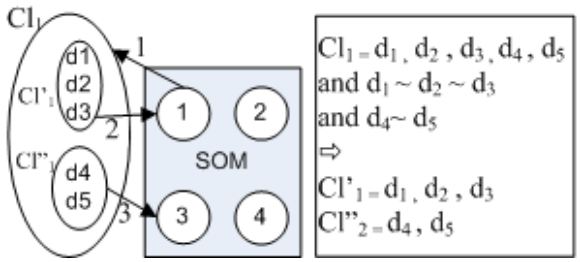

Fig. 4. Phase 2: SOM's Clusters homogeneity validation

As shown on fig. 4, Cl'_k is kept on the current node (BMU) while the remaining documents $\overline{Cl'_k}$ are reprocessed until no homogeny sub-cluster can be found. In case there is another existing sub-cluster(s) $Cl"_k$, it should be moved to a another empty node (with no clusters) within the neighborhood.

PHASE 3: CLUSTERING UNCLUSTERED DOCUMENTS
For every un-clustered column (document) should be found, respecting the rule (1), the best matching cluster if exists. Otherwise as last resort, following the same process should be formed new clusters among the un-clustered documents.

PHASE 4: REMAPPING THE UNCLUSTERED DOCUMENTS
At this point, only left a certain number of semantically unique documents for which neither a matching cluster neither another similar document could be found. Therefore to show vizually their unicity, these documents are reassigned to their first respective *empty* BMU. In case there is no available node, then, they should be mapped to the first BMU having unclusterd documents, in the consequence of which; will be formed new clusters of Un-clustered documents UCl_k.

D. Visualization Update

The map is updated with the redistribution of homogenous clusters of columns as well as the un-clustered ones. As an illustration, the map shown on fig. 6 is the map on fig. 3 after applying the CIBC proposed algorithm to it. In case there is hetergenous clusters UCl_k, the clusters Cl_k with homogenous data, formed in phase 2 and 3, will be differentiated visually by distinguished colors.

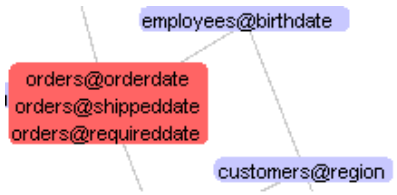

Fig. 5. Clusters visualization: table@column

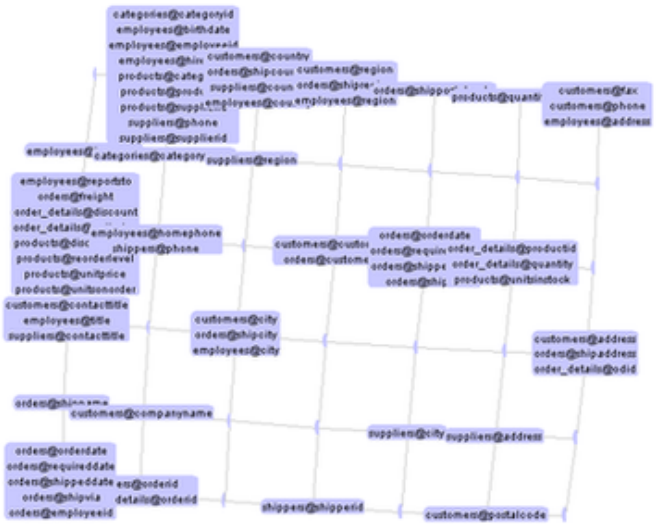

Fig. 6. Updated MAP

IV. EXPERIMENTS

A. Corpus

The proposed algorithms were tested on the demo database available online, named: Northwind. As show on fig. 7, The database has a 77 tables with 4681 terms as vocabulary size, which permits to test the algorithms with and without dimensionality reduction and compare results.

Data Set	Columns	terms	text	num. terms	Cat.
Northwind	77	4681	2154	2527	15

Fig. 7. Unified Vectorization of Textual and Numerical Data

B. Tokenization

The process of breaking a text up into its constituent tokens is known as tokenization. Because we don't make any lingustic pretreatment, i.e., we do not need to apply lemmatization, stemming or stop words elimination the influence of tokenization on the results increases. It's not the

purpose of this paper to evaluate the impact of tokenization, but, it is important to mention some important facts. For example, when using bag of words as method of representation, if the non alphanumeric characters, such as "()-+;.", are not eliminated the results could be affected; The F-measure can drop when using the SOM algorithm up to 15%, and it's even worst when using the hybrid SOM and CIBC algorithm (- 25%). Therefore, all the non alphanumeric characters are eliminated for the experiments.

C. Test Objective

The objective of the tests is to evaluate and at the same time compare the performance of the algorithms of SOM versus the hybrid proposed algorithm (SOM with CIBC). One of the main focus is the homogeneity level of the clusters; because that's the main reason of having developed the CIBC algorithm. Another experimental interest is to select the best text representation, among those proposed in section 2, in order to find the optimal clustering result.

D. Evaluation Measures

One of the most used performance evaluation of unsupervised classifiers in the IR literature, with respect to the known classes for each document, are F-measure and Entropy which are based on Precision and Recall [7]:

$$P = Precision(i,j) = \frac{N_{ij}}{N_j},\ R = Recall(i,j) = \frac{N_{ij}}{N_i}$$

where N_{ij} is the number of members of class i and cluster j, N_j is the number of members of cluster j, and N_i is the number of members of class i.

F-measure distinguishes the correct classification of labels within different classes. In essence, it assesses the effectiveness of the algorithm on a single class and the higher it is, the better is the clustering. It's defined as follow:

$$F(i) = \frac{2PR}{P+R} \Longrightarrow F_c = \frac{\sum_i (|i| \times F(i))}{\sum_i |i|}$$

where; for every class i is associated cluster j which has the highest F-measure, F_c represents the overall F-measure that is the weighted average of the F-measure for each class i, $|i|$ is the size of the class i.

Entropy reflects the homogeneity level of the clusters. The lower is the value of Entropy, the better is the quality of homogeneity of the cluster and vice versa. Subsequently, for every cluster j in the clustering result C we compute the entropy, then, the entropies are summed:

$$E_c = \sum_{j=1}^{N_C} \frac{N_j}{N} \times \left(-\sum \text{Precision(i,j)} \times log\ \text{Precision(i,j)} \right)$$

where N_j is the size of cluster j, and N is the total number of columns (documents).

E. Evaluation

The evaluation of the relevance of the classes formed remains an open problem because of the subjective nature of the task. [7] There are often various relevant groupings for the same data set. For instance, when comparing quantity entries (e.g. 12 05) with a date entry (e.g. 12-02/2005) the data are similar numerically but it does not fit within the purpose of this research which is the integration and the visualization of semantically similar columns. Therefore, these kind of data were classified in separated classes and our hybrid algorithm was significantly penalized (up to 25%) in this sense, but, on the other hand it creates future search perspectives and challenges that are also closer to the industrial needs.

F. Preliminary tests: Without Dimension Reduction

A good classification requires a good presentation[7]. However, the vast number of text representation possibilities presented earlier requires to select the most significant ones to continue further our tests. In this sense firstly will be tested on Northwind DB without dimensionality reduction, different combination of tokenization and vectorization as show in the table (fig 8). Accordingly, there is two tokenization methods: bag of words versus N-gram described earlier. Besides, there is three vectorization techniques: tfidf (text) with histogram (numeric), binary(text) and histogram(numeric), and tfidf (text and numeric) where the numbers are considered as text terms.

Data Set	Classifiers	tfidf + hist.	Bin. + Hist.	tfidf
SOM	Bag of words	58.24	49.85	54.37
Hybrid	Bag of words	**87.64**	74.18	65.07
SOM	Ngram	51.55	45.12	51.26
Hybrid	Ngram	59.75	52.34	60.18

Fig. 8. Preliminary F-measure with different representations

The preliminary tests (fig. 8 - 9) show an evident performance advance of the hybrid algorithm over pure SOM: there is improvement of the quality of clustering by [7.22-29.4]% of F-Measure and [7.93-20.39]% of Entropy enhancement which reflects the homogeneity of clusters. The best results of the hybrid algorithm are when using bag of words as tokenizer, combined with the proposed tfidf and histogram as vectorization method. However, N-gram (3-gram) tokenization does not improve the results when used with proposed unified vectorization.

Data Set	Classifiers	tfidf + hist.	Bin. + Hist.	tfidf
SOM	Bag of words	23.72	26.90	23.65
Hybrid	Bag of words	**8.44**	14.57	**3.26**
SOM	Ngram	28.33	27.83	27.08
Hybrid	Ngram	14.76	19.90	**10.88**

Fig. 9. Preliminary Entropy measure with different representations

G. Tests: With Dimensionality Reduction

Dimensionality reduction was applied for the tests resumed in figures 10 and 11. The size of textual data vocab-

ulary for all vectorization type was originally 2154 terms, then, the dimension was reduced to 1000.

From the results, we can see that the best performance in general is the usage of the proposed Hybrid algorithm of SOM and CIBC. First of all as show on fig.11, CIBC improves the classification precision of SOM by [4.84 - 23.78]% (F-Measure). Secondly, we can see that the proposed integration technique of numerical and textual data works very well, particularly, with bags of words tokenization. Finally, the most important remak is when using the proposed hybrid algorithm (SOM and CIBC) is used with the suggested integration technique (tfidf with histogram), the quality of clusters homogeneity is impressive; sometime even faultlessly as shown on fig. 12. In fact, the clusters homogeinity is improved by [17.49-30.25]% according to the entropy measure.

Another interesting remark, is that the proposed unified vectorization technique (tfidf + histogram) performs better when using bag of words tokenization for textual data rather than N-Gram. This is valid for both processing algorithms. However, N-Gram performs better with the traditional vectorization techniques.

	SOM		Hybrid (SOM + CIBC)	
Tokenization	tfidf + hist.	tfidf	tfidf + hist.	tfidf
Bag of words	54.03	52.93	**71.42**	59.21
3-Gram	51.63	43.83	66.66	56.47
4-Gram	42.88	62.02	66.66	68.28
5-Gram	46.00	58.04	61.00	62.88

Fig. 10. Comparison of the F-score values

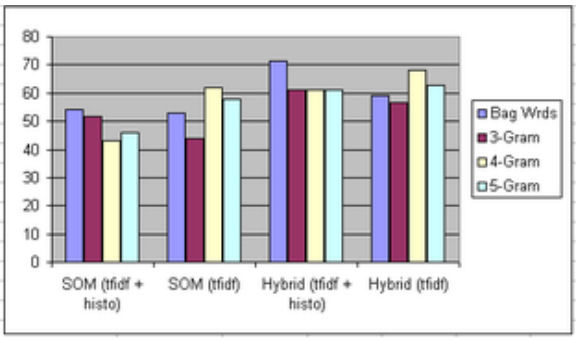

Fig. 11. F-measure comparison depending on tokenization, vectorization and classification algorithm

	SOM		Hybrid (SOM + CIBC)	
Tokenization	tfidf + hist.	tfidf	tfidf + hist.	tfidf
Bag of words	22.44	27.09	1.05	5.86
3-Gram	27.26	32.34	**0.00**	9.77
4-Gram	30.25	21.55	**0.00**	4.06
5-Gram	28.59	25.55	0.90	2.85

Fig. 12. Comparison of the Entropy values

V. Conclusion

In this paper we have presented the concept of integration of unfamiliar heterogenous textual and numerical data, as well as their Automatic Unsupervised Classification and Visualization thraugh the Usage of SOM and its combination with the algorithm CIBC.

We have proposed a new approach to integrate and combine textual with numerical mining by unified vectorization, which, resulted in extracting *meaningful* results. We proposed a database visualization tool that exposes the similarity between columns based on their semantical content and group the homogenous columns which greatly serves the purpose of distributed database integration. This tool is applicable to data integration over web data sources, and also, to tuples classification based on the content. Finally, we proposed a new algorithm named *Common Item Based Classifier* that complements beautifully SOM by improving its precision and enhancing significantly the clusters homogeneity formed by trained SOM map.

In our future work we want to test the proposed application on real cases and consider integrating other data type such as multimedia data type.

References

[1] Miller, R., Haas, L. M., A. Hernández, M., Schema Mapping as Query Discovery, VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases, 2000, 77-88, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[2] Erhard Rahm and Philip A. Bernsteinp, A survey of approaches to automatic schema matching, The VLDB Journal, Vol 10, n. 4, p 334-350, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[3] Kohonen, T., Self-Organizing Maps, Springer-Verlag, Berlin, 2001.

[4] K. Lagus, S. Kaski, and T. Kohonen, Mining massive document collections by the WEBSOM method, Information Sciences, vol. 163, pp. 135-156, 2004.

[5] Bourennani, F., Heterogeneous Data Type Classification using Self Organizing Map and Common Itemset Based Classifier, Master's thesis, Department of Elctrical and Computer Engineering, University of Ontario Institute of Technology, 2009.

[6] Song, M., Wu, YF (eds.), Handbook of Research on Text and Web Mining Technologies, Idea Group Inc, USA (2008)

[7] Amine, A., Elberrichi1, Z., Bellatreche, L., Si- Monet, M. Malki, M., Concept-based clustering of textual documents using SOM, Concept-based clustering of textual documents using SOM, 6th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'2008), Doha, Quatar, 2008

[8] R. Baeza-Yates and R. Ribeiro-Neto, eds., Modern Information Retrieval. Addison Wesley Longman, 1999.

[9] Y. Miao, V. Keelj, and E. Milios. Document clustering using character n-grams: A comparative evaluation with term-based and word-based clustering. In Proceedings of the 14th ACM international conference on Information and knowledge management, (Bremen, Germany 2005).

[10] Han, J., Kamber, M., Data Mining, Second Edition: Concepts and Techniques, Morgan Kaufmann, 2006, pp. 72-97

[11] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 147, 2002.

[12] Back, B., Toivonen, J., Vanharanta, H., Visa, A., 2001. Comparing numerical data and text information from annual reports using self-organizing maps. International Journal of Accounting Information Systems 2(4): 249269.

[13] Pyle, D., Data Preparation for Data Mining, Morgan Kaufman Publishers, San Francisco, 1999.

[14] Salton, G., Automatic Text Processing, Addison-Wesley, 1989.

[15] Fradkin, D., Madigan, D., Experiments with Random Projections for Machine Learning, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, D.C, USA, 2003, pp. 517 - 522