Clustering Relational Database Entities using K-means

Farid Bourennani, Mouhcine Guennoun, Ying Zhu
University of Ontario Institute of Technology, ON, Canada
{farid.bourennani, mouhcine.guennoun, ying.zhu}@uoit.ca

***Abstract*—The fast evolution of hardware and the internet made large volumes of data more accessible. This data is composed of heterogeneous data types such as text, numbers, multimedia, and others. Non-overlapping research communities work on processing homogeneous data types. Nevertheless, from the user perspective, these heterogeneous data types should behave and be accessed in a similar fashion. Processing heterogeneous data types, which is Heterogeneous Data Mining (HDM), is a complex task. However, the HDM by Unified Vectorization (HDM-UV) seems to be an appropriate solution for this problem because it permits to process the heterogeneous data types simultaneously. In this paper, we use K-means and Self-Organizing Maps for simultaneously processing textual and numerical data types by UV. We evaluate how the HDM-UV improves the clustering results of these two algorithms (SOM, K-means) by comparing them to the traditional homogeneous data processing. Furthermore, we compare the clustering results of the two algorithms applied to a data integration problem.**

***Index Terms*—K-means, SOM, Pre-Processing, Data Integration, Heterogeneous data mining.**

I. Introduction

There is much interest from the industry in heterogeneous data mining. This interest seems to be proportional to the heterogeneity of the data, i.e., the more heterogeneous the data types are, and the greater the interest there is in automatically processing it [1]. This is likely due to the availability and abundance of such data. For example, data integration sectors are interested to use heterogeneous data mining in order to detect similar data entities for building data warehouses [2]. Business intelligence sectors and financial institutions are extremely eager to extract coincident clustering from textual and numerical (financial specially) data based on the content. Therefore, several business - and finance related - research projects worked on mining quantitative (numerical) and qualitative (textual) data for comparing companies [3], [4], [5]. Numerical data included the company financial information, and the textual information was business reports from external financial analysts. Some projects try to use heterogeneous data mining for providing tools at the level of strategic decision-making [6], while others use the same logic for automated project management evaluation and automated problem detection [7]. In addition, there have been studies in using data mining for heterogeneous data mining in medical and biomedical fields to classify for example textual and genetic data [8]. In brief, heterogeneous data mining is of great importance in many industrial fields.

Heterogeneous Data Mining by Unified Vectorization (HDM-UV) appeared to be successful when applied to textual and numerical data using Self Organizing-Map (SOM) for data integration applications [1]. Furthermore, when the heterogeneous data types are converted into algebraic models using similar weighting measures, the clustering results are enhanced when SOM is used [2].

In this paper, we validate the fact that simultaneously processing heterogeneous data types is feasible with algorithms other than SOM. Therefore, K-means [10] - a supervised clustering method - is proposed for HDM-UV.

Contribution of the paper

- We demonstrate that HDM-UV is suitable for the k-means clustering method.
- We compare clustering results between SOM and K-means on a new data set, which is a combination of two different databases, for data integration purposes.
- We demonstrate by focusing on the pre-processing phase, i.e. using similar weighting measures of heterogenous data types for building a unified algebraic model, that the k-means clustering results are improved when using HDM-UV.

Outline of the paper

The structure of the rest of the paper is as follows: the Sections II and III discuss the Pre-Processing and the Processing phases respectively. The Section IV is devoted to the experiments, while the Section V concludes the present paper.

II. Pre-processing

In order to implement any clustering technique, it is necessary to transform the input documents into an algebraic model so they can be processed. In this paper, the documents are relational database entities. More precisely, these entities are tables' *columns* in a relational database model. The content of every column is extracted into text file in order to be clustered.

The standard practice in information retrieval is the use of the vector space model (VSM) to represent textual documents [12]. It appears that when two data types, such as numerical and textual, are simultaneously processed by HDM, the use of *unified vectorization (UV)* leads to better convergent semantic clustering results [1]. In spite of good results, the development and the use of similar data weighting measures to represent these heterogeneous data types in a unified VSM matrix improves the clustering results [2]. Let us examine these pre-processing steps.

A. Text Weighting Measure

Most approaches are centered on a vectorial representation of texts using Term Frequency - Inverse Term Frequency(TF-IDF) measure [14], 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 having the term t_k.

B. Numerical Data Weighting Measure

An equivalent representation of TF-IDF for numerical data was introduced for better HDM-UV results. This representation is called Bin Frequency - Inverse Document Bin Frequency (BF-IDBF). It is a combination of histogram and TF-IDF measures. It is important to mention that specifically Equal-Frequency (Equal-Depth) Histogram was used to build the BF-IDBF [13]. The BF-IDBF is defined as follows:

$$\text{BF-IDBF}(b_l, d_j) = \frac{Freq(b_l, d_j)}{\sum_k Freq(b_l, d_j)} \times Log \frac{N_{\text{doc}}}{N_{\text{doc}}(b_l)}$$

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

C. Tokenization of the Data:

As explained previously, the input documents are represented by the VSM matrix. Before building the VSM, these documents have to be tokenized using simple syntactic rules such as white space. For numerical data it is straight forward, every number is a token. For texts, these terms(tokens) separated by spaces are called Bag of Words [12]. Another popular text tokenization method is N-gram. N-gram is a substring of n consecutive characters of a term. For example, the 3-grams of the term "hello" are: "hel", "ell", and "llo". More details regarding the tokenization and the vectorization processes can be found in [2].

D. Data Combination by Unified Vectorization

Once the numerical and textual data is vectorized, the unified VSM model is built by UV as shown in Tab. 1. n is the number of terms t_k in the corpus, m is the number of documents d_j in the corpus, and p is the number of bins(buckets) b_l in the corpus.

It is important to mention that if the data dimension is too large to be processed, then the dimension would be reduced. In addition, every data type's algebraic model is normalized separately before the UV operation. After combining the data, the unified VSM matrix is normalized a second time using the same linear operation.

Tab. 1 Unified Vectorization of Textual & Numerical Data

Docs	Terms				Bins			
	t_1	t_2	…	t_n	b_1	b_2	…	b_p
d_1	w_{11}	w_{12}	…	w_{1n}	v_{11}	v_{12}	…	v_{1p}
d_2	w_{21}	w_{22}	…	w_{2n}	v_{21}	v_{22}	…	v_{2p}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
d_m	w_{m1}	w_{m2}	…	w_{mn}	v_{m1}	v_{m2}	…	v_{mp}

E. Dimensionality reduction

The high dimensionality of the input data is due to the large vocabulary and large amount of numbers which leads to burdensome computations and even restricts the choice of data processing methods. The most widely used way to achieve this dimensionality reduction is the use of Principal Component Analysis (PCA); however, it is computationally costly and it is not feasible on high-dimensional data [16]. Therefore, we adopt another dimensional reduction method Random Projection (RP). RP is simple, fast, and preserves similarity [16]. RP seems to perform better on textual data when the reduced dimensionality is greater than 600. The difference between the average error of RP and SVD (PCA performed directly on the data matrix) is less than 0.025 with 95% confidence interval [15].

RP is applied to the text and numerical VSMs separately, then the two reduced matrixes are combined by UV for processing. RP is defined as follows [16]:

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

where the $R_{[k \times m]}$ is a random matrix, n is the desired reduced dimension, $X_{[k \times m]}$ is the original VSM matrix, and $A_{[k \times n]}$ is the reduced matrix.

III. Processing

Unsupervised and supervised clustering are among the fundamental data mining techniques. The user of unsupervised clustering methods does not need to specify the expected number of clusters, while he must specify the expected number of clusters with the supervised clustering methods. It has already been demonstrated that HDM-UV works well with the unsupervised clustering method SOM [1]. We are aiming to demonstrate that the HDM-UV is applicable to algorithms other than SOM such as K-means. In this paper, we experiment with applying HDM-UV on k-means which is a supervised clustering algorithm. At the same time, we are comparing K-means with SOM for data integration application on a new data set. More details can be found on K-means and SOM in [10][11].

IV. Experiments

Firstly, the experiments serve to validate that the heterogeneous data mining by Unified Vectorization using SOM on a larger data-set produces more convergent clustering results. By running the tests on the *Northila* database, which is a combination of two databases (Northwind [18] and Sakila [17]), this task would be achieved. Secondly, we would like to try a supervised algorithm because so far HDM-UV was tested only on the unsupervised clustering algorithm SOM. This way, it would be possible to re-enforce the fact that heterogeneous data mining by unified vectorization could be achieved by using most likely any supervised or unsupervised algorithm.

The Table 2 shows the number of classes, which are the expected clusters, of the Northila database. It can be observed that the majority of the classes, as well as the files (columns), are of the numerical data type.

Besides, different numerical and textual data representations are tested, i.e., different combinations of tokenizations and vectorizations are used. Accordingly, two tokenization methods for textual data are used: bag of words versus N-gram which were described earlier. Three vectorization methods are compared: TF-IDF (for texts), histogram (for numerical data) and BF-IDBF(for numerical data).

The input data dimension was reduced using RP to 1250 for textual data and 750 for numerical data for a total dimension of 2000. In other words, there is a loss of information when the data is processed by unified vectorization, but impact seems to be minor to bias the results.

A. The corpus

The proposed techniques were tested on the combination, which was slightly adjusted, of two demo databases available online, named Sakila[17] and Northwind [18]. As shown on tab. 2, the combination of these two databases is called Northila. Being larger, the Northila database permits to evaluate a real data integration operation with 165 database columns.

It is important to mention that around 60 % of the files are constituted of purely numerical data such as phone numbers, keys, dates, prices, , etc. The remaining ones are textual or a combination of the two data types.

Tab. 2 Northila Classes

Types	Text	Numeric.	AlphaNum.	Unclassed	Total
Classes	11	20	2	1	34
Columns	40	94	9	22	165

B. Clustering evaluation

The clustering evaluation algorithms is done by using the F-measure. The F-measure is a combination of Precision and Recall measures which are defined bellow. Briefly, the F-measure is calculated with respect of clusters and classes. The classes are the ideal clustering results while clusters are the obtained clustering results.

Precision assesses the predictive power of the algorithm by estimating the ratio of the true positives among the cluster.

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

where N_{ij} is the number of members of the class i which are in the cluster j, N_j is the number of members of cluster j.

Recall is a function of the correctly classified examples (true positives), and the misclassified examples (false negatives).

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

where N_i is the number of members of the class i.

F-measure distinguishes the correct classification of document 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 is defined as follows:

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

where for every class i is associated a 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, and $|i|$ is the size of the class i.

C. Results

As shown in Tab. 3, the highest precision measure is reached, when using SOM, is with the combination of the similar weighting measures BF-IDBF and TF-IDF by UV. It is interesting to observe that the exclusive processing of numerical files using BF-IDBF results are almost as good as the combination of TF-IDF and BF-IDBF. It seems that BF-IDBF is not only good for HDM-UV, but even for pure numerical data processing.

Tab. 3 Precision measure - (SOM processing of Northila)

Tokenization	tfidf	tfidf-histo.	tfidf-bfidbf	histo.	bfidbf
Bag of words	31.01	38.05	**49.97**	-	-
3-Gram	32.25	35.68	46.83	-	-
4-Gram	29.63	39.15	46.70	-	-
5-Gram	31.62	37.15	48.09	-	-
Numeric	-	-	-	38.19	48.37

Furthermore, we were expecting to see the N-gram tokenizations performing better than "bag of words" tokenizations. Hence, it is not really the case. A possible explanation of this higher performance of "bag of words"

over N-grams is that the databases do not contain complex text or long paragraphs. Instead, the columns contains names, addresses, and other similar types of information which are semantically basic. Probably, if the input data was a more complex text data, then the N-grams would have performed better. The comparison of the different textual and numerical data representation measures is recapitulated in Fig. 1

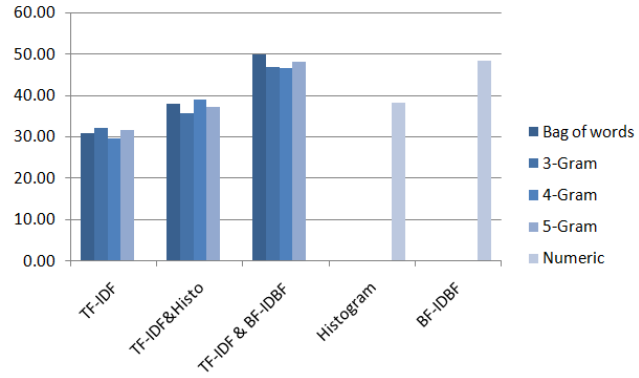

Fig. 1 Precision measures : SOM-based processing of Northila

Regarding the Recall measure (Tab. 4, Fig. 2), the best performance was with the exclusive usage of TF-IDF or BF-IDBF vectorizations. However, the TF-IDF's precision measure performed poorly, that is why the F-measure is more a objective way for evaluating the performance of the data representation methods.

Tab. 4 Recall measures: SOM-based processing of Northila

Tokenization	tfidf	tfidf-histo.	tfidf-bfidbf	histo.	bfidbf
bag-words	84.77	65.40	77.78	-	-
3-Gram	84.98	69.14	73.21	-	-
4-Gram	84.57	69.93	72.63	-	-
5-Gram	84.57	69.83	72.31	-	-
numeric	-	-	-	70.37	84.98

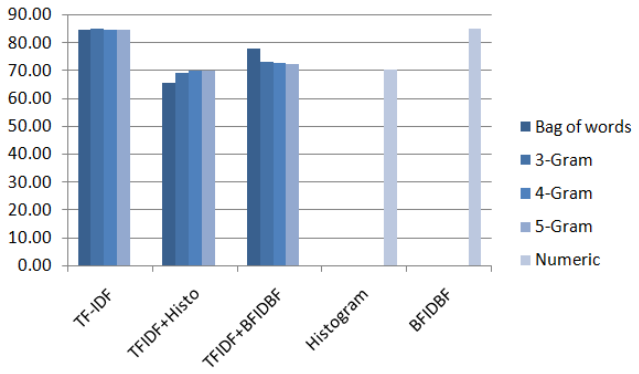

Fig. 2. Recall measures - SOM-based processing of Northila

As shown in Tab. 5 and Fig. 3, the best overall performance is reached with the combination of TF-IDF and BF-IDBF by Unified Vectorization, and the exclusive use of BF-IDBF. However, it is unexpected to see the performance of the exclusive use of BF-IDBF that high. To some extent, it can be explained by the fact that the majority of the classes and the files are of numerical data type; that is probably normal to see the BF-IDBF numerical vectorization measure having the highest score. The problem is that when BF-IDBF is combined with TF-IDF, the F-measure score should be higher than the exclusive usage of the BF-IDBF.

Another reason for this performance of the BF-IDBF over the combination of TF-IDF and BF-IDBF, is that when only BF-IDBF is used for vectorization measure, all the text documents are assigned a weight of 0 because there are no numerical data inside. In other words, all the pure text documents are grouped in one cluster which could enhance the clustering results to a certain degree because the amount of text files is too small. However, this last explanation does not justify everything.

Furthermore, we could assume that the normalization operation is not the best one for HDM-UV in order to give the best clustering results. Consequently, the future research should focus on improving the normalization operations when combining the VSMs from the heterogeneous data types. The normalization operation is very important because the VSMs are normalized three times. The first time, the VSM is normalized before the dimension reduction. Then, the VSM is re-normalized after the dimension reduction. Finally, when the different VSMs, in our case two, are combined, they are re-normalized for the last time before being processed.

Moreover, it can be observed that the BF-IDBF seems to work better than histograms. It is not the purpose of this research to process purely numerical data; however, it is an interesting observation that should be further investigated. It should be reminded that the size of the buckets in our experiments is set to 10 because we found that this size gives the best results. However, we did not do extensive testings to analyze the optimal bucket size because it is not the real scope of the current paper.

Tab.5 F-measures comparison: SOM-based processing of Northila

Tokenization	tfidf	tfidf-histo.	tfidf-bfidbf	histo.	bfidbf
Bag of words	45.39	49.40	**60.64**	-	-
3-Gram	46.75	47.06	57.07	-	-
4-Gram	43.86	50.06	56.67	-	-
5-Gram	46.02	48.49	57.53	-	-
Numeric	-	-	-	49.50	**61.64**

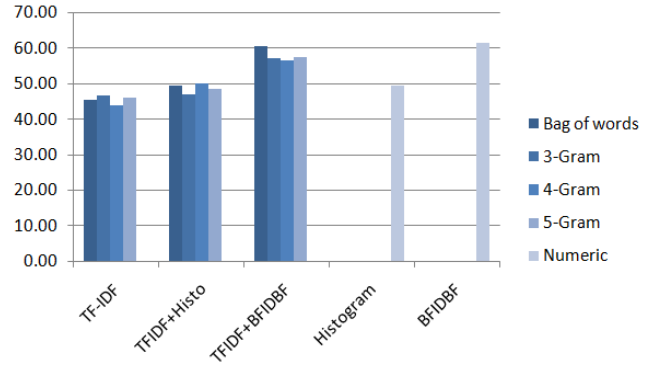

Fig. 3. F-measures: SOM-based processing of Northila

As shown in Fig. 4 and Tab. 6, the best precision measure is reached when using purely the BF-IDBF vectorization measure. This is not only the case for K-means but for SOM as well. Again, the selected normalization method could be the reason for the combined TF-IDF and BF-IDBF measures to not achieve the best precision scores.

Tab. 6 Precision measure
K-means-based processing of Northila

Tokenization	tfidf	tfidf-histo.	tfidf-bfidbf	histo.	bfidbf
Bag of words	23.07	26.33	**35.61**	-	-
3-Gram	18.68	24.48	**35.75**	-	-
4-Gram	23.64	20.59	34.26	-	-
5-Gram	21.30	16.47	27.51	-	-
Numeric	-	-	-	17.25	**36.61**

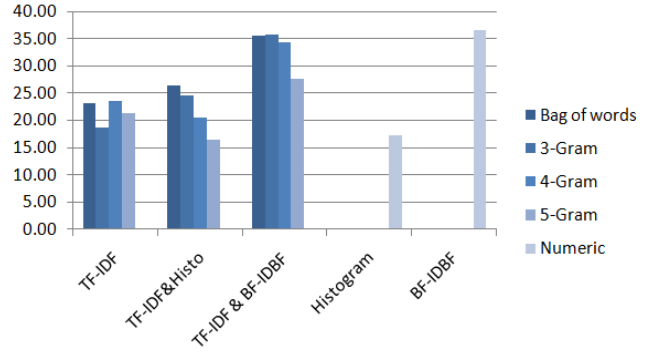

Fig. 4. Precision measure: K-means-based processing of Northila

Regarding the recall measure, as shown in Tab. 7 and Fig. 5, it appears that as expected the proposed combination of the TF-IDF and the BF-IDBF performs better than all the other vectorization measures except when the TF-IDF is exclusively used. However, the TF-IDF obtained a low precision score; therefore, evaluating the machine learning algorithms performance with the F-measure is more objective.

Tab. 7 Recall: K-means-based processing of Northila

Tokenization	tfidf	tfidf-histo.	tfidf-bfidbf	histo.	bfidbf
Bag of words	83.95	81.48	**87.04**	-	-
3-Gram	85.19	82.72	67.90	-	-
4-Gram	**87.04**	77.78	82.72	-	-
5-Gram	85.80	85.19	85.19	-	-
Numeric	-	-	-	82.10	66.04

As shown in Tab. 8 and Fig. 6, the best F-measure score, when using K-means, is the obtained through the combination of the TF-IDF and the BF-IDBF measures. This demonstrates again that having similar measures, such as the TF-IDF and the BF-IDBF, is more appropriate for heterogeneous data mining by Unified Vectorization. This is valid for supervised and unsupervised clustering algorithms.

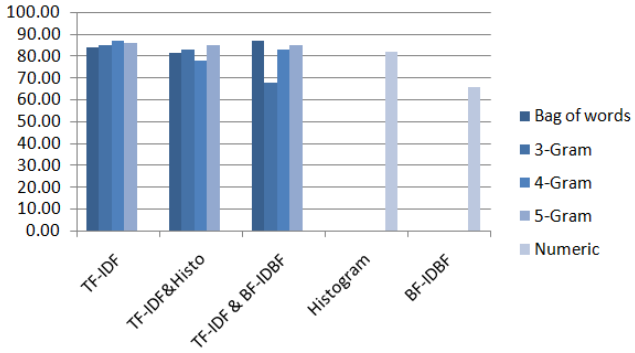

Fig. 5 Recall: K-means-based processing of Northila

Furthermore, it can be observed again that the BF-IDBF measure performs better than the histograms. It would be interesting to re-compare the two measures within the scope of other research projects on purely numerical data. Besides, it is to be noted that BF-IDBF performance is not too far behind the combination of TF-IDF and BF-IDBF which revives the interest of improving the normalization process for better heterogeneous data clustering results.

Tab. 8 F-Measure comparison: K-means-based processing of Northila

Tokenization	tfidf	tfidf-histo.	tfidf-bfidbf	histo.	bfidbf
Bag of words	36.20	39.80	**50.54**	-	-
3-Gram	30.64	37.78	46.84	-	-
4-Gram	37.18	32.56	48.46	-	-
5-Gram	34.12	27.60	41.59	-	-
Numeric	-	-	-	28.51	47.11

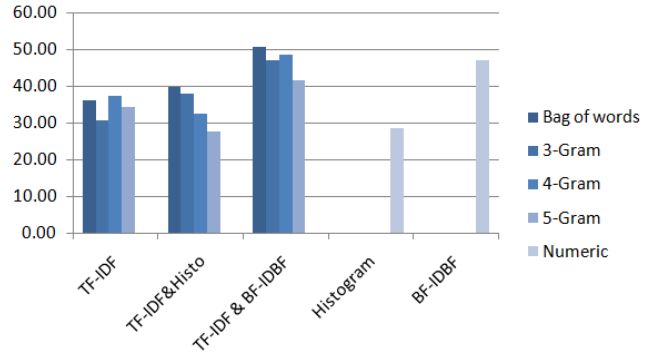

Fig. 6 F-measure comparison - K-means-based processing of Northila

V. Conclusions

In this paper we have demonstrated that Heterogeneous Data Mining by Unified Vectorization is applicable to supervised and unsupervised clustering methods by testing it respectively using K-means and SOM. We proposed a new data-set called Northila, and we have shown that these two clustering methods perform better for heterogeneous textual and numerical data mining by Unified Vectorization on this data-set. These clustering results permitted to perform data integration operations in an automated fashion with more precision.

The results have shown that SOM performs better than K-means on Northila data-set. HDM-UV improved both clustering results, which makes this method suitable for other machine-learning algorithms. Regarding text tokenization methods, "Bag of words" appeared to perform better than N-grams. Finally, the BF-IDBF vectorization measure performed better than histograms for both homogeneous numerical data processing, and heterogeneous textual and numerical data processing.

In our future work, we would like to process other data types such as multimedia data using HDM-UV. Furthermore, we would like to test HDM-UV with other machine learning algorithms. Finally, we would like to compare BF-IDBF with other purely numerical data representation.DataMining

References

[1] Bourennani, F., Pu, K. Q., and Zhu., Y., Visualization and Integration of Databases using Self Organizing Map, International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA09), Cancun, Mexico, 2009. pp. 155-160.

[2] Bourennani, F., Pu, K. Q., and Zhu, Y. Visual Integration Tool for Heterogeneous Data Type by Unified Vectorization. Proceedings of the 10th IEEE International Conference in Reuse and Integration (IRI'09), Las-Vegas, USA, 2009. pp. 132-137.

[3] Eklund, T., Back, B., Vanharanta, H., and Visa, A. Benchmarking International Pulp and Paper Companies Using Self-Organizing Maps. Turku, Finland : TUCS Technical Report No 396, Turku Centre for Computer Science, 2001.

[4] Kloptchenko, A., Eklund, T., Karlsson, J., Back, B., Vanharanta, H., and Visa, A. Combining data and text mining techniques for analysing financial reports. Intelligent Systems in Accounting Finance & Management, Vol. 12, no 1, 2004. pp. 29 - 41.

[5] Back, B., Toivonen, J., Vanharanta, H., and Visa, A. Comparing numerical data and text information from annual reports using self-organizing maps. International Journal of Accounting Information Systems, Vol. 2, no 4, 2001. pp. 249-269.

[6] Rbov, I., Konecn, V., and Matiov, A. Decision Making with Support of Artificial Intelligence. Agricultural Economics, Vol. 51, no 9, 2005. pp. 385-388.

[7] Parvizian, J., Tarkesh, H., Farid, S., and Atighehchian, A. Project Management Using Self-Organizing Maps. Industrial Engineering and Management Systems, the official journal of APIEMS, Vol.5, no 1, 2006.

[8] Hearst, M. A. Untangling Text Data Mining. Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland, USA, 1999. pp. 3-10.

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

[10] MacQueen, J. B. Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, 1967. pp. 281297.

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

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

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

[14] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys, Vol 34, no 1, 2002. pp. 147.

[15] Bingham, E., and Mannila, H. Random projection in dimensionality reduction: Applications to image and text data. Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, USA, 2001, pp. 245 - 250.

[16] 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

[17] Sakila: `http://dev.mysql.com/doc/sakila/en/sakila.html` [accessed: November 24, 2009]

[18] Northwind: `http://www.microsoft.com/downloads/details.aspx?FamilyID=06616212-0356-46A0-8DA2-EEBC53A68034` [accessed: November 24, 2009]