Visual Integration Tool for Heterogeneous Data Type by Unified Vectorization

Farid Bourennani, Ken Q. Pu, Ying Zhu
University of Ontario Institute of Technology
2000 Simcoe Street North, Oshawa, Ontario, Canada, L1H 7K4
{farid.bourennani, ken.pu, ying.zhu}@uoit.ca

***Abstract*—Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data. One of the critical issues of data integration is the detection of similar entities based on the content. This complexity is due to three factors: the data type of the databases are heterogenous, the schema of databases are unfamiliar and heterogenous as well, and the amount of records is voluminous and time consuming to analyze. As solution to these problems we extend our work in another of our papers by introducing a new measure to handle heterogenous textual and numerical data type for coincident meaning extraction. Firstly, to in order accommodate the heterogeneous data types we propose a new weight called Bin Frequency - Inverse Document Bin Frequency (BF-IDBF) for effective heterogeneous data pre-processing and classification by unified vectorization. Secondly in order to handle the unfamiliar data structure, we use the unsupervised algorithm Self-Organizing Map. Finally to help the user to explore and browse the semantically similar entities among the copious amount of data, we use a SOM based visualization tool to map the database tables based on their semantical content.**

***Index Terms*—SOM, Pre-Processing, Data Integration, Information Retrieval (IR), Visual Data Mining**

I. Introduction

As a result of the progress made in hardware technology, never before in history data has been generated at such high volumes as it is today. Now days, it's not surprising to deal with several distributed databases containing terabytes of data for data warehouse building purposes. Joining these multiple databases can significantly increase the data richness. However, exploring and analyzing the vast volumes of data for data integration operations becomes increasingly difficult. For example, identifying similar or semantically equivalent attributes over different databases is essential toward defining data integrity constraints (such as foreign-key constraints). Due to different languages and business practices, these semantically similar database attributes may have been named differently and consequently are difficult to detect by the user. But, in order to maintain a high level of data integrity of the consolidated database, it is important to identify these equivalent attributes. Many automatic schema mapping techniques have been proposed by researchers to facilitate this task[2], [3]. However, due to the fundamental nature of the problem, it is inevitable that mistakes will be made. In brief, a purely automated techniques would be risqué and delegating the task to the user would be too long.

Information visualization and visual data mining are good candidates to deal with the flood of information in semi-automated fashion. The advantage of visual data exploration is that the user is directly involved in the process. There is a large number of information visualization techniques which have been developed over the last decade to support the exploration of large data set. In this paper, we are motivated to extend one of work by using visualization technique to aid the end-user to identify semantically similar data attributes in a semi-automated fashion.

It has been shown that Self Organizing Map (SOM) is appropriate to classify unfamiliar database entities[1]. SOM was used in numerous text visualization related works, and applied in thousands of projects[5]. Also, SOM benefits from fast training algorithms (i.e. batch training [4], [5]). Additionally, SOM based visualization tool appeared to be easy to explore semantically identical or similar database entities due to its inherit low dimensional regular grid layout[1]. However, despite the successful coincident meaning extraction from heterogeneous textual and numerical data types, the classification results were not optimum because of the dissimilar data representation.

In this paper we use SOM visualization tool for data integration with focus on the pre-processing phase because in most real tasks preprocessing is important; perhaps even the most important part of the whole process[14].

Contribution of the paper

- We show that it is possible to meet the challenge of extracting *Coincident Meaning* from *Heterogeneous* textual and numerical data types, by unified vectorization for better mining results.
- We propose a new measure for numerical data, named *Bin Frequency - Inverse Document Bin Frequency (BF-IDBF)*, that permits to pre-process heterogeneous textual and numerical data type efficiently for better clustering, i.e., higher precision and recall.
- We demonstrate that a similar data representation for heterogenous data mining; like the combination of BF-IDBF and TF-IDF measures, can significantly enhance the algorithms classification performance.

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. Pre-processing

In order to implement any classification technique, it is necessary to transform the text documents into an algebraic model so they can be processed. In this paper, the documents are in fact relational database entities. More specifically, these entities or documents refer to table

978-1-4244-4116-7/09/$25.00

columns. The content of every column is extracted into file in order to be processed. Then, the standard practice in IR 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 [15].

Firstly, it's hard to obtain accurate information of semantic relatedness automatically from textual information only [5]. Secondly, the database columns very often are purely numerical or a mix of textual and numerical data like *addresses*. And, it is more complex to extract coincident meaning from heterogenous data [13].

In our previous work[1], we proposed to pre-process other data type such as numerical data separately, then, combine textual with numerical mining by *unified vectorization (UV)* for better concurrent semantic clustering results as illustrated on Fig. 1. In spite of good results, the usage of histograms as representation for numerical data type was was not optimal when compared to the TFIDF measure. The reason is that in the opposite to TF-IDF measure, histogram model does not give a sense of rarity, hence, importance of the number in the corpus. In other words the two representation are not equivalent because they don't reflect exactly the same type of information.

To solve this problem we propose the usage of Bin Frequency - Inverse Document Bin Frequency (BF-IDBF) weight as an alternative for representing the numerical data type. Actually, BF-IDBF model has two advantages. First, it uses the properties of an histogram which are more appropriate for numerical data representation due to the different nature of the data. Secondly, it offers a data representation that is equivalent to TF-IDF measure, in consequence of which, the Machine Learning algorithms perform better when the data is processed by UV.

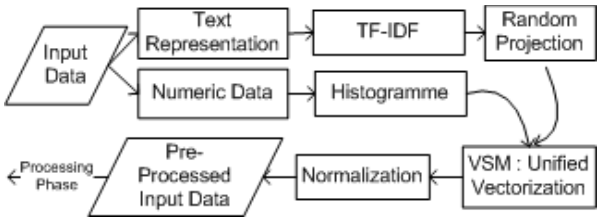

Fig. 1. Preprocessing Phase

A. Treatment of textual Data

In the IR domain, documents are traditionally represented by the VSM. Documents are tokenized using simple syntactic rules such as white space delimiters in English, which, are called also Bag of Words [6] [8]. 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. The documents are vectors in this space.

Another popular text tokenization method, used by text search engines for example, is N-gram. 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. It offers several advantages; it's an easy and fast way to solve the syntax related issues such as misspelling, 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.

B. Textual Data Weighting

Most approaches[6][12] 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 having the term t_k.

C. Treatment of Numerical Data

The numerical data is pre-processed differently, from the textual one, due to the different nature of data. For example, two numbers 6.01 and 5.9 won't be detected as similar 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 pre-process the numeric input data differently so that their value reflect 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[11]. Among them, normalization according to the variance was used to preprocess the financial data for benchmarking using self-organizing map [9].

D. Histogram weighting

In our previous work, Histogram analysis [11] was selected to represent numerical data portion of the database columns. Its dimensional reduction property eases the SOM neural network's learning process.

There are different variants of histograms but Equal-Frequency (Equal-Depth) Histogram [11] 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. 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 Tab. 1.

E. Textual vs. numerical data weighting

It is difficult to extract meaning from the textual and numerical data automatically, some projects tried to do it with SOM but results showed that clusters from qualitative and quantitative analysis did not coincide [13]. Despite this difficulty we were able to extract coincident meaning from textual and numerical data using unified vectorization for SOM data processing. However, the results were not optimum. For example, it's well known that using n-gram as tokenizer for textual data gives better results than "bag of words". But surprisingly when unified vectorization was applied using TF-IDF (n-gram as tokenizer) and histogram, the results were poorer than the combination of TF-IDF (bag of words as tokenizer) and histogram. This can be explained by the fact that the two measures TF-IDF and histogram don't have an equivalent representation of token. Indeed, TF-IDF measures the importance of token in the document as well as its general importance in the corpus. On the contrary, histogram measures only the number of occurrences of the token. For more details and examples refer to [7].

F. BF-IDBF weighting

As an alternative solution to the histogram, we propose an equivalent measure to the TF-IDF for numerical data which is based on the histogram at the same time. First is computed the histogram, then, the BF-IDBF measure is calculated in two steps. The BF serves to estimate the importance of the bin rather than the importance of a number in a document which simplifies significantly the processing time and resources. The BF is defined as follows:

$$\mathrm{BF}(b_l, d_j) = \frac{Freq(b_l, d_j)}{\sum_k Freq(b_l, d_j)}$$

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.

Other variances of histogram could be used as well, but, because of the good results obtained in the previous researches[1], "equal depth" histogram was kept.

The next step is the calculation of the IDBF weight which mainly serves to reduce the weight of the bin if it is not rare in the corpus. In other words, if a certain range of numbers are common to a high amount of documents, the weight is decreased for better documents classification based on the numerical semantical content. The IDBF is defined through a similar formula to IDF calculation as follow:

$$\mathrm{IDBF}(b_l, d_j) = Log\frac{N_{\mathrm{doc}}}{N_{\mathrm{doc}}(b_l)}$$

where N_{doc} is the total number of documents in the corpus, while $N_{\mathrm{doc}}(b_l)$ is the number of documents in the corpus with bin b_l.

Finally, the BF-IDBF is calculated by multiplying the two measures, therefore, the global formula is:

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

As detailed in the experiments section, the SOM based processing when using the BF-IDBF (for numerical data) with TF-IDF (for textual data) permits to achieve better clustering results than the usage of histogram and TF-IDF. Even, the exclusive usage of BF-IDBF can perform better than the combination TF-IDF with histogram when the corpus is composed of higher numerical data volume than the textual.

G. Dimensionality reduction

In textual and numerical application context, the high dimensionality is due to the large vocabulary and numeric data 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 achieve this dimensionality reduction is the Principal Component Analysis (PCA), however, it is computationally expensive and is not feasible on large, highdimensional data[16]. Therefore, another powerful technique that solves these problems is the Random Projection (RP) which is simple, offers clear computational advantages, and preserves similarity[16]. 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[16].

RP is applied to textual and numerical VSM separately, then, the two reduced matrixes are combined by unified vectorization for processing. The RP is when 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.

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}

Tab. 1 Unified Vectorization of Textual and Numerical Data

H. unified vectorization of textual and numeric data

The textual and numerical VSM are combined in order to be processed simultaneously by unified vectorization as shown on tab. 1. This combination permits to meet the challenge of extracting efficiently a coincident

meaning from these *heterogeneous* data. The unified vectorization offers several advantages. Firstly, the training is faster because the heterogeneous data is actioned simultaneously rather then processing it in sequence by data type. Secondly, the thorny task of combining of the results is avoided, and naturally the same SOM trained map is used to classify these heterogeneous data type. In addition, it permits to handle the documents which contain both textual and numerical heteregenous data type. It's important to mention that the unified VSM should be normalized in order to avoid any unjustified influence of one of the two data types during the SOM training phase.

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 plane (2D). The unsupervised classification property of SOM serves to classify completely unfamiliar databases. In essence, despite the unknown databases schemas, the different database respective technologies, the different naming standards (client vs. customer), it would be possible to integrate these databases based on their semantics.

A. Self Organizing Map

The most remarkable capability of SOM is that it produce a mapping of high-dimensional input space onto a low-dimensional (usually 2-D) map, where similar input data can be found on nearby regions of the map. Furthermore, SOM offers all the advantages of visual display for information retrieval which are: 1)the ability to convey a large amount of information in a limited space, 2) the facilitation of browsing and the perceptual inferences on retrieval interfaces, 3) the potential to reveal semantic relationship of terms and documents [10]. These qualities will facilitate the user the exploration of huge amount of database entities and discover similar columns based on the semantics, which, is not possible with the traditional ontology based integration tools.

B. SOM Visualization

A simple way to visualize the clustered database columns is to match every column d_j to its respective Best Matching Unit (BMU) node[4] on the trained SOM's map. This will result in having semantically similar documents grouped on the same Map's node. In addition, the topological distribution of the clusters on the map will reflect the semantical proximity among them. The closer they are on the map, the stronger is their semantical content similarity. Fig. 3 is an illustration of the resulted trained map.

The graphical interface offers several visual options. It is possible to zoom the map, as shown on Fig. 2, to enlarge it, to flip it, or to rotate it.

Fig. 2. Map Zooming : table@column

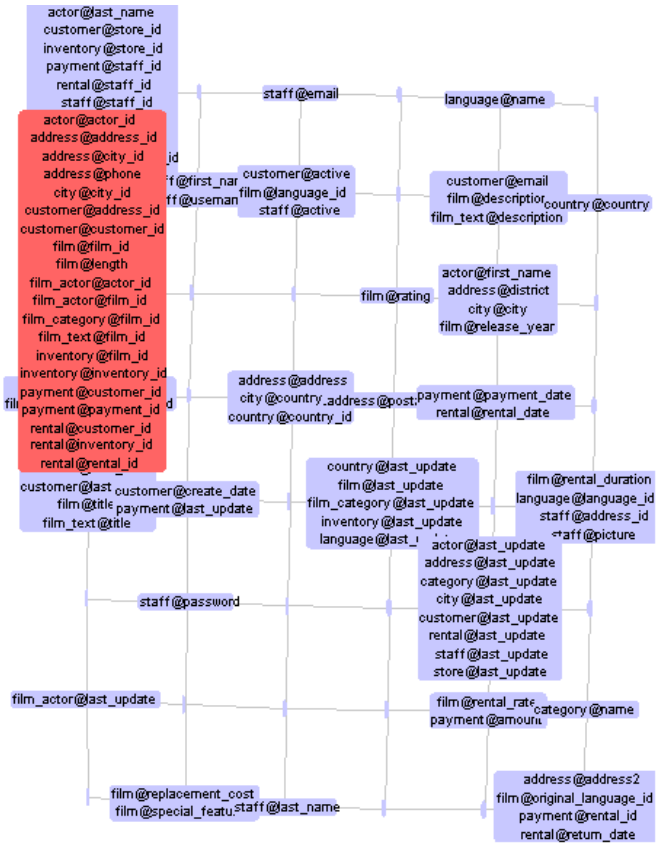

Fig. 3. Trained SOM map

IV. Experiments

A. The corpus

The proposed techniques were tested on a demo database available online, named Sakila[17]. As shown on tab. 2, the database has 89 tables with 22104 numerical and textual terms as vocabulary size.

Data Set	Columns	terms	textual	numeric	classes
Sakila	89	22104	4932	17172	20

Tab. 2. Unified Vectorization of Textual and Numerical Data

B. Tokenization

The process of breaking a text up into its constituent tokens is known as tokenization. In this research, there is no linguistic pretreatment, i.e. lemmatization, stemming or stop words cleaning. Therefore, tokenization has a more significant impact on the results. Even if it's not the purpose of this paper to evaluate the impact of tokenization, however, it is important to mention that all the non alphanumeric characters are eliminated for the experiments because when used they reduce the algorithm performance.

C. Experimental Objectives

The objective of the tests is to evaluate the add value of the BF-IDBF on processing heterogenous data type, and more specifically using SOM classification method. In order to measure the contribution of the BF-IDBF weight to the processing phase, F-measure is used. It is calculated with respect to the known classes for each document, and it is based on Precision and Recall weights[6].

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

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

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

Where N_{ij} is the number of members of class i in the 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 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 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.

D. Forming classes

The evaluation of the relevance of the classes formed remains an open problem because of the subjective nature of the task [6]. 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 they do not fit within the purpose of this research which is the integration of semantically similar columns. Therefore, these kinds of data are classified in separated classes.

E. Experiments

A good classification requires a good presentation[6]. Several representations are tested, i.e. multiple combinations of tokenization and vectorization, are processed through the SOM algorithm. Accordingly two tokenization methods, for textual data, are used: bag of words versus N-gram as described earlier. Besides, three vectorization techniques are compared: TF-IDF (textual data), histogram (numerical data)and BF-IDBF(numerical data).

It's important to mention that around 70 % of the files are constituted of purely numerical data such as keys, dates, prices, phone numbers, etc. The remaining ones are textual or a combination of the two data types.

For every measure, four set of tests were completed as shown on Tab. 3-4-5. Firstly, the database classification using SOM was evaluated without unified vectorization, i.e. either numerical or textual exclusive input data was processed. In this case, the dimension was reduced using RP to 1500. Then, the the heterogenous textual and numerical data type were processed by unified vectorization using different vectorization including BF-IDBF. Therefore, it was possible to estimate the enhancement caused by the proposed measure. Note that in the second case, the dimension was reduced 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, we don't think it has a major impact to bias the results.

Tokenization	TFIDF	TFIDF+HISTO	TFIDF+BFIDBF
Bag of words	26.68	46.23	64.81
3-Gram	21.68	38.15	58.77
4-Gram	30.02	42.72	**65.56**
5-Gram	26.57	47.28	63.94

Tab. 3 Precision measure with different representations

Tokenization	TFIDF	TFIDF+HISTO	TFIDF+BFIDBF
Bag of words	89.89	74.15	86.52
3-Gram	**93.26**	70.79	86.52
4-Gram	92.13	71.91	**88.76**
5-Gram	92.13	71.91	83.14

Tab. 4 Recall measure with different representations

Tokenization	TFIDF	TFIDF+HISTO	TFIDF+BFIDBF
Bag of words	41.15	56.95	74.11
3-Gram	35.18	49.58	69.99
4-Gram	45.29	53.59	**75.42**
5-Gram	41.25	57.05	72.29

Tab. 5 F-measure with different representations

The experiments (Tab. 3, Fig. 4) show that the proposed combination of TF-IDF and BF-IDBF vectorization enhance the precision of SOM significantly by at least 15%. More specifically, the best results are obtained when using 4-gram as tokenizer which is an improvement by almost 20 % of the SOM's precision. Surprisingly, the precision results obtained using exclusively BF-IDBF (54.49%) were better that even the combination of TF-IDF and histogram.

In regards to the Recall measure (Tab. 4, Fig. 5), the best performance was with the exclusive usage of TF-IDF vectorization, however, it's precision was very low and that is why F-measure is a more objective way of comparing these representations. Then, the proposed combination of TF-IDF and the new BF-IDBF vectorization measures follow in the second position. It's interesting to note that the usage of the exclusive BF-IDBF measure with purely numerical data (69.66%) is almost as good as the unified vectorization by TF-IDF and histogram.

Finally, the precision and the recall are combined equally to produce the F-measure which is more objective to compare the different representations. It can be easily observed (Tab. 5, Fig. 6) that 4-gram tokenization

combined with the proposed vectorization using TF-IDF and BF-IDBF overcomes all the other representations. In fact, it performs better than the unified vectorization by TF-IDF and histogram by around 20% and almost *doubles* the performance of the traditional textual representation by TF-IDF. Even the usage of the pure BF-IDBF representation of the exclusively numeric data performs better (61.15%) than the pure TF-IDF or even the combination of TF-IDF and histogram. This demonstrates the beneficial properties of the proposed BF-IDBF measure. It shows as well that the preprocessing step is one of the most important phase in data classification because of the major impact on the machine learning algorithms result. In addition, per induction the proposed combination of TF-IDF and the new BF-IDBF measure can be applied to any other algorithm for probably better results.

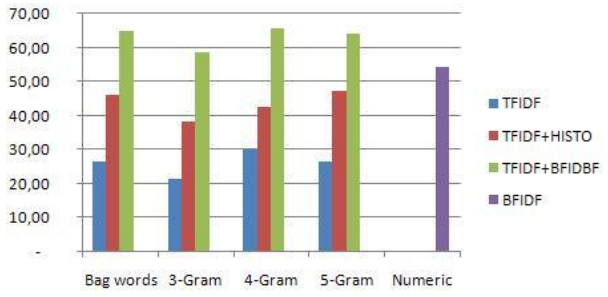

Fig. 4. Precision measures

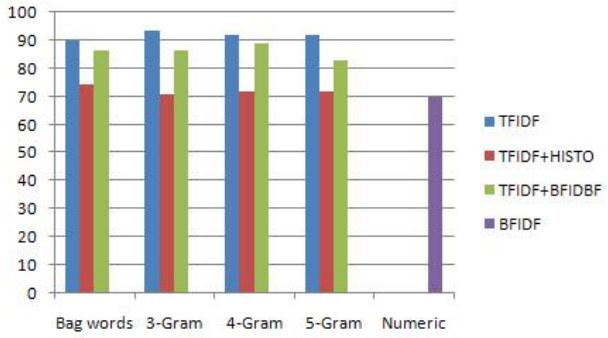

Fig. 5. Recall measures

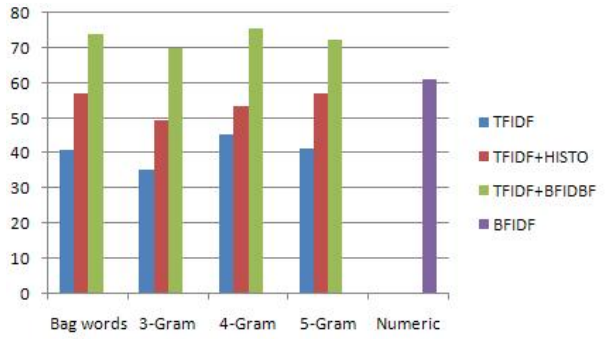

Fig. 6. F-measure measures

V. Conclusions

In this paper we have presented an efficient way to integrate unfamiliar heterogenous textual and numerical data by the usage of SOM based visualization tool with focus on the pre-processing phase. So, we demonstrated through the database visualization tool which exposes the similarity between columns, based on their semantical content, greatly serves the purpose of distributed database integration. This tool is applicable to data integration over web data sources and tuples classification based on the content.

We have proposed a new approach to pre-process and combine heterogenous textual with numerical data mining by unified vectorization, which resulted in extracting better meaningful results. The new suggested weighting measure BF-IDBF improved the precision significantly as well as the recall of the SOM algorithm. It is highly likely that this measure would improve the performance of any other machine learning algorithm.

In our future work, we would like to apply the proposed pre-processing techniques on other data types such as multimedia or meta data. Furthermore, we would like to test the unified vectorization by TF-IDF and BF-IDBF with other machine learning algorithms.

References

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

[2] 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.

[3] 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.

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

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

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

[7] Bourennani, F., Integration of Heterogeneous Data Type Using Self Organizing Maps, Master's thesis, Faculty of Engineering and Applied Science, UOIT, Oshawa, Canada, 2009.

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

[9] Wang, J., Data mining: opportunities and challenges, IGI Publishing, Hershey, USA, 2003, pp. 323-349

[10] Lin, X., Map displays for information retrieval., Journal of the American Society for information Science, 1998, vol. 48: 40-54

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

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

[13] 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.

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

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

[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`