(IJNCAA)

ISSN 2220-9085(ONLINE)
ISSN 2412-3587(PRINT)

INTERNATIONAL JOURNAL OF

New Computer Architectures and their Applications

Volume 7, Issue 2
2017

SDIWC

www.sdiwc.net

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Overview

The SDIWC International Journal of New Computer Architectures and Their Applications (IJNCAA) is a refereed online journal designed to address the following topics: new computer architectures, digital resources, and mobile devices, including cell phones. In our opinion, cell phones in their current state are really computers, and the gap between these devices and the capabilities of the computers will soon disappear. Original unpublished manuscripts are solicited in the areas such as computer architectures, parallel and distributed systems, microprocessors and microsystems, storage management, communications management, reliability, and VLSI.

One of the most important aims of this journal is to increase the usage and impact of knowledge as well as increasing the visibility and ease of use of scientific materials, IJNCAA does NOT CHARGE authors for any publication fee for online publishing of their materials in the journal and does NOT CHARGE readers or their institutions for accessing the published materials.

Publisher

The Society of Digital Information and Wireless Communications
20/F, Tower 5, China Hong Kong City, 33 Canton Road, Tsim Sha Tsui, Kowloon, Hong Kong

Further Information

Website: http://sdiwc.net/ijncaa, Email: ijncaa@sdiwc.net,
Tel.: (202)-657-4603 - Inside USA; 001(202)-657-4603 - Outside USA.

The issue date is October 2017.

IJNCAA

ISSN 2220-9085 (Online)
ISSN 2412-3587 (Print)

Volume 7, Issue No. 2

2017

CONTENTS

ORIGINAL ARTICLES

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 37-44
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Learning Experiences Using Neural Networks and Support Vector Machine (SVM)

Soumya ARACH, Pr. Halima BOUDEN,
Laboratory of Computer Science, Operational Research and Applied Statistics.
University Abdelmalek Essaadi.
Tétouan, Morocco.
soumya.arach@gmail.com , bouden.halima@gmail.com

ABSTRACT

This article is part of the global data mining framework, it addresses the theme of learning and classification, to identify the classes to which objects belong from using some descriptive parameters. They are particularly suited to the problem of automated decision-making. In this article we tried to implement three learning techniques, the Support Vector Machine (SVM), the Neural Networks and the Decision Trees.
This application study aims to compare the results of these three techniques in terms of respecting the performance of the classification used for the contained objects in the data set ''IRIS'' based on the confusion matrix generated by the software weka, which is the tool used to carry out these learning experiences.

KEYWORDS

Data Mining, Machine learning, classification, Support Vector Machine (SVM), Decision trees, Neural Networks.

I. INTRODUCTION:

Classification methods aim to identify the classes to which objects belong of the basis of some descriptive parameters. They apply to many human activities and are particularly suited to the problem of automated decision-making. The classification procedure will be extracted automatically from a set of examples. An example is the description of a case with the corresponding classification. A learning system must, then, from this set of examples, extract a classification procedure, it is a question of extracting a general rule from the observed data.
The procedure generated must correctly classify examples of sample and have good predictive power to correctly classify new descriptions.

The methods used for classification are many, include: the method of Support vector machine (SVM), Neural Networks, decision trees, etc. We present in the rest of this article a study of three techniques SVM, Neural networks and decision trees. These methods have proven their effectiveness in many application areas such as image processing, text categorization and medical diagnostics.

II. MACHINE LEARNING

Machine learning refers to the development, analysis and implementation of methods that allow a machine to evolve through a process of learning, and so perform tasks that are difficult or impossible to fill by more conventional algorithmic means. [4]
Its goal is to automatically extract and exploit this information in a data set.
The learning algorithms can be categorized according to the type of learning they employ: Supervised learning, unsupervised learning and reinforcement.

Unsupervised learning is a type of machine learning algorithm used to draw conclusions from input data of compounds datasets without categorizing responses.
The unsupervised learning method is the most common data partitioning, which is used to perform an exploratory analysis of data to find hidden patterns or clusters in the data. The clusters are designed by means of a similarity measure defined by metrics such as Euclidean distance or probabilistic distance. [5]

Supervised learning: needless to mention here the well-known regression techniques. The most typical method of data mining is certainly that

decision trees: to predict a response Y, either numerical or qualitative, it initially looks for the best score of all data (usually two subsets) after a score performed on predictors and iterating in each of the subsets: the exponential growth of the tree is controlled by cost-complexity type of stop criteria as well as the validation data use that eliminate irrelevant branches. This technique results in very readable decision rules, hence its success, and prioritizes explanatory factors. In contrast in terms of readability, data mining software often offer highly nonlinear methods such as neural networks, support vector machines (SVM), decision trees, etc ... that we are going to apply in this third party to know which of these three methods give the best classification for a set of data. [2] [9]

1. SUPPORT VECTOR MACHINES (SVM)

Support Vector Machines (SVM), also called wide margin separators are supervised learning techniques designed to solve classification problems. Support Vector Machines the concepts relating to the theory of statistical learning and the theory of boundaries of Vapnik and Chervonenkis [16]. The intuitive justification of this method is: if the learning sample is linear separately, it seems natural to separate the elements of the two classes of so that they are as far as possible from the chosen frontier. These famous machines were invented in 1992 by Boser and al, but their denomination by SVM appeared only in 1995 with Cortes and al. [17]

2. NEURAL NETWORKS

Historically the inspiration for neural networks originated however, of the desire to create sophisticated, even intelligent, artificial systems, able to execute operations similar to those performed by the human brain routinely, and to try to improve understanding of the brain.
Most neural networks have a certain capacity for learning, means that they learn from examples. The network can then to be able to generalize that is to say to produce correct results on new cases that had not been presented to him during the during the learning process... [18]

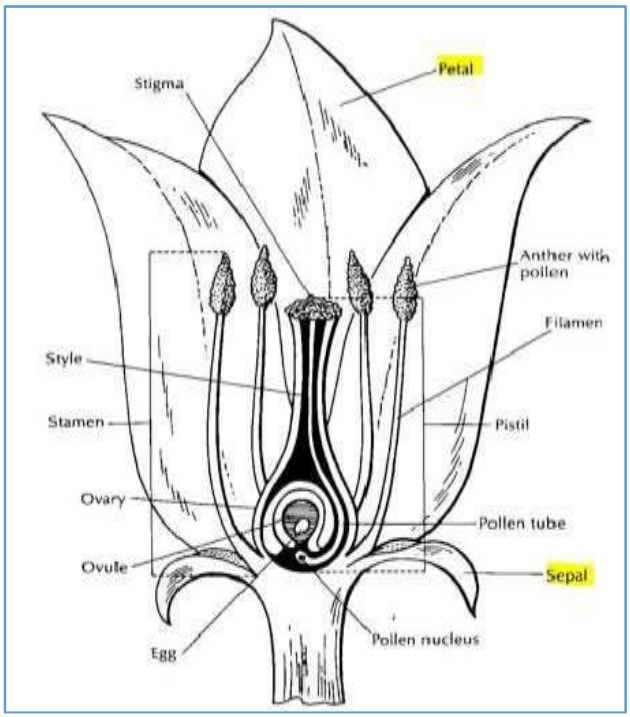
3. DECISION TREES

A decision tree is a diagram representing the possible outcomes of a series of interconnected choices. It allows a person or organization to evaluate different actions based on their cost, likelihood and benefits. It can be used to feed an informal discussion or to generate an algorithm that determines the best choice mathematically. A decision tree usually begins with a node from which several possible outcomes result. Each of these results leads to other nodes, from which emanate other possibilities. The pattern thus obtained is reminiscent of the shape of a tree.

III. APPLICATION STUDY:

The work involves testing different learning systems namely neural networks (PMC), Support Vector Machine (SVM)(SMO) and decision trees (J48) on some appropriate databases and examine how does the performance (rate error, confusion matrix, …)

1. DESCRIPTION OF THE DATABASE



IRIS contains the famous series Fisher iris data. The dataset includes measurements of 150 samples of flowers of all three species of

flowers. Setosa Iris, Iris Virginica, and Iris versicolor [8]

Four characteristics (assigned) were measured for each sample:

- The length of the flower sepals
- The width of sepal flowers
- The length of the flower petal
- The width of flower petal

All 150 samples from the data of the iris Fisher are stored in a single table called measures:
The four columns correspond to the four types of measures: the length of the sepals, width of sepals, petals length and width of the petals, respectively.

- The first **50** rows contain data for Iris Setosa
- The **50** second lines contain data for Iris Virginica
- The **50** third lines contain data for Iris versicolor

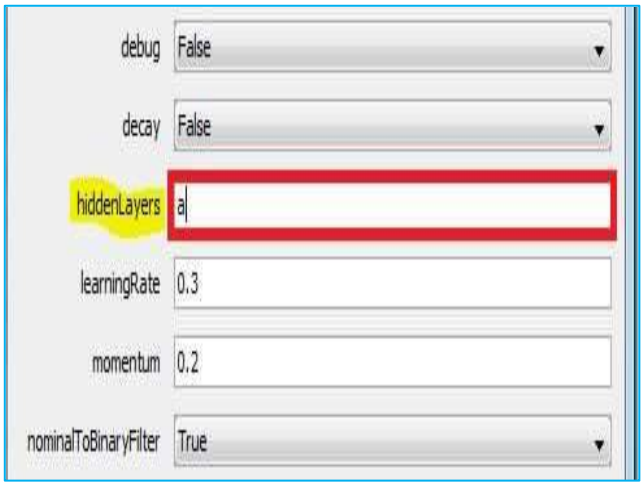
To test these learning systems namely neural networks (PMC), **SVM (SMO)** and decision trees **(J48**), we used the **WEKA** software.
The data file format is the format Weka '.arff.
Once given the game is open we will be providing the following information:

- the number of data and the number of attributes per data,
- the name of each attribute,
- What is the class attribute,
- the number of observed values by attribute,
- the number of missing data for each attribute
- information distribution of attribute values,
- the distribution of each attribute,

2. IMPLEMENTATION OF ALGORITHMS:

2.1 Neural Networks

On the WEKA software [7] we ran the algorithm Neural Networks by changing each time the number of neurons in the hidden layer in Weka changing the Hidden layer attribute that describes the number and size of hidden layers



-Or special values defining a single hidden layer:

a: (number of attributes + number of classes) / 2 t: number of attributes + number of classes
o: number of classes
i: number of attributes

When we are running the neural network algorithm with the first value:

a = (number of attributes + number of classes) / 2)

we obtained the following results:

```
Correctly Classified Instances         146               97.3333 %
Incorrectly Classified Instances         4                2.6667 %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

From this table, we see that 97**, 33%** of the sample were classified correctly. The confusion matrix below, indicates that the errors concerned the "iris-versicolor 'class for which **48** examples are correctly classified **50**, and **48** examples for" Iris- vrginica "that are correctly classified **50** examples.

The second value (t: number of attributes + number of classes)

```
Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

The results are distinguished **96%** were correctly classified, while **4%** were misclassified. The confusion matrix shows that the class **"iris-setosa"** was ranked well in addition to the common errors are the class level **"iris-versicolor"** for which **48** examples **50** are correctly classified, and **48** examples for **"Iris-vrginica"** that are correctly classified **50** examples.

The third value (o: number of classes)

```
Correctly Classified Instances         146               97.3333 %
Incorrectly Classified Instances         4                2.6667 %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

According to this table, it is seen that **97, 33%** of the examples were correctly classified. The matrix of confusion, indicates that the errors related to the class “iris-versicolor” of which 48 examples out of **50** are correctly classified, and **48** examples for “Irises-vrginica” which are correctly classified on **50** examples.

The fourth value (i: number of attributes)

```
Correctly Classified Instances         147               98      %
Incorrectly Classified Instances         3                2      %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  1 49 |  c = Iris-virginica
```

The matrix of confusion, indicates that the errors related to the class “iris-versicolor” for which 48 examples out of 50 are correctly classified, and 49 examples for “Irises-vrginica” which are correctly classified on 50 examples. Thus for the database Iris the best result is got if the number of neurons of the hidden layer is equal to the number of attribute with an error rate of 2% and 3 badly classified examples. (The value “I” is the best).

2.1.a Application of Boosting

Who aims at the improvement of the procedures of decision by overweighting the badly classified units, and by reiterating the process.

We applied the BOOSTING in each value (has, **I, O** and **T**)

a: (number of attributes + number of classes)/2

```
Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  3 47 |  c = Iris-virginica
```

t: number of attributes + number of classes

```
Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %

 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 47  3 |  b = Iris-versicolor
 0  3 47 |  c = Iris-virginica
```

o: number of classes

```
Correctly Classified Instances         142               94.6667 %
Incorrectly Classified Instances         8                5.3333 %

 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 46  4 |  b = Iris-versicolor
 0  4 46 |  c = Iris-virginica
```

i: number of attributes

```
Correctly Classified Instances         147               98      %
Incorrectly Classified Instances         3                2      %

 a  b  c   <-- classified as
50  0  0 |  a = Iris-setosa
 0 48  2 |  b = Iris-versicolor
 0  1 49 |  c = Iris-virginica
```

Here the table summarizing the results obtained by modifying the Hiddenlayer parameter before and after application of the boosting to the method of neural network:

Database	a	o	t	i
% of authority correctly classified	97,33%	97,33%	96%	98%
% of authority correctly classified after	96%	96%	94,66%	98%

After having applied the boosting by using the same number of neurons in the hidden layer and the attributes, one obtains the best error rate by **2%** and **3** badly classified cases.
We notice in this example that the effect of the boosting, on the analysis of the base Iris with the neural networks, **deteriorates** the results for **a, o** and **t** in more, it does not have any effect on i.

2.2 Support Vector Machines

When we carry out the Separating algorithm with Support Vector Machines we got the following results:

```
Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.2311
Root mean squared error                  0.288
Relative absolute error                 52      %
Root relative squared error             61.101   %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      66.6667 %
Total Number of Instances              150

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

- 96% of the examples were classified correctly. The matrix of confusion, indicates that the errors related to the class “iris- versicolor” of which **49** examples out of **50** are correctly classified, and **45** examples for “Irises-vrginica” which are correctly classified on **50** examples.

2.2.a Application of Boosting

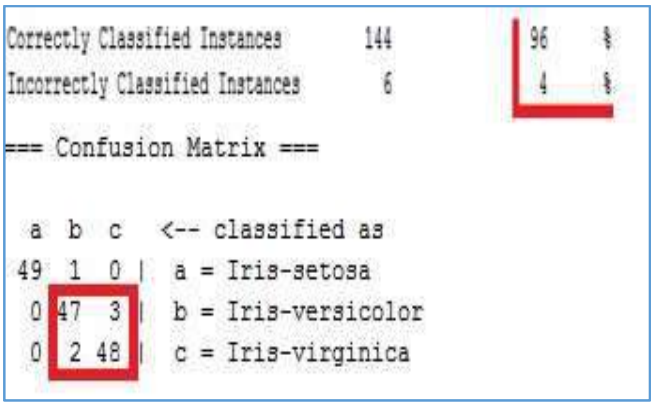
```
Correctly Classified Instances         147               98      %
Incorrectly Classified Instances         3                2      %

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

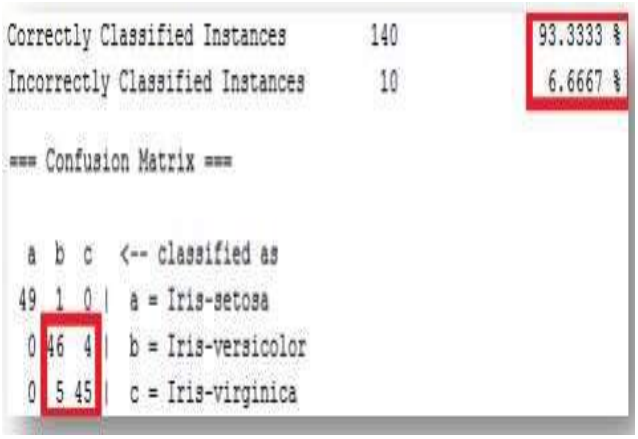
It is noticed that the error rate decreased by **4%** to **2%.** In this case the **boosting improved classification.**

2.3 Decision Trees



The results show that **96%** of the examples were classified correctly.
The matrix of confusion in the bottom, indicates that the errors related to the class "iris-versicolor" of which 47 examples out of **50** are correctly classified, and **48** examples for "Irises-vrginica" which are correctly classified on **50** examples.

2.3.a Application of boosting



The error rate after the application of boosting increased from 4% to 6.66% after applying boosting. In this case the boosting deteriorates the results.

The following table summarizes the results obtained from the iris base.

Cross-Validation						
	RN	**BO**	**SV**	**BO**	**AD**	**BO**
% of instances Correctly classified	98%	98%	96%	98%	96%	93.33%
% of instances incorrectly	2%	2%	4%	2%	4%	6.667%

From this table we see that the neural network is the most appropriate algorithm for this data since there is an error rate of **2%** and **3** examples badlly classified.

In addition there is another method for this analysis is based on the
"use training set"

The same previous steps are followed by **cross validation** using
"use training set".

We get the following table summarizes the results of the application of learning methods using this option:

Use training set						
	RN	BO	SVM	BO	AD	BO
% of instances Correctly classified	98.666	98.666	96.6667	98%	98%	100 %
% of instances incorrectly classified	1.3333	1.3333	1.3333	3.333	2%	0%

We note in the above table that the results of the use of the full data set are much better than the result of **cross-validation (10 folds).**

The J48 (decision tree) and neural networks are the two most appropriate algorithms for this data set.

EVOLUTION AND CONCLUSION

After performing several experiments with different data and three classifications algorithms (neural networks, SVM, decision trees J48), and evaluation learning through the cross-validation method (10) cutting the training set into ten parts and the method use training set using all of the examples for learning.

By examining the matrix of confusion and error rates, according to the two methods of assessment, we note that the best classifiers for both bases is **decision tree (J48)** and **neural networks.**

The Boosting application on SVM (SMO), decision tree (J48) and neural networks can give different effects:

- Improve the results of the algorithms.
- Worsening the results of the algorithms.
- Do not cause any influence on the outcome of the algorithms.

These effects depend on the size of the database, the Boosting application of the small sized data sets effectively improves the results, if not the Boosting has no influence or little influence on the results.
The experimental results seem to prove the following facts:

Decision trees work well if:

- The number of possible values for each attribute is low.
- Class is qualitative value.

Application of Boosting for decision tree (J48) is more efficient and effective than other algorithms (**SVM and neural networks).**
The calculation time for neural networks is generally higher than the calculation time for systems based on decision trees.

REFERENCES

[1] Probabilistic machine learning and artificial intelligence. Zoubin Ghahramani. Nature 521, 452–459(27 May 2015)

[2] Apprentissage artificiel. Antoine Cornuéjols et Laurent Miclet – Eyrolles Collection : Algorithmes – 2e édition – juillet 2011.

[3] Intelligence artificielle. Stuart Russel, Peter Norvig. Pearson Education. Collection Informatique (December 10, 2010)

[4] Machine Learning. Peter Flach. Cambridge University Press. (August 25, 2012)

[5] Computing machinery and intelligence. Alan M. Turing. Mind, 59, 433-460 (1950).

[6] Murphy, K. P. Machine Learning: A Probabilistic Perspective (MIT Press, 2012).

[7] Weka the University of Waikato. Machine Learning Group at the University of Waikato. http://www.cs.waikato.ac.nz/ml/weka/. 14/06/2017

[8] Antoine Cornuéjols et Claudia Marinica. Cours ISI-3 : Apprentissage Artificiel et fouille de données.https://www.lri.fr/~antoine/Courses/Master-ISI/Cours-ISI-3.html. 10/06/2017

[9] Classification supervisée et non supervisée des données de grande dimension. Charles BOUVEYRON1 & Stéphane GIRARD.Revue MODULAD, 2009

[10] R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the Web: Survey and practical guide. Data Mining and Knowledge Discovery, 18:140–181, 2009.

[11] T. M. Mitchell. Machine Learning. McGraw-Hill, New York, NY, 1997.

[12] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning, 36:105–142, 1999

[13] J. Platt, Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods — Support Vector Learning (Cambridge, MA) (B. Sch¨olkopf, C. J. C. Burges, and A. J. Smola,eds.), MIT Press, 1999, pp. 185–208.

[14] T. M. Mitchell, Machine learning, McGraw-Hill, NewYork, 1997

[15] Barber, D. (2012), Bayesian reasoning and machine learning, Cambridge University Press.

[16] V. Vapnik, “The nature of statistical learning theory, ” Springer-Verlag: New York, 1995.

[17] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273–297.

[18] The connection between regularization operators and support vector kernels," Neural Networks, vol. 11, pp. 637–649, 1998.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 45-49
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Data Model Integration

Hassana NASSIRI[1, a] Mustapha MACHKOUR[2, a] Mohamed HACHIMI[3, b]
[a] Laboratory of the Computing Systems and Vision
[b] Laboratory of Engineering Sciences
University Ibn Zohr, Agadir, Morocco
[1] hassana.nassiri@edu.uiz.ac.ma
[2] machkour@hotmail.com
[3] m.hachimi@uiz.ac.ma

ABSTRACT

There has been significant recent interest in data integration and querying heterogeneous data sources. Thus, in our work, we aim to develop a system for querying databases regardless of the nature of their model, especially XML and relational data model as they are increasingly related in practice, Due to that we choose to make them the first models under study in this contribution. In fact, the relational model is the most dominated data model in most organizations, and it has utility widely used to manage and maintain a large volume of data. At the same time, XML is increasingly becoming the lingua franca of data interchange and has received considerable attention due to its multiple benefits. Besides, each one of these models has its specific query languages, and it will be important to ensure a flexible way to access information represented by both technologies. Thus, this paper addresses the problem of accessing data independently of the model used, by using a unique query language. Since users and even developers may not be familiar with multiple query language syntax at a time, we need to facilitate accessing data by making one single query in any query language enough to retrieve data from any data model.

KEYWORDS

Data Model; Relational database; XML; SQL; XPath;

1 INTRODUCTION

This paper addresses the problem of accessing data represented in different data models by using a unique query language. Since users may not be familiar with multiple query language syntax at a time, we need to facilitate accessing data by making one single query in any query language enough to retrieve data on any data model. As a start, the models under study are the relational and XML (eXtensible Markup Language) and we aim to integrate others especially the most used and famous ones. Why exactly these two models? In one hand, Relational databases are still very essential and critical infrastructure in most organizations and have utility widely used to manage and maintain a large volume of data. On the other hand, XML has received considerable attention due to its multiple benefits, especially as it is auto-descriptive, extensible and usable in all fields of applications. Last, they are complementary in practice.

There have been many attempts to query XML and relational data. Moreover, with XML becomes the lingua franca of data interchange increasingly, various research has been done to query XML Database using relational database system [1] [2]. Furthermore, others have been focused on designing general systems to manage XML among other data formats [3]. Such approaches have great opportunities as it has some limitations as well [4]. Our purpose, therefore, is to define a system for querying data stored in different models, such as the relational and XML with any query language of these models. That means that users do not need to know each query language of each data model, one query language is enough and will meet the purpose, even if the language is none of the corresponding ones of the model in question. Hence, our proposal system will be independent of the data model and the query language as well.

The remainder of this work is organized as follows. Section 2 introduces some terms related to our zone. Section 3 describes some related work and compares our approach to them. Section 4 explains the main idea of our

approach. Finally, Section 5 summarizes our contribution.

2 RELATED WORK

Recent RDBMS such as Oracle support some kind of uniform querying of mixed relational and XML data. As mentioned before, Oracle XMLDB technology extends the possibilities of the relational database by offering all the features of an XML database and offers an independent structure for the storage and management of XML data. Furthermore, there has been various work related to XML and relational database and variant studies trying to figure out a link between these models in order to efficiently store and query data, through varied approaches: XML views over Relational, using RDBMS to store XML data and query rewrite and translation. According to the systems studied in literature and based on our understanding of them, these approaches focused on querying data via diverse directions: Relational to XML sense or XML to Relational sense by a translation tool to transform XML queries into SQL or the opposite.

In relational to XML sense, here are some approaches that go with it: the ROX (Relational over XML) project at IBM [13], presents a way to efficiently support Relational over XML and the SQL to XQuery translation approach, and discuss the feasibility of querying natively stored XML data through SQL interfaces. It is an approach that provides access to relational data, based on SQL along relational views over XML. Likewise, The BEA AquaLogic Data Services Platform [14], a unified, service-oriented, XML-based view of data from heterogeneous information sources, which can be queried using XQuery. It proposes a framework that can transform SQL statements to XPath expressions. Also, [15] examines how XML data can be queried using XPath or SQL, and introduces a framework where SQL statements can be transformed to XPath queries that enable users to access XML and relational database through SQL. Again, [16] designs an SQL interface for XML databases that can convert SQL queries to XPath expressions and extract data from XML documents. And, [17] proposes a framework for converting SQL join (Left, Right, and Full) queries into XPath expressions to allow users to access XML database through SQL queries only.

In the other sense, i.e. XML to relational, some techniques are discussed here: [18] discusses the manner to support XML ordered data model using a relational database system, by encoding order as a data value. It proposes three order encoding methods and algorithms for translating ordered XPath expressions into SQL using these encoding methods. In the same token, [19] presents a way to process queries over XML by RDBMS through mapping XML documents/schema to RDBMS schema and use XQuery to retrieve XML documents. It introduces a system to store in and retrieve XML documents from a relational database system. XML documents are translated into tables in the relational database and stored in a shredded schema, and could be queried by the query language XQuery. The input is the user's query, then the XQuery expressions are translated into SQL queries. The results are transformed into XML documents and returned to the user as output. Whereas, [20] presents a methodology for integrating heterogeneous data sources, including relational and tree-structured data sources, under an XML global schema, which is implemented in the Agora data integration system [21], and explains their approach in translating XQuery to SQL. Moreover, [22] discusses the Query translation in the presence of recursive data schemas and provides algorithms for rewriting an XPATH query into an equivalent XPATH query over a recursive DTD. [23] addresses two issues: the translation of XQuery expressions to SQL statements and the development of efficient execution strategies of the resulting queries. The proposed techniques target a relational implementation but it can be used within native XML system too. Additionally, [24] presents BLAS [25] a Bi-LAbeling based XPath processing System, as a generic and efficient system for XML storage and XPath query processing by leveraging relational databases. Also, it represents algorithms for translating XPath query to SQL query.

What makes our approach different and new is that it works on double sense, we mean that there is a way to query Relational over XML and vice versa. In other words, it is possible for users

to access and extract data with either SQL or XPath from heterogeneous models (Relational and XML case), so they do not need to control the use of the two different types of query languages to retrieve data on different database systems. It is an independent system on either data model and query language as well. On the other hand, our approach is valid with or without the need of storing XML in RDBMS also it eliminates the need of additional learning of another language to manipulate hybrid data such as SQL/XML, just SQL or XPath can do the task.

3 PROPOSED SYSTEM

3.1 Objective

Each database query language is specific to a particular data model, for example, SQL to extract data from the relational and XPath or XQuery for querying XML. Then, it is difficult for those users to retrieve data because they need the correspondent query language. Hence, in order to overcome this and make it easier for them to get what they want with less effort, we aim to make one query - no matter what the query language can be - sufficient to retrieve data even if it is none the correspondent data model. Also, the user does not have to store or manage XML data using RDBMS also no need to make any physical changes at the databases level.

We discussed here a way to handle the problem of integrating relational and XML data to support both XPath and SQL queries. Figure 1 explains this goal, it shows that with SQL we can extract data in relational and in XML Model, the same as with XPath.

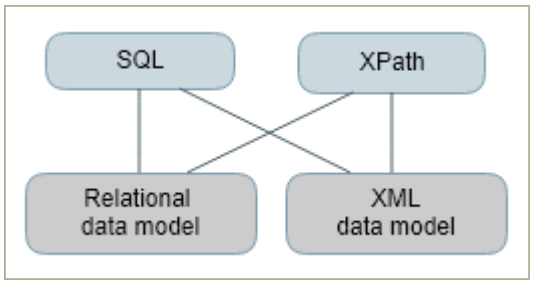


Figure 1. Supporting SQL and XPath to retrieve data from XML and relational data model

3.2 Challenges

Many defies may make the task difficult, the most challenging aspect that needs to be addressed is how to efficiently bridge semantics gaps between these technologies. 'Table 1' shows some of these differences in brief.

Table 1. XML and Relational differences

Relational	XML
Regular structure	Heterogeneous structure
Flat data	Nested elements on several levels
The order has no importance	The Order has an importance
Static schemas	Schemas tend to be more extensible
Always have a schema	May or may not have a schema

3.3 The infrastructure

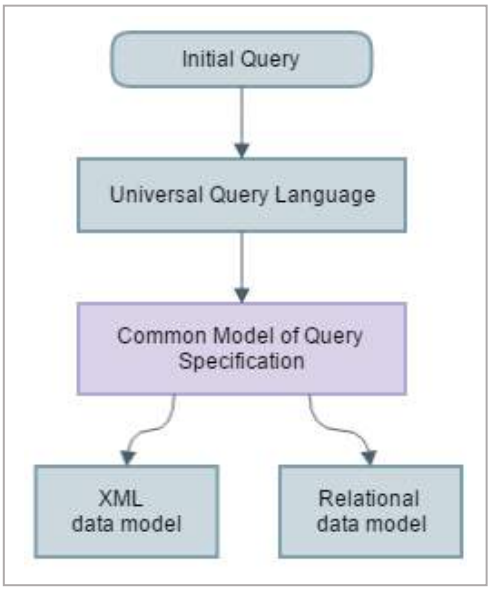


Figure 2. Logical infrastructure of the system

The groundwork for the bulk of our approach involves two phases, as presented in Figure 2. In the first phase, we generate a universal Query language (UQL), where the input is the user Query. In the second one, we identify the query through the Common Model of query specification (CMQS).

The Initial query can be written with either SQL or XPath. UQL presents our Intermediate Query Language (IQL). Why using an IQL? Because it can operate on the lowest level semantics, can increase possibilities to use more

transformations and optimizations, can be an aid to switch between several query languages, and conversion between two languages will be through it.

CMQS is the abstract layer where we specify queries against the specific data model to extract data.

As shown in Figure 3, the procedure begins with one query, which will be decomposed into a set of sub-queries, each query interrogates the suitable model resulting from an answer. Then, the answers to all these queries are recomposed to form an answer to the initial query.

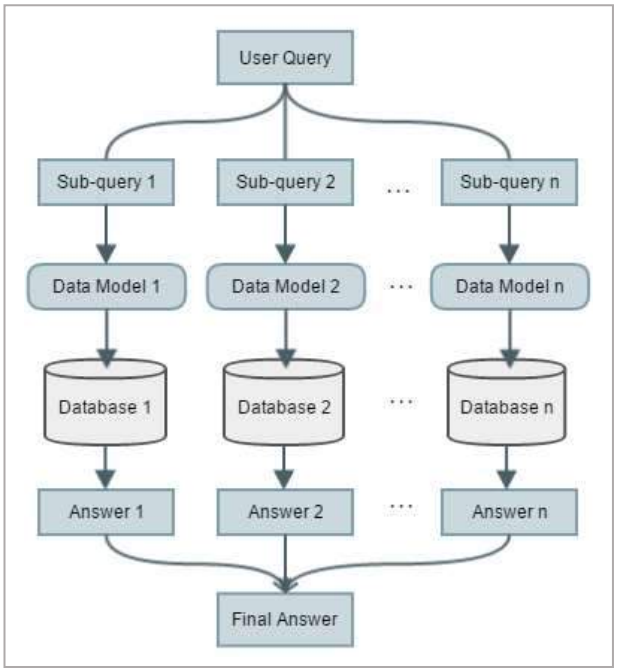


Figure 3. Retrieving data from heterogeneous models

In addition, we will be able to provide the user the answer according to the query language used in the first place. For instance, if the initial query was made using SQL, then the result will be displayed in a tabular form.

4. CONCLUSION & FUTURE WORK

We try to extract data with both SQL and XPath. In fact, even if query languages are specific to a particular data model, we will be able to query a data model with the database query language of the other (with the non-corresponding Query language of the concerning model). Hence, we have built a system that can extract data independently from the query language and the storage model of data.

The relational model is the most data model used to manage data for years. Similarly, XML is rapidly becoming more and more popular and its importance as a standard format for the exchange of information with a management more and more powerful of the documents is very remarkable in the last years. This creates the need of building some bridge between the two. Due to that, we choose to make them the first models under study in this contribution.

In the future work, we will aim to integrate further data models to our system, so it can be independent of any data model as possible as we can, at least for the most known ones.

REFERENCES

1. D. Florescu and D. Kossmann, “Storing and Querying XML Data using an RDMBS,” IEEE Data Eng. Bull., vol. 3, pp. 27–34, 1999.
2. J. Shanmugasundaram, E. Shekita, J. Kiernan, R. Krishnamurthy, E. Viglas, J. Naughton and I. Tatarinov, “A general technique for querying XML documents using a relational database system,” ACM SIGMOD Rec., vol. 30, no. 3, p. 20, 2001.
3. M. Rys, D. Chamberlin, and D. Florescu, “XML and Relational Database Management Systems : the Inside Story,” SIGMOD ’05 Proc. 2005 ACM SIGMOD Int. Conf. Manag. data, pp. 945–947, 2005.
4. J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. De- Witt, and J. Naughton. Relational Databases for Querying XMLDocuments: Limitations and Opportunities. In VLDB, 1999.
5. Oracle Corporation, “Oracle XML DB : Choosing the best XMLType Storage Option for Your Use Case,” no. October, 2009.
6. D. Mann and P. Northwest, “Iso / Iec Jtc 1 / Sc 32 N00575 H2-2000-331R2,” pp. 1–6, 2000.
7. A. Eisenberg and J. Melton, “SQL/XML and the SQLX Informal Group of Companies”, ACM SIGMOD Record, Vol. 30 No. 3, Sept. 2001
8. A. Eisenberg and J. Melton, “SQL/XML is making good progress,” ACM SIGMOD Rec., vol. 31, no. 2, p. 101, 2002.
9. A. Eisenberg, J. Melton, “Advancements in SQL/XML,” SIGMOD Rec., vol. 33, no. 3, pp. 79–86, 2004.
10. R. Murthy and S. Banerjee, “Xml schemas in Oracle XML DB,” Proc. 29th Int. Conf. Very large databases, vol. 29, pp. 1009–1018, 2003.
11. M. Krishnaprasad, Z. H. Liu, A. Manikutty, J. W. Warner, V. Arora, and S. Kotsovolos, “Query Rewrite for XML in Oracle XML DB,” Data Base, 2004.

12. Z. H. Liu, M. Krishnaprasad, and V. Arora, "Native XQuery processing in oracle XMLDB," Proc. ACM SIGMOD Int. Conf. Manag. Data, pp. 828–833, 2005.
13. A. Halverson, V. Josifovski, G. Lohman, H. Pirahesh, and M. Mörschel, "ROX : Relational Over XML," Proc. 30th Int. Conf. Very Large Data Bases, pp. 264–275, 2004.
14. S. Jigyasu et al., "SQL to XQuery translation in the aquaLogic data services platform," Proc. - Int. Conf. Data Eng., vol. 2006, p. 97, 2006.
15. P. M. Vidhya and P. Samuel, "Query translation from SQL to XPath," 2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc., pp. 1749–1752, 2009.
16. H. A. Kore, S. D. Hivarkar, N. K. Pathak, R. S. Bakle, and P. S. S. Kaushik, "Querying XML Documents by Using Relational Database System," vol. 3, no. 3, pp. 5322–5324, 2014.
17. K. Bhargavi and H. S. Chaithra, "Join queries translation from SQL to XPath," 2013 IEEE Int. Conf. Emerg. Trends Comput. Commun. Nanotechnology, ICE-CCN 2013, no. Iceccn, pp. 346–349, 2013.
18. I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and querying ordered XML using a relational database system," 2002 ACM SIGMOD Int. Conf. Manag. Data, SIGMOD'02, no. October, pp. 204–215, 2002.
19. Y. Bin Chiu, H. H. Chen, C. Y. Liu, S. C. Chen, and C. W. Hung, "Efficient Storage and Retrieval of XML Documents Using XQuery," Adv. Mater. Res., vol. 779–780, pp. 1685–1688, Sep. 2013.
20. I. Manolescu, D. Florescu, and D. Kossmann, "Answering XML Queries on Heterogeneous Data Sources.," Vldb, vol. 1, pp. 241–250, 2001.
21. I. Manolescu, D. Florescu, D. Kossmann, F. Xhumari, and D. Olteanu, "Agora: Living with XML and relational," Vldb, pp. 623–626, 2000.
22. W. Fan, J.X. Yu, H. Lu, J. Lu, R. Rastogi, Query translation from XPATH to SQL in the presence of recursive DTDs, in: Proceedings of International Conference on Very Large Data Bases (VLDB), 2005, pp. 337–348
23. D. DeHaan, D. Toman, M. P. Consens, and M. T. Özsu, "A comprehensive XQuery to SQL translation using dynamic interval encoding," Proc. SIGMOD 2003, pp. 623–634, 2003.
24. Y. Chen, S. B. Davidson, and Y. Zheng, "A bi-labeling based XPath processing system," Inf. Syst., vol. 35, no. 2, pp. 170–185, 2010.
25. Y. Chen, S. B. Davidson, and Y. Zheng, "BLAS: An Efficient XPath Processing System," Proc. ACM SIGMOD Int. Conf. Manag. data, pp. 47–58, 2004.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 50-55
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Querying XML and Relational Data

Hassana NASSIRI[1, a] Mustapha MACHKOUR[2, a] Mohamed HACHIMI[3, b]
[a] Laboratory of the Computing Systems and Vision
[b] Laboratory of Engineering Sciences
University Ibn Zohr, Agadir, Morocco
[1] hassana.nassiri@edu.uiz.ac.ma
[2] machkour@hotmail.com
[3] m.hachimi@uiz.ac.ma

ABSTRACT

There has been a growing need for querying heterogeneous data sources, namely XML and Relational databases. Since the relational model is the most data model used to manage data for years. Similarly, the eXtensible Markup Language (XML) is quickly emerging as the de facto standard for data exchange over the Internet. Hence, bridging these two models is surely need. Furthermore, each database system uses a particular query language to manipulate data. So, users need to know each query language of each data model. To this point, we aim to define a system to retrieve data regardless of the nature of the model used and eliminates the burden of learning new languages. In such way, the existing users' knowledge about a query language will be enough and will meet the purpose. Thus, this paper addresses the problem of accessing both XML and relational data, by using a unique query language expressed with whether SQL or XPath. We rely on a new approach in the translation process to convert the user query into the suitable query language according to the nature of the data interrogated.

KEYWORDS

SQL; XPath; Data Model; XML; Relational Database; Translation; Mapping

1 INTRODUCTION

We choose to discuss the eXtensible Markup Language (XML) and relational data model and make them the first models under study in this contribution, as they are increasingly complementary and related in practice. The large volume of the research introduced to integrate these two models is another proof that both relational and XML databases should exist. The Relational Model has long been a successful, dominated and popular Data model, and still widely used by most organizations to manage data. Besides, the Structured Query Language (SQL) is the most popular, basic and the standard query language for managing and querying data. SQL is user-friendly, and almost everything we need to do in manipulating a database can be fulfilled using SQL. Similarly, in the last years, XML is quickly emerging as the de facto standard for data exchange over the Internet. So, bridging these two models is necessary.

Each database system uses a particular query language to manipulate data. So, to query a database, users need to adapt the query language on it, for instance, use SQL to extract data from relational and XPath, for example, to get data from XML. But most of the time, it is a hard task for users to learn new query languages and to control the use of all of them. To this point, we aim to define a system to extract data regardless of the nature of their model. By our system, the existing users' knowledge about a query language will be enough to get what they want. They can use SQL as they can use XPath to retrieve data from both XML and relational data model.

The rest of the paper is organized as follows: Section 2 provides some related works. Section 3 discusses the objectives of the system and explains the translation process to convert queries. Finally, section 4 draws the conclusion.

2 RELATED WORK

There has been considerable interest related to XML and relational database. We focus on those approaches that have proposed a query translation tool to solve the problem of querying different databases. From systems studied in the literature, we have noticed two ways to handle

the problem and met our purpose: (1) translate SQL to an XML query language XPath or XQuery to query XML database (2) the opposite, translates an XML query language to SQL. We will divide them into two parts: from relational to XML [1] [2] [3] [4] [5] [6] and from XML to relational [7] [8] [9] [10] [11] [12] [13]. In the rest of this section, we will give a brief description of each research aforementioned.

[1] discusses a way to query XML documents using relational database system by designing an SQL Interface for XML databases that can convert SQL queries to XPath expressions and extract data from XML documents.

[2] proposes a framework which permits users to access XML databases using SQL. It is an automatic converter of the user's SQL join (Left, Right, and Full) queries into XPath expressions and describes the detailed steps of SQL to XPath conversion along with their algorithms.

[3] examines a way to modify XML data using SQL (INSERT and UPDATE) by proposing a framework to translate these SQL queries to XUpdate expressions, and presents the algorithms to do so. Users can manipulate the XML data through the query language SQL or XUpdate.

[4] introduces a framework to transform SQL statements to XPath expressions, so that accessing XML and relational database can be done using SQL.

[5] proposes a framework that can transform easily SQL statements to XQuery expressions. So users can access both XML and relational database through SQL using their framework.

[6] the ROX explains an approach to translate SQL to XQuery and to query XML data stored in its native format, through SQL interfaces.

[7] uses Relational Database Management System (RDBMS) to store and retrieve XML documents by translating them into tables in RDBMS and storing them in a shredded schema. The user can use XQuery query language, which is translated into SQL Queries. Then, the returned result will be XML documents.

[8] discusses algorithms to translate queries from XPath to SQL. And presents a Bi-LAbeling based System: BLAS [9] for Processing XPath queries over XML data.

[10] proposes an approach to translating a practical class of XPath queries over (recursive) DTDs to SQL queries. Also, it discusses algorithms for rewriting an XPath query over a recursive DTD.

[11] presents a translation of XQuery expressions from a comprehensive subset of XQuery into a single SQL expression.

[12] Presents Agora data integration system [13], a way to integrate relational and tree-structured data sources, in particular XML, under an XML global schema, and to translate queries from XQuery to SQL.

The novelty of our work lies in its ability to retrieve data regardless of the nature of the models with any query language of these models. That means that users can use XPath or SQL to retrieve data from XML or/ and relational databases. In fact, making one query sufficient to retrieve data from heterogeneous databases is among our goals, since users and even developers may not be familiar with many query languages at a time. Using our system, whatever the query posed users can extract data stored in different databases.

3 PROPOSED SYSTEM

As we know, there is a correspondence between the data model and its query language. For instance, we use SQL to extract data from relational and XPath, for example, to get data from XML. That means that users need to know each query language of each data model. Mostly it is a difficult task for users to support all of these query languages because each language has a defined syntax, particular specification and probably difficult to learn.

The principal objective of our system is to define a way to query XML and Relational data sources using one single query expressed with either SQL or XPath as explained in Figure 1. And figure out an efficient method to translate the user query into the suitable query language according to the nature of data interrogated. We think that it is more convenient to build an

intermediate format to pass between phases rather than repeat the whole procedure for each query language and break down the translations process into multiple steps. Each one has a particular task to accomplish. The architecture of our system in a whole is detailed in Figure 2, with all the components, which they are in turn explained in the following sections.

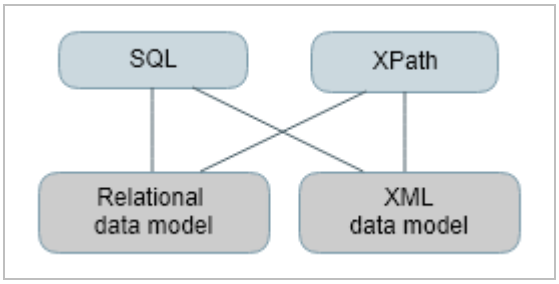


Figure 1. [14] One query to retrieve data from XML and/or relational data models

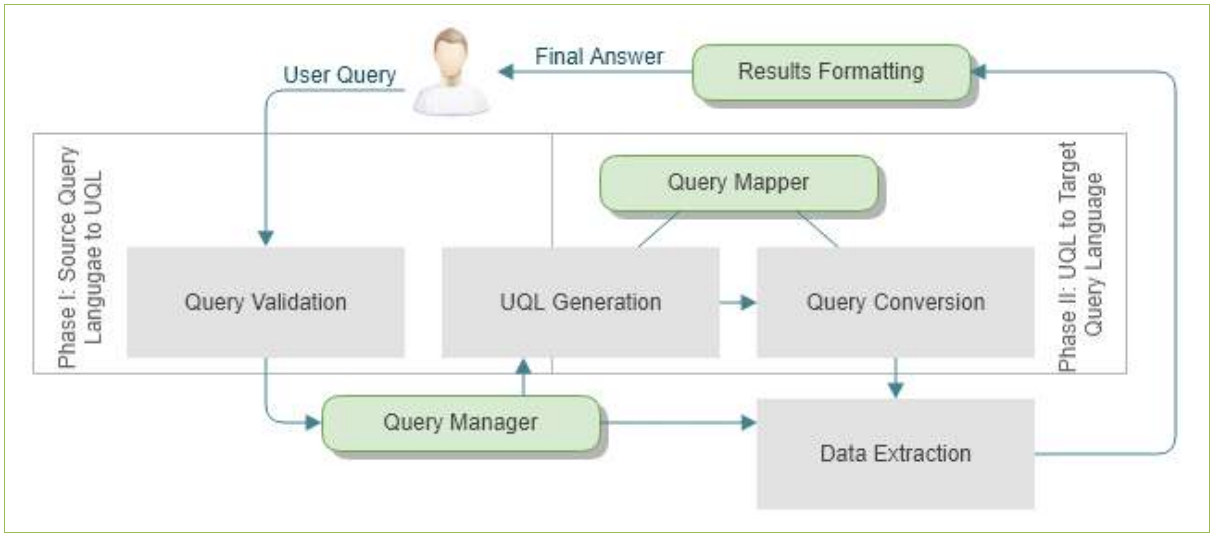


Figure 2. Architecture of the overall system

3.1 Intermediate Representation

We inspired from the compilation notion and adopted the same principle in our approach to translating queries. The system reads the Source Query Language, then translated into an Intermediate Query Language(IQL) which is then translated to its Target Query Language as shown in Figure 3.

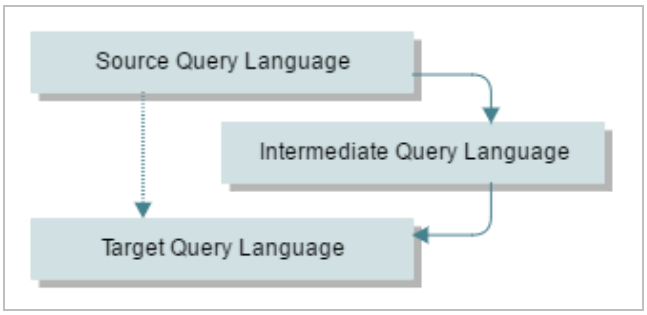


Figure 3. The principal of the translation

Why at all we need an IQL in the translation process, instead of simply translate the query directly into its target and skip the intermediate presentation phase? In fact, if we translate the source language into its target language without generating intermediate language, then for each new language, a full native translation is needed. Using intermediate code eliminates this need by keeping a portion same for all. Then the second part is changed according to the target language. Thus, it becomes easier to add more languages in our system.

The benefits from using an IQL are multiple, we cite for example the fact that it is more general and capable of representing different Query Languages; can operate on the lowest level semantics; can increase possibilities to use more independent transformations and optimizations; can be an aid to switch between several query languages, and conversion between two languages will be through it, also it enables the system to be broken up into multiple components, more manageable and simpler, thus benefiting from modularity.

3.2 Universal Query Language

In our system, we refer to IQL by the universal query language, the representation of a query between the source and target query languages. We know that a good intermediate representation is one that is independent of the source and target languages so that it maximizes its ability to be used in different cases. For that, XML was our candidate to represent our IQL. XML allows separating the contents of the presentation. This makes it possible, for example, to display the same document on different applications or devices without necessarily creating as many versions of the document as we need of representations. Also, it has many qualities such as readability, universality, portability, integrability, and extensibility.

3.3 How it Works?

In this section, we will explain the operation of the system, which consists of four phases as shown in Figure 4: (1) query validation, (2) UQL generation (3) Query conversion and (4) Data extraction.

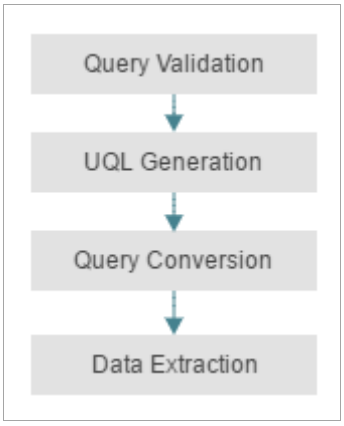


Figure 4. The system's phases

The procedure begins with the user query as expressed in Figure. 5, then verified in the query generation phase, if it is valid we continue the rest of the process, else we ask the user to enter a valid query, this is repeated until getting a valid query. After that comes the query checker role, to decide if the query really should be translated. If that the case we continue the translation process and use the query mapper module to map between each part of the query with the correspondent one in our UQL, else we skip it and pass the query directly to the query executor, then extract data and finally format the result and return the final answer to the user.

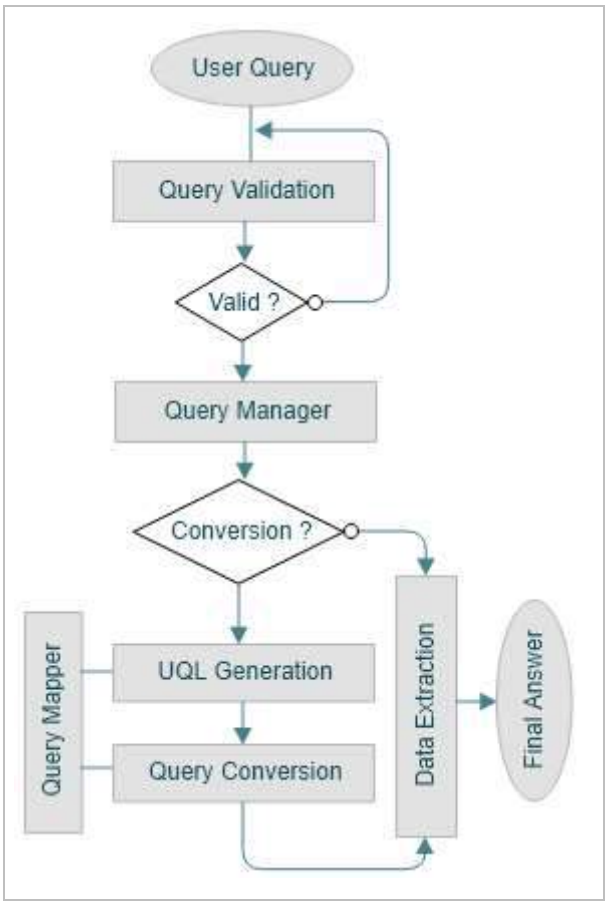


Figure 5. How it works

3.4 Translation process

In this section, we present how to convert queries from a language to another. As shown in Figure. 6 the translator suggested herein maps each input of a query language to an output. To do so the procedure is broken down into multiple phases, and each phase has a defined task. The translation process includes two parts: I. from the source query language to UQL, II. From UQL to target query language. This happens through the first three phases of the system: (1) Query Validation (2) UQL Generation and (3) Query Conversion.

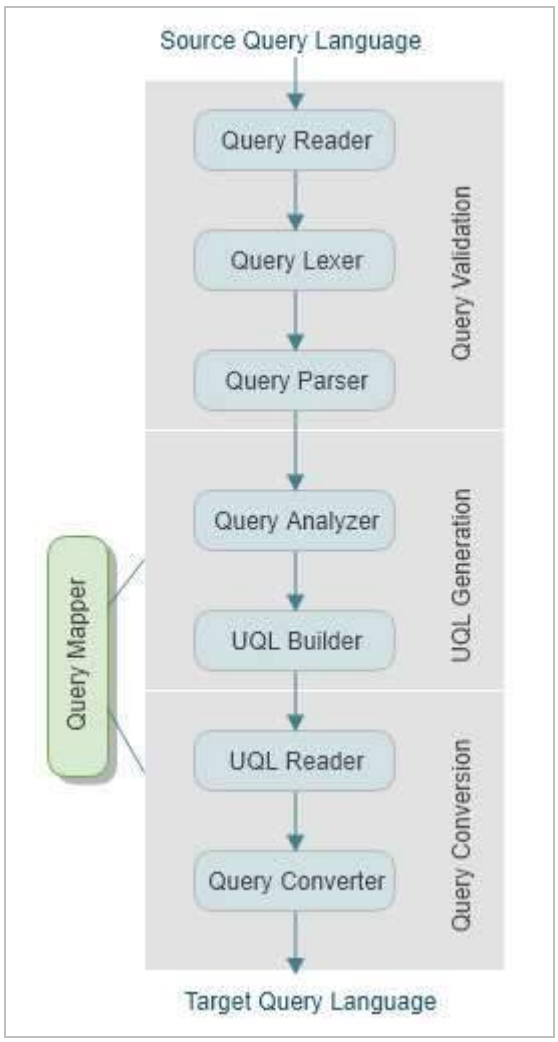


Figure 6. The translation process

3.4.1 Query validation

The Query validation starts with syntax verification of the input given by the user which is analyzed then to extract its portions to appropriate variables. The semantic information's related to the query are captured here.

- Query Reader: provides a uniform interface between users and system, and read their queries.
- Query Lexer: this is the lexical analysis or scanning phase, the lexer converts the groups of characters of the query written by the user into tokens.
- Query Parser: the query parser converts the groups of tokens into a parse tree, a data structure that reflects the syntactic structure of the input query.

3.4.2 UQL generation

The UQL Generation phase is responsible for building the UQL via the information received from the query validation phase. It has two components:

- Query Analyzer: In this step, we split the information extracted and taken it into variables which will be used to proceed further operations.
- UQL Builder: After getting all the needed variables from the query analyzer, it becomes easier to form the UQL by filling each part of it with the suitable information retrieved from these variables using the Query Mapper.

3.4.3 Query Conversion

The Query Conversion phase where actually the translation takes place by the query converter with the aid of the query mapper. Again, it consists of two components.

- UQL Reader: recalls the query mapper again to prepare the UQL parts that should be mapped with the target query language.
- Query Converter: Now that we have our intermediate format, all what we need to do is convert it into the target query language which is chosen according to the nature of data source.

3.5 Query Manager

Not much work is involved in to cases: (i) if the user uses SQL to query relational databases (ii) if he or she uses XPath to query XML database. In other cases, the translation is proceeded. This kind of decision is made by the query checker, it is the one that make sure if we really need to enter the translation process or skip it and directly execute the query and extract data.

3.6 Query Mapper

Contains the steps to link each part of a query to the suitable part of the UQL, and vice versa.

4. CONCLUSION & FUTURE WORK

We try to eliminate the burden of learning new languages so that querying each data model with the correspondent query language is not a problem anymore; users can use whether SQL or XPath to query both XML and relational data. We rely on a new approach in our translation process to convert the user query using an intermediate query language as an intermediate format to pass between the source query language and its target according to the nature of the data interrogated.

XML and relational models were the first models to discuss in our research because of all the attention they received in data integration filed since they have some powerful tools and many merits in either data exchange and data management and extraction. So, until now, only XML and relational data model are supported in our system, but in future work, we aim to enhance the features, and the merits of the system proposed, to integrate other data models and support other query languages.

REFERENCES

1. H. A. Kore, S. D. Hivarkar, N. K. Pathak, R. S. Bakle, and P. S. S. Kaushik, “Querying XML Documents by Using Relational Database System,” vol. 3, no. 3, pp. 5322–5324, 2014.
2. K. Bhargavi and H. S. Chaithra, “Join queries translation from SQL to XPath,” 2013 IEEE Int. Conf. Emerg. Trends Comput. Commun. Nanotechnology, ICE-CCN 2013, no. Iceccn, pp. 346–349, 2013.
3. M. Vidhyap and P. Samuel, “Insert Queries in XML Database,” pp. 9–13, 2010.
4. P. M. Vidhya and P. Samuel, “Query translation from SQL to XPath,” 2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc., pp. 1749–1752, 2009.
5. S. Jigyasu et al., “SQL to XQuery translation in the aquaLogic data services platform,” Proc. - Int. Conf. Data Eng., vol. 2006, p. 97, 2006.
6. A. Halverson, V. Josifovski, G. Lohman, H. Pirahesh, and M. Mörschel, “ROX : Relational Over XML,” Proc. 30th Int. Conf. Very Large Data Bases, pp. 264–275, 2004.
7. Y. Bin Chiu, H. H. Chen, C. Y. Liu, S. C. Chen, and C. W. Hung, “Efficient Storage and Retrieval of XML Documents Using XQuery,” Adv. Mater. Res., vol. 779–780, pp. 1685–1688, Sep. 2013.
8. Y. Chen, S. B. Davidson, and Y. Zheng, “A bi-labeling based XPath processing system,” Inf. Syst., vol. 35, no. 2, pp. 170–185, 2010.
9. Y. Chen, S. B. Davidson, and Y. Zheng, “BLAS: An Efficient XPath Processing System,” Proc. ACM SIGMOD Int. Conf. Manag. data, pp. 47–58, 2004.
10. W. Fan, J. X. Yu, J. Li, B. Ding, and L. Qin, “Query translation from XPath to SQL in the presence of recursive DTDs,” VLDB J., vol. 18, no. 4, pp. 857–883, 2009.
11. D. DeHaan, D. Toman, M. P. Consens, and M. T. Özsu, “A comprehensive XQuery to SQL translation using dynamic interval encoding,” Proc. SIGMOD 2003, pp. 623–634, 2003.
12. I. Manolescu, D. Florescu, and D. Kossmann, “Answering XML Queries on Heterogeneous Data Sources.,” Vldb, vol. 1, pp. 241–250, 2001.
13. I. Manolescu, D. Florescu, D. Kossmann, F. Xhumari, and D. Olteanu, “Agora: Living with XML and relational,” Vldb, pp. 623–626, 2000.
14. H. Nassiri, M. Machkour and M. Hachimi, "Integrating XML and Relational Data", Procedia Computer Science (2017) pp. 422-427.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 56-61
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Introduction to Sociology of Online Social Networks in Morocco. Data Acquisition Process: Results and Connectivity Analysis

Yassine EL Moudene[1], Jaafar Idrais[2] and Abderrahim Sabour[3]
High School of Technology Ibn Zohr University, BP: 33/S 80000 Agadir:Morocco
http://www.esta.ac.ma/
[1]yassine.elmoudene@gmail.com [2]jaafar.idrais@gmail.com [3]ab.sabour@uiz.ac.ma

ABSTRACT

The aim of this paper is to study the Moroccan active community behavior on online social networks, firstly the choice of Facebook as OSN is justified by the statistics that ranks it in the first position compared to other OSNs. The lack of a specific database dedicated to Moroccan community requires the implementation of a data acquisition process. In second part, this paper presents macro data connectivity analysis, visualization was made by applying ForceAtlas2 layout algorithm.

KEYWORDS

Online social networks, Facebook, Data Extraction, Community Detection, Intra-group connectivity, Layout algorithm, ForceAtlas2.

1 INTRODUCTION

Sociology as quite other science saw its axles of influence as well as its objectives evolved or even transferred in the course of this 20 last years, the analysis of the new communication technologies impact NTIC has create a sub-industry devoted, the arriving of web2.0[1] had a prodigious impact since he has allows to people to pass from a simple receiver to a more important role or he participates in the contents production, what allows the generation of huge data quantity in a period of time. The interlocking of the user in the contents generation process also allows measuring importance of an online social network in comparison with other one.

2 ONLINE SOCIAL NETWORKS (OSN)

2.1 Definition

An online social network is a service [2] that allows individuals to:

- ✓ Build a public or semi-public profile in a delimited system.
- ✓ Articulate other users list with which they share a connection.
- ✓ View and browse their connections list and those made by other users in the system.

So an online social network is an Internet community where individuals interact, often through profiles that represent their personalities and their networks.

2.2 OSN Evolution

SixDegrees.com [3] is the first recognizable social networking site launched in 1997. The next wave of OSN began with Ryze.com in 2001. Friendster was launched in 2002 as a social complement to Ryze [4]. As of 2003, many new OSNs have been launched, while MySpace [5] has attracted the attention of the majority of the media in the US and abroad, OSNs growing in popularity worldwide. Friendsk won the attraction in the Pacific Islands, Orkut became the first OSN in Brazil before growing rapidly in India, Mixi adopted a widespread adoption in Japan, LunarStorm took off in Sweden, Dutch users have embraced Hyves, Grono Has captured Poland, Hi5 was adopted in the small countries of Latin America, South America and Europe, and Bebo has become very popular in the United Kingdom, New Zealand and Australia. Facebook[6] debuted in early 2004 as a Harvard OSN only, starting in September 2005, Facebook has grown to include high school students, professionals within the corporate networks and eventually everyone.

2.3 OSNs Popularity Comparison

The world map[1] showing the most popular OSNs

[1]HTTP://VINCOS.IT/WORLD-MAP-OF-SOCIAL-NETWORKS/

by country in 2017, according to Alexa[2] traffic data & SimilarWeb. Facebook is the main social network in 119 of the 149 countries analyzed, but was arrested in 9 territories by Odnoklassniki, Vkontakte and Linkedin. Interestingly, in some countries, such as Botwana, Mozambique, Namibia, Iran and Indonesia, Instagram wins and some African territories prefer LinkedIn. Overall, LinkedIn is in 9 countries, Instagram 7, while VKontakte and Odnoklassniki grow in Russian territories. In China, QZone [7] still dominates the Asian landscape and Japan is the only country where Twitter is the leader.

3 FACEBOOK

3.1 Definition

Founded in 2004, Facebook's mission [8] is to empower people to share and make the world more open and connected.
%People use Facebook to stay connected with friends and family, to discover what is happening in the world and to share and express what is important to them. Facebook also allows people to share their opinions, ideas, photos and videos, and other activities with Audiences ranging from their closest friends to the general public

3.2 Facebook Usage Statistics

At December 31, 2016, Facebook has 17,048 employees [9], with 1.87 Billion [9] Users, Facebook is the most popular social network in the world. This large number of users allows having huge productions in information every minute or even every second.

- ✓ Daily Active Users (DAU): Facebook defines DAU [9] as a registered user of Facebook who logged in and visited Facebook via his website or a mobile device, or used his Messenger application.
- ✓ Monthly Active Users (MAU): Facebook defines a MAU [9] as a registered user of Facebook who has logged in and visited Facebook via our website or mobile device, or Uses the Messenger application within the last 30 days from the date of the measurement.

[2] HTTP://WWW.ALEXA.COM

Table 1. Number of Facebook Active Users In 2016

	Average (Md)	Mobile (Md)
DAUs	1.23	1.15
MAUs	1.86	1.74

Global DAUs increased by 18%, from 1.04 billion in December 2015 to 1.23 billion on averages in December 2016. In June 27th, 2017, Marck Zuckurberg publishes on the official page that Facebook contain 2 billion users accompanied by a picture [10] of those users. The evolution among the Facebook users can be pointed out by comparing the previous picture with this [11] published in 2013.

3.3 Facebook in Morocco

The proportion [12] of Moroccan Internet users increased slightly from 2014, from 56.8% to 57.1% in 2015. 17.8 million[12] Moroccans connected to the Internet in the last three months of the year, Year 2015.Facebook arrives at the top of social networks frequently visited with a percentage of 93%, followed by whatsApp with 85% .On table 2, global statistics of different OSNs used in Morocco are given:

Table 2. Most Used OSNs in Morocco

OSN	% Users	Users Number
Facebook	93.0	16 554 000
Whatsapp	84.9	15 112 200
Skype	19.8	3 524 400
Instagram	19.8	3 524 400
Google+	18.1	3 221 800
LinkedIn	5.4	961 200
Twitter	11.7	2 082 600

4 SEMI-SUPERVISED EXTRACTION PROCESS

Online social networks are part of the Web, but their data representations are very different from the general web pages. Web pages that describe an individual, a page, a group, in an OSN are generally well structured, as they are usually generated automatically, unlike general web pages that could be written by a person. The lack of fact that we seek to profile the behavior of the Moroccan community from interactions and reactions analysis has necessitated the creation of a data set adequate for this purpose.

4.1 Inaccessible and Inadequate DataSet

Before starting data retrieval, a search for Existing Datasets, for a such study, was initially done, although the existing Datasets are inaccessible [13][14], they are not adequate to the studied problem, thus comes the need for Defines a semi-supervised data acquisition process for the collection of a specific database meeting the requested requirements.

4.2 Semi-Supervised Extraction Process

To extract this data, a queued data structure to store the groups list and pending pages to be crawled is set up. Initially, by referring to the social media analysis Websites as SocialBakers [15] a list of most consulted pages and groups by the Moroccan community feeds the starting list of the process. Then, at each stage, the first entry in the queue is finalized: all publications, comments on publications, sub-comments (comment comments), and reactions (towards publications and comments) for this entry (Group or page) are scanned to extract the different types of information. The figure 1 display the semi-supervised collection process followed.

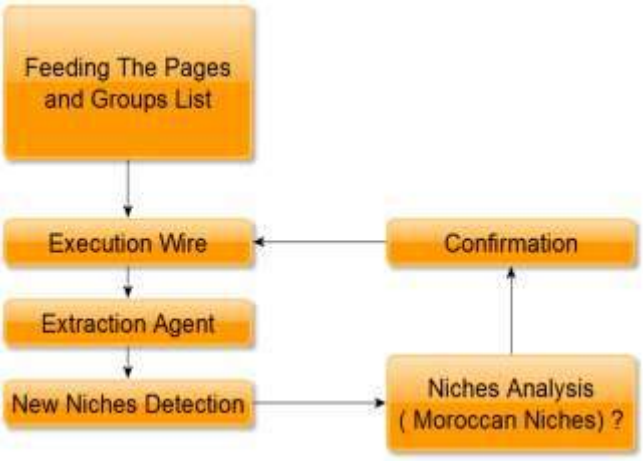


Figure1. Le processus semi-supervisé d'acquisition des données

4.3 Extracting Data Description

4.3.1 Extracted Data Structure

An user can stick to a group, Like A Page, Publish a publication, Comment on a publication or on a comment, React to a publication or to a comment, Share a publication, or Tag (identify) another user in a publication or a comment. So, by going through these different types of information, different connections graphs between all users can be constructed. To assure the coherence and the data structure evolution, a MySQL database is used for the agents. It allows to unfold new agents without being subjected of time to stop or to shape, while supporting a proportional increase of performances. It is important to note that the failure of one agent does not affect the operation of others.

4.3.2 Extraction speed

However, in order to guarantee a browsing speed, we have set up a set of machines called Extraction Agent, each of these agents takes care of the extraction of the various information related to specific groups, at the end of the extraction of all types of information from of a group (page), This latter is marked as exploring, to avoid a re-run of the group (page).

4.3.3 Accessible Data

Online social networks are inherited from complex dynamic systems, the faculty of making accessible only a visible part and hiding a large part of it like an iceberg. During the data extraction, some limitations are imposed by Facebook.

4.3.4 Information Dissipation

Despite that a serious study has not yet been started on the extracted data, the system has triggered the disappearance of a set of nodes (group, pages, feeds, etc).

5 RESULTS AND PRELIMINARY ANALYSIS OF CONNECTIVITY

5.1 Extracted Data Statistics

Since the launch of the data extraction script on 01 January 2017 at the level of the various agents, up to the time of the writing of these lines, 2000 Moroccan groups and pages have been explored, this operation has made it possible to recover different Types of information cited above, the database is a size of 15 Gigabyte and contains about 100 Millions

records. Table 3 detail different extracted information:

Table 3. Extracted Information Global Statistics

Table	Records number	Size (en Mib)
Users	13 096 530	1 900
Groupes/Pages	2 164	1,2
Members	4 662 253	212
Feeds	5 733 444	3 600
Comments	21 652 626	5 400
Reactions	49 551 978	4 000
Tags	1 061 392	144
Total	95 760 387	15 256,2

5.2 Intra-Group Connectivity

The extracted data is the product of user behavior within Facebook. Using this information, the relationship between the groups (pages) is measured (via the calculation of the common user number between these groups) and then visualized (using the ForceAtlas2 algorithm).

5.2.1 Visualization with ForceAtlas2 Algorithm

ForceAtlas2 [16] is a spatialization algorithm implemented in the Gephi software. It allows for a force-directed layout. The nodes repel like charged particles, while the edges attract their nodes, like the springs. These forces create a movement that converges to a balanced state.

ForceAtlas2 Algorithm

Input: Undirected graph G = (V, E), iterations, gravitational and repulsive force scalars fg and fr.
Output: A position pv ∈ R2 for each v ∈ V.

```
globalspeed ← 1.0
   For all v ∈ V do ➔ Initialize variables
      pv = random ()
      fv = (0.0, 0.0) T ➔ Net force on node v
      f'v = (0.0, 0.0) T➔f'v is fv of preceding iteration
End For
For i = 1→ iterations do
   BH.rebuild()➔ (Re)build Barnes-Hut tree
   For all a ∈ V do
      fv ← fv-pv.➔ (Strong) Gravity
      fv ← fv+kr.BH.force_at(pv). ➔Repulsion
      For all w ∈ neighbors (v) do
            fv←fv+ (pv-pw)/|pv-pw|. ➔ Attraction
      End For
   End For
   UpdateGlobalSpeed ()
   For all v ∈ V do
      pv ← localspeed(v) * fv ➔Displacement
      f'v ← fv
      fv ← (0.0, 0.0)
   End For
End For
Function LOCALSPEED (v).➔For a node v
   return globalspeed / (1.0+√(globalspeed+swing(v)))
End Function
Function SWING (v) ➔ For a node v
   return |fn-f'n|
End Function
```

5.2.2 Visualization Results

The graphs in Figures 2, 3, 4 and 5 are weighted graphs, the nodes represent the pages (groups) and the stops are the number of the common users having carried out the studied activity within the two nodes. For better readability, we intend to visualize the strongly linked nodes E (V, W)> 1000.

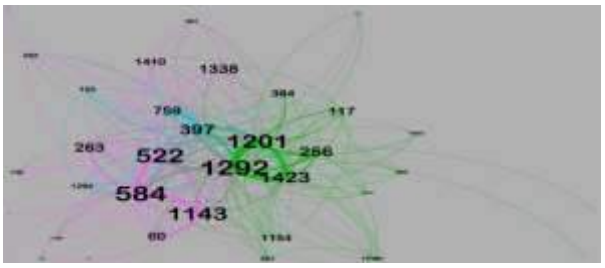


Figure 2. Connection between groups by users who have commented within these groups

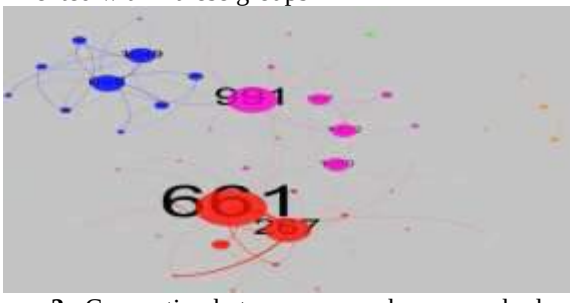


Figure 3. Connection between groups by users who have published within these groups

Figure 4. Connection between groups by users who have reacted to comments

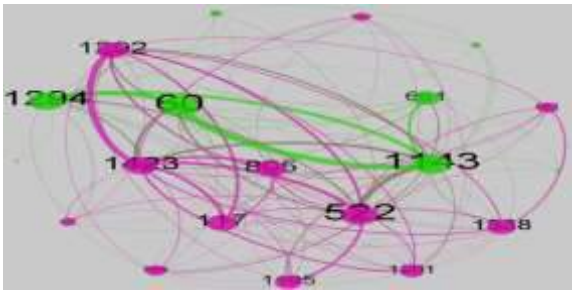


Figure 5. Connection between groups by users who have reacted to publications

For each Type studied, we notice the variation of the number of community extracted, the publications, being the source of the other informtaions, generated us 11 communities.

Table 4. Description of Group Connectivity Results

Type	Nodes	Edges	Degree	Modula-rity	Commun-nity
Comm	37	225	6.081	0.175	3
feeds	67	107	1.597	0.621	11
Like Comm	20	71	3.55	3.55	3
Like Pub	22	119	5.409	5.409	2

5.3 Reactions Statistics

Since Wednesday 24th February 2016, Facebook offers reactions in order to offer the much-claimed alternative to the simple button I like. The table 3 shows a clear view on the extracted reactions distribution.

Table 5. Global Statistics of Extracted Information

Type	Before 24/02/2017	After 24/02/2017
Angry	1	235
HAHA	543	581638
LIKE	6965483	27594050
LOVE	1091	780799
SAD	0	532
THANKFUL	0	0
WOW	50	106603

The button "LIKE" has received a large use percentage (95%), which is normal because before the appearance of these new reaction buttons (before 24/02/2016), All the reactions were a simple "LIKE", hence the interest of neglecting the comparison of this button with the different reaction types. So by neglecting the "LIKE" button we notice that two buttons are the most used, namely "LOVE" (53%) and "HAHA" (39.5%), reflecting a social trend of the Moroccan population, to be emotional and humorous. This observation is validated by the negligible number of the reaction type "Angry" (235 reactions) and of the reaction type "SAD" (532 reactions).

5.4 Publications Statistics

Table 6. Global Statistics of Extracted Information

Type	Feeds	Reaction	Coments	Share
Photos	2020197	12779890	7374995	138346468
Vidéos	701269	4282558	1334805	23976143
Links	1595325	2767789	1530594	18206035
Events	74833	290521	806053	111450
Normal	1441698	7034893	4158864	52453299

To make the publication more visible, the Facebook user enriched it with content: Image, video, event, Link or just leave text. Among all the types of contents cited, the photos are the visual cues that gave rise to the more feedback from users.

CONCLUSION

OSNs are among the most intriguing phenomena of recent years. In this work a description of data extraction from Facebook was made via the semi-supervised process, after general statistics for of Moroccan community Facebook use was given, a macro intra-group connectivity analysis was carried out using graph theory concepts. The visualization is done by applying the spatialization algorithm ForceAtlas2. A general macro-description of the extracted data was developed in the second part, which requires in-depth studies and analyzes in order to extract indicators on the social behavior of the studied population as well as to profile the Moroccan users, those studies are the future axes to be undertaken.

REFERENCES

1. T. Stenger, A. Coutant, “Les réseaux sociaux numériques : des discours de promotion à la définition d’un objet et d’une méthodologie de recherché”, 2010
2. D.M. Boyd, N.B. Ellison, “Social Network Sites: Definition, History, and Scholarship”, 2008
3. I. Ahmed, T.F. Qazi, “A look out for academic impacts of Social networking sites (SNSs): A student based perspective”, 2011
4. K. Danias, A. Kavoura, “The role of social media as a tool of a company’s innovative communication activities”, 2013
5. M.A. Urista, Q. Dong, K.D. Day, “Explaining why young adults use MySpace and Facebook through uses and gratifications theory”, 2009
6. N.B. Ellison, C. Steinfield, C. Lampe, “The Benefits of Facebook Friends Social Capital and College Students’ Use of Online Social Network Sites”, 2007
7. V. Apaolaza, P. Hartmann, J. He, J.M. Barruti, “Shanghai adolescents’ brand interactions on the Chinese Social Networking Site Qzone: A Uses and Gratifications Approach”, 2015
8. Facebook Reports Fourth Quarter and Full Year 2016 Results https://s21.q4cdn.com/399680738/files/doc_financials/2016/Q4/Facebook-Reports-Fourth-Quarter-and-Full-Year-2016-Results.pdf
9. Annual report pursuant to section 13 or 15(d) of the securities exchange act of 1934 For the fiscal year ended December 31,2016 http://d18rn0p25nwr6d.cloudfront.net/CIK-0001326801/80a179c9-2dea-49a7-a710-2f3e0f45663a.pdf
10. Carte facebook publiée en 27/06/2017: https://www.facebook.com/photo.php?fbid=10103832396388711&set=a.941146602501.2418915.4&type=3&theater
11. Carte facebook publiée en 24/09/2013: https://www.facebook.com/photo.php?fbid=10101026493146301&set=a.941146602501.2418915.4&type=3&theater
12. Enquête de collecte des indicateurs TIC auprès des ménages et des individus pour l’année 2015 https://www.anrt.ma/sites/default/files/publications/enquete_tic_2015_fr.pdf
13. P. De Meo, E. Ferrara, G. Fiumara, A. Provetti, “On Facebook, most ties are weak”, 2014
14. Laboratory for Web Algorithmics DataSets : http://law.di.unimi.it/DataSets.php
15. SocialBakers. 2017. Facebook Statistics by Country: https://www.socialbakers.com/statistics/facebook/pages/total/morocco/
16. M. Jacomy, T. Venturini, S. Heymann, M. Bastian, ForceAtlas2, “a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software”, 2014

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 62-67
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Text Classification Using Time Windows Applied to Stock Exchange

Pavel Netolický, Jonáš Petrovský, František Dařena, Jan Žižka

Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

pavel.netolicky@mendelu.cz, jonas.petrovsky@mendelu.cz, frantisek.darena@mendelu.cz, jan.zizka@mendelu.cz

ABSTRACT

Each day, a lot of text data is generated. This data comes from various sources and may contain valuable information. In this article, we use text classification to discover if there is a connection between textual documents (specifically Facebook posts) and changes of the S&P 500 stock index. The index values and documents were divided into time windows according to the direction of the index value changes. In the first experiment, we used a batch processing approach to put the documents from all windows into one data set and a classification accuracy of 62% was achieved. In the second experiment, we used a data stream approach to divide documents into twelve data sets created from two neighboring windows and we achieved an accuracy of 68%. This indicates that posts, which companies write on their Facebook pages, are partially related to the performance of the stock index. Taking the concept change into account also enables better quantification of this relationship.

KEYWORDS

Machine Learning, Classification, Text Mining, Stock Exchange, Time Windows, Data Streams

1 INTRODUCTION

A huge amount of data is constantly being generated by people and organizations. The speed of data creation is rapidly growing and we use the term "data stream" for the constant flow of new data [1].

Data streams may be of various data types (text, image, numeric) and come from different application areas (computer networks monitoring, scientific experiments, internet search, social networks etc.). In comparison to batch processing (for which we have all data available at once), data streams processing needs a different approach, because classical approaches are not effective or even feasible [2].

In this article, we will focus on the connection between text documents published on the Internet and movements of stock prices (represented by a composite value of stock index). Some research in this area uses structured (quantitative) data to analyze the impact of data on stock prices [3]. Unstructured data (like text) may provide us with another complementary information with additional hard-to-quantify knowledge [4].

Behavioral finance theory says that emotions may deeply influence behavior and decision making of individuals as well as whole human societies [5]. This means that the prices on capital markets are (more or less) influenced by emotions, moods and opinions of market participants [6]. These attributes are often contained in text documents and therefore we decided to use text data for our research.

[7] examined the connection between the content of messages posted to a discussion board and movements of the Czech stock index. We will expand this approach further by focusing on the US stock market, using a larger number and another type (Facebook

posts) of text data and treating stock prices and related text documents as data streams divided into time windows as we suppose that the reasons of stock price changes evolve in time.

2 CURRENTLY USED METHODS

To model the behavior of a stock price with a relation to the content of text data we can use classification in a way that we examine the direction of the change of the stock price to create classes. This approach was used for example by [6]. The problem can be seen as text classification – given a text, decide its class (direction of the price movement). However, we must overcome two problems. The first problem is the definition of classes. [8] used a threshold value of 1% price change for the class determination. The second problem lies in choosing correct features. Many studies used just single words and this simple unigram bag-of-words model provided good results in [8].

There exist a wide range of supervised learning algorithm that can be uses for the text classification. An interesting approach is described in [9] – it focuses on sentence-level sentiment analysis of movie reviews. They used the cosine normalization, Term Presence, and Smoothed delta IDF as weighting schemes and the Recursive Neutral Tensor Network algorithm to achieve an accuracy of 87.60%. [10] used Naïve Bayes and SVM as algorithms and unigrams, bigrams, unigrams with bigrams, and unigrams with POS (Parts-of-speech) as features. The bigrams showed a lower accuracy then unigrams – the reason is that the resulting vectors were very sparse. All in all, the type of features used in the bag-of-words model has a little (maximal 2–3%) impact on the accuracy.

3 DATA AND METHODOLOGY

The goal of the work was to examine whether the content of text documents published on the Internet has any connection with stock price movements. We decided to use Facebook posts from company pages as the text data, because it has been a very rarely used data source for this area of research, we have lots of available data, and it might bring new interesting insights.

3.1 Stock prices

In our research, the values of the S&P 500 Index were used to represent stock prices. The index values reflect stock prices of the selected blue chip (large and famous) companies on the US stock market. The historical values of the index were downloaded from the website *investing.com*. For each trading day, we have a closing (end-of-day) numeric value of the S&P 500 Index available.

3.2 Text data

As the text data, posts from Facebook pages of the companies from the S&P 500 Index were used. In total, we examined 431 company pages. The company's Facebook page contains a sequence of documents arranged according to their publication time. These short postings are created by the company representatives. Figure 1 shows an example of a post on the Intel's page. A post may be commented by Facebook users. However, the comments were not used in the analysis.

In total, 138,713 Facebook posts published between 1. 1. 2015 and 15. 10. 2016 were used.

Figure 1. Example of a Facebook post

3.3 Classification methodology

We used text classification to predict whether the given document is connected with an upward or downward movement of the S&P 500 Index.

We examined the time series of the S&P 500 Index values between 1. 1. 2015 and 15. 10. 2016 and found time intervals (windows) in which the change (either positive or negative) of the index value between the first and the last day of the interval was at least 5%. In total, 24 such windows were found. In 12 of them, the index value grew and in 12 it declined. The length (a number of days) of the time windows varied between 4 and 30. Then, each document was, based on the time window in which it was published, assigned a class: 1 (up) for the positive index value change, 2 (down) for the negative one. Figure 2 shows an example of time windows between 1. 1. 2016 and 1. 4. 2016 with the assigned classes.

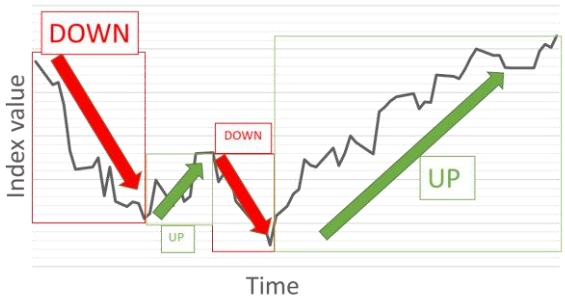


Figure 2. Classification classes identified in the time series of the stock index values

We decided to perform two types of experiments with different data sets used for the classification. In the first experiment, documents from all 24 windows were put into one data set. The results for this experiment are presented in section "Batch approach".

In the second experiment, we divided the documents into 12 data sets. Each data set consisted of the documents from two neighboring windows: one with an upward movement and one with a downward movement. The windows represented two classes for the classification. The results for this experiment are presented in section "Neighboring windows approach".

Text pre-processing and conversion

The raw text of each document was processed by a Python script as follows:

1. Remove all whitespace.
2. Lowercase all letters.
3. Tokenize the document – get words (using *TreebankWordTokenizer*).
4. Filter words – minimal length of three letters, exclude numbers.

The edited text was converted into a structured format by using a Python library called *scikit-learn* and its *Vectorizer* class. Only words that occurred at least 5 times in the whole document collection were included in the resulting vector representation.

The documents were converted to a bag-of-words representation using three different weighting schemes for the term-document matrix [11, p. 21–26]:

- Term Presence (TP): 1 if a term was present in a document, 0 if not.
- Term Frequency (TF): number of times a term was present in a document.
- TF-IDF: TF (local weight) multiplied by the IDF (global weight).

Classification

The converted data was split into the training (60%) and testing (40%) set. Each bag-of-words representation was processed by 10 classifiers (with default settings – no parameter optimization was made) in *scikit-learn*. The classifier's performance was evaluated by the achieved accuracy (proportion of the correctly classified instances on all examined instances [10, p. 268]) on the test set.

4 RESULTS AND DISCUSSION

One set of the text data (Facebook posts) together with the S&P 500 Index values was used to prepare the data for classification. The class-labelled data set was processed using the three weighting schemes (TP, TF, TF-

IDF) by 10 classification algorithms. In total, 30 classification results were obtained.

4.1 Batch approach

Tables 1, 2, 3 and 4 show the results for classification of the data set including all windows.

Table 1. Facebook posts – data statistics.

Total samples	Class 1 samples	Class 2 samples	Number of words
138,713	85,286	53,427	33,397

Table 2. Facebook posts – classification results.

Accuracy	Precision	Recall	F1 score
0.623	0.605	0.623	0.614

Table 1 shows the statistics about the data used for the classification. It is obvious that the data set was quite unbalanced (with more documents marked with index value going up). Table 2 shows the best classification results. The highest accuracy (62%) was achieved with the TF-IDF weighting scheme and the Multinomial Naïve Bayes classification algorithm.

Table 3 tells us that the used weighting scheme was not very important. However, we can see that the highest average accuracy was achieved by TF-IDF.

Table 3. The comparison of average accuracies achieved for each weighting scheme.

Weighting scheme	Average accuracy from all experiments
TF-IDF	0.598
TP	0.587
TF	0.586

Table 4 shows for each classifier the average accuracy from all experiments. We can see that the decision tree classifiers "ExtraTreesClassifier" and "RandomForestClassifier" performed the best with the accuracy around 61%.

Table 4. The comparison of average accuracies achieved by each classifier.

Classifier	Average accuracy
ExtraTreesClassifier	0.613
RandomForestClassifier	0.609
MultinomialNB	0.602
LogisticRegression	0.601
BernoulliNB	0.598
LinearSVC	0.597
MLPClassifier	0.596
DecisionTreeClassifier	0.564
NearestCentroid	0.533

4.2 Neighboring windows approach

Tables 5, 6, 7, and 8 show the results achieved for the 12 data sets consisting of two neighboring windows.

Table 5. Facebook posts – neighboring windows: data statistics.

Data set no.	Class 1 samples	Class 2 samples	Total samples	Data set balance ratio	Number of words
1	477	520	997	0.917	691
2	1,156	719	1,875	1.608	1,523
3	4,293	953	5,246	4.505	3,627
4	1,187	2,008	3,195	0.591	2,350
5	8,971	2,207	11,178	4.065	6,523
6	3,052	4,802	7,854	0.636	4,995
7	3,454	3,998	7,452	0.864	4,684
8	8,399	15,292	23,691	0.549	10,441
9	5,198	2,211	7,409	2.351	4,646
10	1,918	2,828	4,746	0.678	3,191
11	26,111	8,139	34,250	3.208	13,269
12	21,070	9,750	30,820	2.161	12,116

Table 5 shows the statistics about the data used for the classification. Because the length of the windows was variable, the numbers of documents greatly vary. It is also visible that most of the data sets are imbalanced. This should be taken into account when evaluating the results.

Table 6. Facebook posts – neighboring windows: classification results.

Data set no.	Accuracy	Precision	Recall	F1 score
1	0.584	0.602	0.584	0.593
2	0.615	0.598	0.615	0.606
3	0.808	0.653	0.808	0.722
4	0.639	0.733	0.639	0.683
5	0.803	0.807	0.803	0.805
6	0.646	0.653	0.646	0.650
7	0.554	0.553	0.554	0.553
8	0.674	0.669	0.674	0.672
9	0.721	0.727	0.721	0.724
10	0.618	0.614	0.618	0.616
11	0.800	0.782	0.800	0.791
12	0.698	0.702	0.698	0.700
Average	0.680	0.674	0.680	0.676

According to Table 6, the average accuracy (as well as the F1 score) was 68%. The best accuracy (as well as F1 score) was achieved for data sets 3 (72%), 5 (80%), and 11 (79%). The reason for this might be that they have a balance ratio around 4 (with more documents marked with index value going up).

Table 7. The comparison of average accuracies achieved with different weighting schemes applied to the neighboring windows of the Facebook posts.

Data set no.	TP	TF	TF-IDF
1	0.539	0.534	0.534
2	0.555	0.556	0.570
3	0.744	0.753	0.769
4	0.626	0.625	0.639
5	0.735	0.736	0.765
6	0.610	0.609	0.620
7	0.534	0.534	0.536
8	0.633	0.629	0.646
9	0.663	0.661	0.675
10	0.568	0.562	0.578
11	0.754	0.747	0.766
12	0.637	0.644	0.658
Average	0.633	0.633	0.646

From Table 7 can be seen that the highest average accuracy provided the TF-IDF weighting scheme (+1% in comparison to TP and TF).

Table 8. The comparison of average accuracies achieved by different classifiers applied to the neighboring windows of the Facebook posts.

Data set no.	Classifier	Avg. accuracy
1	NearestCentroid	0.587
2	LogisticRegressionCV	0.578
3	LogisticRegression	0.748
4	LogisticRegressionCV	0.633
5	LogisticRegression	0.784
6	MultinomialNB	0.630
7	ExtraTreesClassifier	0.553
8	SGDClassifier	0.656
9	MultinomialNB	0.701
10	LogisticRegressionCV	0.595
11	ExtraTreesClassifier	0.788
12	SGDClassifier	0.673

Table 8 shows the classifier that achieved the highest accuracy for each data set. We can see that most of the times the Logistic Regression (5 times) achieved the best result. Among the other classifiers, the Multinomial Naïve Bayes classifier, Extra Trees Classifier, and Stochastic Gradient Descent (SGD) Classifier were the most successful twice and the Nearest Centroid was the best only once.

5 CONCLUSION

The goal of the work was to examine whether the content of text documents published on the Internet (specifically Facebook posts) has any connection with stock price movements. We used the values of the S&P 500 Index and divided them into 24 time windows with either growing or decreasing index value trend. Subsequently, we examined (using the classification accuracy) the connection between the documents' content and the trend of the index value in the time window in which was the document published.

Two types of experiments were performed. In the first one, the documents from all 24 windows were put into one data set and we achieved an accuracy of 62%. The second experiment, in which we divided the documents into 12 data sets formed from two neighboring windows, provided better results – the average accuracy was 68%. Moreover,

for three data sets the accuracy was even higher – 72%, 79% and 80%. This means that classifying data from the neighboring windows brings on average better results than using only one data set. This might be related to the concept drift [12] phenomenon which requires a further investigation for this specific domain.

The achieved accuracy around 70% tells us that the posts which companies write on their Facebook pages are partially related to the performance of the whole stock index.

It must be noted that we did not optimize the parameters of used classification algorithms. By doing this, we might achieve a slightly higher accuracy.

This area could be further researched in various directions. Firstly, the analysis may be performed on more types of documents (e.g., newspaper articles). Secondly, the class assigning method may be enriched by using various thresholds of the index value changes (not only 5%). Thirdly, it might be interesting to examine not the whole stock index, but the stock prices of the individual companies instead.

ACKNOWLEDGEMENT

This research was supported by the Czech Science Foundation [grant No. 16-26353S ”Sentiment and its Impact on Stock Markets”] and Internal Grant Agency of Mendel University [No. PEF_DP_2017001 “Searching for semantic information and gaining knowledge from text data streams with new machine learning methods”] and Internal Grant Agency of Mendel University [No. PEF_DP_2017022 “Acquiring, filtering and analyzing of texts for stock markets”].

REFERENCES

[1] Aggarwal, C. C. Data Streams: Models and Algorithms. 2007. Springer

[2] Gama, J. Knowledge discovery from data streams. CRC Press, 2010.

[3] Petrovský, J., Netolický, P. and Dařena, F. Examining Stock Price Movements on Prague Stock Exchange Using Text Classification. International Journal of New Computer Architectures and their Applications (IJNCAA). Vol. 7 No. 1. (2017). pp. 8-13. ISSN 2412-3587.

[4] Sven S. Groth, Jan Muntermann. An intraday market risk management approach based on textual analysis. Decision Support Systems. Volume 50. Issue 4. March 2011. Pages 680-691

[5] Colm Kearney, Sha Liu. Textual sentiment in finance: A survey of methods and models. International Review of Financial Analysis. Volume 33. May 2014. Pages 171–185.

[6] Bollen, J., Mao, H. and Zeng, X. Twitter mood predicts the stock market. Journal of Computational Science. 2011. vol. 2. no. 1. p. 1–8.

[7] Kaplanski, G. and Levy, H. Sentiment and stock prices: The case of aviation disasters. Journal of Financial Economics. 2010. vol. 95. no. 2. p. 174–201.

[8] Lee, H., Surdeanu, M., MacCartney, B. and Jurafsky, D. On the Importance of Text Analysis for Stock Price Prediction. In: LREC. 2014. p. 1170-1175

[9] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011. vol. 1. p. 142–150.

[10] Go, A., Bhayani, R. and Huang, L. Twitter sentiment classification using distant supervision. CS224N Project Report. Stanford. 2009. vol. 1. p. 12.

[11] Weiss, S. M., Indurkhya, N. and Zhang, T. Fundamentals of Predictive Text Mining. London: Springer. 2010. ISBN 978-1-84996-225-4.

[12] Lindstrom, P, Delany, S. J., Mac Namee, B. (2010) Handling Concept Drift in Text Data Stream Constrained by High Labelling Cost. Florida Artificial Intelligence Research Society Conference (FLAIRS). Florida, 19-21, May.

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

The *International Journal of New Computer Architectures and Their Applications* aims to provide a forum for scientists, engineers, and practitioners to present their latest research results, ideas, developments and applications in the field of computer architectures, information technology, and mobile technologies. The IJNCAA is published four times a year and accepts three types of papers as follows:

1. **Research papers**: that are presenting and discussing the latest, and the most profound research results in the scope of IJNCAA. Papers should describe new contributions in the scope of IJNCAA and support claims of novelty with citations to the relevant literature.
2. **Technical papers**: that are establishing meaningful forum between practitioners and researchers with useful solutions in various fields of digital security and forensics. It includes all kinds of practical applications, which covers principles, projects, missions, techniques, tools, methods, processes etc.
3. **Review papers**: that are critically analyzing past and current research trends in the field.

Manuscripts submitted to IJNCAA **should not be previously published or be under review** by any other publication. Plagiarism is a serious academic offense and will not be tolerated in any sort! Any case of plagiarism would lead to life-time abundance of all authors for publishing in any of our journals or conferences.

Original unpublished manuscripts are solicited in the following areas including but not limited to:

- Computer Architectures
- Parallel and Distributed Systems
- Storage Management
- Microprocessors and Microsystems
- Communications Management
- Reliability
- VLSI