A Text Classifier Model for Categorizing Feed Contents Consumed by a Web Aggregator

H.O.D. Longe
Department of Computer Science,
University of Lagos

Fatai Salami
Department of Computer Science
Bells University of Technology, Ota, Nigeria

***Abstract*—This paper presents a method of using a Text Classifier to automatically categorize the content of web feeds consumed by a web aggregator. The pre-defined category of the feed to be consumed by the aggregator does not always match the content being consumed and categorizing the content using the pre-defined category of the feed curtails user experience as users would not see all the contents belonging to their category of interest. A web aggregator was developed and this was integrated with the SVM classifier to automatically categorize feed content being consumed. The experimental results showed that the text classifier performs well in categorizing the content of feed being consumed and it also affirmed the disparity in the pre-defined category of the source feed and appropriate category of the consumed content.**

Keywords—feed; aggregator; text classifier; svm

I. INTRODUCTION

Web feeds provide a way for websites especially those that are frequently updated to provide up to date information to their users. Feeds are provided in either RSS or Atom format.

Users who are interested in consuming the content of feeds use an aggregator software called feed reader. Aggregator software can either be a windows or a web application and it collects feed contents from various sources in one view. With a feed reader, a user can have the latest content of his/her favourite website in one place; thereby reducing time spent checking different websites. A spin-off of feed readers is web aggregation sites. A web aggregation site is a website that has content from various feeds in one place. This makes it easier for users to view contents from various websites at once. It also removes the overhead of having to build the content of a feed aggregator by the user. Popular aggregation websites include newsnow.com, kicknews.com.

When aggregators have to categorize the content consumed from feeds, they either use a predefined category that has been registered for the source of the feed or try to get the category from the meta-data supplied with the feed content. Using the predefined category of the source brings up scenario in which the category does not match the actual content being consumed. In some cases also, the category supplied in the meta-data would not match any of the categories set up in the aggregator.

The categorization of content from feeds can be achieved via the use of Text Classifiers. Text Classifiers are algorithms that are used to carry out Text Categorization (TC). In formal terms, taking a document di from a set of documents D and categories {c1, c2, c3}, text categorization is assigning a category ci to document di [11]. Example of text categorization algorithms include; K Nearest Neighbour (KNN), Naïve Bayes (NB), Support Vector Machines (SVM).

In TC, documents may be classified in such a way that it can only belong to one category (single-label categorization) or can belong to multiple categories (multi-label categorization) [15]. Multi-label categorization is better suited to aggregators because the content consumed from a feed can belong to multiple categories. Example, a story about a Nigerian footballer getting married to a Nollywood (Nigerian movie industry) actress can rightly belong to categories about sports, gossip and entertainment.

The paper is organized as follows. Section II contains a review of existing literatures in the field of Text Categorization. It is followed by system architecture and software design in Section III. The categorization process is discussed in Section IV and implementation and evaluation of the system is in Section V. Conclusion is made in Section VI.

II. RELATED WORK

There are two main approaches to building text classifiers – Knowledge Engineering (KE) approach and Machine Learning (ML). Knowledge Engineering (KE) used to be very popular. It involves manually defining a set of rules encoding knowledge from experts to place texts in specified categories. KE gradually lost its popularity in the 1990's to Machine Learning (ML) approach which involves building automatic text classifier by learning the characteristics of the categories of interest from a set of pre-classified texts [18].

In deciding whether to use Machine learning or Knowledge Engineering approach to text classification, sentences in Dutch Law were classified using both Machine Learning technique and Knowledge engineering approach [7]. SVM and pattern based KE were implemented and was found that SVM attained accuracy of up to 90%.

A Scientific News Aggregator that gathered news from both Atom and RSS feeds of about 1000 web journals was developed in [19]. NB classifier was used to classify the news coming from the different sources into stipulated categories of interest. Since a relatively large part of the RSS/Atom feed was already manually classified from the originating news source, the key idea implemented for classifying was to use the classifier in a mixed mode: as soon as already classified scientific news by a scientific news source was seen, the classifier switched to training mode; the remaining unclassified

scientific news was categorized with the classifier in categorizing mode.

Multi-label classification was implemented by [4]. A ranking function was used to compute the relevancy of all predefined categories to the news item. The contents of <title>, <description> and <link> elements were retrieved and used as features. Normalized term frequency method was used to determine the weight of individual feature in the vector space.

SVM was used by [12] to classify news articles into three categories; Sports, Business and Entertainment. The vector representation of features serves as entry point into the SVM classifier. The SVM classifier was implemented using LIBSVM - an integrated software for support vector classification, regression and distribution estimation [one-class SVM] with the support for multiclass classification.

Categorization of news text using SVM and ANN was carried out in [2]. In the overall comparison of SVM and ANN algorithms for the data set that was used, the results for both recall and precision over all conditions indicate significantly differences in the performance of the SVM algorithm over the ANN algorithm and since SVM is a less (computationally) complex algorithm than the ANN, they concluded that SVM is preferable at least for the type of data examined, i.e., many short text documents in a relatively few well populated categories.

A method of Text Categorization on web documents using text mining and information extraction based on the classical summarization techniques was proposed in [9]. First, web documents are pre-processed by removing the html tags, meta-data, comment information, images, bullets, buttons, graphics, links and all other hyper data in order to establish an organized data file, by recognizing feature terms like term frequency count and weight percentage of each term. Experimental results showed that this approach of Text Categorization is more suitable for Informal English language based web content where there is vast amount of data built in informal terms. The method significantly reduced the query response time, improved the accuracy and degrees of relevancy.

In [16], rough set theory was used to automatically classify text documents. After pre-processing text documents and stemming the features, they used specific thresholds of 10%, 8%, 6% and 4% to reduce the size of the feature space based on the frequency of each feature in that text document. Thereafter, their model used a pair of precise concepts from the rough set theory that are called the lower and upper approximations to classify any test text document into one or more of main categories and sub-categories of interest. The rough set theory produced accuracy of up to 96%.

SVM and NB classifiers were used to categorize Arabic texts in [1]. In the Arabic dataset that was used, each document was first processed to remove digits and punctuation marks and then some letters were normalized after which stop words were removed. They used three parameters for their evaluation – precision, recall and F1 and SVM outperformed NB with respect to all the evaluation parameters.

The combination of SVM and Elitist Genetic Algorithm (EGA) was applied to the classification of Chinese text by [10]. Genetic Algorithms (GA) are used to determine the values of parameters such as the regularization parameter (C) when used in combination with SVM. However, it is possible that some better solution found in previous steps may be lost because of the genetic operation in traditional GA. This led to introduction of memory to keep track of the better solutions that would have otherwise been discarded. Elite survival strategy is employed in combing algorithm, EGA-SVM. The results obtained in their evaluation showed that the EGA-SVM outperformed GA-SVM and ordinary SVM.

Ttext categorization was used to detect intrusion by [9]. KNN classifier was used for the classification. System processes were taken as documents to be classified and system calls were taken as distinct words. The tf-idf text categorization weighting technique was adopted to transform each process into a vector. Their preliminary result showed that the text categorization approach is effective in the detection of intrusive program behaviour.

SVM was used as the classification algorithm in this paper because it has high dimensional input space, understands that there are few irrelevant features and tries to use as many features as possible, the documents' vectors are sparse and most text categorization problems are linearly separable [6].

III. Proposed Web Aggregator System Architecture

The architecture as shown in Fig. 1 consists of a user that makes request to view information from the aggregator, an application server which serves the pages and connects the system to the internet, a feed database that contains the information about registered feeds, training data for the Categorization engine and retrieved contents by the Content Retrieval engine. It also includes a Content Retrieval Engine which retrieves new contents from the registered feeds and a Categorization Engine which carries out the categorization process.

A. The Feed Database

This consists of six entities. The Entity Relationship Diagram (ERD) presented in Fig. 2 shows all the entities in the Feed Database and the relationship between them. The entities in the Feed database are: Category – contains the categories used in the aggregator, Feed – registered feeds to retrieve contents from, Post – contents retrieved from registered feeds, PostCategory – categories assigned to the retrieved content by the Categorization engine, PostView – a count of the number of times a particular post has been viewed and TrainingPost – retrieved contents that would be used to train the categorization engine.

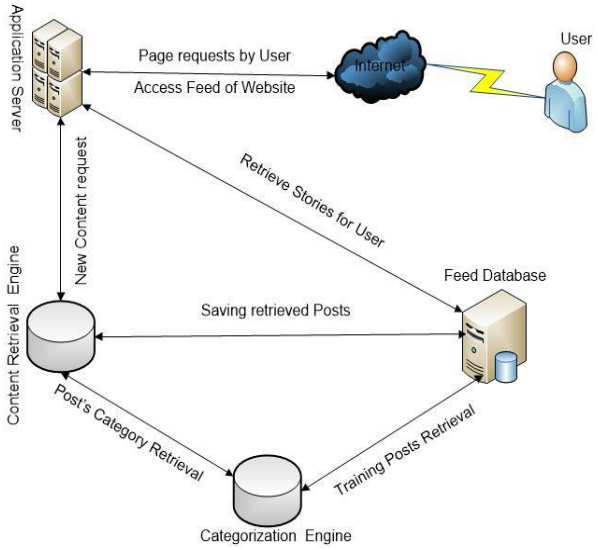

Fig. 1. Architecture for Web Aggregator

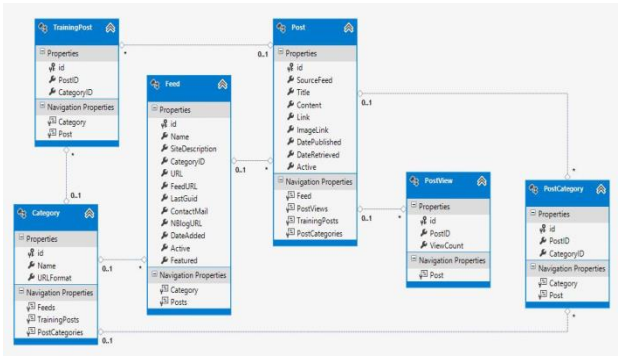

Fig. 2. Entity Relationship Diagram of Feed database

B. The Content Retrieval Engine

It retrieves most recent yet to be retrieved contents from the registered feeds.

The steps to retrieve new content are as follows:

1) Retrieve all Feeds to be polled for content from Feed database and store as ListFeeds.

2) Set ListPost as list of posts to be added to database, ListUpdate as list of Feeds to update their LastGuid and ListPostCategory as Categories determined for the Contents.

3) For each Feed in ListFeeds repeat steps 4 to 16.

4) Download XML of Feed.

5) Determine type of Feed and adjust tags to examine appropriately.

6) Set LatestGuid = Guid of the most recently published content in the feed, usually the first.

7) If LatestGuid = LastGuid of the Feed, Go to next Feed in ListFeeds else continue to 8.

8) Set count = 0, maxCount = maximum number of posts to retrieve and PostGuid = null.

9) If PostGuid = LastGuid of the Feed or count >= maxCount; Add LatestGuid and Feed to ListUpdate then Go to next Feed in ListFeeds ELSE select content as Post.

10) Process Post to remove all unnecessary HTML tags.

11) Add processed Post to ListPost.

12) Set ListCat as categories determined for the Post by the Categorization Engine.

13) Add ListCat to ListPostCategory.

14) Set count = count + 1.

15) Set PostGuid = Guid of Post.

16) Go to 9.

17) Save ListPost and ListPostCategory to Feed database and update Feed table using ListUpdate.

C. The Categorization Engine

This makes use of SVM classifier to classify contents. The literatures reviewed showed that the SVM is one of the best classifiers available hence its choice for this paper. The Categorization Engine builds SVM model which is required for categorization using the Posts saved to the TrainingPost table in Feed Database. The TrainingPost table had 1020 manually categorized posts which were retrieved from some Nigerian blogs and websites. The spread of the training posts among the various categories is presented in Table I.

The Categorization Engine also determines the categories that best fits a post retrieved by the Content Retrieval Engine. Since the project looks at the possibility of placing a retrieved content in more than one category, SVM multi-label classification class is employed. The result returns a list of possible categories for the retrieved content.

TABLE I. TRAINING POST SPREAD AMONGST CATEGORIES.

Category	Number of Training Data
Business	104
Current Affairs	130
Education	92
Entertainment	124
Gossip	134
Jobs	109
Personal	80
Politics	65
Science & Technology	80
Sports	102

IV. OVERVIEW OF THE CATEGORIZATION PROCESS

The text categorization process can be divided into seven sub processes – Read document, Tokenize text, Stemming, Stop words removal, Vector representation of text, Feature Selection and/or Feature Transformation (Dimensionaliity Reduction) and Learning Algorithm. The Feature Selection and/or Feature Transformation phase was not used in this paper because the contents of Feeds are usually a summary and often times already have few features. A diagrammatic representation of the categorization process is shown in Fig. 3.

The Read Document phase was achieved by supplying the categorization engine with string representation of content to be categorized. Tokenization of Text removed punctuation marks and separated the text into individual words.

Stop Words removal involved removing words with little semantic meaning from the tokens. The list of stopwords used in this paper was gotten from [17].

The stemming process involves getting the stem terms for words. This is done by removing the suffix from words. The Porter Stemmer is a conflation Stemmer developed by Martin Porter and it is based on the idea that the suffixes in the English language are majorly made up of a combination of smaller and simpler suffixes. The Porter Stemmer Algorithm is widely used and it is probably the stemmer most widely used in IR research [8].

The vector space representation involves converting the words in the text to be categorized into SVM matrix representation of words. The general format of the vector space representation for SVM is:

<label> <index>:<value> <index>:<value>

<label> is the number representation of the category of the text to be classified. A random category amongst the legal categories can be selected. The id value in the Category table of Feed Database is used to represent the categories. <index> is the number representing the stemmed word and <value> is the tf.idf value of the word. The <index> values are arranged in alphabetical order.

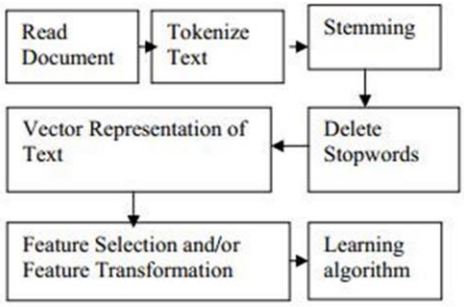

Fig. 3. Text Categorization Process (Source: [5]).

The learning algorithm that was used in this work is the SVM algorithm. There are several implementations of the SVM algorithm. LibSVM.Net which is the .Net implementation of LibSVM [3] was used in this project. Modification was made to LibSVM.Net to allow it accept string inputs instead of the default text document. LibSVM first builds a Model using the vector space representation of the training data along with a set of parameters.

A. Vector Space Representation Process

The algorithm used to carry out the vector space representation process is as follows:

1) Initialize BagOfWords = combination of all ListStemWords for all training data arranged alphabetically.

2) Initialize ListCategorizingWords = ListStemWords for the text that is to be categorized arranged alphabetically.

3) Initialize string VSR which would hold the vector space.

4) For each word W in ListCategorizingWords.

5) If W exists in BagOfWords go to 6 ELSE go to next W.

6) Set string S = W's index in BagOfWords + ":".

7) Calculate the tf.idf frequency of W as ti.

8) Set string S = S + ti + " ".

9) Set VSR = VSR + S.

10) Go to next W.

11) Return VSR.

V. Implementation and Evaluation

A. Web Aggregator User Interface

The web aggregator developed called "NBlogs" was based on the concept of responsive design. A responsive website is a website that automatically adjusts the screen size to fit the size of the screen from which it is being viewed from whether a desktop, a tablet pc or a smart phone. Twitter bootstrap package was used in the design to achieve responsiveness. Fig. 4 shows what the home page of NBlogs looks like in a desktop browser while Fig. 5 shows the same home page on a smaller screen. C# programming language was used in coding the business logic. NBlogs runs on .Net's MVC framework. MSSQL server was used to house the Feed Database.

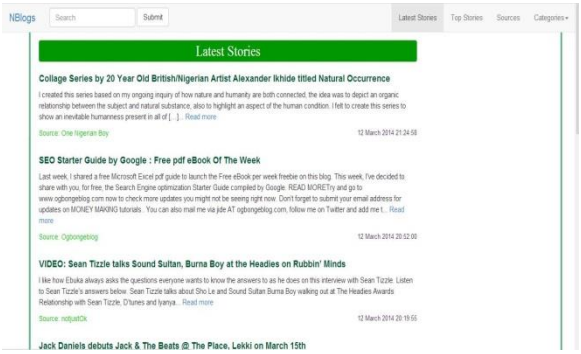

Fig. 4. Web Aggregator Home Page on Desktop browser.

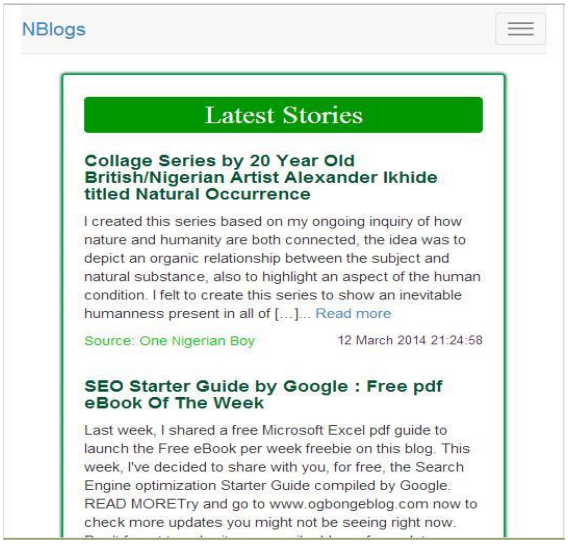

Fig. 5. Web Aggregator Home Page on smaller screen

B. Performance Evaluation of Categorization Algorithm

The evaluation of classifiers can be carried out using metrics such as precision, recall and F-Measure.

Recall is the proportion of real positive cases that are actually predicted as positive while Precision is the proportion of Predicted Positive cases that are correctly Real Positives (Powers, 2011).

$$\text{Recall} = r = \frac{TP}{TP + FP}$$

$$\text{Precision} = p = \frac{TP}{TP + FN}$$

Where:

TP = True Positive – predicted the right category for a story.

FP = False Positive – predicted category is wrong category for a story.

FN = False Negative – category was not rightly predicted for a story.

A total of one hundred and fifty (150) stories were retrieved from feeds to test the Categorization Engine. The stories were categorized into one hundred and ninety six (196) categories. The result of categorization including the TP, FN, FP, p and r is presented in Table 2. The bar graph of p and r is presented in Fig. 6.

F-Measure is the harmonic mean of the recall and precision with interval between 0 and 1 with a high F-Measure indicating a high quality classifier. The micro-averaged F-Measure is computed over all categories and it is achieved by summing the individual precision and recall scores for the categories. The macro F-Measure score is first computed over the individual categories before an average is taken (Ozgur, Ozgur, and Gungor, 2005).

Micro-Averaged F-Measure can be calculated as:

$$\frac{2(sr * sp)}{sr + sp}$$

Where:

sr =

$$\frac{\sum_{i=1}^{N} TPi}{\sum_{i=1}^{N} (TPi + FPi)}$$

sp =

$$\frac{\sum_{i=1}^{N} TPi}{\sum_{i=1}^{N} (TPi + FNi)}$$

N = number of categories.

Macro-Averaged F-Measure can be calculated as:

$$\frac{\sum_{i=1}^{N} FMi}{N}$$

Where

N = number of categories

FM_i =

$$\frac{2(ri * pi)}{ri + pi}$$

ri = recall of category i.

pi = precision of category i.

The Micro-Average F-Measure computed from Table 2 above is *0.731457801* while the Macro-Average F-Measure computed from the same table is *0.721934751*. The F-Measure values indicate a high quality classifier.

C. Effect of Text Classifier on Post Categories

Table 3 presents the distribution of posts after categorization has been carried out. PC is the number of posts that were categorized to be in the same category as the category registered for the feed while PD is the number of posts that were categorized in a different category to the category of the feed. %PD is the percentage of posts for that category that were placed in a different category. Overall, 68% of retrieved feed content were placed in a different category compared to the category of the source feed.

TABLE II. CATEGORIZATION RESULT

Category	True Positive (TP)	False Positive (FP)	False Negative (FN)	Precision (p)	Recall (r)
Business	8	9	3	0.73	0.47
Current Affairs	6	6	1	0.86	0.50
Education	5	0	2	0.71	1.00
Entertainment	25	8	6	0.81	0.76
Gossip	6	7	7	0.46	0.46
Jobs	13	0	4	0.76	1.00
Personal	27	11	11	0.70	0.71
Politics	3	1	0	1.00	0.75
Science & Technology	38	5	14	0.73	0.88
Sports	12	6	4	0.75	0.67

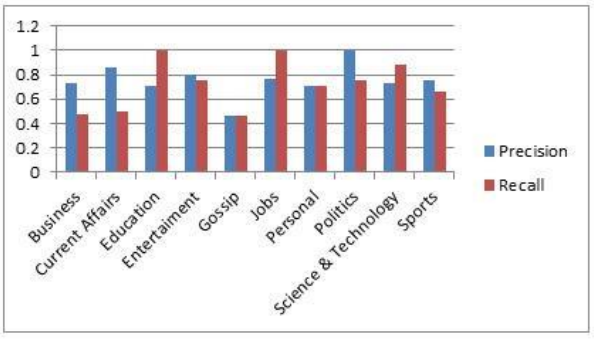

Fig. 6. Precision and Recall Grapgh

TABLE III. CATEGORIZING USING FEED CATEGORY AGAINST CATEGORIZATION ALGORITHM

Feed Category	PC	PD	% PD
Business	3	18	86
Current Affairs	1	22	96
Education	5	1	16
Entertainment	21	38	64
Gossip	2	13	87
Jobs	8	12	60
Personal	4	44	91
Politics	0	3	100
Science & Technology	34	24	41
Sports	12	22	64

VI. CONCLUSION

In this paper, text categorization algorithm was used to categorize the contents of feed consumed by a web aggregator. With training data obtained from the feeds of Nigerian websites, a SVM model was constructed to carry out the categorization.

The result obtained showed that the categorizer is of a high quality with a Micro-Average F1 measure of 0.731457801 and Macro-Average F1 measure of 0.721934751 and it further showed that it is not reliable to categorize contents consumed from a feed using the pre-defined category of the Feed as 68% of the feed content retrieved was placed in a different category by the SVM classifier.

The use of text categorization algorithm in web aggregators would improve user experience as they would be able to more easily access stories of interest to them.

REFERENCES

[1] Alsaleem, S., 2011. Automated Arabic Text Categorization Using SVM and NB. International Arab Journal of e-Technology, Volume 2, No. 2, pp. 124-128.

[2] Basu, A., Watters, C. and Shepherd, M., 2003. Support Vector Machines for Text Categorization.

[3] Chih-Chung Chang and Chih-Jen Lin, 2011. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[4] Darabi, M., Adeli, H. and Tabrizi, N., 2012. Automatic Multi-Label Categorization of News Feeds.

[5] Ikonomakis, M., Kotsiantis, S. and Tampakas, V., 2005. Text Classification Using Machine Learning Techniques. Wseas Transactions on Computers, Volume 4, Issue 8, pp. 966-974.

[6] Joachims, T., 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. University of Dortmund Computer Science Department.

[7] Krabben, K., 2010. Machine Learning vs. Knowlegde Engineering in Classification of Sentences in Dutch Law. BSc. Universiteit Van Amsterdam

[8] Lancaster University, 2014. What is Porter Stemming?.

[9] Liao, Y. and Vemuri, R.V., 2002. Using Text Categorization Techniques for Intrusion Detection. In: USENIX Association, 11th USENIX Security Symposium. San Francisco, California, USA August 5-9, 2002.

[10] Liu, X. and Fu, H., 2012. A Hybrid Algorithm for Text Classification Problem.

[11] Manne, S. and Sameen, F.S., 2011. A Novel Approach for Text Categorization of Unorganized data based with Information Extraction. International Journal on Computer Science and Engineering (IJCSE), Volume 3, Issue 7, pp. 2846-2854.

[12] Mayor, S. and Pant, B., 2012. Document Classification Using Support Vector Machine. International Journal of Engineering Science and Technology (IJEST), Volume 4, No.04, pp. 1741-1745.

[13] Ozgur, A., Ozgur, L. and Gungor, T., 2005. Text Categorization with Class-Based and Corpus-Based Keyword Selection. P. Yolum et al.(Eds.): ISCIS 2005, LNCS 3733, pp. 606–615, 2005.

[14] Powers, D.M.W., 2011. Evaluation: From Precision, Recall and F-Measure to ROC,Informedness, Markedness & Correlation. Journal of Machine Learning Technologies ISSN: 2229-3981 & ISSN: 2229-399X, Volume 2, Issue 1, 2011, pp-37-63.

[15] Rafi, M., Hassan, S. and Shaikh, M.S., 2011. Content-based Text Categorization using Wikitology. National University of Computer and Emerging Sciences (NU-FAST) Karachi, Sindh, Pakistan.

[16] Sadiq, A.T. and Abdullah, S.M., 2013. Hybrid Intelligent Techniques for Text Categorization. International Journal of Advanced Computer Science and Information Technology (IJACSIT), Volume 2, No. 2, pp. 23-40.

[17] Savoy, J., 2005. IR Multilingual Resources at UniNE. Universite de Neuchatel.

[18] Sebastiani, F., 2001. Machine Learning in Automated Text Categorization.

[19] Shaikh, F., and Rajawat, A., 2012. Approach for Developing Scientific News Aggregators Using ATOM Feeds. International Journal of Electronics and Computer Science Engineering (IJECSE), Volume1, Number 4, pp. 2279-2284.