A Comparative Analysis of Various Classifications in Vector Space Model with Absolute Pruning

Nandni Patel
MTech
GGITS, Jabalpur

Santosh Vishwakarma
GGITS
Jabalpur

ABSTRACT

Text Classification is an important problem in text mining used to categorize an undefined label. In this work, various classification models have been evaluated after pre-processing of the text dataset. The pre-processing steps include tokenization, stop word removal and stemming, after which different term weight scheme have also been implemented. Various pruning techniques have also been implemented to get the maximum count of the terms. Based on this analysis, we summarized that Naïve Bayes method gives the highest accuracy while comparing with other state of the art text classifiers.

Keywords

Text Classification Models, Pruning Methods, Vector Space Model, Absolute Pruning

1. INTRODUCTION

Text mining is defined as the process of examining large collection of unstructured text documents to get the desired information. It is a part of Data mining. Text mining consists of two phases one is Text refining that transform unstructured text into intermediate form (document based IF) and other is knowledge refining (Knowledge based IF) that gathers information from intermediate form (IF).The diagram below shows the Text retrieval framework:

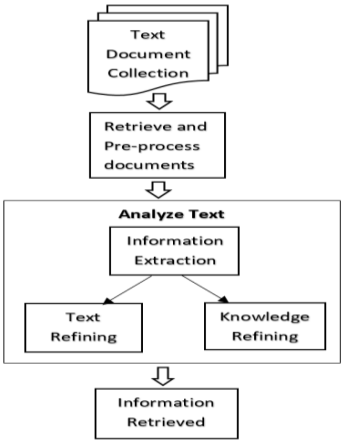


Fig-1: Framework for Text mining

We are using Vector Space Model which is an algebraic model to represent text as vector with definite weighted frequency. It is used in information filtering, information retrieval, indexing and relevancy rankings. In this model both document and query represented as vectors. The vector space model process can be divided into three stages: first one is Indexing in which words are extracted from text, second is weighting in which weighting of indexed terms is performed to increase the chances of relevant retrieved document, and last one is ranking which ranks the document according to degree of similarity with query in decreasing order.

To assign a weight for each vector in second stage of vector space model various weighting schemes are available as TF-IDF, Term -frequency, Term-occurrence, and Binary term-occurrence. In our analysis TF-IDF found better among them. TF-IDF is a numerical measure the importance of a vector or word in a document of a corpus. The value of TF-IDF increases in a proportion of occurrence of a word in a document. It is the most popular term weighting scheme. The formula for assigning weight for a term iin document j in TF-IDF is:

$$\mathrm{W}i,j \;=\; \mathrm{TF}i,j \times \log\frac{N}{df_{\mathrm{i}}}$$

Where TF$i,j$ is number of occurrence of *i* in *j*, df_i is number of documents containing *i* and N is total number of documents.

Stemming is the process of replacing all the variants of a word into its root word. It is the document pre-processing step. We applied various stemming algorithm and analysed that Porter algorithm is the better one. Porter stemming removes the inflexions and other morphology from the end of the English word. Porter stemming algorithm is strictly defined and not amendable. To overcome this new algorithm is developed known as snowball. Porter algorithm is appropriate for the work where an experiment needs to be exactly repeatable.

We also used various pruning methods in our dataset. Pruning is the process of eliminating those leaf nodes of a decision tree who do not add to the discriminative power. Absolute pruning found better than other pruning methods.

This paper consists following sections: In section-I we discussed about the introduction, section-II will discuss about the literature review, section-III will discuss about the methodology, section-IV reports the Result and Analysis and in section-V the paper is concluded by explaining the best classification model based on our analysis and the future scope.

2. LITERRATURE REVIEW

Zhai et al. [1] studied the problem of language model smoothing in the context of information retrieval. They explore various text weighting scheme along with text normalization. They also examined three interpolation-based smoothing methods.

Recently Reel et al. [2] introduce a new term weighting scheme that does not require access to the general document corpus and that considers information from the users' personal document collections.

Hans-Peter Frei [4] stated that information retrieval (IR) as a discipline is not a new term. It was also used in past but recently IR researchers have started to study real-life problems and have realized that encountered data is highly unstructured, heterogeneous, hypermedia and of varying quality.

Ronan Cummins and Colm O'Riordan [5] stated an evolution of evolved term-weighing schemes on short, medium and long TREC queries. The most effective scheme identified is Okapi-tf and the evolved global scheme.

Ronan Cummins and Colm O'Riordan [6] stated Term weighting scheme play a vital role in the performance of many Information Retrieval models. The vector space model is one such model in which the weights applied to the document terms are of crucial importance to the accuracy of the retrieval system.

Rong et al. [7] proposed algorithms for automatic term weighting that exploit document category information. Vishwakarma et al. [15] performs various experiments based on n-grams on the FIRE Corpus under different weighting schemes.

3. METHODOLOGY

We performed our experiments in RapidMinerStudioversion7.5. This tool is an open source tool, written in java and is available for all major operating systems. Also, it does not require any programming skills for its use. Its main advantage is that it works with both structured and unstructured documents. Several plug-ins are already available but if we want more it is easily extensible. It is applicable in multimedia mining, distributed data mining, machine learning and predictive analysis etc. As explained this tool includes all the features required for the analysis of FIRE dataset.

FIRE stand for Forum for Information Retrieval and Evaluation. It is an Indian organization for research in information retrieval. It works on the languages of South Asian countries. We performed our experiments on collecting 100 documents from FIRE-2011 dataset. FIRE dataset adapts TREC document style format. We have separate file for each document which contains three fields named -DOC, DOCNO, and TEXT.

As shown in Fig 2, step-1 of designing process we used four operators in Min process window, they are Process Document from File that generates word vectors from a text collection stored in multiple files. In this process, we are using two Process-Documents from File operators. First operator taking Training corpus and second is using Test corpus. Second is Validation operator which performs a simple validation i.e. randomly splits up the Example Set into a training set and test set and evaluates the model. Validation operator performs a simple validation i.e. randomly splits up the Example Set into a training set and test set and evaluates the model. Performance operator is used for performance evaluation.

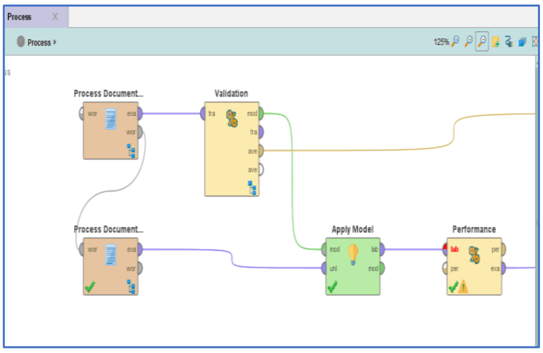


Fig 2. Step-1 shows the main process wizard [1]. In this wizard, we have used four operators.

As in Fig 3, Step-2 is Pre-processing or Process Document from File wizard. It is performed as:

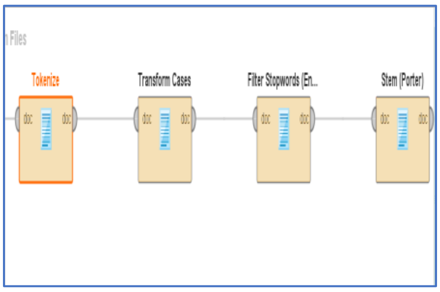


Fig 3. Step-2.Process Document from File

There are some Pre-processing steps which are used by information retrieval system before assigning ranks to the documents. Pre-processing is the process of incorporating a new document into an information retrieval system. It has various stages like tokenization that breaks the documents into tokens, transform case converts the token into either upper case or lower case, Stop words eliminationfilter out the words that have no values for retrieval purpose, Stemming perform replacement of all the variations of the words with its root word are applied on each document to determine how well documents satisfy information needs.

Fig 4 shows the validation step which includes training and testing as:

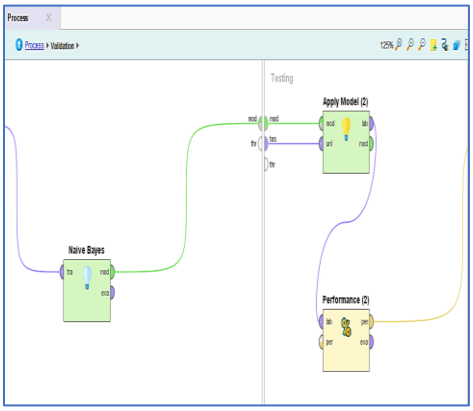


Fig 4. Step-3.Validation

The classification models used in Training wizard are: Random forest[1], this operator generates a set of a specified number of random trees , k-NN[2] operator generates a k-Nearest Neighbour model from the input Example Set, Generalized linear model[3]executes GLM algorithm using H2O 3.8.2.6, classification by regression model[4] builds a polynomial classification model through the given regression learner and Naïve Bayes[5] model, A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. The output port delivers the Naive Bayes classification model. This classification model can now be applied on unseen data sets for prediction of the label attribute. the goal of learning P(X│Y)where X = [X1… ;Xn], the Naive Bayes algorithm makes the assumption that each Xi is conditionally independent of each of the other Xĸ'S given Y, and also independent of each subset of the other Xĸ'S given Y.

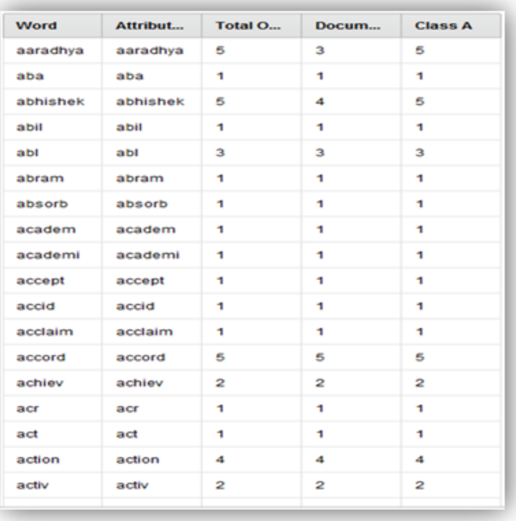
$$P(X1,\dots\dots Xn)\,\big|\,Y = \prod_{i=1}^{n} P(Xi\,\big|\,Y)$$

In Testing part of Validation wizard we used Apply model and Performance operators which are explained Step-2.

Different pruning methods are also applied in our dataset. Absolute pruning was found better among them.

4. ANALYSIS AND RESULT

As mentioned in section III, the dataset has been pre-process with several weighting schemes in the initial phase. A snapshot of the vector creation is shown as following:



Word	Attribut...	Total O...	Docum...	Class A
aaradhya	aaradhya	5	3	5
aba	aba	1	1	1
abhishek	abhishek	5	4	5
abil	abil	1	1	1
abl	abl	3	3	3
abram	abram	1	1	1
absorb	absorb	1	1	1
academ	academ	1	1	1
academi	academi	1	1	1
accept	accept	1	1	1
accid	accid	1	1	1
acclaim	acclaim	1	1	1
accord	accord	5	5	5
achiev	achiev	2	2	2
acr	acr	1	1	1
act	act	1	1	1
action	action	4	4	4
activ	activ	2	2	2

Fig 5. Word list of processed documents using TF-IDF weighing scheme.

The word count for various weight schemes is shown in Table 1, and it shows that TF-IDF gives the highest number of terms.

Vector Creation	Word Count
Binary Term-occurrence	5556
Term- Frequency	5557
Term-occurrence	5558
TF-IDF	5558

Table-1

The graph for the same have been plotted below in Fig.6

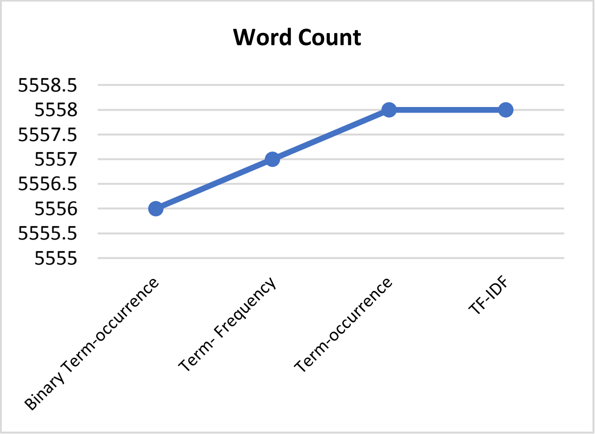


Fig-6

In the next step, several pruning methods have been applied and word count has been recorded as following in table 2:

Prune Method	Word Count
Percentual	358
By Ranking	5344
Absolute	**5555**
None	5558

Table-2

A graph is made based on above table's data between prune methods and word count. The graph shown below-

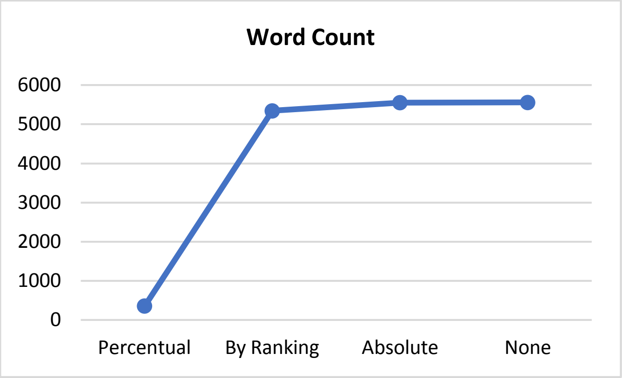


Fig-7

Based on the results of Table 2, it has been found that absolute pruning method has the highest number of terms after pre-processing. After this step, we applied various state of the art classifiers and compared their efficiency. The accuracy of the various classifiers are shown as following,

Models	Accuracy
Random Forest	43.33%
k-NN	73.33%
Generalized Linear Model	83.33%
Classification by Regression	86.67%
Naïve Bayes	**86.67%**

Table-3

A graph is plotted between Different classification models and Accuracy we got in each model respectively. The graph is as follows:

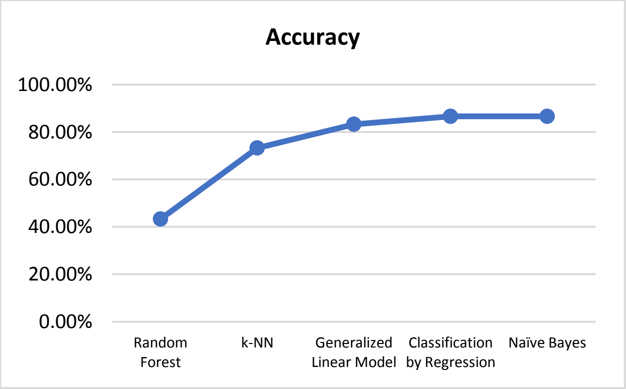
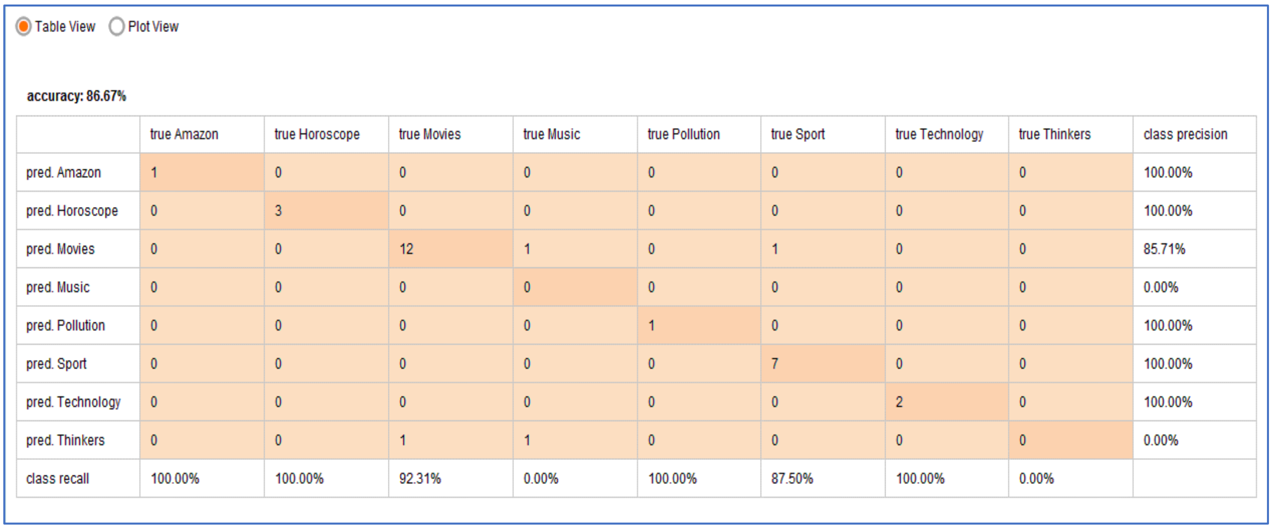


Fig-8: Accuracy of various classifiers

As we can see in the graph that Naïve Bayes and classification by Regression we got the maximum accuracy hence they are the best model of classification for our dataset.

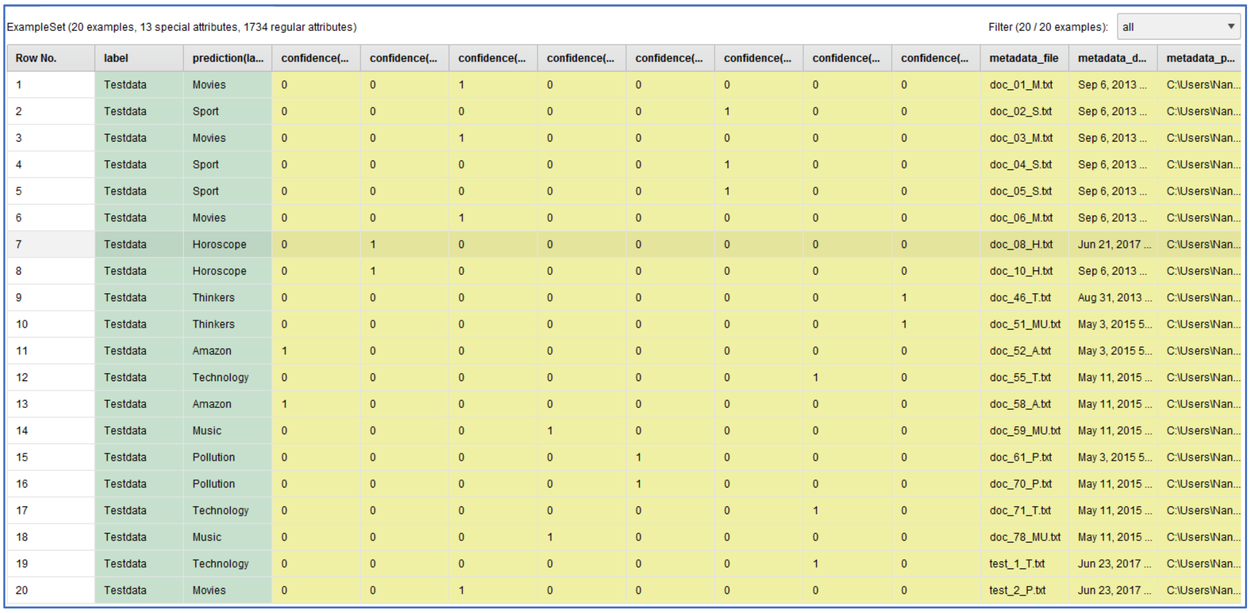
Also the Performance Vector and result of unlabele data have been shown in Fig 9 and Fig 10 respectively.



accuracy: 86.67%

	true Amazon	true Horoscope	true Movies	true Music	true Pollution	true Sport	true Technology	true Thinkers	class precision
pred. Amazon	1	0	0	0	0	0	0	0	100.00%
pred. Horoscope	0	3	0	0	0	0	0	0	100.00%
pred. Movies	0	0	12	1	0	1	0	0	85.71%
pred. Music	0	0	0	0	0	0	0	0	0.00%
pred. Pollution	0	0	0	0	1	0	0	0	100.00%
pred. Sport	0	0	0	0	0	7	0	0	100.00%
pred. Technology	0	0	0	0	0	0	2	0	100.00%
pred. Thinkers	0	0	1	1	0	0	0	0	0.00%
class recall	100.00%	100.00%	92.31%	0.00%	100.00%	87.50%	100.00%	0.00%	

Fig-9 Performance Vector Using Naïve Bayes classification model, showing the accuracy for each class of testing dataset in terms of Precision and Recall.



ExampleSet (20 examples, 13 special attributes, 1734 regular attributes)

Row No.	label	prediction(la...	confidence(...	confidence(...	confidence(...	confidence(...	confidence(...	confidence(...	confidence(...	confidence(...	metadata_file	metadata_d...	metadata_p...
1	Testdata	Movies	0	0	1	0	0	0	0	0	doc_01_M.txt	Sep 6, 2013 ...	C:\Users\Nan...
2	Testdata	Sport	0	0	0	0	0	1	0	0	doc_02_S.txt	Sep 6, 2013 ...	C:\Users\Nan...
3	Testdata	Movies	0	0	1	0	0	0	0	0	doc_03_M.txt	Sep 6, 2013 ...	C:\Users\Nan...
4	Testdata	Sport	0	0	0	0	0	1	0	0	doc_04_S.txt	Sep 6, 2013 ...	C:\Users\Nan...
5	Testdata	Sport	0	0	0	0	0	1	0	0	doc_05_S.txt	Sep 6, 2013 ...	C:\Users\Nan...
6	Testdata	Movies	0	0	1	0	0	0	0	0	doc_06_M.txt	Sep 6, 2013 ...	C:\Users\Nan...
7	Testdata	Horoscope	0	1	0	0	0	0	0	0	doc_08_H.txt	Jun 21, 2017 ...	C:\Users\Nan...
8	Testdata	Horoscope	0	1	0	0	0	0	0	0	doc_10_H.txt	Sep 6, 2013 ...	C:\Users\Nan...
9	Testdata	Thinkers	0	0	0	0	0	0	0	1	doc_46_T.txt	Aug 31, 2013 ...	C:\Users\Nan...
10	Testdata	Thinkers	0	0	0	0	0	0	0	1	doc_51_MU.txt	May 3, 2015 5...	C:\Users\Nan...
11	Testdata	Amazon	1	0	0	0	0	0	0	0	doc_52_A.txt	May 3, 2015 5...	C:\Users\Nan...
12	Testdata	Technology	0	0	0	0	0	0	1	0	doc_55_T.txt	May 11, 2015 ...	C:\Users\Nan...
13	Testdata	Amazon	1	0	0	0	0	0	0	0	doc_58_A.txt	May 11, 2015 ...	C:\Users\Nan...
14	Testdata	Music	0	0	0	1	0	0	0	0	doc_59_MU.txt	May 11, 2015 ...	C:\Users\Nan...
15	Testdata	Pollution	0	0	0	0	1	0	0	0	doc_61_P.txt	May 3, 2015 5...	C:\Users\Nan...
16	Testdata	Pollution	0	0	0	0	1	0	0	0	doc_70_P.txt	May 11, 2015 ...	C:\Users\Nan...
17	Testdata	Technology	0	0	0	0	0	0	1	0	doc_71_T.txt	May 11, 2015 ...	C:\Users\Nan...
18	Testdata	Music	0	0	0	1	0	0	0	0	doc_78_MU.txt	May 11, 2015 ...	C:\Users\Nan...
19	Testdata	Technology	0	0	0	0	0	0	1	0	test_1_T.txt	Jun 23, 2017 ...	C:\Users\Nan...
20	Testdata	Movies	0	0	1	0	0	0	0	0	test_2_P.txt	Jun 23, 2017 ...	C:\Users\Nan...

Fig-10 ExampleSet Using Naïve Bayes classification model showing Prediction for each document and its relative Class.

5. CONCLUSION

This work has been carried out to analyse the best vector creation, stemming algorithm, pruning method and classification model with the FIRE dataset which contains corpus of various resources. We found that TF-IDF weighing scheme with absolute pruning method gives best performance under the state of the art Naïve Bayes classification model. In future we aim to perform our experiment for large dataset of FIRE as TREC data set. This combination may also be applied as opinion mining and various sentiment analysis experiments.

6. REFERENCES

[1] Zhai, Chengxiang, and John Lafferty. "A study of smoothing methods for language models applied to ad hoc information retrieval." ACM SIGIR Forum. Vol. 51. No. 2. ACM, 2017.

[2] Beel, Joeran, Stefan Langer, and Bela Gipp. "TF-IDuF: A Novel Term-Weighting Scheme for User Modeling based on Users' Personal Document Collections." Proceedings of the 12th Conference. 2017.

[3] Deng, Zhi-Hong, Kun-Hu Luo, and Hong-Liang Yu. "A study of supervised term weighting scheme for sentiment analysis." Expert Systems with Applications 41.7 (2014): 3506-3513.

[4] Frei, Hans-Peter. "Information retrieval-from academic research to practical applications." In: Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas. 1996.

[5] Cummins, Ronan, and Colm O'Riordan. "An evaluation of evolved term-weighting schemes in information retrieval." Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005

[6] Cummins, Ronan, and Colm O'Riordan. "Determining general term weighting schemes for the vector space model of information retrieval using genetic programming." 15th Artificial Intelligence and Cognitive Science Conference (AICS 2004). 2004.

[7] Jin, Rong, Joyce Y. Chai, and Luo Si. "Learn to weight terms in information retrieval using category information." Proceedings of the 22nd international conference on Machine learning. ACM, 2005.

[8] Reed, Joel W., et al. "TF-ICF: A new term weighting scheme for clustering dynamic data streams." Machine Learning and Applications, 2006.

[9] Ljiljana Dolamic & Jacques Savoy UniNE at FIRE 2010: Hindi, Bengali, and Marathi IR

[10] Paul McNamee and James Mayfield, Character N-gram Tokenization for European Language Text Retrieval. Information Retrieval, 7:73-97, 2004.

[11] Mierswart al, "Rapid prototyping for complex data mining tasks", In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 935–940. ACM, 2006.

[12] Land Sebastian and Fisher Simon,"RapidMiner in academic use", 2012 www.rapid-i.com.

[13] Mierswa, I. et al "YALE: Rapid Prototyping for Complex Data Mining tasks", in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-06), pp. 935-940, 2006.

[14] Paolo Palmerini, "On performance of data mining: from algorithms to management systems for data exploration", Technical Report, Universit`a Ca' Foscari di Venezia, 2004.

[15] Monolingual Information Retrieval using Terrier: FIRE 2010 Experiments based on n-gram indexing Vishwakarma Santosh K., et al. "Monolingual Information Retrieval using Terrier: FIRE 2010 Experiments based on n-gram indexing." Procedia Computer Science 57 (2015): 815-820.

[16] "Text mining: The state of the art and the challenges." Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases. Vol. 8, 1999.