UACEE International Journal of Advances in Computer Science and its Applications – IJCSIA
Volume 3 : Issue 3 [ISSN 2250 – 3765]
Publication Date : 09 September 2013

A Text Mining Approach for Automatic Classification Of Web Pages

Surabhi Lingwal, Bhumika Gupta

Abstract—**Today the web contains a huge amount of information provided as html and xml pages and their number is growing rapidly with expansion of the web. In Web text mining, the text extraction and filtering of extracted content is the foundation of text mining. Automatic Classification of text is a semi-supervised machine learning task that automatically classify a given document to a set of pre-defined categories based on its features and text content. This paper explains a generic strategy for automatic classification of web pages that deals with unstructured and semi-structured text. This work classified the datasets into different labeled classes using kNN and Naïve Bayesian classification techniques. The experimental evaluation concluded that kNN has better accuracy, precision and recall value as compared to Naïve Bayesian classification. This paper presents a unified approach that is able to provide robust classification and validation of web pages to different categories.**

Keywords—**accuracy, automatic classification, cosine similarity, kNN, Naïve Bayes, precision, recall, tf-idf**

I. Introduction

Much of the Web content data is unstructured text data or free text such as news stories. Text mining can be considered as an instance of Web content mining. Text mining deals with the preprocessing of documents it includes content extraction, stop word removal, stemming which reduces words to their morphological roots. Automatic Text Classification involves assigning a text document to a set of pre-defined classes automatically, using a machine learning technique [9]. The classification is usually done on the basis of significant words or features extracted from the text document. Since the classes are pre-defined it is a supervised machine learning task. Due to the presence of large amount of information available on the web, efficient classification and retrieval of relevant content has gained significant importance. This paper explains the generic strategy for automatic text classification which includes steps such as pre-processing (eliminating stop-words [6][8], stemming [11] etc.),content extraction, and modeling using appropriate machine learning techniques (Naïve Bayes, Decision Tree, Neural Network, Support Vector Machines, k-Nearest Neighbor, etc). Automatic text classification has several useful applications such as classifying text documents in electronic format; spam filtering; improving search results of search engines; opinion detection [10] and opinion mining from online reviews of products, movies or political situations; and text sentiment mining [3].

II. Related Work

The authors of [4] proposed a data treatment strategy to generate new discriminative features, called compound-features, for the sake of text classification. These c-features are composed by terms that co-occur in documents without any restrictions on order or distance between terms within a document. The idea was that, when c features are used in conjunction with single- features, the ambiguity and noise inherent to their bag-of-words representation are reduced. In this paper [1] the authors proposed a news web page classification method (WPCM). Each news web page is represented by the term-weighting scheme. As the number of unique words in the collection set is big, the principal component analysis (PCA) has been used to select the most relevant features for the classification. Then the final output of the PCA is combined with the feature vectors from the class-profile which contains the most regular words in each class. These words are weighted then, using an entropy weighting scheme. The fixed number of regular words from each class will be used as a feature vectors together with the reduced principal components from the PCA. These feature vectors are then used as the input to the neural networks for classification. The authors of [7] describe text mining technique for automatically extracting association rules from collections of textual documents. The technique called, Extracting Association Rules from Text (EART). It depends on keyword features for discover association rules amongst keywords labeling the documents. In this work, the EART system ignores the order in which the words occur, but instead focusing on the words and their statistical distributions in documents. The main contributions of the technique are that it integrates XML technology with Information Retrieval scheme (TFIDF*)* and use Data Mining technique for association rules discovery.

Surabhi Lingwal
G.B.P.E.C Pauri Garhwal
India
surabhi.lingwal@gmail.com

Bhumika Gupta
Dept. of Computer Science & Engg, G.B.P.E.C Pauri Garhwal
India
Bhumikamit6@gmail.com

III. Text Mining

Text mining is the discovery of previously unknown information or concepts from text files by automatically extracting information from several written sources using computer software. Text mining on Web adoptive technique include classification, clustering, association rule and sequence analysis etc. Among them, classification is a kind of data analysis form, which can be used to gather and describe important data set. In Web text mining, the text extraction and the characteristic express of its extraction contents are the foundation of mining work, the text classification is the most important and basic mining method.

A. *Text Preprocessing*

The goal of text preprocessing phase is to optimize the performance of the next phase. This phase begins with the transformation process of the original unstructured documents [7]. This transformation aims to obtain the desired representation of documents in XML format. After that, the documents are filtered to eliminate the unimportant words by using a list of stop words and after word stemming. The resulting documents are processed to provide basic information about the content of each document.

1) **Extraction:** In extraction process, required information is extracted by checking maximum text density from the text contents from a web page. By this process, noises from the web page is removed.

2) **Tokenization:** Extraction is followed by tokenization of text content into tokens of words. Sometimes filtering of tokens is also required for some words.

3) **Stopword removal:** Stop word are the words that are evenly distributed in documents corpus are among the most frequent words in a language. Removal of such words from the index saves space and does not damage retrieval effectiveness. Stop words belongs to several word groups such as conjunctions, prepositions, adverbs etc [13].

4) **Stemming :** Stemming is a fundamental step in processing textual data preceding the tasks of information retrieval, text mining, and natural language processing. The common goal of stemming is to standardize words by reducing a word to its base [12].

5) **Select attributes and set role :** It is a preprocessing step that provide filtering of attributes like missing values and role of learning is set, based on labels of web pages.

6) **Tf–idf :** Term frequency-inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. A document in the vector space model is represented as a weight vector, in which each component weight is computed based on some variation of TF or TF-IDF scheme. In this method, the weight of a term t_i in document d_j is the number of times that t_i appears in document d_j, denoted by f_{ij}. Let N be the total number of documents in the system or the collection and df_i be the number of documents in which term t_i appears at least once[2]. Let f_{ij} be the raw frequency count of term t_i in document d_j. Then, the normalized term frequency (denoted by tf_{ij}) of t_i in d_j is given by:

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots\ldots, f_{|V|j}\}} \qquad (1)$$

where the maximum is computed over all terms that appear in document d_j. If term t_i does not appear in d_j then $tf_{ij} = 0$. Recall that |V| is the vocabulary size of the collection.

The inverse document frequency (denoted by idf_i) of term t_i is given by:

$$idf_i = \log\left(\frac{N}{df_i}\right) \qquad (2)$$

The intuition here is that if a term appears in a large number of documents in the collection, it is probably not important or not discriminative. The final TF-IDF term weight is given by:

$$w_{ij} = tf_{ij} \times idf_i \qquad (3)$$

B. *Classification*

Classification of web pages is performed using two different techniques kNN and Naïve Bayes. After classification, performance of these techniques is determined by applying learning models and accuracy of the two techniques is calculated and compared.

1) K-Nearest Neighbor

kNN is a lazy learning method in the sense that no model is learned from the training data. Learning only occurs when a test example needs to be classified [2]. When a test instance d is presented, the algorithm compares d with every training example in D to compute the similarity or distance between them. The k most similar (closest) examples in D are then selected. This set of examples is called the k nearest neighbors of d. d then takes the most frequent class among the k nearest neighbors. The key component of a kNN algorithm is the distance/similarity function, which is chosen based on applications and the nature of the data. For text documents, cosine similarity is a popular choice.

a) Algorithm kNN(D, d, k):

```
1. Compute the distance between d and every example
   in D;
2. Choose the k examples in D that are nearest to d,
   denote the set by P is a subset of D;
3. Assign d the class that is the most frequent class in P
   (or the majority class).
```

Figure 1. The k-nearest neighbor algorithm

b) Cosine similarity

One way to compute the degree of relevance is to calculate the similarity of the query q to each document d_j in the document collection D. There are many similarity measures. The most well-known one is the cosine similarity, which is the cosine of the angle between the query vector q and the document vector d_j:

$$cosine(\mathbf{d}_j,\mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q}\rangle}{\|\mathbf{d}_j\| \times \|\mathbf{q}\|} = \frac{\sum_{i=1}^{|V|} w_{ij}\times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2}\times\sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}. \qquad (4)$$

The dot product of the two vectors is another similarity measure:

$$sim(dj \bullet q) = < dj \bullet q > \qquad (5)$$

2) Naïve Bayesian

The naïve Bayesian learning method for text classification is derived based on a probabilistic generative model. It assumes that each document is generated by a parametric distribution governed by a set of hidden parameters [2]. Training data is used to estimate these parameters. The parameters are then applied to classify each test document using Bayes' rule by calculating the posterior probability that the distribution associated with a class would have generated the given document. Classification then becomes a simple matter of selecting the most probable class. Specifically, the naïve Bayesian classification treats each document as a "bag" of words.

C. *Performance*

1) Accuracy

The accuracy of a classification model on a test set is defined as:

$$Accuracy = \frac{No.of\ correct\ clasifications}{Total\ No.of\ test\ cases} \qquad (6)$$

where a correct classification means that the learned model predicts the same class as the original class of the test case.

2) Precision

It is the ratio between the number of relevant documents returned originally and the total number of retrieved documents returned after eliminating irrelevant documents [5]. Here the relevant documents indicate the required documents which satisfy the user needs.

$$Precision = \frac{Relevant\ \cap Retrieved\ Originally}{Retreived\ after\ Refinement} \qquad (7)$$

3) Recall

It is the ratio between the number of relevant documents returned originally and the total number of relevant documents returned after eliminating irrelevant documents [5].

$$Recall = \frac{Relevant\ \cap Retrieved\ Originally}{Relevant\ after\ Refinement} \qquad (8)$$

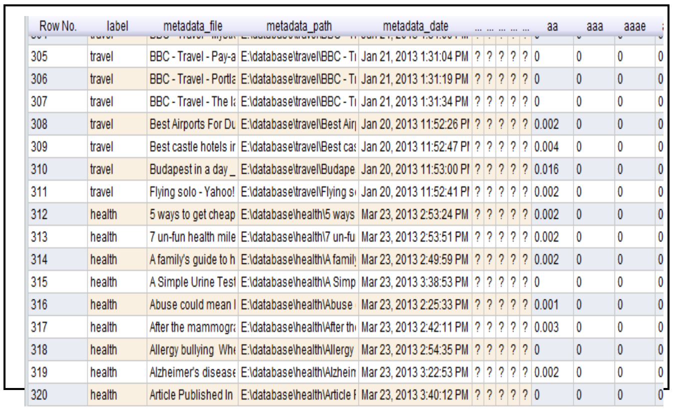
IV Experimental Evaluation

A. *Dataset*

The dataset is consist of nearly 800-900 web pages taken from different news sites like BBC, Yahoo, CNN, ABP, NASA science, etc., regarding different fields as business, health, sports, science, environment, technology, travel, entertainment, etc. The experimental evaluation takes place on rapidminer.

B. *Preprocessing of Documents*

The collection of web pages are preprocessed, the text content are extracted from the web pages removing <p>, <br>, <i>, <b>, span tags and ignoring non html tags. The retrieved text is then tokenized to smaller words, stop words are removed, and stemming process reduces the word to their root terms. The fig.2 below shows the data view of processed documents along with the metadata information and tf-idf value of different terms in different documents.



Row No.	label	metadata_file	metadata_path	metadata_date	...	...	...	...	...	aa	aaa	aaae	
305	travel	BBC - Travel - Pay-a	E:\database\travel\BBC - Tı	Jan 21, 2013 1:31:04 PM	?	?	?	?	?	0	0	0	0
306	travel	BBC - Travel - Portla	E:\database\travel\BBC - Tı	Jan 21, 2013 1:31:19 PM	?	?	?	?	?	0	0	0	0
307	travel	BBC - Travel - The l:	E:\database\travel\BBC - Tı	Jan 21, 2013 1:31:34 PM	?	?	?	?	?	0	0	0	0
308	travel	Best Airports For Du	E:\database\travel\Best Air|	Jan 20, 2013 11:52:26 PI	?	?	?	?	?	0.002	0	0	0
309	travel	Best castle hotels ir	E:\database\travel\Best ca:	Jan 20, 2013 11:52:47 PI	?	?	?	?	?	0.004	0	0	0
310	travel	Budapest in a day _	E:\database\travel\Budape	Jan 20, 2013 11:53:00 PI	?	?	?	?	?	0.016	0	0	0
311	travel	Flying solo - Yahoo!	E:\database\travel\Flying s	Jan 20, 2013 11:52:41 PI	?	?	?	?	?	0.002	0	0	0
312	health	5 ways to get cheap	E:\database\health\5 ways	Mar 23, 2013 2:53:24 PM	?	?	?	?	?	0.002	0	0	0
313	health	7 un-fun health mile	E:\database\health\7 un-fu	Mar 23, 2013 2:53:51 PM	?	?	?	?	?	0.002	0	0	0
314	health	A family's guide to h	E:\database\health\A famil	Mar 23, 2013 2:49:59 PM	?	?	?	?	?	0.002	0	0	0
315	health	A Simple Urine Test	E:\database\health\A Simp	Mar 23, 2013 3:38:53 PM	?	?	?	?	?	0	0	0	0
316	health	Abuse could mean l	E:\database\health\Abuse	Mar 23, 2013 2:25:33 PM	?	?	?	?	?	0.001	0	0	0
317	health	After the mammogr:	E:\database\health\After th	Mar 23, 2013 2:42:11 PM	?	?	?	?	?	0.003	0	0	0
318	health	Allergy bullying Wh	E:\database\health\Allergy	Mar 23, 2013 2:54:35 PM	?	?	?	?	?	0	0	0	0
319	health	Alzheimer's disease	E:\database\health\Alzhein	Mar 23, 2013 3:22:53 PM	?	?	?	?	?	0.002	0	0	0
320	health	Article Published In	E:\database\health\Article F	Mar 23, 2013 3:40:12 PM	?	?	?	?	?	0	0	0	0

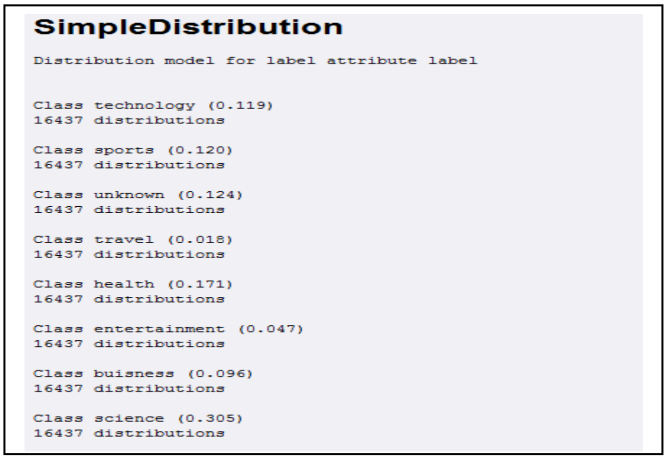
Figure 2: Data view of processed documents

C. *Classification of Web Pages: Result and Evaluation*

After preprocessing of documents, the web pages are classified to different labels based on supervised learning. In my work two different learning classification model is used to classify dataset into different labeled classes. These are: Naïve Bayesian and kNN algorithm.

1) Naïve Bayesian

In Naïve Bayesian classification, Laplace correction is involved to smoothing the probability of infrequently occurring words. The fig 3. below shows the text view of dataset distribution through Naïve Bayesian classification.



```
SimpleDistribution

Distribution model for label attribute label


Class technology (0.119)
16437 distributions

Class sports (0.120)
16437 distributions

Class unknown (0.124)
16437 distributions

Class travel (0.018)
16437 distributions

Class health (0.171)
16437 distributions

Class entertainment (0.047)
16437 distributions

Class buisness (0.096)
16437 distributions

Class science (0.305)
16437 distributions
```

Figure 3 . Dataset distribution through Naïve Bayesian

The fig 4 below is a plot view which shows the density of a word science in different labels using naïve bayes.

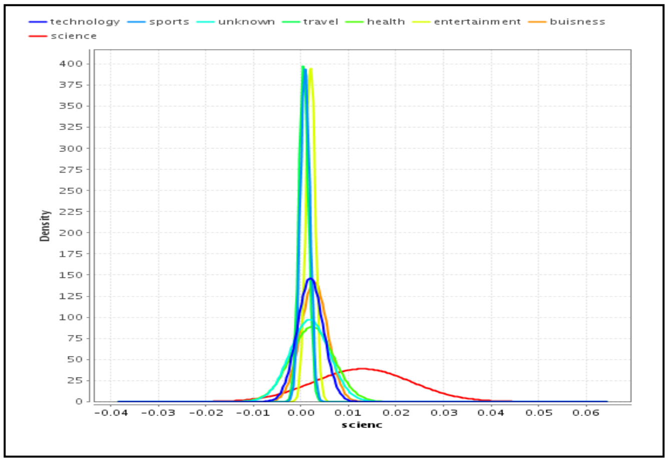
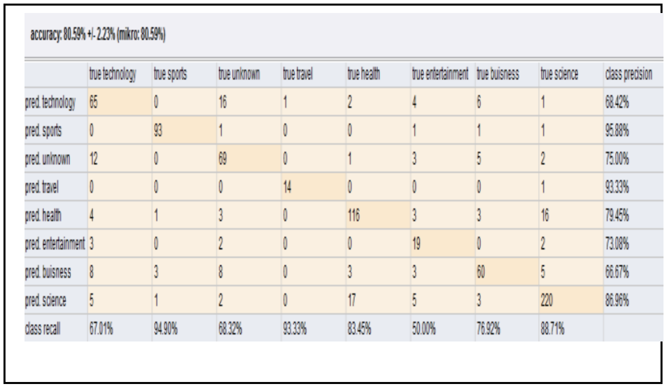


Figure 4. Density of word ‘science’ in different labels

Accuracy of Naïve Bayesian classification is 80.59% as shown in fig 5

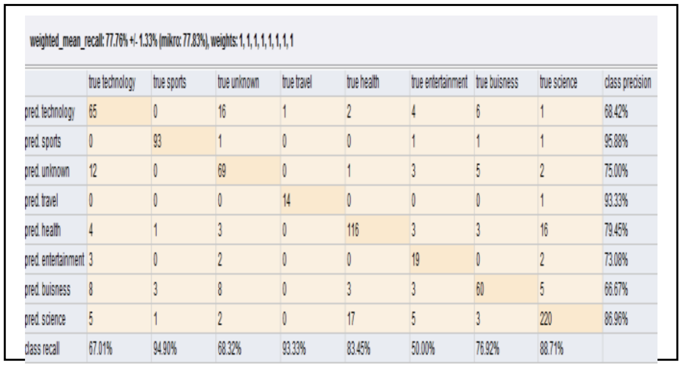


accuracy: 80.59% +/- 2.23% (mikro: 80.59%)

	true technology	true sports	true unknown	true travel	true health	true entertainment	true buisness	true science	class precision
pred. technology	65	0	16	1	2	4	6	1	68.42%
pred. sports	0	93	1	0	0	1	1	1	95.88%
pred. unknown	12	0	69	0	1	3	5	2	75.00%
pred. travel	0	0	0	14	0	0	0	1	93.33%
pred. health	4	1	3	0	116	3	3	16	79.45%
pred. entertainment	3	0	2	0	0	19	0	2	73.08%
pred. buisness	8	3	8	0	3	3	60	5	66.67%
pred. science	5	1	2	0	17	5	3	220	86.96%
class recall	67.01%	94.90%	68.32%	93.33%	83.45%	50.00%	76.92%	88.71%	

Figure 5 Accuracy of Naïve Bayesian classification

Computed Recall value of Naïve Bayesian classification is 77.76%as shown in fig 6.

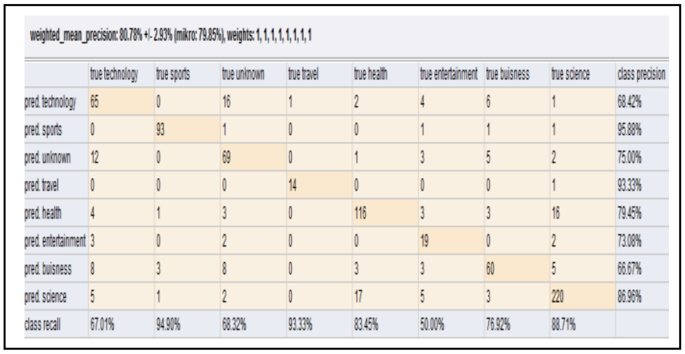


weighted_mean_recall: 77.76% +/- 1.33% (mikro: 77.83%), weights: 1, 1, 1, 1, 1, 1, 1, 1

	true technology	true sports	true unknown	true travel	true health	true entertainment	true buisness	true science	class precision
pred. technology	65	0	16	1	2	4	6	1	68.42%
pred. sports	0	93	1	0	0	1	1	1	95.88%
pred. unknown	12	0	69	0	1	3	5	2	75.00%
pred. travel	0	0	0	14	0	0	0	1	93.33%
pred. health	4	1	3	0	116	3	3	16	79.45%
pred. entertainment	3	0	2	0	0	19	0	2	73.08%
pred. buisness	8	3	8	0	3	3	60	5	66.67%
pred. science	5	1	2	0	17	5	3	220	86.96%
class recall	67.01%	94.90%	68.32%	93.33%	83.45%	50.00%	76.92%	88.71%	

Figure 6. Recall value of Naïve Bayesian classification

Precision value calculated using Naïve Bayesian is 80.78% as shown in fig 7.



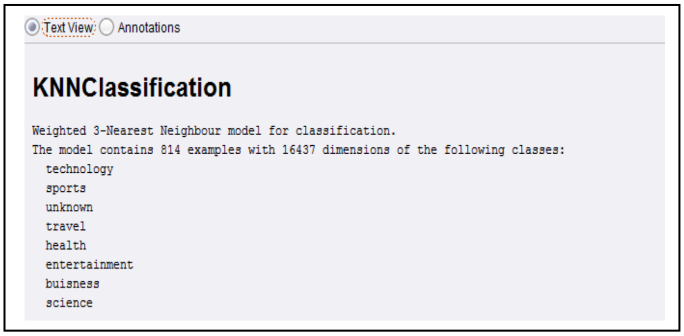
weighted_mean_precision: 80.78% +/- 2.93% (mikro: 79.85%), weights: 1, 1, 1, 1, 1, 1, 1, 1

	true technology	true sports	true unknown	true travel	true health	true entertainment	true buisness	true science	class precision
pred. technology	65	0	16	1	2	4	6	1	68.42%
pred. sports	0	93	1	0	0	1	1	1	95.88%
pred. unknown	12	0	69	0	1	3	5	2	75.00%
pred. travel	0	0	0	14	0	0	0	1	93.33%
pred. health	4	1	3	0	116	3	3	16	79.45%
pred. entertainment	3	0	2	0	0	19	0	2	73.08%
pred. buisness	8	3	8	0	3	3	60	5	66.67%
pred. science	5	1	2	0	17	5	3	220	86.96%
class recall	67.01%	94.90%	68.32%	93.33%	83.45%	50.00%	76.92%	88.71%	

Figure 7. Precision value computed through Naïve Bayesian

2) K-Nearest Neighbor

In kNN classification, k is taken as 3 and the classification is numerically computed using cosine similarity. The fig 8. below shows the text view of how the web pages are distributed to different labels using kNN classification.

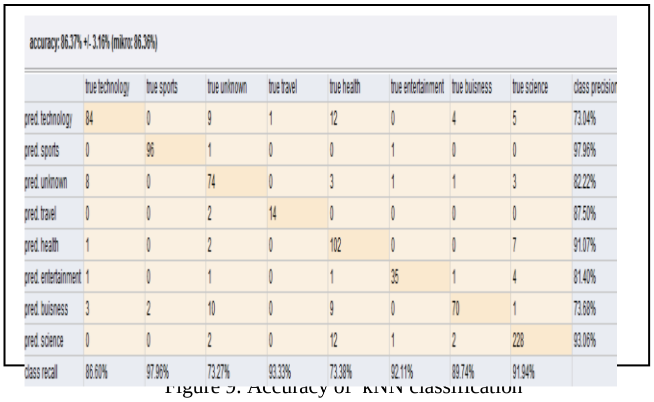


```
KNNClassification

Weighted 3-Nearest Neighbour model for classification.
The model contains 814 examples with 16437 dimensions of the following classes:
  technology
  sports
  unknown
  travel
  health
  entertainment
  buisness
  science
```

Figure 8. Text view of kNN distribution

Accuracy of kNN learning classification is 86.37% as depicted in fig 9. below:

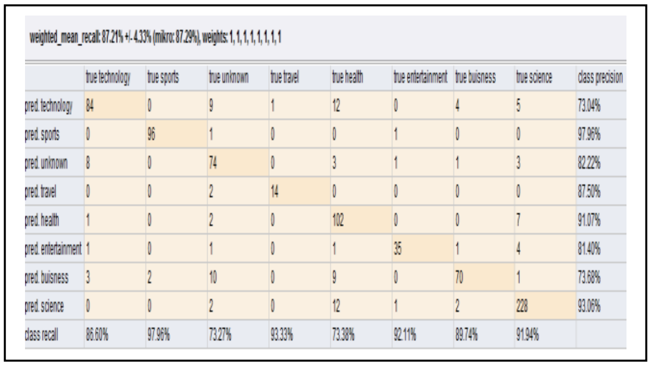


accuracy: 86.37% +/- 3.16% (mikro: 86.36%)

	true technology	true sports	true unknown	true travel	true health	true entertainment	true buisness	true science	class precisio
pred. technology	84	0	9	1	12	0	4	5	73.04%
pred. sports	0	96	1	0	0	1	0	0	97.96%
pred. unknown	8	0	74	0	3	1	1	3	82.22%
pred. travel	0	0	2	14	0	0	0	0	87.50%
pred. health	1	0	2	0	102	0	0	7	91.07%
pred. entertainment	1	0	1	0	1	35	1	4	81.40%
pred. buisness	3	2	10	0	9	0	70	1	73.68%
pred. science	0	0	2	0	12	1	2	228	93.06%
class recall	86.60%	97.96%	73.27%	93.33%	73.38%	92.11%	89.74%	91.94%	

Figure 9. Accuracy of kNN classification

Recall percentage computed is 87.12% as depicted in fig 10. below:

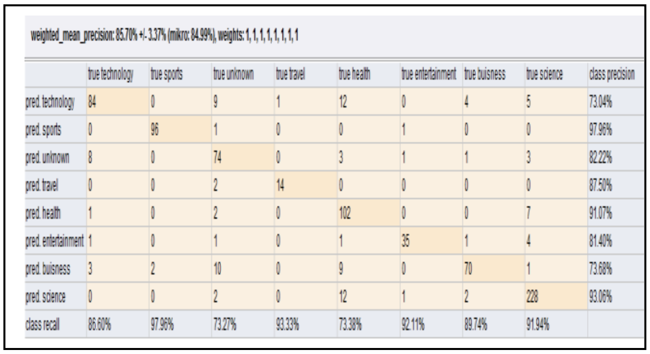


weighted_mean_recall: 87.21% +/- 4.33% (mikro: 87.29%), weights: 1, 1, 1, 1, 1, 1, 1, 1

	true technology	true sports	true unknown	true travel	true health	true entertainment	true buisness	true science	class precision
pred. technology	84	0	9	1	12	0	4	5	73.04%
pred. sports	0	96	1	0	0	1	0	0	97.96%
pred. unknown	8	0	74	0	3	1	1	3	82.22%
pred. travel	0	0	2	14	0	0	0	0	87.50%
pred. health	1	0	2	0	102	0	0	7	91.07%
pred. entertainment	1	0	1	0	1	35	1	4	81.40%
pred. buisness	3	2	10	0	9	0	70	1	73.68%
pred. science	0	0	2	0	12	1	2	228	93.06%
class recall	86.60%	97.96%	73.27%	93.33%	73.38%	92.11%	89.74%	91.94%	

Figure 10. Recall value of kNN classification

Precision of kNN classification is 85.70% as shown in fig 11. below:



weighted_mean_precision: 85.70% +/- 3.37% (mikro: 84.99%), weights: 1, 1, 1, 1, 1, 1, 1, 1

	true technology	true sports	true unknown	true travel	true health	true entertainment	true buisness	true science	class precision
pred. technology	84	0	9	1	12	0	4	5	73.04%
pred. sports	0	96	1	0	0	1	0	0	97.96%
pred. unknown	8	0	74	0	3	1	1	3	82.22%
pred. travel	0	0	2	14	0	0	0	0	87.50%
pred. health	1	0	2	0	102	0	0	7	91.07%
pred. entertainment	1	0	1	0	1	35	1	4	81.40%
pred. buisness	3	2	10	0	9	0	70	1	73.68%
pred. science	0	0	2	0	12	1	2	228	93.06%
class recall	86.60%	97.96%	73.27%	93.33%	73.38%	92.11%	89.74%	91.94%	

Figure 11. Precision value of kNN classification

Conclusion

With the increasing proliferation of information on web, it is required to have an efficient technique that can efficiently classify web pages. In this work I generated an automatic classification of web pages using two different learning classification method i.e, kNN and Naïve Bayesian and it can be concluded from the above results that kNN classification method produces better accuracy, precision and recall value as compared to Naïve Bayesian. Accuracy of kNN classification is 86.37% while that of Naïve Bayesian is 80.59% which is also good but less accurate classification than kNN. This work can be further extended by comparing it with other classification algorithm or by using a different software.

References

[1] A. Selamat , S.Omatu, “Web Page Feature Selection And Classification Using Neural Networks,” Information Sciences, vol 158, pp 69-88, 2004.

[2] B.Liu,, “Web Data Mining – Exploring Hyperlinks, Contents, and Usage Data,” Springer, December 2006.

[3] Durant K. T., Smith M. D., “Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection.,” Springer-Verlag Berlin Heidelberg. LNAI 4811, pp. 187- 206, WebKDD.2006.

[4] F. Bifigueiredo , L. Rocha, T. Couto, T. Salles , M. Gonc- alves, W. MeiraJr.B, “Word co-occurrence features for text classification.,” Information Systems, Elseveir.,vol 36, pp 843-858 , 2011.

[5] G. Poonkuzhali, K. Thiagarajan, K. Sarukesi and G.V. Uma, ”Signed Approach for Mining Web Content Outliers,” World Academy of Science, Engineering and Technology. 56, 2009.

[6] H. Lili.,H. Lizhu, “ Automatic identification of stopwords in Chinese text classification.,” In proceedings of the IEEE international conference on Computer Science and Software Engineering, pp. 718 – 722, 2008.

[7] H. Mahgoub, D. Rösner, N. Ismail, F. Torkey, “A Text Mining Technique Using Association Rules Extraction,“ International Journal of Information and Mathematical Sciences, vol 4, 2008.

[8] Kim S., Han K., Rim H., Myaeng S. H. ,”Some effective techniques for naïve bayes text classification,” IEEE Transactions on Knowledge and Data Engineering. vol. 18, no. 11, pp. 1457-1466., 2006.

[9] M.K. Dalal, M.A. Zaveri, “Automatic Text Classification: A Technical Review’” International Journal of Computer Applications. vol 28- No.2, pp 0975 -8887, August 2011.

[10] M. M. S. Missen, and M. Boughanem, “Using WordNet’s semantic relations for opinion detection in Blog,” Springer-Verlag Berlin Heidelberg, LNCS 5478, pp. 729-733, ECIR 2009.

[11] Porter M. F., “An algorithm for suffix stripping.,” Program, 14 (3). pp. 130-137, 1980.

[12] N. H. Ali, N. S. Ibrahim, “Porter Stemming Algorithm for Semantic Checking,” ICCIT 2012.

[13] R. A. Shalabi, G. Kanaan, J.M..Jaam, A. Hasnah, E. Hilat, “Stop Word Removal Algorithm for Arabic Language,” IEEE. 2004.