Automated Classification of Web Sites using Naive Bayesian Algorithm

Ajay S. Patil, B.V. Pawar

Abstract— Subject based web directories like Open Directory Project's (ODP) Directory Mozilla (DMOZ), Yahoo etc., consists of web pages classified into various categories. The proper classification has made these directories popular among the web users. The exponential growth of the web has made it difficult to manage human edited subject based web directories. The World Wide Web (WWW) lacks a comprehensive web site directory. Web site classification using machine learning techniques is therefore an emerging possibility to automatically maintain directory services for the web. Home page of a web site is a distinguished page and it acts as an entry point by providing links to the rest of the web site. The information contained in the title, meta keyword, description and in the labels of the anchor (A HREF) tags along with the other content is a very rich source of features required for classification. Compared to the other pages of the website, webmasters take more care to design the homepage and its content to give it an aesthetic look and at the same time attempt to precisely summarize the organization to which the site belongs. This expression power of the home page of a website can be exploited to identify the nature of the organization. In this paper we attempt to classify web sites based on the content of their home pages using the Naïve Bayesian machine learning algorithm.

***Index Terms*- classification, machine learning, Naïve Bayesian algorithm, Web mining.**

I. Introduction

The World Wide Web (WWW) service started in the year 1991 and is gaining popularity day by day. Netcraft's January 2012 survey, estimates about 584 million web sites on the web and out of which, nearly 175.2 million are active. The number of users using the Internet is rapidly increasing. Internet World Stats reveals that the world Internet usage growth has increased by 480.4% during 2000-2011. In order to locate information from millions of web sites the Internet users use various search tools broadly classified as: 1. Crawler based Search Engines (SE) e.g., Google, Bing, Yahoo etc., 2. Subject Directories like DMOZ (Directory Mozilla), Librarians Internet Index (LII) etc. and 3. Meta Search Engines e.g., Metacrawler, Clusty etc. The crawler based search engines and subject directories have data repository of their own, whereas meta search engines do not maintain such data repositories. The meta search engines depend on indices of other SE's and subject directories to answer user queries. The crawler based search engines have considerably large data indexed in their databases as compared to the subject directories. The subject directories are manually edited by editors. Subject directories are popular due to the proper classification of data in several categories. Manual classification is expensive to scale and is highly labor intensive. Directory Mozilla (DMOZ) [1] i.e., dmoz.com has 93,431 editors for one million categories and has indexed 4.98 million websites, which is only 2.5% of the total active web sites available on the Internet today. Therefore there is a need to automate the process of creating and maintaining subject directories. Current search engines fail to answer user queries like listing organizations related to a particular business or situated in a particular region etc. Queries of such type can be answered if websites were to be classified (web site directory) according to the different categories. Web site directory shall be further helpful in improving the quality of search results, web content filtering, development of knowledge bases, building efficient focused crawlers or vertical (domain specific) search engines.

This paper describes Naïve Bayesian (NB) approach for the automatic classification of web sites based on content of home pages. The NB approach, is one of the most effective and straightforward method for text document classification and has exhibited good results in previous studies conducted for data mining. The rest of the paper is organized as follows. Section II reviews previous work on the machine learning and classification. Section III and IV discusses the classification of web pages and Naïve Bayes Theorem respectively, Section V presents our approach of classifying websites based on home pages using NB technique. Section VI discusses the results of our experiment. The last section summarizes the paper and gives some directions for future research.

II. Related work

This section briefly reviews related work on text classification with special emphasis on classification of web pages. In the early days, classification was done manually by domain experts. But very soon, classification was also carried out in semi-automatic or automatic manner. Some of the approaches for text-categorization include statistical and machine leaning techniques like k-Nearest Neighbor approach [2], Bayesian probabilistic models [3]-[4], inductive rule learning [5], decision trees [4],[6], neural networks [7],[8] and support vector machines [9],[10]. While most of the learning methods have been applied to pure text documents, there are numerous publications dealing with classification of web pages. Pierre [11] discusses various practical issues in automated

Manuscript received January 04, 2012; revised January 19, 2012. This work is supported in part by the University Grants Commission, New Delhi, India under its Research Project Scheme for Teachers..

Ajay S. Patil is with the North Maharashtra University, Jalgaon (MS), India (+91+257-2257453; +91-9423975215 e-mail: aspatil@ nmu.ac.in).

B. V. Pawar is with the North Maharashtra University, Jalgaon (MS), India (+91+257-2257451; e-mail: bvpawar@ nmu.ac.in).

categorization of web sites. Machine and statistical learning algorithms have also been applied for classification of web pages [12]-[15]. In order to exploit the hypertext based organization of the web page several techniques like building implicit links [16], removal of noisy hyperlinks[17], fusion of heterogeneous data[18], link and context analysis[19] and web summarization[20] are used. An effort has been made to classify web content based on hierarchical structure [21].

III. Classification of Web Pages

Classification of web content is different in some aspects as compared with text classification. The uncontrolled nature of web content presents additional challenges to web page classification as compared to traditional text classification. The web content is semi structured and contains formatting information in form of HTML tags. A web page consists of hyperlinks to point to other pages. This interconnected nature of web pages provides features that can be of greater help in classification. First all HTML tags are removed from the web pages, including punctuation marks. The next step is to remove stop words as they are common to all documents and does not contribute much in searching. In most cases a stemming algorithm is applied to reduce words to their basic stem. One such frequently used stemmer is the Porter's stemming algorithm [22]. Each text document obtained by application of procedures discussed above is represented as frequency vector. Machine learning algorithms are then applied on such vectors for the purpose of training the respective classifier. The classification mechanism of the algorithm is used to test an unlabelled sample document against the learnt data. In our approach we deal with home pages of organizational websites. A neatly developed home page of a web site is treated as an entry point for the entire web site. It represents the summary of the rest of the web site. Many URLs link to the second level pages telling more about the nature of the organization. The information contained the title, meta keyword, meta description and in the labels of the A HREF (anchor) tags are very important source of rich features. In order to rank high in search engine results, site promoters pump in many relevant keywords. This additional information can also be exploited. Most of the homepages are designed to fit in a single screen. The factors discussed above contribute to the expression power of the home page to identify the nature of the organization.

IV. Bayes Theorem

Consider *D={d1,d2,d3,…dp}* to be a set of documents and *C={c1,c2,c3,….cq}* be set of classes. Each of the *p* number of documents in *D* are classified into one of the *q* number classes from set *C*. The probability of a document d being in class c using Bayes theorem is given by:

$$c_{map} = \arg\max_{c \in C} P(c \mid d) = \arg\max_{c \in C} \frac{P(c)P(d \mid c)}{P(d)}$$

As *P(d)* is independent of the class, it can be ignored.

$$= \arg\max_{c \in C} P(c)P(d \mid c) \quad \text{.....(1)}$$

A. Naïve Bayesian Assumption

Assuming that the attributes (terms) are independent of each other,

$$P(d \mid c) = P(t_1 \mid c)P(t_2 \mid c)P(t_3 \mid c)...P(t_{n_d} \mid c) = \prod_{1 \le k \le n_d} P(t_k \mid c)$$

Replacing (1),

$$c_{map} = \arg\max_{c \in C} P(c \mid d) = \arg\max_{c \in C} P(c)P(d \mid c)$$

$$= \arg\max_{c \in C} P(c) \prod_{1 \le k \le n_d} P(t_k \mid c)$$

P(c), the prior probability of *c* is calculated as:

$$P(c) = \frac{N_c}{N}$$

Where N_c is number of training documents in class *c*, *N* is the number of training documents. *P(c|d)* is called the posterior probability of *c*, as it reflects our confidence that *c* holds after we have seen *d*.

$$P(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

Here, T_{ct} is the number of occurrences of *t* in *D* from class c, and $\sum_{t' \in V} T_{ct'}$ is the total number of terms in *D* from class c.

B. Laplacean Smoothing

A term-class combination that does not occur in the training data makes the entire result zero. In order to solve this problem we use add-one smoothing or Laplace smoothing. The equation after adding Laplace's correction becomes:

$$P(t \mid c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'}) + |V|}$$

C. Underflow Condition

Multiplying lots of probabilities, which are between *0* and *1* by definition, can result in floating-point underflow. Since *log(xy) = log(x) + log(y)*, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities. Class with highest final un-normalized log probability score is the most probable.

V. Experimental Setup

This section discusses the setup of the entire experiment. To start with we require collection of homepages of websites pre-classified into different categories. We obtain the dataset from these pre classified pages by cleaning them by removing HTML tags, scripts, style sheets etc. contained in them. The dataset is then subjected to training and testing the classifier. All of this is briefly discussed as given below.

A. Creation of Data Set

Our data set consisted of home pages in HTML (Hypertext Markup Language) format belonging to 10 different categories mentioned in Table I. In order to create the dataset, services of various search engines and subject directories were used. The popular search engines like Google, Bing, Altavista etc., were submitted keyword based queries and then the results obtained were examined. If any of the links in the results pointed to a homepage, we visually examined its contents with the help of internet browser software. Home pages that belonged to the categories of our interest were saved in respective directories. Since our classification is text based, home pages of few web sites that

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

were implemented in technologies such as Flash or made used of other plug in applications were not included in the dataset. Home pages other than those in English and having size less than 200 characters were also ignored. Hereafter we refer to these home pages as documents for simplicity. The entire dataset was then independently subjected to two annotators for moderation. Documents were removed from the dataset wherever there was a disagreement between the annotators. The dataset consisted of 4887 documents in ten different categories.

B. Cleaning HTML Documents

The Jericho HTML Parser (Version 3.2) [23] was used to extract the HREF (hyperlink) label, TITLE, META DESCRIPTION and META KEYWORD and all BODY text containing in each document of the dataset. The Jericho HTML Parser is an open source library released under both the Eclipse Public License (EPL) and GNU Lesser General Public License (LGPL). This library is available online on the internet at http://jerichohtml.sourceforge.net. The Jericho HTML Parser 3.2 is a powerful java library allowing analysis and manipulation of parts of an HTML document, including server-side tags. One advantage of using this library is that the presence of badly formatted HTML does not interfere with the parsing of the rest of the document. In many web site images were as buttons to be clicked in place of hyperlinks or images were used to display name of the organization. Such information is a very important feature for classification purpose, however our experiment concentrates on text based retrieval so such graphical text we ignored.

The standard stop word list used in Bow [24] was used. Bow is a library of C code useful for writing statistical text analysis, language modeling and information retrieval programs. Words in plural format were converted into their singular version using similar approach. We applied stemming by constructing a map of words and their relevant stems. We did not use stemming algorithm such as Porter as in most cases stemming totally changes the meaning of the word and in some cases it is undesired. e.g., In case of the book category the word “book” refers to a physical book, whereas in the hotel category the word “booking” which also stems to “book”, refers to “reservation”.

TABLE I
DATA SET

Category	Total Samples	Training Samples	Test Samples
Academic Institutions	503	453	50
Hotels	470	423	47
Book Sellers/Publisher	485	437	48
Health Care	511	460	51
Sports	476	428	48
Automobiles	495	445	50
Tours & Travel	475	427	48
Computer	502	452	50
Banking	490	441	49
Domestic Appliances	480	432	48
Total	4887	4398	489

C. Vocabulary Generation

Common features that are part of every web site were considered as stop features (About Us, Home, All Rights Reserved, Contact Us, Feedback etc). Such words are similar to regular stop words but specific to home pages. Some of these home page specific stop words are mentioned in table II. Such words were also considered as stop words and therefore were removed from the dataset.

TABLE II
HOME PAGE SPECIFIC STOP WORDS

Information, login, view, browser, website, web, online, search, keyword, designed, copyright, rights, reserved, click, search, welcome, email, click, contact, developed, mail, home, page, feedback, webmaster …

It was also observed that webmasters inflated the title, meta description and keyword tags with multiple keywords. We normalized such repeating keywords to reduce the impact of site promotion techniques applied by webmasters. This step was performed during the cleaning phase. A vector called as vocabulary containing the most relevant words (features) was created for the experiment. The relevant words or features were those words that occurred more than seven times in the entire training set. Thus very rare words and also the very common words (stop words) were eliminated. The vocabulary count was 4500 for 4,398 documents of the training set. We term this vocabulary set as *V*. The next section discusses training and testing of the classifier.

D. Training the Classifier

The K fold strategy (with $k=10$) was followed to decide the number of training and testing examples. Nine folds i.e., 4398 examples were used as the training set to build the classifier and the remaining fold 489 examples were used to test the classifier for accuracy. The prior probability for each category is 1/10 (as there are 10 categories). The posterior probability $P(w_k|c)$ was calculated as follows. All documents that belonged to respective categories were parsed and a hash table was prepared for each category. All words in the vocabulary served as keys of the hash table. The values of the hash table were the word occurrence frequency (n_k) in all documents belonging to that category. The total word count (including repeats) for each category termed as n was also calculated. The posterior probability with Laplace’s correction was calculated using the formula $P(w_k|c) = (n_k +1) / (n + |Vocabulary|)$. Partial feature sets generated as an effect of the experiment for academic and sports categories are given in Table II and III.

TABLE II
SAMPLE FEATURE SET FOR ACADEMIC INSTITUTIONS CATEGORY

university, school, department, syllabus, student, alumina, placement, examination, result, principal, chancellor, campus, registrar, library, study, course, information, education, PG, center, technology, conference, administration, workshop, science, commerce, faculty, programme, academic,….

TABLE III
SAMPLE FEATURE SET FOR SPORTS CATEGORY

cricket, sports, score, goal, stadium, ground, kit, ball, umpire, referee, stumps, hockey, football, badminton, wrestling, player, commentary, highlight, victory, win, won, team, wicket, field, game, match, penalty, corner, kick, service, court, seed, scorecard, tour, champion,…

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

E. Testing the Classifier

In order to classify a document say *X*, the probabilities of a given category are looked up in the hash table and multiplied together. The category producing the highest probability is the classification for document *X*. Only the words found in *X* would be looked up in the hash table. Also if a word in *X* is absent in the original vocabulary (built from training set) the word is ignored. The equation used to classify *X* is $C = arg\ max\ (\ P(c)\ \Pi\ P(wk|c)\ )$. The Naïve Bayes algorithms to train and test the classifier are as given below:

ALGORITHM NB TRAINING

1. Let V be the vocabulary of ALL words in the documents in D

2. For each category $c_i \in C$
   - Let Di be the subset of documents in D in category c_i
   - $P(c_i) = |D_i| / |D|$
   - Let T_i be the concatenation of all the documents in D_i
   - Let n_i be the total number of word occurrences in T_i
   - For each word $w_j \in V$
     - Let n_{ij} be the number of occurrences of w_j in T_i
     - Let $P(w_j \mid c_i) = (n_{ij} + 1) / (n_i + |V|)$

ALGORITHM NB TESTING

1. Given a test document *X*
2. Let *n* be the number of word occurrences in *X*
3. Return the category:

$$arg\ max_{c_i \in C}\ P(c_i) = \prod_{i=1}^{n} P(a_i \mid c_i)$$

where a_i is the word occurring at the i^{th} position in *X*

VI. EXPERIMENTAL RESULTS

Table IV shows the results obtained when nine folds i.e., 4398 examples were used as the training set to build the classifier and the remaining fold 489 examples were used to test the classifier for accuracy. We use recall [25], precision [25] and F-measure [25] to verify the accuracy of our classification approach. F-measure is the harmonic mean of recall and precision. Recall, Precision and F-Measure are calculated as follows:

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$$

$$\text{F-measure} = \frac{(2 \text{ x Recall x Precision})}{(\text{Recall + Precision})}$$

Using the three measures, we observe that the average precision is 89.09%, average recall is 89.04%, whereas the F-Measure is 89.05%. Thus, classification of web sites is possible by examining the contents of their home pages.

TABLE IV
CLASSIFICATION ACCURACY

Category	Precision	Recall	F-Measure
Academic Institutions	93.36	89.46	91.37
Hotels	89.29	90.43	89.86
Book Sellers	88.66	88.66	88.66
Hospitals	85.13	85.13	85.13
Sports	93.36	88.66	90.95
Automobiles	88.65	89.90	89.27
Tours & Travel	90.21	89.26	89.73
Computer Dealers	86.44	87.65	87.04
Banks	88.89	89.80	89.34
Domestic Appliances	86.93	91.46	89.14
Average	89.09	89.04	89.05

Number of Training Examples and Accuracy

The classifier was subjected to training and testing in 9 steps each time increasing the input by 50 documents. Graph 1 depicts the number of training examples versus the accuracy in terms of average F-measure. The accuracy of the classifier was very poor i.e., about 45%, when only 50 documents were supplied as training data. The accuracy increases each time when the classifier is supplied with additional learning data. The classifier achieved an accuracy of 89% when nearly 450 documents were supplied as input in each category.

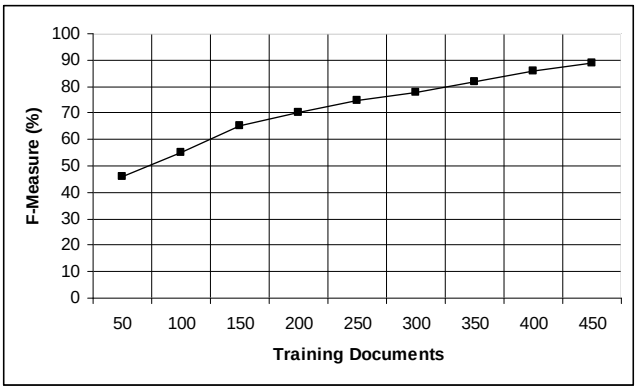

Fig 2. Number of training documents versus accuracy.

Thus, the accuracy of the classifier depends on the number of training documents and in order to achieve high accuracy, the classifier should be supplied with sufficiently large training documents.

VII. CONCLUSION AND FUTURE WORK

The NB approach used in this paper exploits the richness of features of a home page of a website for classification into industry type category. It categorizes the web pages into very broad categories. NB approach for classification of home pages for the ten categories considered above yielded 89.05% accuracy. It is also observed that the classification accuracy of the classifier is proportional to number of training documents. The results are quite encouraging. This approach can be used by search engines for effective categorization of websites to build an automated website directory based on type of organization. However in this experiment, only distinct and non hierarchical categories are considered. The same algorithm could also be used to classify the pages into more specific categories (hierarchical classification) by changing the feature set e.g. a web site that is academic may be further classified into school, college or a university website.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

REFERENCES

[1] DMOZ open directory project. [Online]. Available: http://dmoz.org/
[2] G. Guo, H. Wang and K. Greer, "An kNN model-based approach and its application in text categorization", *5th Int. Conf., CICLing* Springer, Seoul, Korea, 2004, pp. 559-570.
[3] A. McCallum and K. Nigam, "A comparison of event models for Naïve Bayes text classification", in *AAAI/ICML-98 Workshop on Learning for Text Categorization*, 1998, pp. 41-48.
[4] D.D. Lewis and M. Ringuette, "A Classification of two learning algorithms for text categorization", in *Proc. of 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994, pp. 81-93.
[5] S.T. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization", in *Proc. of the 17th Int. Conf. on Information and Knowledge Management (CIKM'98)*, 1998, pp. 148-155.
[6] C. Apte and F. Damerau and S. M. Weiss, "Automated learning of decision rules for text categorization", *ACM Trans. on Information Systems*, Vol. 12, no.3, pp. 233-251, 1994.
[7] S. Wermter, "Neural network agents for learning semantic text classification", *Information Retrieval*, Vol. 3, no. 2, pp. 87 - 103, Jul 2000.
[8] A.S. Weigend, E.D. Weiner, and J.O. Peterson, "Exploiting hierarchy in text categorization", *Information Retrieval*, Vol. 1, no. 3, pp.193-216, 1999.
[9] E. Leopold, and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?", *Machine Learning,* Vol. 46, no. 1-3, pp. 423-444, 2002
[10] D. Bennett and A. Demiritz, "Semi-Supervised support vector machines", *Advances in Neural Information Processing Systems*, Vol. 11, pp. 368-374, 1998.
[11] J. M. Pierre, "Practical issues for automated categorization of web sites.", in *Electronic Proc. of ECDL 2000 workshop on the Semantic Web*, Lisbon, Portugal, 2000.
[12] A. Sun, E. Lim and W. Ng, "Web classification using support vector machine", in *Proc. of the 4th Int. workshop on Web information and data management*, McLean, Virginia, USA, 2002, pp. 96 – 99.
[13] Y. Zhang and B. F. L. Xiao, "Web page classification based on a least square support vector machine with latent semantic analysis", in *Proc. of the 5th Int. Conf. on Fuzzy Systems and Knowledge Discovery 2008*, Vol. 2, pp. 528-532,
[14] O. Kwon and J. Lee, "Web page classification based on k-nearest neighbor approach", in *Proc. of the 5th Int. Workshop on Information Retrieval with Asian languages*, Hong Kong, China, 2000, pp. 9-15.
[15] S. Dehghan and A. M. Rahmani, "A classifier-CMAC neural network model for web mining", in *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 2008*, Vol. 1, pp. 427-431.
[16] S. Dou, S. Jian-Tao , Y. Qiang and C. Zheng, "A comparison of implicit and explicit lin
[17] ks for web page classification", in *Proc. of the 15th International Conference on World Wide Web*, Edinburgh, Scotland, 2006 , pp. 643–650.
[18] S. Zhongzhi and L. Xiaoli, "Innovating web page classification through reducing noise", *Journal of Computer Science and Technology*, Vol. 17, no. 1 , pp. 9–17, Jan. 2002
[19] Z. Xu, I. King and M. R. Lyu, "Web page classification with heterogeneous data fusion", in *Proc. of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, 2007, pp. 1171 – 1172,
[20] G. Attardi, A. Gulli, and F. Sebastiani, "Automatic web page categorization by link and context analysis", in *Chris Hutchison and Gaetano Lanzarone (eds.), Proc. of THAI'99, 1999*, pp. 105-119.
[21] S. Dou, C. Zheng, Y. Qiang, Z. Hua-Jun, Z. Benyu, L. Yuchang and M. Wei-Ying, "Web-page classification through summarization", in *Proc. of the 27th annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Sheffield, United Kingdom, 2004, pp. 242 - 249.
[22] S. Dumais and H. Chen, "Hierarchical classification of web content", in *Proc. of the 23rd annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Athens, Greece, 2000, pp. 256 - 263.
[23] M.F. Porter, "An algorithm for suffix stripping", *Program*, Vo.14, no. 3, pp. 130-137, Jul. 1980.
[24] The Jericho HTML Parser Library Version 3.2, [Online]. Available : http://www.jerichohtml.sourceforge.net
[25] The BOW or libbow C Library [Online]. Available: http://www.cs.cmu.edu/~mccallum/bow/
[26] C. J. van Rijsbergen. (1979). Information Retrieval (2nd ed.) London: Butterworths,[Online].Available:http://www.dcs.gla.ac.uk/Keith/Preface.html
[27] T. M Mitchell. (1997). Machine Learning McGraw-Hill Companies, Inc.

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)