ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/330878207

Website Classification Using Word Based Multiple N -Gram Models and Random Search Oriented Feature Parameters

Conference Paper · December 2018

DOI: 10.1109/ICCITECHN.2018.8631907

CITATIONS
6

READS
1,361

4 authors:

Ashadullah Shawon
Frontier Semiconductor
11 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Syed Tauhid Zuhori
University of Alberta
22 PUBLICATIONS **85** CITATIONS
SEE PROFILE

Firoz Mahmud
Rajshahi University of Engineering & Technology
36 PUBLICATIONS **186** CITATIONS
SEE PROFILE

Md. Jamil-Ur Rahman
University of Alberta
11 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Theoretical analysis of algorithms View project

RUET OJ View project

All content following this page was uploaded by Ashadullah Shawon on 06 February 2019.

The user has requested enhancement of the downloaded file.

2018 21st International Conference of Computer and Information Technology (ICCIT), 21-23 December, 2018

Website Classification Using Word Based Multiple N-Gram Models And Random Search Oriented Feature Parameters

Ashadullah Shawon
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
shawonashadullah@gmail.com

Syed Tauhid Zuhori
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
tauhid.ruet04@gmail.com

Firoz Mahmud
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
fmahmud.ruet@gmail.com

Md. Jamil-Ur Rahman
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
jamilruet13@gmail.com

***Abstract*—Website classification is a convenient starting point for building an intelligent web browser and social networking sites that can understand the favorite categories of a user and also detect adult or harmful websites perfectly. Classifying the websites using the information of the Uniform Resource Locator (URL) is an important and fast technique. A perfect result is needed for URL classification to make it usable in the real world applications. So we have proposed an improved approach for URL classification that is able to provide a better result. We have introduced the word-based multiple n-gram models for efficient feature extraction and multinomial distribution for Naïve Bayes classifier under the Random Search pipeline for hyperparameter optimization that finds the best parameters of the URL features. The experimental result of our research is compared with the result of previous research works and we have shown a better result than the existing result. Our experimental result provides 88.77% in recall and 87.63% in F1-Score which is the best performance so far.**

Keywords— Multiple N-gram Models, Random Search, URL Classification, Website Classification, Multinomial Naïve Bayes, Web Mining

I. INTRODUCTION

Generally, websites can be classified using the contents such as title, description, Meta keywords and link structure of the web graph [1, 2, 3]. But these factors are not feasible when we want to classify the websites in real time without visiting, fetching and downloading web pages. In these circumstances, URL based website classification has some advantages over the contents based classification because of speed, real-time classification, classification before loading the web page and classification when content is hidden [4]. According to this paper [5], website classification using URL features is now the recent trend and this is the most advantageous as there is no need to preprocess large contents of the web pages. URL itself is quite informative, human readable and human can get the hints of the category from URL [6]. As an example, http://xbox360.gamespy.com/xbox-360 is from Games category. It is easy to find that the URL is from Games category by showing the **Xbox** and **game** words. Besides the URL of every website is unique and creators always set URL name according to website category. So URL classification can play a significant role in the real world applications such as managing the user's post or messages in social networking, forums, and emails. As an example, a user may not like to see adult contents, games and sports-related website links on his news feed or message box. In this situation, social networking websites or forums can get the category of shared URLs by URL classification and block those types of links for that user. Search engines can also index the websites more quickly and categorize the search results. Moreover, it will be easy for email providers to classify the messages containing URLs by URL classification. According to classified URLs, email providers can distribute the messages to primary, social, promotion or spam folder. Web browsers can also recommend websites accurately to users by website classification. That's why we became motivated to research on website classification. Another reason for our interest in website classification is the largest dataset DMOZ [7] which was previously known as Open Directory Project (ODP). This large dataset contains 1562808 URLs (1.5 million) with 15 categories. This dataset has become a widely used dataset because almost maximum researchers do their experiment of URL classification using DMOZ dataset. Before the creation of DMOZ dataset, researchers used to do the experiment by a small dataset which is known as WebKB [8]. WebKB contains 8282 pages with 7 categories.

Our contributions are word based multiple n-gram models rather than character based single n-gram model [4], best feature extraction methods and parameters finding for URL features using hyperparameter optimization and the best classifier model for URL classification. The details of contributions are the following: (i) we have introduced the Random Search technique that finds the best feature extraction method and the parameters of both features and classifier. (ii) Maximum computation power is provided for Random Search. (iii) We have proposed a word based multiple n-gram splitting models to look for n-gram relationship at multiple scales. (iv) We have selected multinomial distribution of Naïve Bayes classifier instead of Gaussian distribution. (v) We have evaluated our experimental result by the latest evaluation metric and compare our result with previous research papers

978-1-5386-9242-4/18/$31.00 ©2018 IEEE

result and finally (vi) we are able to show a better result than previous research works.

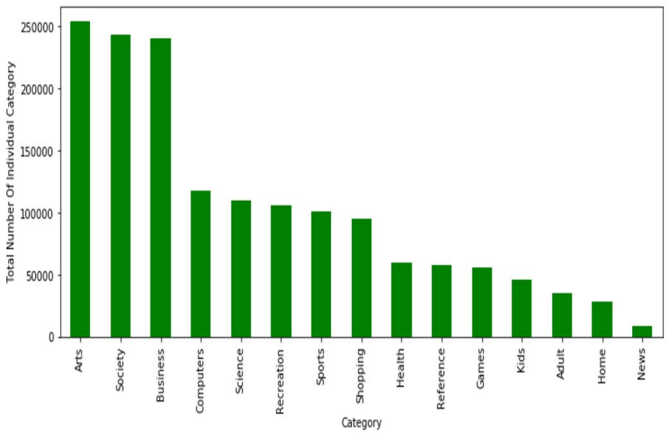
The paper is organized as follows: The related works are presented in Section II. The dataset and proposed method are described in III, IV. Feature extractions and classifier models are described in section V, VI. Section VII, VIII presents parameter tuning and experimental result analysis and finally, conclusion and future works are mentioned in Section IX.

II. Related Work

There are several research works on web page classification. The early research works focused on mainly link structure based web page classification and feature extraction technique [1, 2]. Four kinds of neighbors (parents, children, siblings and spouses) based categorization was introduced by Xi. Qi, and B. D. Davison [1]. After 3 years they classified web pages based on HTML content and generated summaries of the pages [2]. Another web content based research proposed feature extraction of web pages by latent semantic analysis and showed the result based on title, meta-data, content body, description of web pages by achieving highest 82% accuracy for description based classification [3]. But content-based classification will not work when the contents are hidden. So Beykan et al. [4] proposed tokens as features and n-gram as features for the URL classification in 2009. They showed 82.44% accuracy in F1-score for 15 categories by SVM classifier on DMOZ dataset. Segmentation of URLs by information content reduction and finite state transducer were introduced by Kan [6] by achieving 43.2 % F1-score on WebKB dataset. In 2005, Kan presented 50% accuracy on DMOZ dataset containing 27,252 URLs by Maximum Entropy (ME) classifier [9]. In 2013, n-gram character-based web page classification on DMOZ dataset for 14 categories was proposed and this research paper [10] showed 78.54% accuracy in F1-score by SVM classifier. After 4 years, in 2017, the same authors experimented on WebKB dataset and got 79% F1-score [11]. Beykan et al. elaborated on their previous research by experimenting with other 4 different datasets and compared human predicted accuracy with machine predicted accuracy [12]. In 2015, T. Abdullah et al. [13] proposed an n-gram character-based language model for feature extraction and showed 82.72% accuracy in F1-score by Naive Bayes classifier on DMOZ dataset for 15 categories. Recently in 2017, a research work [14] was published in IEEE intelligent system that showed online URL classification for large-scale streaming environments.

III. Dataset

In Section I, we briefly described the dataset of URL classification. In this Section, our experimental dataset description and visualization are shown. We have selected the DMOZ dataset as our experimental dataset because it is the largest, contains 1562808 (1.5 million) URLs and 15 categories. Besides from section II, it is cleared that most researchers have used DMOZ. Our experiment analysis is based on classifying the 15 categories of this dataset. These categories are Adults, Arts, Business, Computers, Games, Health, Home, News, Recreation, Reference, Science, Society, Shopping, Sports, and Regional. All of these URLs are based on the English language. Fig. 1 shows the dataset visualization.



.Fig. 1. DMOZ dataset visualization

IV. Proposed Method

Our proposed method defines the improvement of previous research method [4, 13]. The improvement of both feature extraction and classifier is considered in our approach. Fig. 2 shows our proposed method for URL classification.

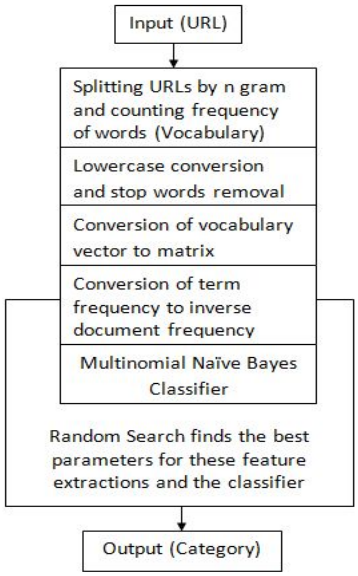


Fig. 2. The proposed method of URL classification.

We have described our proposed feature extraction method, a classifier model and the hyperparameter optimization method in Section V, VI and VII.

V. Feature Extraction

Classifier model cannot understand the strings or words directly [4]. So we have to convert the strings into feature vectors and matrix. Feature vectors are compatible with any kind of classifier. That's why feature extraction is very important.

A. Vocabulary By Word Based Multiple N-gram Models

The multiple n-gram models are inspired by the future work of the recent research [13]. The URL contains different kinds of punctuations (dot, slash, double slash, and colon). As punctuations are not meaningful for important features, we have removed all kinds of punctuations. Then some words from the URLs are extracted by n-gram. But our approach is not like the previous approaches [4, 10, 13]. Our proposed n-gram is word-based multiple models with different values of n. The different values of n are called a range and the range is (1, 2). The reason for choosing (1, 2) is described in Section VII. The range helps to split the URLs by both unigram and bigrams. Only a single word is not enough for understanding the category of a URL. Sometimes two words that together give us the more correct features than one word or represent another meaning. So both unigrams and bigrams are applied. The final step of vocabulary building is counting the number of occurrences of those extracted words. Table I shows the vocabulary of a URL (http://computer.org/) by n-gram range.

TABLE I. VOCABULARY BY N-GRAM RANGE

N-gram range (1,2)	Occurrence
computer	1
computer org	1
http	1
http computer	1
org	1

The second bigram (computer org) of Table I is more meaningful than unigram (computer) because the URL belongs to a computer organization.

B. Removal of Stopwords

After vocabulary building, we found some common words for all URLs that are not meaningful (http, https, ftp, www etc.). These are called "stop words". So, all kinds of stop words are removed from our feature words. All characters are also converted to lowercase.

C. Term-Frequency (TF) and Vector Space Model (VSM)

Term-frequency (TF) is an important approach for extracting the weights of the important words from text [16]. Term frequency is represented as a Vector Space Model (VSM) that enables computers to understand the words [17]. Term frequency measures the occurrence of vocabulary in the whole document. If the vocabulary term is t, the whole document is d, the frequency is f and count of in every data is c then term frequency is represented as Eq. 1.

$$tf(t,d) = \sum_{c \in d} f(c,t) \qquad (1)$$

The term-frequency is converted to a document vector by the following Eq.2.

$$v_{d_n} = (tf(t_1,d_1), tf(t_2,d_2) \ldots\ldots tf(t_n,d_n)) \qquad (2)$$

t_n is i th term of vocabulary t. For example, if d_1 is http://computer.org/ and d_2 is https://king.com/games/ then according to Eq.2 the document vectors are shown for both d_1 and d_2 in Table 2. The sizes of the vectors are always the same because these vectors are converted to a matrix.

TABLE II. DOCUMENT VECTORS

t / d	computer	computer org	king	games	org
d_1	1	1	0	0	1
d_2	0	0	1	1	0

The document vectors are converted into document sparse matrix [17]. The sparse matrix that has the maximum element with zeroes is very memory efficient during calculation. So the converted sparse matrix is presented in Eq. 3

$$M_d = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad (3)$$

D. Term-Frequency Inverse Document Frequency(TF-IDF)

TF-IDF is a logarithmic scaling process that reflects the important words of the whole document [18]. The most frequent items are not always the most important feature. For an example, there are lots of links from different categories of Wikipedia domain in our dataset that start with en.wikipedia.com/wiki/. So the occurrence of Wikipedia and wiki are more than other words. But Wikipedia and wiki words need to be downscale as they are not important for features. That's why TF-IDF downscale less important words and upscale the important words. If the whole document is D, an individual document is d, corpus size is |D|, a word is t then Eq.4 defines IDF.

$$idf(t,d) = log \frac{|D|}{f(t,d)} \qquad (4)$$

Here $f(t,d)$ is the number of times t appears in d [18]. So the final TF-IDF is obtained by Eq.5.

$$tfidf(t,f,d) = tf(t,d) \times idf(t,d) \qquad (5)$$

The sparse matrix from Eq. 3 will be converted to Eq. 6. The matrix is defined by M_i . The similarities among documents are measured by cosine similarity which is described in the previous research [17, 4.4].

$$M_i = \begin{bmatrix} 0.577 & 0.577 & 0 & 0 & 0.577 \\ 0 & 0 & 0.577 & 0.577 & 0 \end{bmatrix} \qquad (6)$$

So we have shown the feature extraction of the URL classification step by step. We have changed the previous character-based n-gram approach [10, 13] to the word-based n-gram approach.

VI. MULTINOMIAL NAÏVE BAYES

After the feature extraction, the extracted features are needed to be classified by a classifier that gives the best performance for the text classification. According to previous research [19], we have found that the Naïve Bayes classifier performs better than other classifiers for text classification. So we need not find by our self. There are also some varieties of Naïve Bayes classifier such as Gaussian distribution, multinomial distribution etc. The performance comparison among them was also done by a previous research [20] and it showed that the performance of multinomial distribution is higher than the Gaussian distribution. Multinomial distribution captures word frequencies in documents and cares about multiple features that are frequently occurred [20]. But Gaussian distribution refers to conditionally independence of each feature. If vocabulary is V, each document is d_i, features are θ, words are w_t and word occurrence is N_{it} then Eq. 7 defines the multinomial distribution of the Naive Bayes classifier.

$$P(d_i|\theta) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|\theta)^{N_{it}}}{N_{it}!} \quad (7)$$

Finally, the probability of a category c_j is derived from Eq. 8. Eq.7 and Eq. 8 are described elaborately in the previous research [20].

$$P(c_j|d_i) = \frac{P(d_i|c_j) \times P(c_j)}{P(d_i)} \quad (8)$$

There is also some other distribution such as Bernoulli which is also compared with multinomial distribution finding the best performance of multinomial from the same research.

VII. RANDOM SEARCH

The feature extraction algorithms and the classifier contains a large number of parameters with different values. These parameters may depend on the corpus and maximum time these parameters are taken from previous research or manually with few parameters which may lead to less optimal accuracy [21]. So the technique that finds automatically best parameters from all parameters is called hyperparameter optimization or parameter tuning and the algorithm that we have used for parameter tuning is Random Search [22]. The finding of the best parameters is quite difficult because it takes a huge time and computational power. So previous research works mentioned in Section II did not apply parameter tuning for URL classification. There are also another algorithm which is known as Grid Search but Grid Search needs more time computational power than Random Search. Random search finds the parameters randomly with a small fraction of computational time, finds better models by effectively searching a larger, less-promising configuration space [22]. It is performed by evaluating n uniformly random points in the hyperparameter space and select the one producing the best parameters [23]. The parameters that we have found by Random Search are n-gram range, TF-IDF and additive smoothing parameter [24] which assigns the non-zero probability to the words that are not present in the sample. In Naïve Bayes, the smoothing parameter is also called alpha. The default value of alpha is 1 and 0 means no smoothing. After applying Random Search, it takes approximately 6 iterations to find the best parameters. Table III shows the all parameters list for 6 iterations and Table IV shows the best parameters.

TABLE III. ALL PARAMETERS LIST

n-gram range	TF-IDF use	alpha
(1,1)	True	0.01
(1,2)	True	0.01
(1,1)	False	0.01
(1,2)	False	0.01
(1,1)	True	0.001
(1,2)	True	0.001

TABLE IV. THE BEST PARAMETERS BY RANDOM SEARCH

n-gram range	TF-IDF use	alpha
(1,2)	True	0.001

According to Table IV, we have found the best parameters by Random Search. Our feature extraction and classifiers can show their best performance by using these parameters.

VIII. EXPERIMENTS AND RESULTS ANALYSIS

A. Experimental Environment

The experiment of our research has been processed by quite highly configured computer with Tesla K80 GPU and 12GB of RAM. Our URL classification system has been implemented in Python 3.

B. Training And Testing Dataset

There are 1562808 URLs of 15 different categories in DMOZ (ODP) dataset. Among 1562808 URLs, 1532808 URLs have been taken as a training dataset and 30000 URLs have been taken as the testing dataset. In the testing dataset, each category contains 2000 URLs and thus 15 categories contain 30000 URLs. So, the testing dataset is a balanced dataset. The training and testing splitting of the dataset is followed by previous research works [4, 13]. In previous research, there are 1000 URLs for each category for balanced DMOZ (ODP) testing dataset. But we have increased the testing dataset to check our approach more accurately. Fig. 3 shows our testing dataset for the experiment.

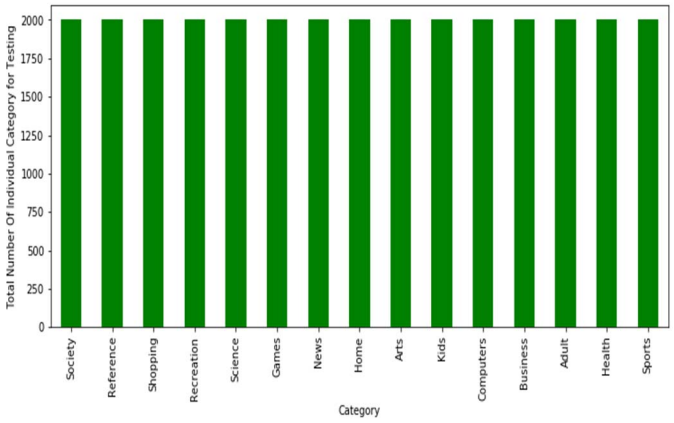


Fig. 3. Testing dataset

C. Results

We have evaluated our result by calculating the precision, recall, and F1-score of our testing dataset. We have also shown the precision, recall and F1-score of every individual category so that we can analyze in which categories our method is showing better or poor performance. Table V shows the performance of our word based multiple n-gram model and Random Search oriented feature parameters experimental result.

TABLE V. OUR EXPERIMENTAL RESULT

Category	Precision	Recall	F1-score
Adult	99.22%	19.10%	32.03%
Arts	55.29%	91.50%	68.93%
Business	75.92%	99.80%	86.24%
Computers	94.02%	96.70%	95.34%
Games	96.91%	95.65%	96.28%
Health	98.94%	97.80%	98.37%
Home	97.18%	92.95%	95.02%
Kids	93.72%	71.60%	81.18%
News	99.41%	84.35%	91.26%
Recreation	95.00%	98.85%	96.89%
Reference	86.78%	94.55%	90.50%
Science	94.36%	95.30%	94.83%
Shopping	97.63%	99.00%	98.31%
Society	87.28%	99.80%	93.12%
Sports	97.83%	94.65%	96.21%
Average/Total	**91.30%**	**88.77%**	**87.63%**

From Table V, we have analyzed that "Adult" category shows poor performance in recall and F1-score. So, URL classification method needs more improvement on the Adult category to classify correctly. We have also presented a confusion matrix for better understanding. The values of the confusion matrix are rounded up without percentage. Fig. 4 shows the confusion matrix.

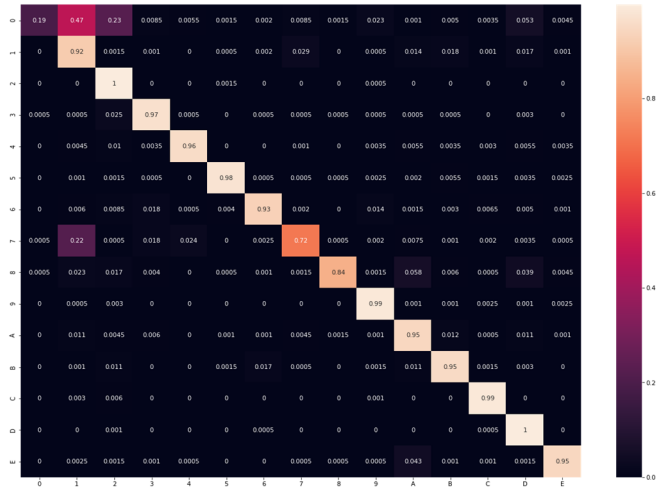


Fig. 4. Confusion Matrix

D. Results Comparison with Previous Research

The recent research [13] of URL classification showed their result in F1-score for character-based single n-gram feature extraction model with Naïve Bayes (NB) classifier and the early research showed character based all gram model with SVM classifier [4]. Our feature extraction model which was mentioned as a future work of the last research [13] is a word-based multiple n-gram models and classification model is Multinomial Naïve Bayes (MNB) classifier. Then the parameters of our model are optimized by Random Search. But previous research works did not optimize the parameters. So, these are the reasons for our better performance than previous research works. Table VI shows the F1-score comparison with previous research.

TABLE VI. F1-SCORE COMPARISON WITH PREVIOUS RESEARCH

Category	SVM + all gram [4]	n-gram LM +NB [13]	Multiple n-grams + Random Search +MNB
Adult	87.60%	87.58%	32.03%
Arts	81.90%	82.03%	68.93%
Business	82.90%	82.71%	**86.24%**
Computers	82.50%	82.79%	**95.34%**
Games	86.70%	86.43%	**96.28%**
Health	82.40%	82.49%	**98.37%**
Home	81.00%	81.13%	**95.02%**
Kids	80.00%	81.09%	**81.18%**
News	80.10%	79.01%	**91.26%**
Recreation	79.70%	80.22%	**96.89%**
Reference	84.40%	83.37%	**90.50%**
Science	80.10%	82.52%	**94.83%**
Shopping	83.10%	82.48%	**98.31%**
Society	80.20%	81.66%	**93.12%**
Sports	84.00%	85.30%	**96.21%**
Average/Total	82.44%	82.72%	**87.63%**

According to Table VI, we have analyzed that the improvement of our approach is 4.91% of the latest research and it is clear that our proposed method performs better than the existing method. Thus our research has successfully contributed in the area of website or web page classification.

IX. Conclusion

In this paper, we proposed a better approach than the existing approach for website classification from URL. We worked on future work of previous research and proposed word based multiple n-gram models. We introduced the hyperparameter optimization for the URL classification which was not applied in the previous research. The precision, recall, and F1-score of our research are 91.30%, 88.77%, and 87.63% respectively and our result is the highest so far. As future work, we believe that a balanced training dataset and higher value of n-gram can improve our result. We propose to improve our research in this direction.

References

[1] X. Qi and B. D. Davison, "Knowing a web page by the company it keeps," In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 228-237. ACM, 2006.

[2] Xi. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM computing surveys (CSUR)* 41, no. 2 (2009): 12.

[3] D. Shen, Z. Chen, Q. Yang, H. Zeng, B. Zhang, Y. Lu, and W. Ma, "Web-page classification through summarization," In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 242-249. ACM, 2004.

[4] E. Baykan, M. Henzinger, L. Marian, and I. Weber, "Purely URL-based topic classification," In *Proceedings of the 18th international conference on World wide web*, pp. 1109-1110. ACM, 2009.

[5] M. I. Devi, R. Rajaram, and K. Selvakuberan, "Machine learning techniques for automated web page classification using URL features," In *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, vol. 2, pp. 116-120. IEEE, 2007.

[6] M.-Y. Kan, "Web page classification without the web page," In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 262-263. ACM, 2004.

[7] DMOZ, Open Web Directory Project. [Online]. Available: https://dmoztools.net/. [Accessed: 01-August-2018].

[8] WebKB, The 4 University Dataset. [Online]. Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/. [Accessed: 01-August-2018].

[9] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using URL features," In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 325-326. ACM, 2005.

[10] R. Rajalakshmi and C. Aravindan, "Web page classification using n-gram based URL features," In *Advanced Computing (ICoAC), 2013 Fifth International Conference on*, pp. 15-21. IEEE, 2013.

[11] R. Rajalakshmi and S. Xaviar, "Experimental study of feature weighting techniques for URL based webpage classification," *Procedia Computer Science* 115 (2017): 218-225.

[12] E. Baykan, M. Henzinger, L. Marian, and I. Weber, "A comprehensive study of features and algorithms for URL-based topic classification," *ACM Transactions on the Web (TWEB)* 5, no. 3 (2011)

[13] T. Abdallah and B. Iglesia, "URL-based web page classification: With n-gram language models," In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pp. 19-33. Springer, Cham, 2014.

[14] N. Singh, N. S. Chaudhari, and N. Singh, "Online URL Classification for Large-Scale Streaming Environments," in *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 31-36, Mar.-Apr. 2017.

[15] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," *Ann arbor mi* 48113, no. 2 (1994): 161-175.

[16] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management* 24, no. 5 (1988): 513-523.

[17] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research* 37 (2010): 141-188.

[18] J. Ramos, "Using tf-idf to determine word relevance in document queries," In *Proceedings of the first instructional conference on machine learning*, vol. 242, pp. 133-142. 2003.

[19] S. L. Ting, W. H. Ip, and A.Tsang, "Is Naive Bayes a good classifier for document classification?," *International Journal of Software Engineering and Its Applications* 5, no. 3 (2011): 37-46.

[20] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," In *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1, pp. 41-48. 1998.

[21] C. H. A. Koster and J.G. Beney, "On the importance of parameter tuning in text categorization," In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics*, pp. 270-283. Springer, Berlin, Heidelberg, 2006.

[22] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research* 13, no. Feb (2012): 281-305.

[23] A. Johnson, Common Problems in hyparparameter optimization. [Online]. Available: https://blog.sigopt.com/posts/common-problems-in-hyperparameter-optimization. [Accessed: 01-August-2018].

[24] D. Valcarce, J. Parapar, and Á. Barreiro, "Additive smoothing for relevance-based language modelling of recommender systems," In *Proceedings of the 4th Spanish Conference on Information Retrieval*, p. 9. ACM, 2016.

View publication stats