International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Block-Based Feature Selection Method for Classification of Web Pages

Azween Abdullah[1], Sandeep Kumar M[2], Prabhu J[3], Balamurugan Balusamy[4]

Submitted: 08/05/2023 **Revised**: 17/07/2023 **Accepted**: 06/08/2023

Abstract: Webpage Classification is one of the methods for retrieving useful information that can be used for many purposes like searching, organizing, and spam filtering, and so on. Most of the existing web page classification algorithms focus on extracting the entire data however recent works focus on selective retrieval that could improve the efficiency of the classification. In this paper, we propose a block-wise feature selection algorithm that can segment a web page into blocks and finally filter out all non-important blocks. We introduce three features namely 1) keyword weighting, 2) block segmentation and, 3) similarity measures for improving the efficiency of the classification process. We select blocks that are very crucial in the classification process. Since the useless blocks are removed, the feature space is reduced and the accuracy is increased. The semantic words are also eliminated and the subset of most relevant features is choosing for building the classification model. The results shows an improved classification results as the relationship between the features and the target variable is understood easily. To demonstrate the efficiency of the proposed model, we compared it with other top machine learning classifiers. Two datasets are used in our experiment. The experimental results showed that our proposed work with four machine learning models and obtained up to 95% accuracy which is 11.7% more than existing models

Keywords: *Web classification, Feature extraction, Blocks Segmentation, Spam Filtering, Semantics Word, Classifier.*

1. Introduction

Web page classification is one of the recent popular methods particularly used for searching, spam detection, topic categorization, and many other applications. Much private organization uses web page classification for analyzing the web page contents for blocking malicious contents. Web page classification is a supervised classification method in which a web page is assigned one or more classes by a machine learning model. Traditional web page classification was done using URL (Uniform Resource Locator)-based filtering. A standard database was created and updated regularly. String-based methods such as regular expression were used to extract and find the pattern between the data present in the database and the web page URL. Many problems exist in this approach such as regular updation, difficult to map the patterns, and so on. Moreover, the contents in the web page are different from the URL pattern most of the time. In this case, the false positive and false negative may be very high. Hence, an alternative approach should be implemented in the web page classification domain. Machine learning models are one of the best alternatives for URL based classification.

Web page classification is done using four stages as shown in fig 1. Preprocessing stage is the first and the most important stage in the classification [1]. This stage is responsible for removing misleading and unwanted information from the corpus by performing operations such as stemming, stop word removal, punctuation removal and so on.

[1]*Faculty of Applied Computing, Perdana University, Kuala Lumpur, Malaysia .azween@perdanauniversity.edu.my*
[2]*School of Computing Science & Engineering, Galgotias University, Uttar Pradesh 203201, India, sandeepkumarm322@gmail.com*
[3]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India. j.prabhu@vit.ac.in*
[4]*Associate Dean-Student Engagement, Delhi-National Capital Region (NCR), Shiv Nadar University, India. kadavulai@gmail.com*

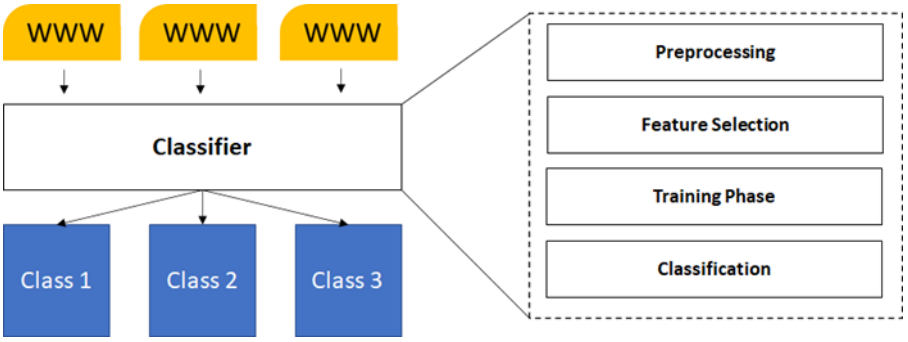

Fig 1: Stages of Web Classification

The web page classification is done in the following manner. The number of end classes is fixed then uniqueness for each class is determined. Uniqueness is a set of features that can determine a class without the help of any other features, for example, in training set, if there are 5 features and suppose if two features can identify a class A, then the two features are unique to A. The uniqueness states the features that help the classifiers to identify a class for a web page. The second step is to seperate the input documents into two sections, first the training set and then the testing set. Training set is used by the classifier to understand the uniqueness for each document. The validation of the learning is calculated by the testing set. Once the validation results are satisfying, the classifier is ready for the deployment. The classifier then analyses an input document and finds the probability for all the classes. The end class is the one which is having the highest probability. There are lots of research works using many powerful machine learning models such as Support Vector Machines (SVM) [4], Naive Bayes (NB) [5], k Nearest Neighbour (kNN) [6], Logistic Regression (LR) [7] and so on. Machine learning models naturally have the ability to find the features of a class, but in many cases, the features are not uniformly distributed across the corpus or there are lots of duplicate or error full features. Hence a good feature selection algorithm is used in the Preprocessing stage to find and remove the irrelevant features from the input document. This step will significantly increase the performance of the classification. There are three types of feature selection algorithm called as filter, wrapper and embedded [2]. Filter based feature selection algorithm picks features based on occurrences and frequencies of terms in a web page. This type of algorithm produces same set of features for all the classifiers. Pearson correlation and Analysis of Variance are some examples of filter based feature selection. The next type of feature selection is wrapper based. These are combined with a classifier to pick the best sub set of features. Wrapper based algorithms start with zero set and increments iteratively to find the optimal sub sets that maximizes the accuracy of the classifier. Wrapper based feature selection methods are classifier dependent, that means unlike filter based methods, these algorithms produces different features for each classifier. Hybrid based feature selection is combination of the other two types. It inherits both the advantages and disadvantages of both methods.

Almost all of the existing works fully depend on term frequencies [3]. However, term frequencies give good performance at many text document, but for web pages depending on term frequencies alone is not a good idea, because the contents in the heading tag is more important in the contents in the paragraph tag. Hence the position will super seed the term frequencies in case of web document. This study considers the position level of each term to represent the document more accurately.

In this paper, we propose three stages of feature extraction in the web page which adds more details about the topic which could help the classifier to understand the semantics of the webpage and assigns the correct class to the web page. The three stages are as follows. First, we run a minor classifier that could pick up the positive and negative features for a given class. Then we assign different ranks to both the positive and negative features. This allows a clear margin between the overlapping features. The second stage is identifying and removing the common blocks such as headers, footers, ads. Finally, the semantic is captured using the NLTK package to find and remove the semantic words.

The main contribution of the paper is as follows

1. To separate the positive and negative overlapping features to create a good margin to improve the classification performance.

2. To find and eliminate irrelevant parts of the web page by identifying the common blocks.

3. Remove the semantically redundant features from the feature space.

The rest of the paper is as follows. The section 2 briefs few related works based on web classification. Section 3 presents the working of the proposed model. Section 4 compares the proposed model with other machine learning models. The discussion is present in section 5.

Finally section 6 contains the conclusion and future work.

2. Related Work

A research work proposed by Ashokkumar et al [7] uses the web page classification for various services such as enterprises, industry, and government. Their proposed method extracts knowledge from user interest and market trends. Accuracy has been improved by introducing a new validation method from cross verification from social media. The deep learning-based RNN model was used in [8] in an information retrieval application for recommending appropriate web pages for users. The Meta tags such as title, keywords are given high importance in their work. For improving the accuracy, transfer learning is also adopted. Their proposed model had obtained an accuracy of 85%.Web page classification can also be improved by using swarm optimization such as a research work [9]. The web page is scanned to identify all the tags. The HTML tags are then ranked by assigning weights. The results show that identifying proper tags and ranking them produces better accuracy. However, common tags such as the p tag are very common in a web page and still more optimization is required to rank those common tags. Web page classification often requires crawling the entire content and if the web page is very long or deeply nested, the crawling process seems to be difficult. Research work by [10] aims to solve this problem by introducing URL matching which eliminates the need of crawling all the web pages on a single website. URL matching has lots of inaccuracies as described in [11] so hence URL matching should not be the only criteria for web classification.URL can hold more information that is sufficient to classify a web page most of the time. For example the URLs such as tutorial point, easy tuts can have web pages related to education. A supervised classification model was developed by [12] uses tokenization for extracting tokens from the URL. Feature selection techniques such as length, non-alphabetical symbols are used as maximization algorithms [6].

Besides URL, semantics is also one of the aspects to determine the class of a web page. [13] Proposes a semantic method of classifying web documents using both length and breadth of a web page. Deep nestings of the web page can be detected using this approach. They have used the n-gram model to classify massive web page contents. Another research work [14] focuses on terms and tags in a web page for classification. They had used a genetic algorithm that considers tags and terms as separate entities which increases the end classification accuracy. Up to 95% accuracy was obtained by separating both tags and terms. A fingerprint of a web page [27] is used to mine marketing and customer analysis. They have used web logs to extract the needed information from both desktop and mobile held devices. The maximum of 92.2 F1 score was obtained by their proposed method. Time based information can be added to increase their research performance. Time based information is used in another research [28] for spam detection in web pages. Fuzzy based information is also added to increase the accuracy. Recent advances in spam detection using web page classification uses deep learning approach such as [29] where they use incremental algorithm to rank web page, their research performance outperform exiting works by 7%. Although there are many deep learning research works recently increasing, still there are many improvements needed as proposed by [30] which includes considering data augmentation with permutations and showed 41% increase in accuracy.

Feature extraction using ensemble methods [32] or by using multiple layers [33] can also increase the performance of the classifiers in various fields such as Sentimental analysis [33], review classification. The process of selecting the appropriate features can be done using sub set analysis where iteratively each feature is either inserted or removed from the selection set, then the best combination is found and used for the classification process. The feature selection is incorporated in a teaching evaluation where both machine learning and deep learning models were used. 15,000 instances are inputted into these machine learning models to train and obtain the knowledge in the dataset. The authors uses ensemble-based model to combine multiple feature selection methods such as term-presence, term-frequency, and term-inverse document frequency schemes. Four ensemble models are compared and found that the ensemble features help to increase the performance.

Ensembling the features in web page is challenging when it comes to heterogenous blogs and contents. A proper relationship should be found between each feature selection models is necessary to prevent high false rate, especially when all the feature selection models are running over different set of pages. Few other works such as [15] [16] uses various weighting schemes for classification which ranks each tag or term in web pages for feature selection [17,5]. Other meta information such as multimedia can also be used as feature selection method for web page classification. [31] Shows a novel method to include multimedia contents for web page classification. Some notable works uses unique concepts for selecting features such as under sampling, word embedding, term weighting, topic modeling, association

mining and hybrid clustering. Table 1 presents a few more research works in the field of web classification.

Most of the exiting works focus on finding out features in the web document for increasing the classification model. However, exploiting the full structure of a web page is not focused in the recent literature because many researches works aims to develop a genetic classification model which can be used for all text inputs. Moreover, web pages contain some unique properties such as block wise content segregation, links, and semantic wise features and so on. In this paper, we consider developing a classifier which is dedicated for web pages alone thus fully exploiting the contents of a web page.

3. Web Classification Model

Our proposed classification model works in three stages known as keyword weighting, block-level segmentation, and similarity measure. Fig 2 shows the flow of each stage. These stages are explained in the below subsections.

Table 1: Comparision of few recent works

Reference	Architecture	Discussion	Comments
[18]	Vector Model Representation	The DOM tree is converted to a vector based model to find and eliminate the duplicate contexts.	Sentence level comparison can improve the accuracy of the classification
[19]	Term Weighting	Sensitive or important words are time de- pendent. The proposed model aims to find the sensitive features automatically.	Web pages with shorter text elements are difficult to find the sensitive features.
[20]	LSTM Model	The actual text present in the document is combined with the functional text to produce more feature space.	Higher crawl rate is required
[21]	Fusion	The proposed method uses meta data along with the textual contents for improvising the performance of classification	Adding proper weights to meta data can improve the performance of the classification.
[22]	Feature extraction	The correct and relevant image features are extracted by the proposed method. This process helps to remove the noise from the training phase.	Some images are displayed only when the user clicks on some elements; hence the complete feature scanning may improve the accuracy.

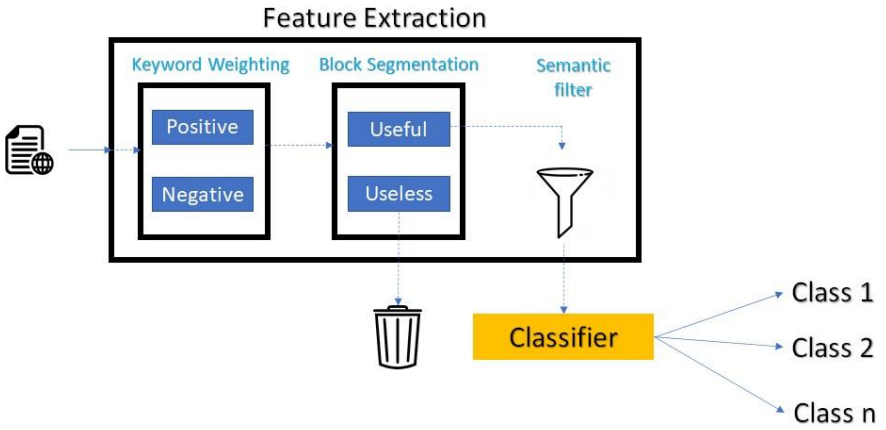

Fig 2: Architecture of the proposed method

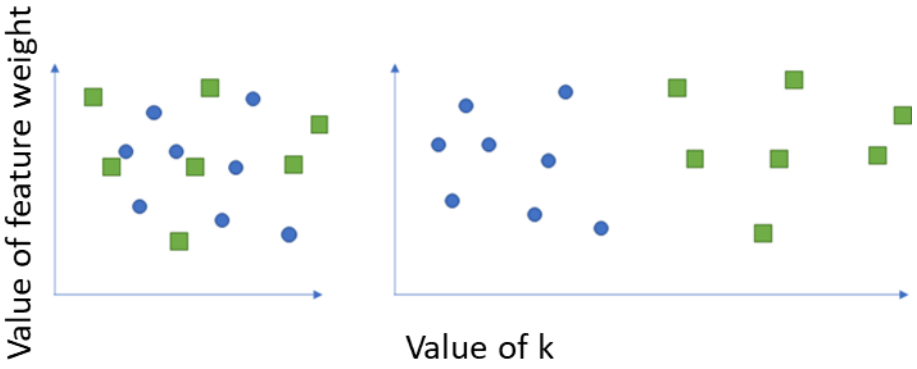

Fig 3: Domain Mapping

3.1 Keyword Weighting

The most important problem faced by all the classification models is to place a margin to separate positive and negative classes. However, in most cases, the placement of margin at the correct position is difficult and that creates Lots of false positives and false negatives. A good performance can be obtained when the position of margin is optimal. Moreover, due to the scatter of data, there will not be any optimal position of the margin. In this case, we develop a keyword weighting that maps all the documents from the original domain to a newly modified domain. This domain will have a good separation of data points. The Fig 3 shows how the data points are separated in the new domain. The X axis in the Fig 3 represents the value of k (eq 4) and the Y axis represents the value of feature weight (eq 3). Let us assume that the binary classification model for the experiment. This is not a constraint as the proposed method can be also scalled to the multi class classification. Two disjoint sets are constructed using Pearson correlation feature selection (as per 1) algorithm in the training set.

$$r = \frac{\sum(X_i - \overline{X_i})(Y_i - \overline{Y_i})}{\sqrt{(X_i - \overline{X_i})^2 (Y_i - \overline{Y_i})^2}} \quad (1)$$

X_i – Represents the probability of the current document containing highly rich features of class i

Y_i – Represents the probability of all the highly rich features in the class i

These two sets represent highly rich features that can identify a class. For example, in science web page detection, the positive set contains features like mass, weight, novel, design, and so on. Similarly, the negative set contains features like swimming, chatting, cook, music and so on.

After generating the feature sets, each feature are then calculated its weight as shown in equation 2.

$$W_f = \frac{frequency\ of\ f}{n} \quad (2)$$

W_f – Represents the weight of the current feature f.

f– Represents the frequency of f in the current document.

n – Represents the total number of features in the current document.

The frequency of all the features present in the web document is then used to plot the document in the new domain. The formula for mapping the web document in the new domain is 3.

$$Position = \frac{\sum_{\forall} f \in feature\ set W_f}{|feature\ set|} \quad (3)$$

To create a boundary, a constant value k is added for each class. In our experiment, we considered the value of k as 10 (in positive class) and 20 (in negative class). This allows the documents of different classes to get separated and a margin can be drawn at the optimal position. The constant value is added at the frequency as shown in equation 4.

$$W_f = \frac{frequency\ of\ f}{n} + k \quad (4)$$

k – The constant value to shift the features across the feature space

3.2 Block Level Segmentation

The location of features is very important as it affects the frequency a lot. For example, the features at the header or footer of a website are present in all the documents and thus it has a high frequency and a low meaning. These words can be removed from the feature list to increase the classification performance. For this purpose, we identify blocks in the webpage and eliminate all the common blocks. Fig 4 shows the components of a web page, in which the blocks at red

color are irrelevant for classification and can be removed. The block in blue color contains more valid information than other blocks. To identify and detect the blocks, we use regex to match nav, ad units, header and footer, and so on. Sometimes a webpage may contain a header and footer with the standard header and footer tag. In this case, it is difficult to find the header and footer content and remove them. In this subsection, we identify repeatable blocks across all web pages and remove them.

Each block is extracted by using the DOM (Document Object Model) tree and a hash value is calculated based on the contents. This hash value is then inputted into the dictionary and finally, if the hash value is present across 80% of the documents, they are removed. Algorithm 1 shows the working of the block level selection.

3.3 Semantic Measure

Two unique words are said to be semantically the same if the meaning of both the words is the same. The weight of two semantic words should be the same.

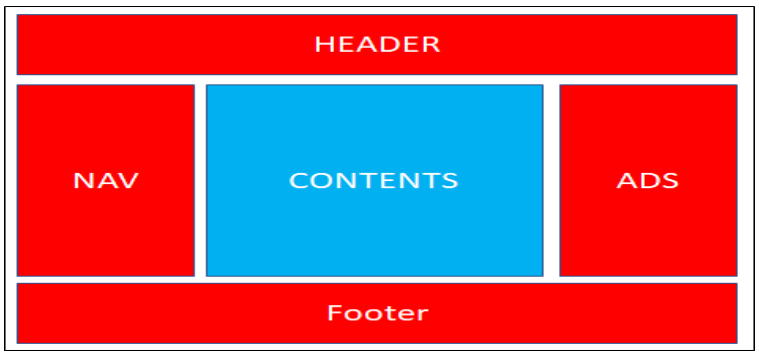

Fig 4: Components of a web page

Algorithm 1: Block level Segmentation

```
procedure BLOCKFILTERING
        blockdict ← Empty dictionary
        begin:
for each web page d in the corpus do
        DOM ←  DOM Tree of the webpage
 for each child element e in DOM do
        c ← Content of e
        h←  Hash of c
if h is present in blockdict then
        blockdict[h]++
    else
    blockdict[h]=1
let common block=empty
TH=80% of total web pages in the corpus
  for each key k in blockdict do
      if  blockdict[k]<TH then
          add k to common block
       Remove all common block
```

We make use of NLP techniques to extract the POS (Parts of Speech) of each sentence in the webpage. We have used an NLP-based textual similarity [24] to find and extract the meanings of each sentence. This will be helpful to identify and eliminate the semantics in the web page.

The efficiency of the proposed method is dependent on how the high features are extracted from the web page. This is done by identifying the key parts in the web page and then extracting unique features that can help in classifying the document. The extracted features are filtered based on semantic similarities

4. Experiment Results

To evaluate the performance of the proposed method, we took two standard datasets and took 25K webpages from them for the classification. Clue web 12 dataset contain

web pages extracted between February 10, 2012 and May 10, 2012. To generate heterogeneity, various English countries are selected to generate the dataset such as United States, United Kingdom, Canada, Australia, Ireland, and New Zealand. DBPedia is constructed from Wikipedia website, the dataset constantly tracks the changes from the Wikipedia website and updates regularly. The details of the datasets are shown in table 2. We have implemented the three stages in the Support Vector Machine classifier and compared them with other standard machine learning algorithms such as Logistic Regression, K Nearest Neighbor, SVM, and Naive Bayes. The validation process is done using 10 fold validations. The number of classes that we have used in the experiment are 2, 3, 4 and 5. Python 3.6 and sklearn library for the implementation purposes. We run all data on an Intel Core i5-11400H computer with 16GB RAM. All the algorithms are coded on a Windows 10 operating system.

Table 2: Dataset Descriptions

Name	**Number of Documents**
Clue Web 12 [24]	25k
DBpedia [25]	25k

We have used two standard datasets, the first one is ClueWeb 12 which was created to improve the research works done in the field of information retreieval, text classification and various NLP based research tasks. This dataset contains about 733, 019, 372 documents from English language. The duration of the web pages that are contained in the dataset is between Feb 10, 2012 and May 10, 2012. This dataset is the next version of the previois datasets such as clue web 09. The second dataset used is DBPedia. This dataset contains around 6 million documents out of which 5.2 million documents are already classified into various classes such as persons, places, albums, films, video games, organization, species, diseases and so on. This dataset is a mini version of the English edition of the Wikipedia.

The performance of the proposed method is evaluated using four parameters accuracy, precision, recall and F1 score which uses four performance metrics True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The equations for the parameters are shown in equation 5, 6, 7 and 8 respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FP} \quad (7)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

Since each positive and negative term was shifted to a new domain based on the weights, a clear margin has been placed which reduces the chance for overlapping. This is one of the reasons for maximum performance. Fig 5 and 6 show the results of the four parameters accuracy, recall, precision, and F1 score for the datasets ClueWeb12 and DBpedia respectively.The Figs 5 and 6 show the results when the number of classes is set to 6. In our experiment, the classification is done from binary to 6 classes. The proposed method showed superior in all the cases while compared with other machine learning models. The Fig 7 and 8 show the accuracy results of the models with respect to the number of classes used for classification. The proposed model shows superiority while compared with other machine learning models. The blocks and the semantics were identified correctly and the higher important terms are given more priority than lower important terms and thus the classer is able to judge the class of the web page more efficiently than the other machine learning models.

Table 3 and 4 shows the comparative analysis of the classification results which are analyzed with the other four machine learning models. LR classifier produces the lowest results with 80% accuracy whereas the proposed model produces 91% accuracy in ClueWeb12 dataset. The lowest and highest accuracies in DBpedia dataset are LR and the proposed model respectively with the corresponding accuracies 82% and 95%. As the nature of web page has lots of sections, a good classifier should pick the correct features under each section as it will affect the performance of the classifier massively. The existing machine learning models treats all features equally across the sections; thus, they produce less accuracy. The proposed machine learning model is able to identify the correct features each section, weights them accordingly before performing the classification. This is the main reason why the proposed model has better accuracy than the other exiting machine learning models. Moreover, the common contents across all the webpages such as header, footer are removed completely in the proposed classification model. There are two advantages from removing the contents; firstly, the size of the input decreases which increases the speed and

effectiveness of the classification; secondly, the uniqueness content of a web page is used for the classification which allows the classifier to learn more knowledge in the feature space. The main advantage of the proposed method over the existing models is the block identification. All the relevant information is kept in a single block. If there is a break in the flow of concept, then that information are kept in a separate block. For example, in a science article, all the introduction information may be present in the first block; the next few blocks may contain more information about the concept and may contain equations and Figs. The concept may be concluded in the last block. Hence identification of blocks will certainly increase the accuracy of the classification.

F1-Score is one of the good measures to determine the performance of a machine learning model. The F1-score is the harmonic mean of both precision and recall, which means when both precision and recall are high, and then the F1-score is also high. If the precision and recall are low, then the F1-Score is also low, whereas if any one of precision or recall is high, then the F1-Score is medial. F1 provides a good metric during the imbalanced training model. The proposed model is able to obtain a maximum of 95%.

Table 3: Performance of DBpedia Dataset

Algorithm	Accuracy	Precision	Recall	F1
kNN	0.85	0.97	0.84	0.9
LR	0.82	0.98	0.81	0.89
SVM	0.85	0.65	0.88	0.74
NB	0.92	0.92	0.98	0.95
Proposed	0.95	0.98	0.92	0.95

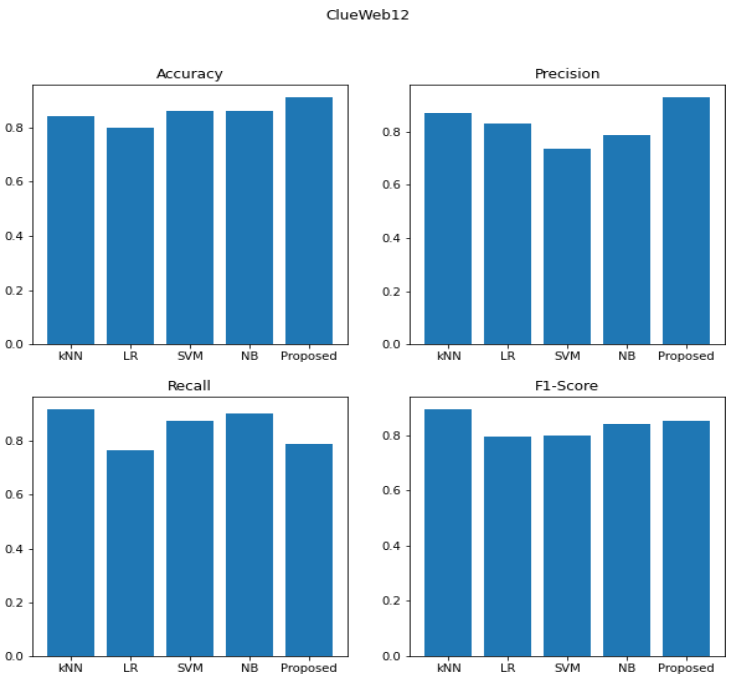

Fig 5: Performance on ClueWeb12 dataset

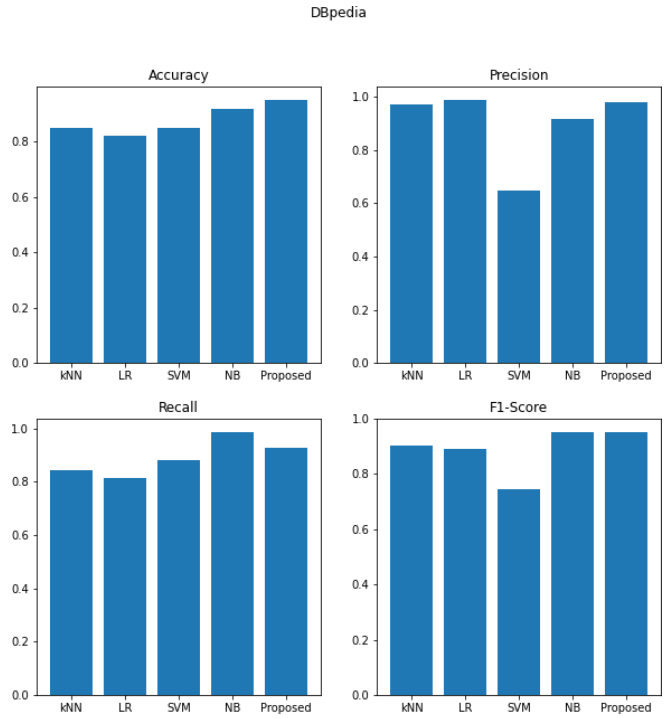

Fig 6: Performance on DBpedia dataset

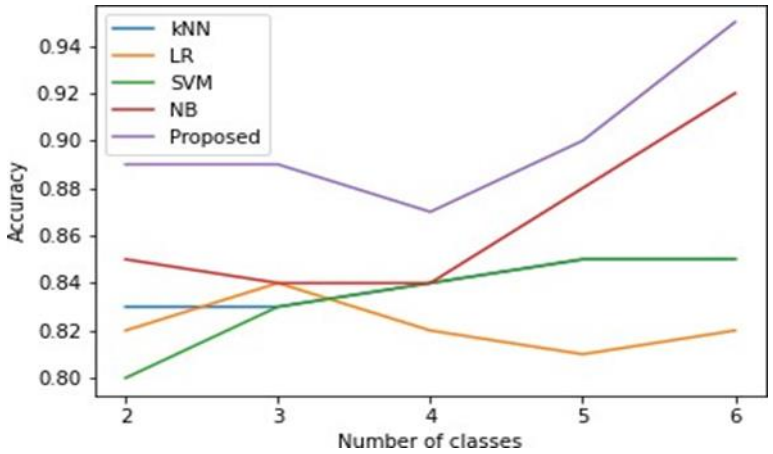

Fig 7: Performance on DBpedia dataset

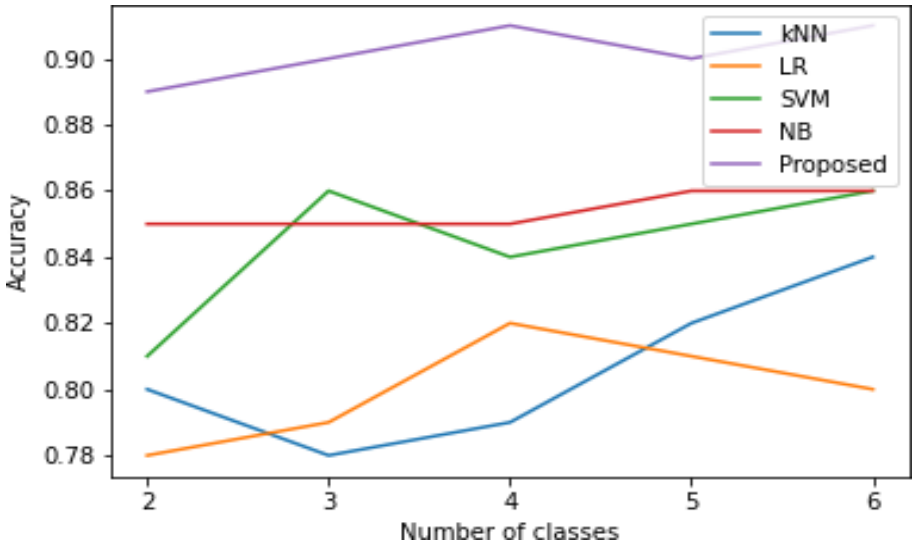

Fig 8: Performance on ClueWeb12 dataset

Table 4: Performance of ClueWeb12 dataset

Algorithm	Accuracy	Precision	Recall	F1
kNN	0.84	0.87	0.91	0.89
LR	0.8	0.83	0.7	0.79
SVM	0.86	0.74	0.87	0.8
NB	0.85	0.79	0.9	0.84
Proposed	0.91	0.92	0.78	0.85

Table 5: Accuracy comparison with respect to different block counts

Dataset	Number of Blocks				
	10	15	20	25	30
DBpedia Dataset	0.9	0.9	0.91	0.94	0.95
ClueWeb12 dataset	0.86	0.87	0.87	0.89	0.91

Fig 9 and table 5 presents the accuracy of the proposed method when varied number of blocks is used. The different block counts used in the experiment are 10, 15, 20, 25 and 30. The proposed model offers the accuracy of 0.86, 0.87, 0.87, 0.89 and, 0.91 respectively for ClueWeb12 dataset and the accuracy of 0.9, 0.9, 0.91, 0.94, 0.95 respectively for DBpedia Dataset. It was noted that the higher the number of blocks, the easier to absorb the class relationship and hence the proposed model produces the highest accuracy when the number of blocks is set to 30.Semantics can disturb the accuracy of a classification badly. During our experiment, we have fed purposefully few semantic words into the input space. As predicted, the proposed method was able to identify and separate all the semantic words out of the input space; where as other models struggle to identify this error. Nave Bayes is the only existing model which could remove few of the words.

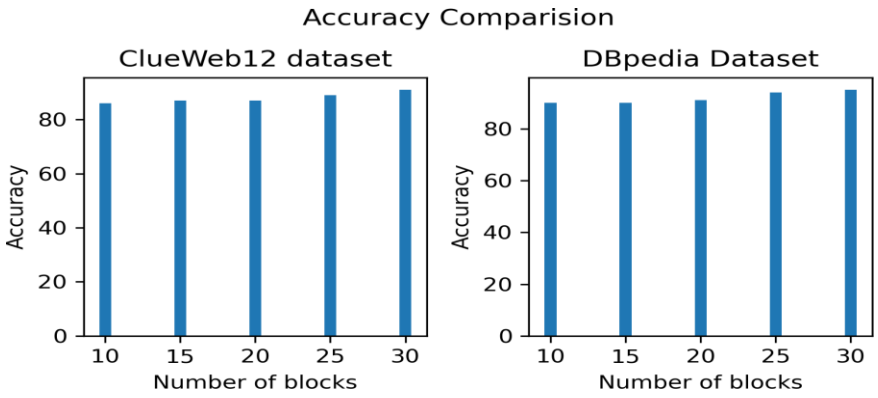

Fig 9: Accuracy and block counts

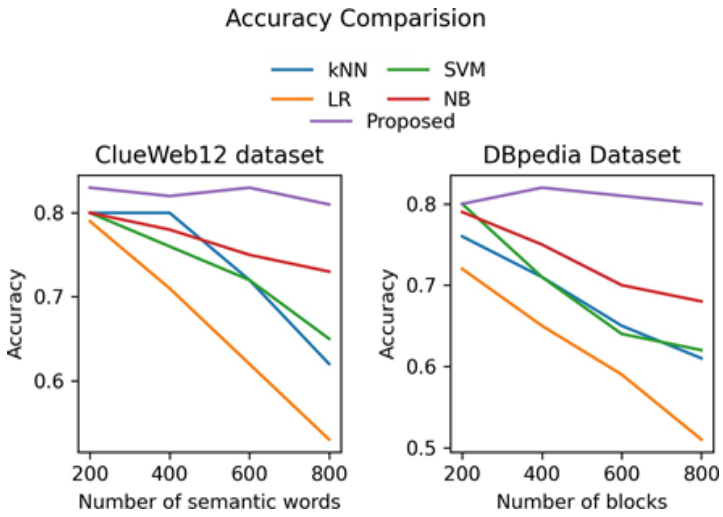

Fig: 10 Semantic Errors in the experiment

Fig 10 presents how the various models accuracy was affected when there is a presence of semantics. The performance of the proposed method can be increased by tuning the validation method. We have used K fold validation (k=10) and compared our work with few existing works. In this comparison, we have considered two related works which deals with web page classification. The first one [10] extracts link text and gives more importance to the connections and the second one [25] uses noun words and prioritize then over other terms. Our proposed methods used block level ranking thus eliminates the noises in other two words (those links or nouns present in useless blocks).One more reason is our proposed method can able to identify the usefulness of links and noun words in normal blocks also, this is done by semantic phase.Thus,we executed the models proposed by the two works (With 10-flod validation) and compared our proposed model under the same criteria. Our model was able to produce better accuracy in both the datasets. Table 6 shows the Comparision results of the three models. In the table 6, both the previous papers used feature extraction technique such as considering link text and noun text however, they failed to detect the importance based on text occurrence. For example, a text present in both header and introduction section should be treated differently. Our proposed classification model detects the features based on the position in the web page and assigns importance accordingly, thus, the performance of the proposed model is higher than the other two works.

Table 6: Comparison of the proposed model with existing works

	DBpedia Dataset				**ClueWeb12 Dataset**			
	Accuracy	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
[10]	0.82	0.77	0.94	0.85	0.85	0.80	0.97	0.88
[25]	0.84	0.71	0.95	0.82	0.80	0.64	0.9	0.75
Proposed	0.9	0.93	0.93	0.93	0.92	0.94	0.94	0.94

5. Discussions

As the number of web pages increases day by day, it is very difficult to manually read and classify the web pages. Hence, it is a mandatory that an automatic machine learning model should be build and can be used to classify web pages. Once the classification is complete, then it makes the searching and summarizing very easy. Unlike normal text documents, web pages contain much additional useful information (such as head text, link text, para headings and so on) that can be used for classification. This research work focuses on using this information to add hints to machine learning based text classification models to improve the accuracy and deal with the separation of the web pages. The practical usage of the proposed model is as follows.

1. Easy representation of web documents.
2. Faster searching.
3. Used to summarize a web document.
4. Extracting the sensitive and critical information from a web document.
5. Sentimental analysis.

The proposed model uses the above mentioned optimization techniques and the results show the efficiency of the proposed method. The proposed method is compared with the two existing research works and the accuracy of 0.9 is obtained which is 9.7561% more than the two works. The model is also compared with popular machine learning models and the accuracy of 91% was observed which 13% is more than the lowest accuracy obtained by the existing machine learning model.

The proposed method is compared using the following criteria

1. The number of blocks considered in the classification

2. Reducing the number of features by omitting the blocks

3. Extracting rich features from non-overlapping blocks.

6. Conclusion and Future Work

In this paper, we considered a web page classification problem and proposed a novel web page classification method that comprises three unique stages. We adopted domain shift which can create a compatible space for the margin to be placed for classifying the positive and negative samples. Furthermore, we select blocks that are very crucial in the classification process. Since the useless blocks are removed, the feature space is reduced and the accuracy is increased. Finally, the semantic words are also eliminated and the subset of most relevant features is choosing for building the classification model. The results shows an improved classification results as the relationship between the features and the target variable is understood easily. To demonstrate the efficiency of the proposed model, we compared it with other top machine learning classifiers. Two datasets are used in our experiment. The experimental results showed that the proposed model outperforms the other four approaches. In the future, we plan to extend the research work by considering deep nesting and further optimize the classification using link-based features.

References

[1] Ashokkumar, P., Arunkumar, N., & Don, S. (2018). Intelligent optimal route recommendation among heterogeneous objects with keywords. *Computers & Electrical Engineering*, *68*, 526-535.

[2] Cheng, M. Y., Kusoemo, D., & Gosno, R. A. (2020). Text mining-based construction site accident classification using hybrid supervised machine learning. *Automation in Construction*, *118*, 103265.

[3] Srivastava, S. K., Singh, S. K., & Suri, J. S. (2019). Effect of incremental feature enrichment on healthcare text classification system: A machine learning paradigm. *Computer methods and programs in biomedicine*, *172*, 35-51.

[4] Palanivinayagam, A., & Nagarajan, S. (2020). An optimized iterative clustering framework for recognizing speech. *International Journal of Speech Technology*, *23*(4), 767-777.

[5] Galitsky, B. (2013). Machine learning of syntactic parse trees for search and classification of text. *Engineering Applications of Artificial Intelligence*, *26*(3), 1072-1091.

[6] Palanivinayagam, A., & Sasikumar, D. (2020). Drug recommendation with minimal side effects based on direct and temporal symptoms. *Neural Computing and Applications*, *32*(15), 10971-10978.

[7] Ashokkumar, P., & Don, S. (2019). Link-Based Clustering Algorithm for Clustering Web Documents. *Journal of Testing and Evaluation*, *47*(6), 4096-4107.

[8] Nigam, C., & Sharma, A. K. (2020). Experimental performance analysis of web recommendation model in web usage mining using KNN page ranking classification approach. *Materials Today: Proceedings*.

[9] Buber, E., & Diri, B. (2019). Web Page Classification Using RNN. *Procedia Computer Science*, *154*, 62-72.

[10] Lee, J. H., Yeh, W. C., & Chuang, M. C. (2015). Web page classification based on a simplified swarm optimization. *Applied Mathematics and Computation*, *270*, 13-24.

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

[12] Li, H., Xu, Z., Li, T., Sun, G., & Choo, K. K. R. (2017). An optimized approach for massive web page classification using entity similarity based on semantic network. *Future Generation Computer Systems*, *76*, 510-518.

[13] Özel, S. A. (2011). A web page classification system based on a genetic algorithm using tagged-terms as features. *Expert Systems with Applications*, *38*(4), 3407-3415.

[14] Chen, R. C., & Hsieh, C. H. (2006). Web page classification based on a support vector machine using a weighted vote schema. *Expert Systems with Applications*, *31*(2), 427-435.

[15] De Smedt, J., Lacka, E., Nita, S., Kohls, H. H., & Paton, R. (2021). Session stitching using sequence fingerprinting for web page visits. *Decision Support Systems*, 113579.

[16] Chatterjee, M., & Namin, A. S. (2021). A fuzzy Dempster–Shafer classifier for detecting Web

spams. *Journal of Information Security and Applications*, *59*, 102793.

[17] Kang, J., & Choi, J. (2008, September). Block classification of a web page by using a combination of multiple classifiers. In *2008 Fourth International Conference on Networked Computing and Advanced Information Management* (Vol. 2, pp. 290-295). IEEE.

[18] Xu, G., Yu, Z., & Qi, Q. (2018). Efficient sensitive information classification and topic tracking based on tibetan Web pages. *IEEE Access*, *6*, 55643-55652.

[19] Ye, H., Cao, B., Peng, Z., Chen, T., Wen, Y., & Liu, J. (2019). Web services classification based on wide & Bi-LSTM model. *IEEE Access*, *7*, 43697-43706.

[20] Yang, Z., Gui, Z., Wu, H., & Li, W. (2019). A latent feature-based multimodality fusion method for theme classification on web map service. *IEEE Access*, *8*, 25299-25309.

[21] Uzun, E., Özhan, E., Agun, H. V., Yerlikaya, T., & Buluş, H. N. (2020). Automatically Discovering Relevant Images From Web Pages. *IEEE Access*, *8*, 208910-208921.

[22] Semantic-textual-similarity-nlp(2020).URL https://www.kaggle.com/bhrt97/semantic-textual-similarity-nlp

[23] Clueweb12. URL https://lemurproject.org/clueweb12/

[24] J.Holze,S.Hellmann, E.Starke.dbpedia dataset (2021).URL https://www.dbpedia.org/

[25] Smedt, J. D., Lacka, E., Nita, S., Kohls, H., & Paton, R. (2021). Session stitching using sequence fingerprinting for web page visits. Decision Support Systems, 150, 113579. doi:10.1016/j.dss.2021.113579

[26] Chatterjee, M., & Namin, A. S. (2021). A fuzzy Dempster–Shafer classifier for detecting Web spams. Journal of Information Security and Applications, 59, 102793. doi:10.1016/j.jisa.2021.102793

[27] Palanivinayagam A, El-Bayeh CZ, Damaševičius R. Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review. *Algorithms*. 2023; 16(5):236. https://doi.org/10.3390/a16050236

[28] Perdices, D., Ramos, J., García-Dorado, J. L., González, I., & López de Vergara, J. E. (2021). Natural language processing for web browsing analytics: Challenges, lessons learned, and opportunities. Computer Networks, 198, 108357. https://doi.org/10.1016/j.comnet.2021.108357

[29] Rinaldi, A. M., Russo, C., & Tommasino, C. (2021). A semantic approach for document classification using deep neural networks and multimedia knowledge graph. Expert Systems with Applications, 169, 114320. doi:10.1016/j.eswa.2020.114320

[30] Toçoğlu, M. A., & Onan, A. (2020). Sentiment analysis on students' evaluation of Higher Educational Institutions. Advances in Intelligent Systems and Computing, 1693–1700. https://doi.org/10.1007/978-3-030-51156-2_197.

[31] Jiang, X., Li, L., & Gao, G. (2022). Efficient secure and verifiable KNN set similarity search over outsourced clouds. High-Confidence Computing, 100100. https://doi.org/10.1016/j.hcc.2022.100100

[32] Vu, D.-H., Vu, T.-S., & Luong, T.-D. (2022). An efficient and practical approach for privacy-preserving Naive Bayes classification. Journal of Information Security and Applications, 68, 103215. https://doi.org/10.1016/j.jisa.2022.103215

[33] Wichitaksorn, N., Kang, Y., & Zhang, F. (2022). Random feature selection using random subspace logistic regression. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4108579

[34] Kachhwaha, R. ., Vyas, A. P. ., Bhadada, R. ., & Kachhwaha, R. . (2023). SDAV 1.0: A Low-Cost sEMG Data Acquisition & Processing System For Rehabilitatio. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 48–56. https://doi.org/10.17762/ijritcc.v11i2.6109

[35] Carmen Rodriguez, Predictive Analytics for Disease Outbreak Prediction and Prevention , Machine Learning Applications Conference Proceedings, Vol 3 2023.