SMU Data Science Review

Volume 6
Number 1 *Spring 2022*

Article 4

2022

Web Page Multiclass Classification

Brian Gaither
Southern Methodist University, gaitherb@mail.smu.edu

Antonio Debouse
Southern Methodist University, adebouse@mail.smu.edu

Catherine Huang
catherine_huang@mcafee.com

Web Page Multiclass Classification

Brian Gaither[1], Antonio Debouse[1], Catherine Huang, PhD[2]
[1] Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
gaitherb@mail.smu.edu, adebouse@mail.smu.edu, catherine_huang@mcafee.com

Abstract. As the internet age evolves, the volume of content hosted on the Web is rapidly expanding. With this ever-expanding content, the capability to accurately categorize web pages is a current challenge to serve many use cases. This paper proposes a variation in the approach to text preprocessing pipeline whereby noun phrase extraction is performed first followed by lemmatization, contraction expansion, removing special characters, removing extra white space, lower casing, and removal of stop words. The first step of noun phrase extraction is aimed at reducing the set of terms to those that best describe what the web pages are about to improve the categorization capabilities of the model. Separately, a text preprocessing using keyword extraction is evaluated. In addition to the text preprocessing techniques mentioned, feature reduction techniques are applied to optimize model performance. Several modeling techniques are examined using these two approaches and are compared to a baseline model. The baseline model is a Support Vector Machine with linear kernel and is based on text preprocessing and feature reduction techniques that do not include noun phrase extraction or keyword extraction and uses stemming rather than lemmatization. The recommended SVM One-Versus-One model based on noun phrase extraction and lemmatization during text preprocessing shows an accuracy improvement over the baseline model of nearly 1% and a 5-fold reduction in misclassification of web pages as undesirable categories.

1 Introduction

As the volume of content hosted on the Web continues to grow larger, individuals and organizations are increasingly seeking capabilities to prevent access to undesirable content such as pornography and protect themselves from malicious content hosted on the Web. Web pages can fall into many different categories such as finance, marketing, business, and pornography, to name a few. Accurately classifying websites into the appropriate categories enables individuals and organizations to filter unwanted categories. By filtering such unwanted categories, homes and businesses can prevent access to websites belonging to such categories and therefore reduce risk of exposure to potential malicious content such as malware and phishing attacks or simply prevent viewing such websites.

Through media coverage of identity theft and large-scale cyber security incidents plaguing various industries, homes and businesses are becoming more and more aware of the pervasiveness of undesirable online content and the potential damage malware and phishing attacks can cause. Attacks such as drive-by-downloads are an example "attack scenario which can add malicious extensions, inject malicious ads into search results and even steal credentials in some cases" [1]. As a result, these internet users rely more heavily on online security services that can provide website categorization to restrict access to undesirable and potentially malicious sites to assist in their safe online navigation. Some businesses who specialize in developing cyber security products employ automatic website categorization techniques to provide such web filtering services. Such services can block access to websites flagged for certain categories such as pornography or drugs [2].

There are multiple use cases for effective web site categorization. For example, some parents and businesses may choose to prevent access to certain categories of websites such as pornography or gambling. For home use, parental controls can be put in place to block access to such sites. For businesses, gateway appliances can be utilized to prevent access to certain websites. Additionally, certain web page categories may be more likely to serve malicious content such as files or scripting that may cause harm to a computer or information on that computer [3]. Some websites are not legitimate sites at all but are made to look like popular legitimate sites for phishing purposes. In a phishing attack, the attacker may send the victim an email from a spoofed email address that looks legitimate with a request to click a link to a website whose URL appears familiar and benign but, perhaps, a character may be missing or out of place. The website will closely resemble a trusted site such as a banking website. The fake banking website will require the victim to log in and provide sensitive information. When the victim enters their login credentials and sensitive information, the attacker has now stolen this information and can use it for nefarious purposes [4].

If an application uses the categorization of a web page to prevent access to certain types of websites such as pornography, it should be done so with very few false positives. Legitimate businesses whose web traffic is impeded due to being incorrectly classified would be very unhappy and likely to demand recategorization to reclaim traffic lost to their website. Therefore, misclassifying a web page as pornography when it is truly a legitimate business would carry a higher weight than misclassifying an educational website as business, which would still allow access to that website. Additionally, if a user of such an application is unable to browse a legitimate site due to misclassification, this would lead to a loss of trust and a decrease in value of the application.

Web page classification can be conducted using text-based methods, image-based methods or combined image and text-based methods. Additionally, web page classification is a multi-label and multi-class classification problem. Web page classification is also a multi-lingual problem. This paper focuses on single-label and text-based categorization for English language only. This paper discusses current state-of-the-art techniques utilized in web page categorization including text normalization, feature engineering, and categorization algorithms used in the Literature

Review section. Many of the techniques used in the literature utilize standard preprocessing techniques such as stemming, dropping all upper-case characters to lower case, and removal of non-alpha characters [7][12][13]. Various dimensionality reduction techniques have been applied in the current state-of-the-art using algorithms such as Information Gain and ReliefF [14]. Classification algorithms generally utilize Support Vector Machine (SVM) and Multinomial Naïve Bayes. This study will show how combining text preprocessing techniques such as key word extraction and noun-phrase extraction, feature engineering, dimensionality reduction and ensemble methods improve overall accuracy of the web page categorization while ensuring minimal errors in misclassification of websites such as pornography and drugs. The Methods section introduces the reader to the dataset used for this study as well as the techniques leveraged to achieve improved accuracy.

2 Literature Review

Although some rudimentary work had been done prior, Alan Turing's 1950 article titled "Computing Machinery and Intelligence" [5] really championed the significance of researching natural language processing (NLP) and infamously created the Turing Test; a test of a computer's ability to have a conversation with a human and the human failing to discern if they are talking with another human or not. Like many of Alan Turing's innovations in computer science, this laid a foundation providing support and guidance resulting in current NLP abilities to give automated voice responses or provide web page classifications. Turing's work laid the foundation which eventual lead to Christopher Manning and Hinrich Schutze publishing "Foundations of Statistical Natural Language Processing" in 1999, which has been cited in over 15,000 articles to date. They also claim this book to be "the first comprehensive book in statistical natural language processing (NLP) to appear" [6].

Hashemi [7] thoroughly surveys various methodologies in the literature and describes how the approach to web page classification has evolved. This study highlights the various techniques by which the problem has been tackled and notes limitations as well as potential future research. Hashemi [7] highlights that though web page classification is not a new endeavor, because of recent improvements in computing power and memory space, there has been an increase in researchers tackling the problem. However, web page classification is still in its infancy as the problem is complex and web page content is quite diverse, not to mention the computational costs involved. Hashemi [7] identifies that text classification is largely aimed around term frequency counts of words that occur within the text to form a feature vector and then using those feature vectors for model training. However, a problem with using term frequencies as feature vectors is their sparseness. This is especially true when dealing with text that is rather short. Therefore, feature selection is utilized to reduce

dimensionality focused on including only features that help to maximize separation between document classes [7].

Salton et. al. [8] suggests TFIDF as an alternative to the term-frequency feature vector. Term Frequency-Inverse Document Frequency (TFIDF) is intended to give lower weights to terms appearing in many documents and higher weights to terms that appear in only a few documents. TFIDF is calculated by multiplying a term's frequency (TF) by an inverse document frequency (IDF) factor. Where w_{ij} is the weight of the i-th term (t_i) in the j-th document (d_j), $TF(t_i, d_j)$ is the frequency of term t_i in document d_j, d is the number of documents, and $DF(t_i)$ is the number of documents containing the term t_i[8]. An important note about using TFIDF for classification to remember is that the IDF is based on the training set used to train the classifier. Thus, when classifying new text using the trained classifier, the term frequency from the document must be multiplied by the IDF derived from the training set [8].

Hashemi [7] identified several limitations and possible future directions to potentially improve web page classification results. Though his paper covers both text-based and image-based approaches, the text-based limitations are referred to herein. The paper specifies that surrounding words provide context and are mostly ignored in text classification. Considering surrounding terms that indicates a term's context could enable improvements in web page classification. For example, the word nickel can mean a 5-cent coin or could also mean a type of defense in American football. The words surrounding "nickel" within the document could help provide such context to the model. Another area of potential research is the use of the distribution and structure of text in the web page HTML tags, as well as hyperlinks. Hashemi reviews many studies in the literature and identifies the most common classifier used is the SVM [7].

Boser et al. [9] introduces the concept of the SVM. The SVM classification technique finds the optimal margin between the training patterns and the decision boundary on separable data. The paper describes the training algorithm that maximizes the margin between training samples and class boundaries. Only the supporting patterns, known as support vectors, that are closest to the decision boundary are used the resulting classification function. These support vectors are a very small subset of the original training data. The SVM was originally designed for binary classification, however, the one-versus-one is a technique used for multi-class classification that builds N(N-1) classifiers. Limitations of the SVM are the lack of transparency in results caused by a high number of dimensions [10]. Rung-Ching Chen et al. [11] expanded on SVM by adding a latent sematic analysis and web page feature selection to create a "weighted voting support vector machine" by combining latent semantic analysis (LSA) with back propagation network (BPN). An WVSVM resulted in an F-value (ensemble of precision and recall) of 93% compared to running separate model of LSA-SVM of 91% and BPN of 73% [9].

Altay et al. [12] propose a novel keyword extractor to be used for document classification. While traditional methods use a term frequency-based approach, this

paper proposes a new method termed keyword density for determining feature weights. Additionally, this research studies various machine learning techniques such as SVM, Extreme Learning Machine (ELM), and Maximum Entropy (MaxEnt). This study uses a keyword density extractor library designed by Comodo Group. The paper references the Comodo website, however, as of the time of this writing, details about the keyword density extractor are not available on the Comodo website. The approach used for keyword density considers HTML tags of a webpage and using the frequency of the keywords and the tags, a score is applied which is used as the density. The findings of this study were that using an RBF kernel for SVM yields the best accuracy with linear-SVM and MaxEnt showing similar results. After applying methods such as stemming, special character removal, article extraction and dimensionality reduction, the novel keyword extraction and density scoring is performed [12].

Thamrongrat et al. [13] performed research comparing a one-versus-one SVM classification and voting algorithm against a one-versus-all SVM classification and voting algorithm. Additionally, the research compares four different feature selection techniques such as ReliefF, Information Gain, Chi Square, and Gain Ratio. The one-versus-one SVM approach trains a SVM classifier on training samples belonging to two classes only with the number of classification sets equating to (No. of classes * (No. of classes -1))/2. The outputs of the SVMs are then used to determine the number of votes one by each class with the class having the maximum number of votes assigned to the test pattern. The one-versus-all approach trains a SVM classifier for each class against the rest of the classes. For example, class 1 is the positive class, and all remaining classes are the negative class. There would be a classifier constructed for each class that exists in the training set. This research utilized the TFIDF method to assign term weight values in the term document matrix which is then used for the feature vectors. Next, this paper selects only terms that meet a specified threshold for document frequency. There were separate processes to derive the text features and title features, then combining both the text and title features. The paper concludes that ReliefF feature selection yields the highest F-measure compared to Information Gain, Chi Square, and Gain Ratio. The one-versus-all SVM classification strategy yields a higher F-measure than the one-versus-one SVM technique. Using both text and title features with a one-versus-all voting algorithm gives the highest F-measure [13].

Robnik-Šikonja et al. [14] provide a study of the ReliefF algorithm for the purpose of feature selection. Relief algorithms can detect conditional dependencies indicating a relationship between attributes and provide a unified view on the attribute estimation in regression and classification. Another benefit of Relief algorithms is that their estimates are easy to interpret. Typically, Relief algorithms are used as a feature selection technique applied prior to model training. Feature selection is important in any machine learning endeavor to avoid including noisy, irrelevant features that provide little value. At its core, feature reduction is aimed at selecting only a small subset of total available features that is necessary to describe the target class. Therefore, it's important to understand how feature selection algorithm's function and ensure the

selected features adequately aid to solve the task of classification or regression. The conclusion of this study is that Relief algorithms are generally successful at creating estimates that accurately detect conditional dependencies [14].

Kobayashi et al. [15] describes text mining techniques and the sequential steps involved. The paper identifies the following steps: training data preparation, text preprocessing, feature transformation, application of classification techniques, and validation. This paper is useful for researchers who are new to text classification and natural language processing techniques in general. Kobayashi et al. delve into text preprocessing techniques such as tokenization, stop word removal, differences between stemming and lemmatization in normalizing text, removal of special characters and making all character's lower case. The paper also touches on the topic of text transformation such as using vector space model and TFIDF, however, consecutive characters or parts of speech are potential areas to experiment with when classification model performance is poor. As with other research papers, reducing dimensionality is important and various methods are suggested such as setting a cutoff on the scores derived from the text transformation step with the goal of removing rare terms that are introducing noise. Additionally, the paper recommends supervised scoring methods as approaches to reduce dimensionality such as chi-squared, mutual information, information gain, and Gini index. A recommended starting point for removing noisy terms is to first calculate the document frequency of each term and terms belonging to the lower 5th and upper 99th percentiles are filtered out. Kobayashi et al. also suggests combining different methods of classification techniques such as Naïve Bayes, SVM, gradient boosted trees, Random Forest and choosing the pair with the lowest error rate [15].

Rajalakshmi et al. [16] proposes including a URL-based classifier along with a rejection framework that can be used as a first-level filter in a multistage classifier with feature extraction of content from the web page that can be done in later stages. The compelling argument for such an approach from the authors is that the classification of a URL is less costly than the extraction of a web page's content and subsequent processing required before classification can be performed. This becomes important in cases where a website must be blocked before the classification can be completed. It is noted, however, that URL classification is challenging as a URL itself contains many compound words, abbreviations or nonmeaningful and part-of-a-word terms while some URLs do not contain any related information about a page, and hence pose more challenges for classification. The paper proposes a multi-level classification approach with each level using more and more information for the purposes of classifying the web page. Level 1 uses URL features alone, level 2 uses additional information such as web page title and anchor text, level 3 uses web page content, and level 4 uses the contents of sibling pages. The classification process stops at the level by which a confident classification is achieved and only continues if the classification is not achieved. Kan, M.Y., & Thi, [17] also explores using the URL to classify instead of analyzing the entire webpage. The researchers separated the URL into significant

sections –which they called “meaningful chunks”- then added other features like orthographic and component. They used a supervised maximum entropy model to analyze results based on binary, multi-class, and hierarchical classification. The methods lead to faster processing time and highlight the benefit of great preprocessing techniques of the data. Another major benefit of using the URL to create a classification is that it is highly encouraged for URL names that are easy to remember and have some language that hint at the content of the webpage, which assumes a higher chance for patterns to be created and trends modeled [18].

Bo et al. [19] performed a study of various feature selection methods and classification techniques to propose a set of recommendations based on the model performance results. This paper includes a feature selection technique not covered in previous literature on the topic that is called Symmetrical Uncertainty. Thamrongrat et al. [13] had performed a study evaluating ReliefF for feature selection for multi-class classification and concluded it resulted in the highest F-measure in the study. However, in that study, Symmetrical Uncertainty was not included as a competing method. Therefore, this paper is helpful in gaining insight into the two feature selection methods compared side-by-side. This paper evaluates ReliefF and Symmetrical Uncertainty for feature selection as well as multiple classification techniques such as Hidden Naïve Bayes, Naïve Bayes, Complement Class Naïve Bayes, Support Vector Machine, K-Nearest Neighbor, and C4.5 Decision Tree algorithm. The results of the experimentation indicate that the combination of Symmetrical Uncertainty and Hidden Naïve Bayes yields the highest accuracy and F-measure.

Qiang et al. [20] introduces a framework called “Strong-to-weak-to-strong” (SWS) that transforms a “strong” learning algorithm to a “weak” algorithm by decreasing its iterative number of optimizations while preserving its other characteristics like geometric properties. To improve the text classification performance, the kernel trick is used to so that the weak algorithm works well in high dimensional space. This paper suggests using a minimax probability machine (MPM) proposed by Lanckreit et al., which tries to minimize the probability of misclassification of data. The MPM uses Mahalanobis distances to involve geometric information of the vector while SVM ignore such information once the support vectors are identified. The algorithm proposed uses an iterative least-squares approach with less iterations to solve the kernelization. The multiclass classifier obtained by the algorithm becomes much stronger of a learning algorithm according to the reported classification performance. Comparing the SWS algorithm to other traditional algorithms such as SVM yields a slight improvement in the per class precision/recall breakeven point on the ten largest categories in the data set used for experimentation. The max improvement is 5% with the max degradation per class being 2% [20].

Singhal et al. [30] contends that in the medical field there is a pressing need to take unstructured data from multiple sources with nuanced meaning variations for words and emotions that are specifically unique to the healthcare industry. To address this

issue, the researchers used multi-class classification along with a weak supervised learning approach to improve categorization of text relating to the healthcare industry. The goal was to improve upon traditional supervised learning model by addressing the need to self-learn and improve. Their solution combined two basic levels of NLP and machine learning. The first layer uses current heuristics and domain knowledge to categorize and create an annotated training set. Then using TF-IDF for feature extraction and that were trained using logistic regression and linear SVMs [30].

Buber et al. [31] approached the objective of improving upon current methods of web page classification by using a 5-layered RNN model on the meta tag information, the attributes that describe the web page data like title, descriptions, etc. The researchers used a Recurrent Neural Network on the textual properties of web pages to develop a classification model. As previously mentioned, the benefit of using meta tags is that the administrator of that website uses this information to explain the purpose and function of the web page which incentivizes them to try and distinguish their website as much as possible and leading to strong possibilities for signals to be modeled. This experiment's 85% accuracy rate did not lead to a significant improvement compared to previous analysis but the use of "transfer learning" did reduce the consumed system resources. Early experiments that used deep neural networks along with pre-trained words embedded have shown promise in identifying meaningful syntactic and semantic regularities [32]. Given the complexity caused by the different languages and cultures' sentence structures and multiple connotations, Mikolov et al. objective was to use neural networks to better train models to parse out content using context to help the model distinguish the meaning of the words and terms leading, with the hope of improving accuracy. Although Mikolov and his partners' model did not add any significant improvement to accuracy it is a remarkably interesting concept to explore as another method of feature creation to improve signaling. Many natural language processing models involves at least some pre-processing to formulate the data that can then be modeled to identify important features that can be used to train a model that uses those signals/important features to make classification predictions [25]. This paper also investigates various preprocessing techniques to find unique opportunities to create additional features with the hope of making a significant accuracy improvement over other current multi-class webpage classification models.

This paper hypothesizes that by first extracting noun phrases, then performing lemmatization along with contraction expansion, removing special characters, removing extra white space, lower casing and removing stop words, the models will have more relevant terms from which to use for feature engineering, thus yielding better accuracy. As described in the introduction, it's not only important to have a model with higher accuracy but also a model that minimizes misclassification of web pages as undesirable categories such as pornography and drugs. Additionally, utilizing feature reduction using chi squared and SelectKBest, this paper will identify an optimal subset of features that reduces the size of the TFIDFVectorizer object to be held in memory yet will generalize well to unseen data.

3 Methods

3.1 Data

This research utilizes a hand labelled dataset comprised of 41,876 English language websites. The data was produced by a cyber security company using human researchers for labelling. There are eight different attributes in the dataset including the URL, response code, category, language, title, summary, key words, and website content. The URL is the URL of the website including the domain and top-level domain only. The response code for all records is 200 which means the scraping mechanism was successfully able to reach the site. The category is the label or target value for which this research will use to train and test the models. The language for all records is 'en' which means English. The title contains the title of the website in the meta tags. The summary is the description of the website in the meta tags. The key words are the key words from the meta tags. The website content is the actual content text on the website and all HTML tags have been removed. The data is comprised of eight different categories and each category consists of the number of observations shown in Table 1: Categories and the number of records below.

Category	Label Code	No. of Records
Business	bu	11,731
Drugs	dr	1,595
Education	ed	3,894
Marketing	mk	8,775
Online Shopping	os	6,789
Sports	sp	2,535
Pornography	sx	6,557

Table 1: Categories and the number of records

The data was split into a train, test, and holdout sets for model development, testing and validation. The train set is roughly 70% of the data, the test set is 20% of the data, and the holdout set is 10% of the data. The holdout set was created to compare competing models and is used as a validation data set. The test set is used to test a trained model and tune the model. The resulting data sets generated along with their various record counts by category are shown below in Figure **1**: Train, test, and holdout set sizes.

Target Label	Train Count	Test Count	Holdout Count
bu	8187	2355	1189
mk	6177	1709	889
os	4743	1393	653
sx	4544	1345	668
ed	2723	783	388
sp	1783	494	258
dr	1145	307	143
Total	**29302**	**8386**	**4188**

Figure 1: Train, test, and holdout set sizes

3.2 Noun Phrase Text Extraction and Additional Text Preprocessing

Preprocessing was performed on the raw data for further analysis and feature engineering. First, the title, key words, summary, and content fields were combined. Next, nouns phrases were extracted from the combined content and added as a new attribute to the data for deeper analysis. This paper utilizes the Spacy library along with the en_core_web_sm trained pipeline to extract all noun phrases from all web pages. This is a key preprocessing step that aims to differentiate a marked improvement above and beyond the current baseline approach of using full document text and standard preprocessing techniques. Noun phrases were selected as a technique to identify the subjects discussed in each web page. Then, the data was further cleaned using the following techniques: removing extra new lines, removing accented characters, expanding contractions, lemmatization, removing special characters, removing extra white space, lower casing, and removing stop words. The sequence of the steps can be found below in Figure 2: Text Preprocessing Steps. Further exploratory data analysis was performed on the resulting terms to identify the top 25 most frequently occurring words in each category as shown in Table **4**: Top 25 Occurring Terms by Category in the Appendix of this paper. The resulting terms are then used for feature engineering using TFIDF and us unigrams only. This is referred to as Noun-Based text in the methods outlined below.

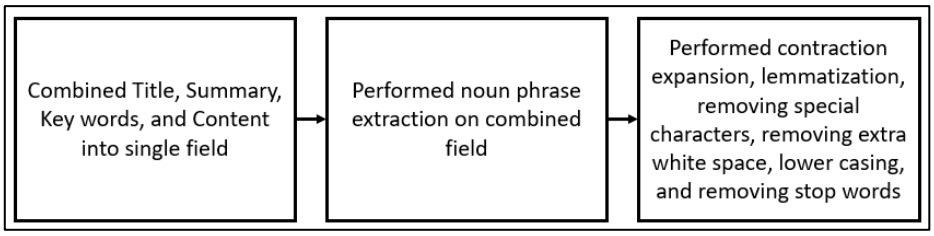


Figure 2: Text Preprocessing Steps

Additionally, and separately, a python library named YAKE [33] was utilized to examine the combined content and derive key words from the text. YAKE utilizes text statistics to extract key words from text. Key words extracted from the content using YAKE were saved as a new attribute to the data set for further analysis and feature engineering.

3.3 TFIDF

TFIDF for feature engineering was utilized in this study as it provides additional information above and beyond term frequency only. Equation 1: Calculating TFIDF below specifies the TFIDF calculation. Hashemi states that TFIDF considers a terms occurrence in each document as well as how often the term appears in all documents in the corpus. Thereby giving less weight to terms appearing in many documents and more weight to terms which appearing in fewer documents [7]. The scikit-learn library [34] using TfidfVectorizer was used to transform the features from the Noun Phrases and Key Words to TFIDF feature vectors. The optimal TFIDFVectorizer parameter for min_df was identified to be 25 with the optimal max_df identified to be .9 by iterating through combinatorial permutations.

$$w_{ij} = TF(t_i, d_j) \times IDF(t_i)$$

$$IDF(t_i) = \log\left(\frac{d}{DF(t_i)}\right)$$

Equation 1: Calculating TFIDF

3.4 Dimensionality Reduction

After performing TFIDF feature transformation on the resulting terms using TFIDFVectorizer, the resulting matrix width was 344,693. Separately, TFIDF feature transformation was performed on the key words and the resulting matrix width was 54,224. The two resulting matrices will be used to develop two separate models for comparison against the baseline model explained further in this paper. The high dimensionality of the matrixes warranted an approach to dimensionality reduction given the size of the training data is only 29,302 rows. Using scikit-learn's feature selection algorithm SelectKBest with chi2, the dimensionality of the data was reduced to only 4,100 features wide for the noun phrase matrix and 2,000 features wide for the key word matrix. To identify the optimal features for each matrix, the original TFIDF matrix was calculated, then a loop was performed to compare model accuracy changes as the number of features selected using SelectKBest increased. The effect of selecting the optimal number of features is shown below in Figure 3: Effect of No. of Features on Accuracy.

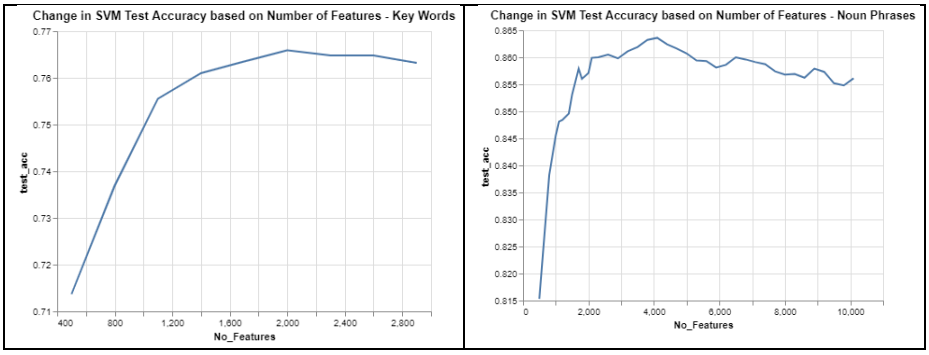


Figure 3: Effect of No. of Features on Accuracy

Another key benefit of performing feature selection is the ability to rebuild the TFIDFVectorizer matrix using only the identified significant features as the resulting matrix is a large sparse feature matrix. Once the optimal features are identified, the selected features' associated terms can be retrieved and a new, smaller TFIDFVectorizer matrix can be created. This will reduce the overall size of the TFIDF Object that must be loaded in memory when performing inference when the model is deployed.

3.5 Support Vector Machine Linear Kernel Model on Noun-Based Text

An SVM model with a linear kernel was trained using the scikit-learn svm.SVC model which supports generation of prediction probabilities. The model applies weights to the drugs and pornography categories to minimize misclassifications of web pages to these categories. The features used included the TFIDF vectorization of the individual terms resulting post the text preprocessing steps as described above in section 3.2 and Figure **2**: Text Preprocessing Steps using unigrams only. That is, the single terms and their associated TFIDF values are used as the features. As mentioned in the conclusion of this paper, future research could be performed to determine if using extracted noun phrases intact as multi-term features rather than unigrams will yield both a reduction in feature dimensionality as well as improved accuracy. When evaluating performance on the test data, the overall accuracy is 0.8637, the accuracy on the pornography category is 0.9948 and the accuracy on the drugs category is 0.9479. When evaluating performance on the holdout data, the overall accuracy is 0.8649, the accuracy on the pornography category is 0.9925 and the accuracy on the drugs category is 0.9650. This model (svm_lin) yields an overall improvement against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table.

3.6 Support Vector Machine Non-Linear (RBF) Kernel Model on Noun-Based Text

An SVM model using a non-linear 'RBF' kernel was trained using the scikit-learn svm.SVC model with kernel='rbf'. The model applies weights to the drugs and pornography categories to minimize misclassifications of web pages to these categories. Again, the features used included the TFIDF vectorization of clean noun phrase terms using unigrams only. When evaluating performance on the test data, the overall accuracy is 0.8641, the accuracy on the pornography category is 0.9911 and the accuracy on the drugs category is 0.9381. When evaluating the performance on the holdout data, the overall accuracy is 0.8682, the accuracy on the pornography category is 0.9940 and the accuracy on the drugs category is 0.9441. This model (svm_rbf) yields an overall improvement against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table.

3.7 Support Vector Machine Linear Kernel Model on Extracted Keyword Text

An SVM model using a linear kernel was trained using the extracted key words from the YAKE library and their TFIDF vectorization. The specific algorithm used is the scikit-learn svm.SVC model which supports generation of prediction probabilities. The model also uses weights for the drugs and sex categories to minimize misclassifications of web pages to these categories. The resulting accuracy was slightly worse than the approach using noun-based text. When evaluating the performance on the test data, the overall accuracy is 0.7645, the accuracy for the pornography category is 0.9703 and the accuracy for the drugs category is 0.8958. When evaluating the performance on the holdout data, the overall accuracy is 0.7720, the accuracy for the pornography category is 0.9805 and the accuracy on the drugs category is 0.9091. This model (svm_key_lin) yields an overall degradation against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table.

3.8 Linear Stochastic Gradient Descent Model on Noun-Based Text

As a comparison, a linear SDG classifier model was trained using the noun phrase features. This model also used class weights to protect against misclassification of websites as pornography and drugs, early stopping set to true, loss function was modified_huber and number of iterations with no change was set to 5. The resulting accuracy of the model on the test data was comparable to the SVM on noun phrases with an overall accuracy of .8651. The test accuracy for the pornography category is 0.9970 and the test accuracy for the drugs category is 0.9511. When evaluating performance on the holdout data, the overall accuracy is 0.8649, the accuracy for the pornography category is 0.9940 and the accuracy for the drug category is 0.9720. This model (sgd_lin) yields an overall improvement against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table.

3.9 Ensemble Voting Classifier Model on Noun-Based Text

A scikit-learn Voting Classifier was trained and tuned using four input models: Logistic Regression using l2 penalty with max iterations of 100, the SVM linear, linear SGD classifier using l2 penalty, optimal learning rate and hinge as loss, and the SVM with 'rbf' kernel. The model was tuned using a grid search using soft and hard voting and an array of various weighting permutations to identify the final model which uses hard voting and 2 votes for the SGD classifier and a single vote for the remaining input models. When evaluating the performance on the test data, the overall accuracy is 0.8655, the accuracy for the pornography category is 0.9889 and the accuracy for the drugs category is 0.9056. When evaluating the performance on the holdout data, the overall accuracy is 0.8670, the accuracy for the pornography category is 0.9925 and the accuracy for the drug category is 0.9091. This model (ens_svm) yields an overall improvement against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table.

3.10 SVM One-versus-Rest Model on Noun Based Text

A scikit-learn OneVsRestClassifier (also known as one-versus-all) was trained using the SVM linear model. Class weights are not used for this model, nor are they supported for this classifier type. This classifier uses an approach that results in fitting a separate classifier for each class in the training set. Therefore, there are seven different classes in the dataset used for this research resulting in seven different classifiers. When evaluating the performance on the test data, the overall accuracy is 0.8678, the accuracy on the pornography class is 0.9918 and the accuracy on the drug category is 0.9381. When evaluating the performance on the holdout data, the overall accuracy is 0.8644, the accuracy on the pornography class is 0.9925 and the accuracy on the drug category is 0.9371. This model (ovr_lin) yields an overall improvement against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table.

3.11 SVM One-versus-One Model on Noun Based Text

A scikit-learn OneVsOneClassifier was trained using an SVM with linear kernel model. Class weights are not used for this model, nor are they supported for this classifier type. This classifier fits one classifier per class pair with the class that receives the most votes being selected as the predicted class. This results in fitting n_classes * (n_classes -1)/2 classifiers. The dataset for this research contains seven classes resulting in 21 classifiers being fitted in this model. When evaluating the performance on the test data, the overall accuracy is 0.8632, the accuracy for the pornography class is 0.9859 while the accuracy for the drug class is 0.9251. When evaluating the performance on the holdout data, the overall accuracy is 0.8660, the accuracy on the pornography category is 0.9895 and the accuracy on the drug category is 0.9301. This model (ovo_lin) yields an overall improvement against the baseline model (svm_lin_base) as can be seen in Figure 5: Model Comparison Table. In fact,

this model provides an improvement across all model metrics while also minimizing the number of false positives on pornography (fp sx) and drug (fp dr) categories. As a result, this is the recommended model.

3.12 Baseline SVM Linear Kernel Model Using Standard Text Preprocessing Techniques

The baseline model for comparison is an SVM linear kernel model trained on an optimally selected set of features of 3,800 terms. The text pre-processing performed includes the typical techniques included in the Hashemi [7] article including expanding contraction, stemming, special character removal, changing all text to lower case, removal of stop words and removal of any extra whitespace and new lines. The optimal number of features for the baseline model were identified by iterating through a SelectKBest algorithm using chi square as was done to identify the optimal features for the candidate models implemented in this research. The accuracy of the model using multiple feature sizes is visualized in Figure 4: Optimal Features for Baseline Model below where the optimal 3,800 feature size can be identified.

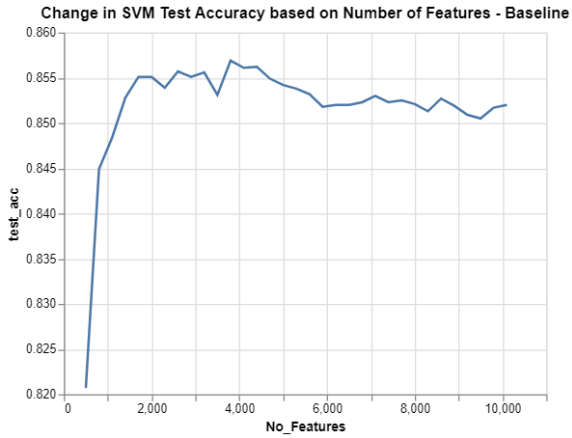


Figure 4: Optimal Features for Baseline Model

4 Results

As mentioned previously, the key difference in the preprocessing steps used to implement the candidate models for this research is to first extract noun-phrases as defined in the Methods section above that describe what the subject matter of each webpage is and then perform lemmatization instead of stemming along with the other forementioned preprocessing steps. This paper hypothesizes that by first extracting noun phrases and performing lemmatization, the models will have more relevant terms

from which to use for feature engineering, thus yielding better accuracy and minimize False Positives for undesirable categories such as pornography and drugs.

For the comparison charts and tables below, the following code names of the candidate models and their meanings is outlined in Table 2: Model Code Names and Descriptions.

Model Code Name	**Model Description**
svm_rbf	SVM non-linear model using 'rbf' kernel
svm_lin	SVM linear kernel
sgd_lin	Stochastic Gradient Descent model
ovo_lin	SVM One-versus-One model using linear kernel
ovr_lin	SVM One-versus-Rest model using linear kernel
ens_svm	Scikit-learn Voting Classifier with four input models logistic regression, svm_lin, svm_rbf, sgd_lin using hard voting with weighting (1,1,2,1) respectively
svm_key_lin	SVM using linear kernel and only using key words
svm_lin_base	Baseline SVM using standard text-preprocessing techniques

Table 2: Model Code Names and Descriptions

The following metrics have been collected across each of the models for evaluation and comparison as outlined in Table 3: Comparison Metrics Defined:

Column Name	**Description**
test_acc	Model accuracy on the test set
test_precision	Model precision on the test set
test tpr/recall	Model True Positive Rate on the test set
test F1 Score	Model F1 score on the test set
sx_acc	Model accuracy on the pornography category in the test set
dr_acc	Model accuracy on the drug category in the test set
ed_acc	Model accuracy on the education category in the test set
sp_acc	Model accuracy on the sports category in the test set
mk_acc	Model accuracy on the marketing category in the test set
os_acc	Model accuracy on the online shopping category in the test set
miss_sx	How many times did the model misclassify a web page as pornography in the test set?
miss_dr	How many times did the model misclassify a web page as drugs in the test set?

Table 3: Comparison Metrics Defined

A crucial success metric for this experiment is developing a classification model that avoids misclassifying web pages as pornography or drugs due to the potential ramifications that could arise as mentioned in the introduction section of this paper. Therefore, the two metrics "fp_sx" and "fp_dr" play a major role in determining a successful model. However, overall model performance as measured by the remaining metrics in the table must show improvement against the baseline model to be considered for recommendation.

The comparison of the models to the baseline model is shown below in Figure 5: Model Comparison Table.

Test Data

model_name	Test acc	Test precision	test tpr/recall	test F1 Score	sx acc	dr acc	ed acc	sp acc	mk acc	os acc	fp sx	fp dr
svm_rbf	0.8641	0.8664	0.8641	0.8647	0.9911	0.9381	0.8799	0.8785	0.7970	0.7854	7	5
svm_lin	0.8637	0.8668	0.8637	0.8645	0.9948	0.9479	0.8723	0.8684	0.8016	0.7832	10	7
sgd_lin	0.8651	0.8655	0.8651	0.8650	0.9970	0.9511	0.8838	0.8947	0.7689	0.7940	23	7
ovo_lin	0.8619	0.8662	0.8619	0.8632	0.9859	0.9251	0.8697	0.8704	0.8022	0.7861	4	2
ovr_lin	0.8678	0.8682	0.8678	0.8678	0.9918	0.9381	0.8863	0.9008	0.7765	0.8040	12	5
ens_svm	0.8655	0.8670	0.8655	0.8659	0.9888	0.9055	0.8825	0.8866	0.7800	0.7968	8	2
svm_lin_base	0.8560	0.8587	0.8560	0.8567	0.9918	0.9349	0.8685	0.8563	0.7905	0.7739	13	11
svm_key_lin	0.7645	0.7654	0.7645	0.7636	0.9703	0.8958	0.7050	0.7287	0.6536	0.6597	83	42

Hold Out Data

model_name	Holdout acc	Holdout precision	holdout tpr/recall	holdout F1 Score	sx acc	dr acc	ed acc	sp acc	mk acc	os acc	fp sx	fp dr
svm_rbf	0.8682	0.8693	0.8682	0.8685	0.9940	0.9441	0.8634	0.8605	0.7942	0.7979	6	2
svm_lin	0.8649	0.8665	0.8649	0.8652	0.9925	0.9650	0.8711	0.8372	0.8009	0.7795	4	7
sgd_lin	0.8653	0.8650	0.8653	0.8649	0.9940	0.9720	0.8686	0.8915	0.7627	0.7979	10	6
ovo_lin	0.8660	0.8687	0.8660	0.8669	0.9895	0.9301	0.8711	0.8411	0.8043	0.7948	1	1
ovr_lin	0.8644	0.8644	0.8644	0.8642	0.9925	0.9371	0.8763	0.8915	0.7649	0.8009	4	4
ens_svm	0.8670	0.8678	0.8670	0.8671	0.9925	0.9091	0.8737	0.8643	0.7739	0.8070	3	1
svm_lin_base	0.8598	0.8609	0.8598	0.8600	0.9940	0.9510	0.8505	0.8798	0.7897	0.7703	12	9
svm_key_lin	0.7720	0.7715	0.7720	0.7705	0.9805	0.9091	0.7242	0.6977	0.6659	0.6432	30	24

Figure 5: Model Comparison Table

The comparison of each model based on test and hold out accuracy is shown below in Figure 6: Model Accuracy Comparison. The best accuracy of all models on the test data is the SVM One-Versus-Rest model with nearly a 1% improvement in accuracy when compared to the baseline model. However, the best accuracy on the hold out data is the SVM non-linear "rbf kernel" model, again with nearly a 1% improvement compared to the baseline model. The poorest performing model on both the test data and the hold out data is the SVM keyword model with the baseline model in the second to last position.

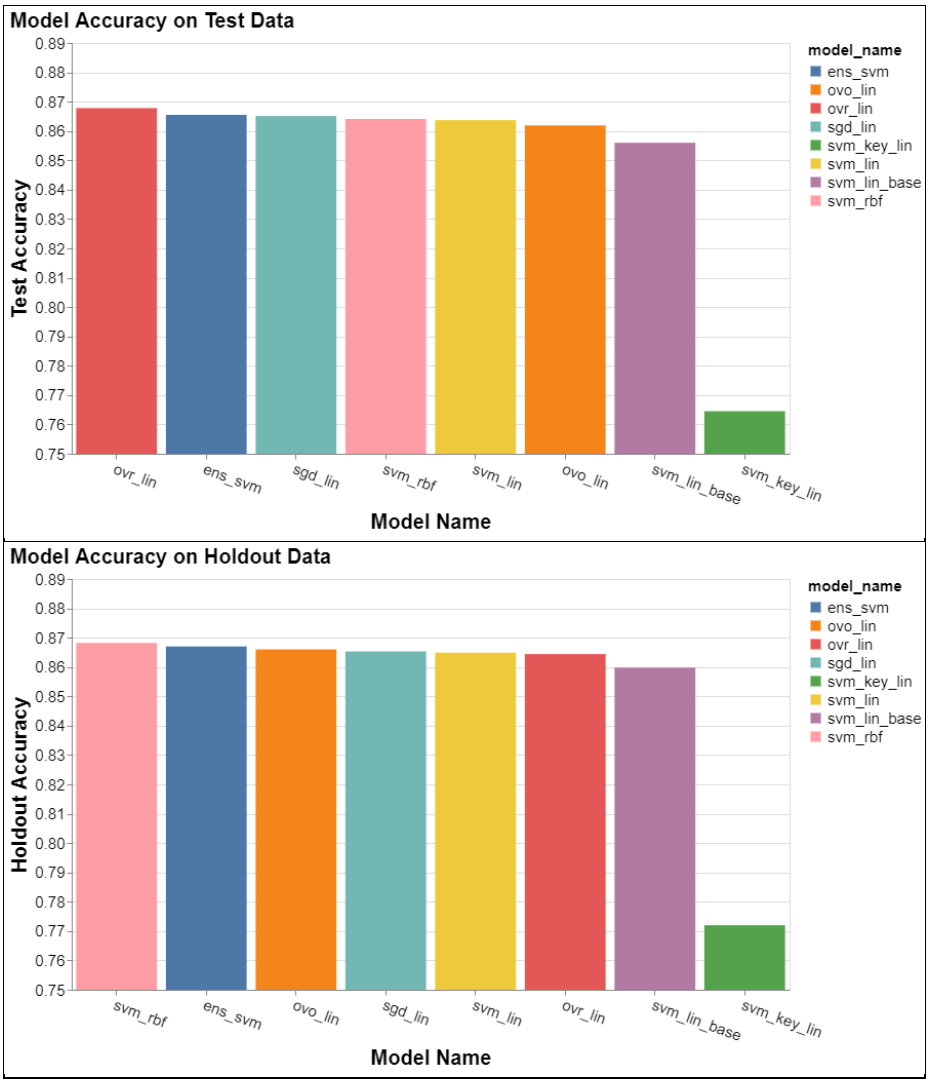


Figure 6: Model Accuracy Comparison

When comparing accuracies across each model on the pornography and drug categories, the best performing model is the SGD Linear model as can be seen below in Figure 7: Scatter Plot of Pornography and Drug Accuracy by Model. However, accuracy alone cannot be the final determining factor when judging the best model as the objective is improving overall accuracy and minimizing the number of misclassifications of web pages as undesirable categories such as pornography and drugs.

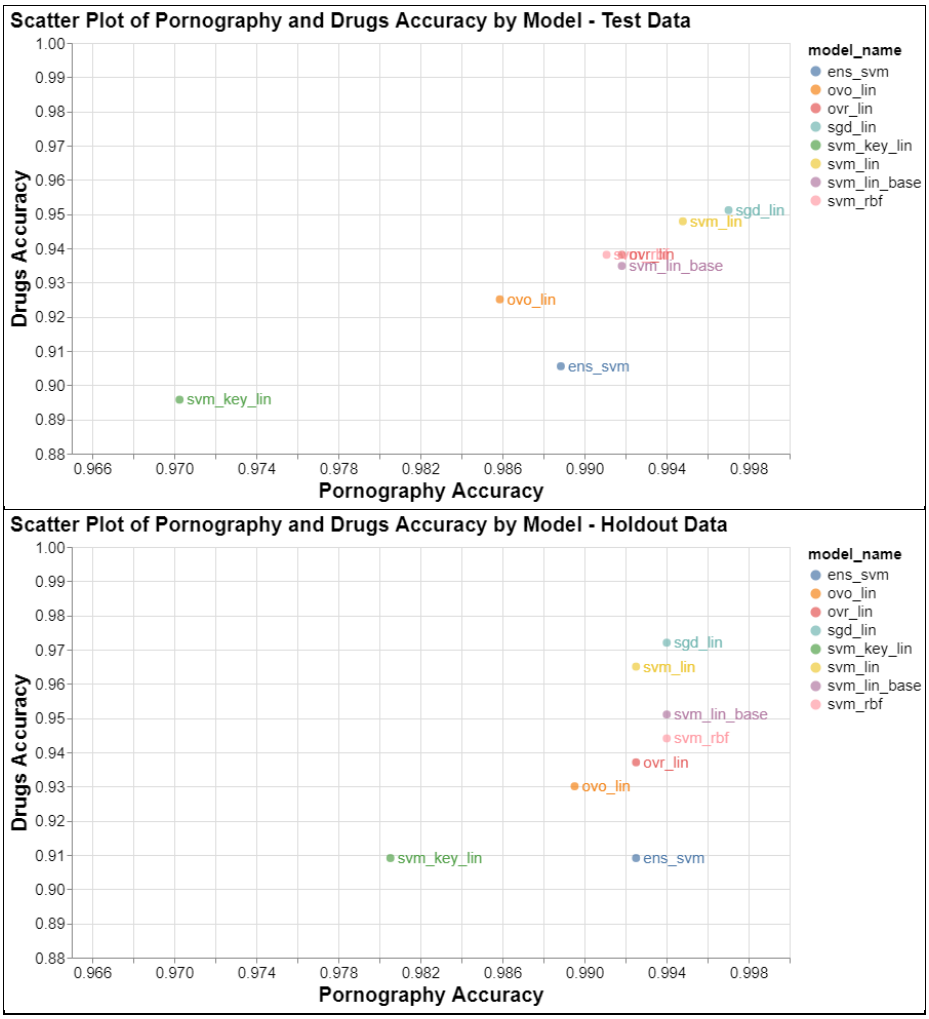


Figure 7: Scatter Plot of Pornography and Drug Accuracy by Model

When examining the models based on how often they wrongly classify a webpage as pornography or drugs, the best performing model to minimize such misclassifications is the One-versus-One SVM model. Looking at the test data, the One-versus-One model only misclassified a web page as pornography four times and only misclassified a web page as drugs two times in the test dataset which is made up of 8,386 observations. When examining the model performance on the hold out data with 4,188 observations, the model only misclassifies one web page as pornography and one web page as drugs. See Figure 8: Comparison of Misclassification of sites as Pornography and Drugs.

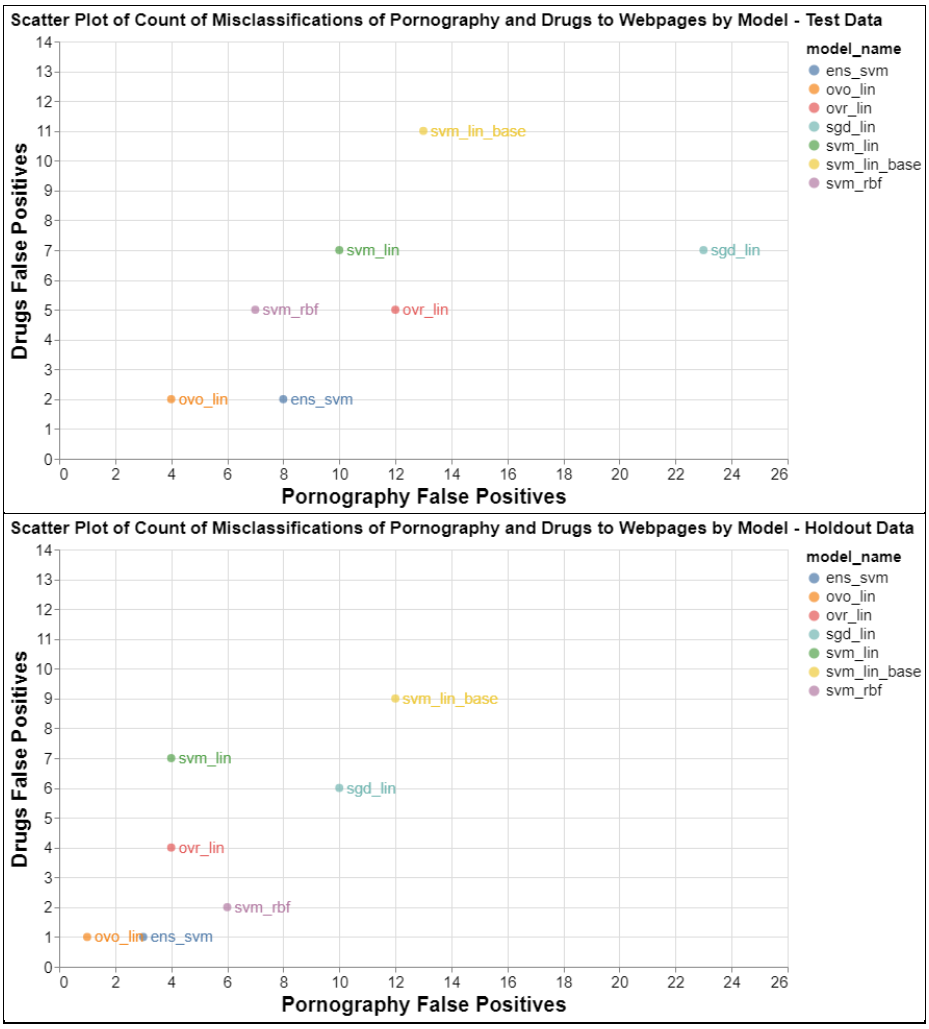


Figure 8: Comparison of Misclassification of sites as Pornography and Drugs

The results shared above support the hypothesis that by first extracting noun phrases, then performing lemmatization along with contraction expansion, removing special characters, removing extra white space, lower casing and removing stop words, the models will have more relevant terms from which to use for feature engineering, thus yielding better accuracy. Except for the model based on keywords only (svm_key_lin), all models showed improved accuracy above and beyond the baseline model (svm_lin_base). Additionally, all models except the keyword-based model and the SGD model (sdg_lin) reduced false positive categorization on pornography and drug categories.

5 Discussion

The table below in Figure 9: Differences between Candidate Models and Baseline Model shows the differences (candidate model minus baseline model) between each candidate model and the baseline model. Overall, though the SVM One-versus-One model does not have the overall best accuracy, it has the best performance when measuring overall accuracy and precision improvement over the baseline model while significantly reducing the number of instances a web page is misclassified as pornography or drugs. The improvement in accuracy on the test data yielded by the One-versus-One model is over .5% as compared to the baseline model with a .75% improvement in precision. On the hold out data, the SVM One-versus-One model overall accuracy is again over .5% compared to the baseline model with a .78% improvement in precision with a significant improvement in reducing the number of false positives on web pages to the undesirable categories.

Test Data

model_name	accuracy diff	precision diff	tpr diff	f1 Score diff	sx acc diff	dr acc diff	ed acc diff	sp acc diff	mk acc diff	os acc diff	fp sx diff	fp dr diff
svm_rbf	0.0081	0.0077	0.0081	0.0080	-0.0007	0.0033	0.0115	0.0223	0.0064	0.0115	-6	-6
svm_lin	0.0077	0.0081	0.0077	0.0078	0.0030	0.0130	0.0038	0.0121	0.0111	0.0093	-3	-4
sgd_lin	0.0091	0.0068	0.0091	0.0083	0.0052	0.0163	0.0153	0.0385	-0.0217	0.0201	10	-4
ovo_lin	0.0059	0.0075	0.0059	0.0065	-0.0059	-0.0098	0.0013	0.0142	0.0117	0.0122	-9	-9
ovr_lin	0.0118	0.0095	0.0118	0.0111	0.0000	0.0033	0.0179	0.0445	-0.0140	0.0302	-1	-6
ens_svm	0.0095	0.0083	0.0095	0.0092	-0.0030	-0.0293	0.0140	0.0304	-0.0105	0.0230	-5	-9
svm_key_lin	-0.0915	-0.0933	-0.0915	-0.0931	-0.0216	-0.0391	-0.1635	-0.1275	-0.1369	-0.1141	70	31

Hold Out Data

model_name	accuracy diff	precision diff	tpr diff	f1 Score diff	sx acc diff	dr acc diff	ed acc diff	sp acc diff	mk acc diff	os acc diff	fp sx diff	fp dr diff
svm_rbf	0.0084	0.0084	0.0084	0.0085	0.0000	-0.0070	0.0129	-0.0194	0.0045	0.0276	-6	-7
svm_lin	0.0051	0.0056	0.0051	0.0052	-0.0015	0.0140	0.0206	-0.0426	0.0112	0.0092	-8	-2
sgd_lin	0.0055	0.0041	0.0055	0.0049	0.0000	0.0210	0.0180	0.0116	-0.0270	0.0276	-2	-3
ovo_lin	0.0062	0.0078	0.0062	0.0069	-0.0045	-0.0210	0.0206	-0.0388	0.0146	0.0245	-11	-8
ovr_lin	0.0046	0.0035	0.0046	0.0042	-0.0015	-0.0140	0.0258	0.0116	-0.0247	0.0306	-8	-5
ens_svm	0.0072	0.0069	0.0072	0.0071	-0.0015	-0.0420	0.0232	-0.0155	-0.0157	0.0368	-9	-8
svm_key_lin	-0.0878	-0.0894	-0.0878	-0.0895	-0.0135	-0.0420	######	-0.1822	-0.1237	-0.1271	18	15

Figure 9: Differences between Candidate Models and Baseline Model

The poorest performing candidate model is the SVM Keyword model. This model uses only keywords extracted from the web page content and the TFIDF vector uses only 2,000 terms as compared to the other candidate models which utilize the noun phrase extraction technique with a TFIDF vector that uses 4,100 terms. When using these models to run inference on new unseen data in production, the TFIDFVectorizer object created during training will be required to load into memory. There could be benefits in some applications to use a lower performing model when memory constraints require it. But, in this bench test, the keyword model is not recommended as is.

7 Ethics

This research utilizes data only from English language web pages and, as such, the results may not be applicable to all languages. In addition to data only being from English language web pages, there is the potential for bias if there is no adjustment for the variance across the content used by diverse web creators to create websites in the same categorizations. For example, two sports web pages could have vastly different content depending on the targeted audience of the designer(s). The majority language on ESPN web pages will most likely lead to correctly classifying it as a sports website, in contrast to the raunchier Barstool Sports website's content that could potentially be classified as pornography. It is important to review the misclassified predictions to analysis if any human bias is being imported into the algorithms.

Another ethical concern is the usage by governmental entities to use filtering technology to censor content that opposes or challenges the current structure. Rulers have been silencing their opposing' voices since humans have created hierarchical social structures. After a war, the winning side would burn and destroy history from the losing side. Nicolaus Copernicus was famously persecuted for his Heliocentric theory that suggested the Sun was the center of the universe and the Earth revolves around it. This evidence-based theory directly opposed the previously widely held belief, and those in power acted to make sure the Heliocentric theory was not accepted by the majority.

6 Conclusion

This paper illustrates that performing an additional step of noun phrase extraction and lemmatization improves overall accuracy against the baseline model which uses traditional text-preprocessing techniques. Additionally, the SVM One-Versus-One modeling technique is found to yield the best performance when considering the goal of minimizing misclassification of websites to undesirable categories such as pornography and drugs. The improvement in accuracy yielded by the One-versus-One model is over .5% as compared to the baseline model with a .75% improvement in precision. Thus, with these results and assessment, the hypothesis posed in this research paper is accepted.

This research highlights that it is important to understand the objective of the problem to be solved when comparing competing models. There are several competing models that show improved accuracy, precision, and even improved accuracy on individual classes. However, in this example, the best model is not the model with only improved overall accuracy and precision that is the winner, it's the model with improved accuracy and precision that minimizes the classification of web pages as the undesirable categories. In this case, the SVM One-Versus-One model shows improvements across the range of model metrics against the baseline on both the test and holdout data and is the recommended model.

For this research, the TFIDFVectorizer was applied using unigrams post the text preprocessing steps taken to first perform noun-phrase extraction and then subsequently performing lemmatization along with contraction expansion, removing special characters, removing extra white space, lower casing and removing stop words. For future research, applying a tokenization approach where the complete noun phrases are used as features instead of unigrams may prove to be beneficial in providing additional feature reduction as well as boosting model performance.

References

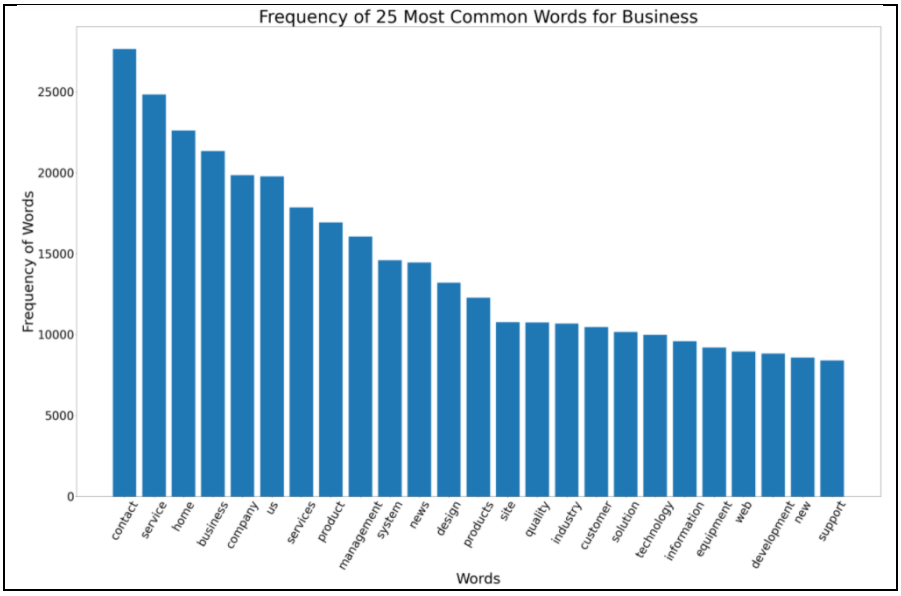
1. Winder, D. (2020, December 13). Microsoft Warns of Massive New 'Drive-By-Attack' Targeting Chrome, Edge, Firefox Users. Forbes. https://www.forbes.com/sites/daveywinder/2020/12/13/microsoft-warns-of-massive-new-drive-by-attack-targeting-chrome-edge-firefox-users/?sh=682f5ae31f79
2. Jenic, I. (2021, March 31). 10 Best Antivirus with Web Filtering [2021 Guide]. Windows Report. https://windowsreport.com/antivirus-website-blocker-web-filtering/
3. Albright, D. (2016, Jan 18). Which Websites are Most Likely to Infect You with Malware? https://www.makeuseof.com/tag/websites-likely-infect-malware/
4. FBI. Scams and Safety – Spoofing and Phishing. FBI.gov. https://www.fbi.gov/scams-and-safety/common-scams-and-crimes/spoofing-and-phishing
5. Turing, A. M. (2009). Computing machinery and intelligence. In Parsing the turing test (pp. 23-65). Springer, Dordrecht.
6. Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT press.
7. Hashemi, M. (2020). Web page classification: a survey of perspectives, gaps, and future directions. Multimedia Tools and Applications, 1-25.
8. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.
9. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152).
10. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.
11. Rung-Ching Chen, Chung-Hsun Hsieh (2006), "Web Page Classification based on a support Vector Machine using a weighted vote schema", Expert Systems with Applications, Vol. 31, issue 2, pp:427-435.
12. Altay, B., Dokeroglu, T., & Cosar, A. (2019). Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection. Soft Computing, 23(12), 4177-4191.
13. Thamrongrat, P., Preechaveerakul, L., & Wettayaprasit, W. (2009, August). A novel Voting Algorithm of multi-class SVM for web page classification. In 2009 2nd IEEE International Conference on Computer Science and Information Technology (pp. 327-331). IEEE.

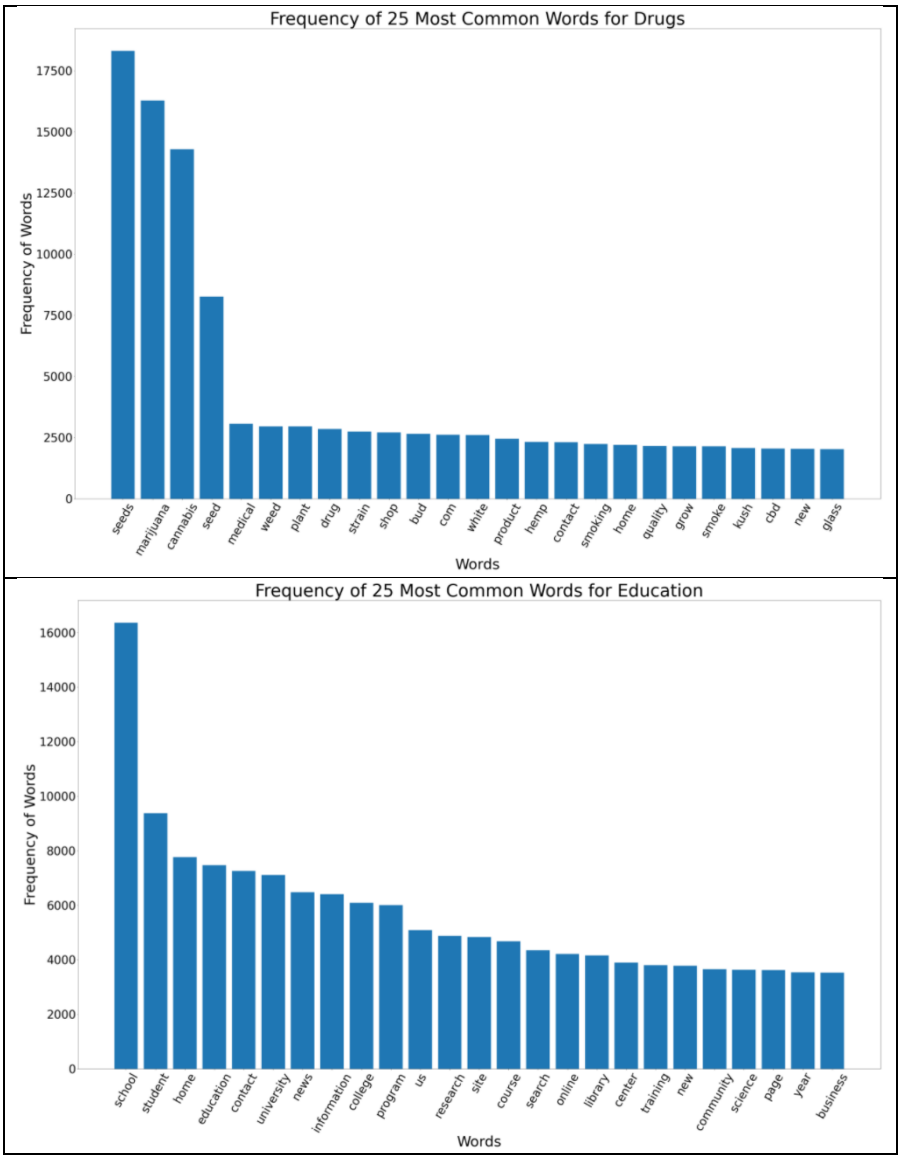
14. Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. Machine learning, 53(1), 23-69.
15. Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihok, G., & Den Hartog, D. N. (2018). Text classification for organizational researchers: A tutorial. Organizational research methods, 21(3), 766-799.
16. Rajalakshmi, R., & Aravindan, C. (2018). A Naive Bayes approach for URL classification with supervised feature selection and rejection framework. Computational Intelligence, 34(1), 363-396.
17. 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)
18. Verma, R., & Das, A. (2017, March). What's in a url: Fast feature extraction and malicious url detection. In Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics (pp. 55-63).
19. Bo, S., Qiurui, S., Zhong, C., & Zengmei, F. (2009, March). A study on automatic web pages categorization. In 2009 IEEE International Advance Computing Conference (pp. 1423-1427). IEEE.
20. Qiang, Q., & He, Q. (2006, February). A multiclass classification framework for document categorization. In International Workshop on Document Analysis Systems (pp. 474-483). Springer, Berlin, Heidelberg.
21. Qi, X., & Davison, B. D. (2009). Web page classification: Features and algorithms. ACM computing surveys (CSUR), 41(2), 1-31.
22. 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).
23. Patil, D., & Patil, J. (2018). Feature-based malicious url and attack type detection using multi-class classification. The ISC International Journal of Information Security, 10(2), 141-162.
24. Crammer, K., Dredze, M., & Kulesza, A. (2009, August). Multi-class confidence weighted algorithms. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (pp. 496-504).
25. Pal, K., & Patel, B. V. (2020). Multi-Class Document Classification: Effective and Systematized Method to Categorize Documents. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 7(7), 118-123.
26. Amin, S., Neumann, G., Dunfield, K., Vechkaeva, A., Chapman, K. A., & Wixted, M. K. (2019). MLT-DFKI at CLEF eHealth 2019: Multi-label Classification of ICD-10 Codes with BERT. In CLEF (Working Notes).
27. Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). Perceiver: General perception with iterative attention. arXiv preprint arXiv:2103.03206.
28. Sahoo, D., Liu, C., & Hoi, S. C. (2017). Malicious URL detection using machine learning: A survey. arXiv preprint arXiv:1701.07179.
29. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.
30. Singhal, S., Hegde, B., Karmalkar, P., Muhith, J., & Gurulingappa, H. (2021). Weakly Supervised Learning for Categorization of Medical Inquiries for Customer Service Effectiveness. Frontiers in research metrics and analytics, 6, 683400. https://doi.org/10.3389/frma.2021.683400

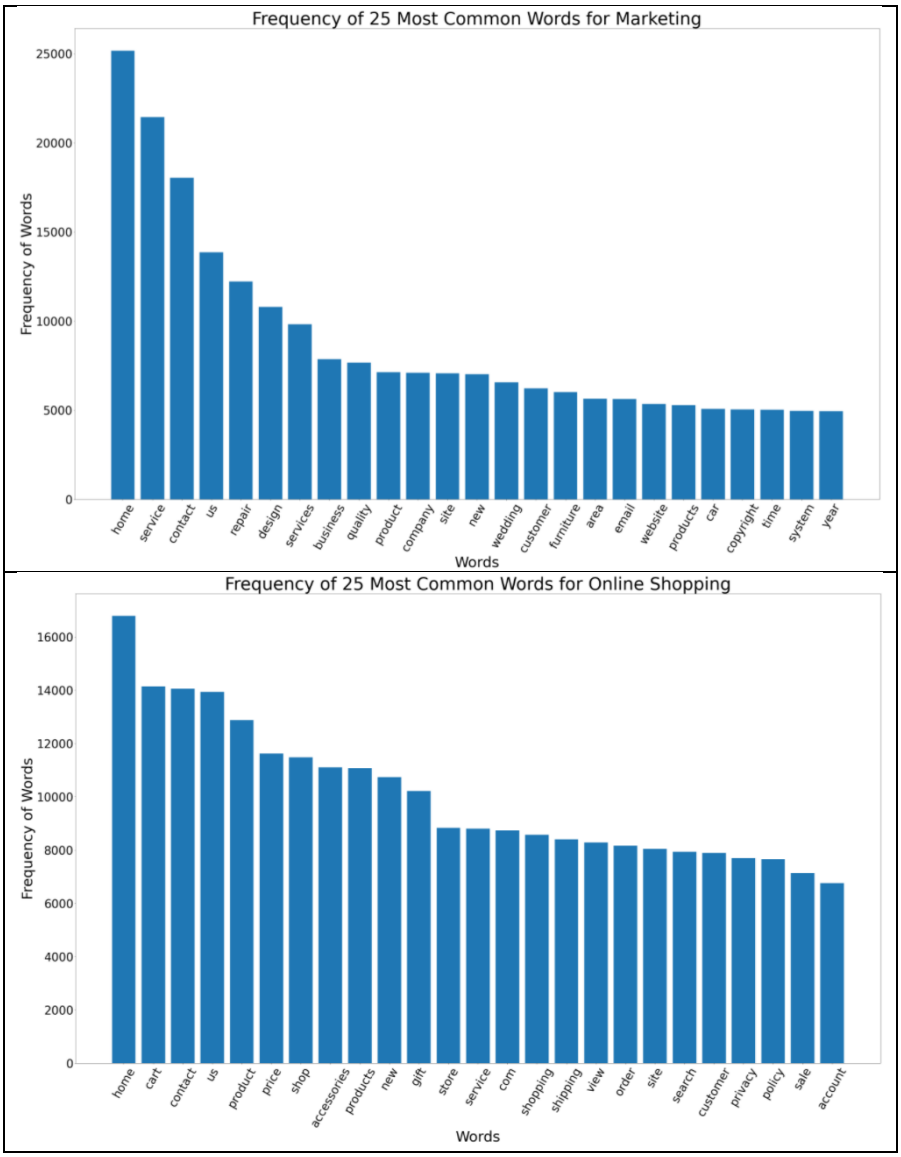
31. Buber, E., & Diri, B. (2019). Web Page Classification Using RNN. Procedia Computer Science, 154, 62-72. Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic
32. Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. NAACL.
33. Campos, R., Mangaravite, V., Pasquali, A., Jatowt, A., Jorge, A., Nunes, C. and Jatowt, A. (2020). YAKE! Keyword Extraction from Single Documents using Multiple Local Features. In Information Sciences Journal. Elsevier, Vol 509, pp 257-289.
34. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.
35. Pal, K., & Patel, B.V. (2020). Multi - Class Document Classification: Effective and Systematized Method to Categorize Documents. International journal of scientific research in science, engineering and technology, 118-123.
36. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.
37. 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.

Appendix

Top 25 occurring terms in each category after performing noun phrase extraction on web page content:




Frequency of 25 Most Common Words for Drugs
Frequency of Words
17500
15000
12500
10000
7500
5000
2500
0
seeds
marijuana
cannabis
seed
medical
weed
plant
drug
strain
shop
bud
com
white
product
hemp
contact
smoking
home
quality
grow
smoke
kush
cbd
new
glass
Words
Frequency of 25 Most Common Words for Education
Frequency of Words
16000
14000
12000
10000
8000
6000
4000
2000
0
school
student
home
education
contact
university
news
information
college
program
us
research
site
course
search
online
library
center
training
new
community
science
page
year
business
Words


Frequency of 25 Most Common Words for Marketing
Frequency of Words
Words
home
service
contact
us
repair
design
services
business
quality
product
company
site
new
wedding
customer
furniture
area
email
website
products
car
copyright
time
system
year
Frequency of 25 Most Common Words for Online Shopping
Frequency of Words
Words
home
cart
contact
us
product
price
shop
accessories
products
new
gift
store
service
com
shopping
shipping
view
order
site
search
customer
privacy
policy
sale
account

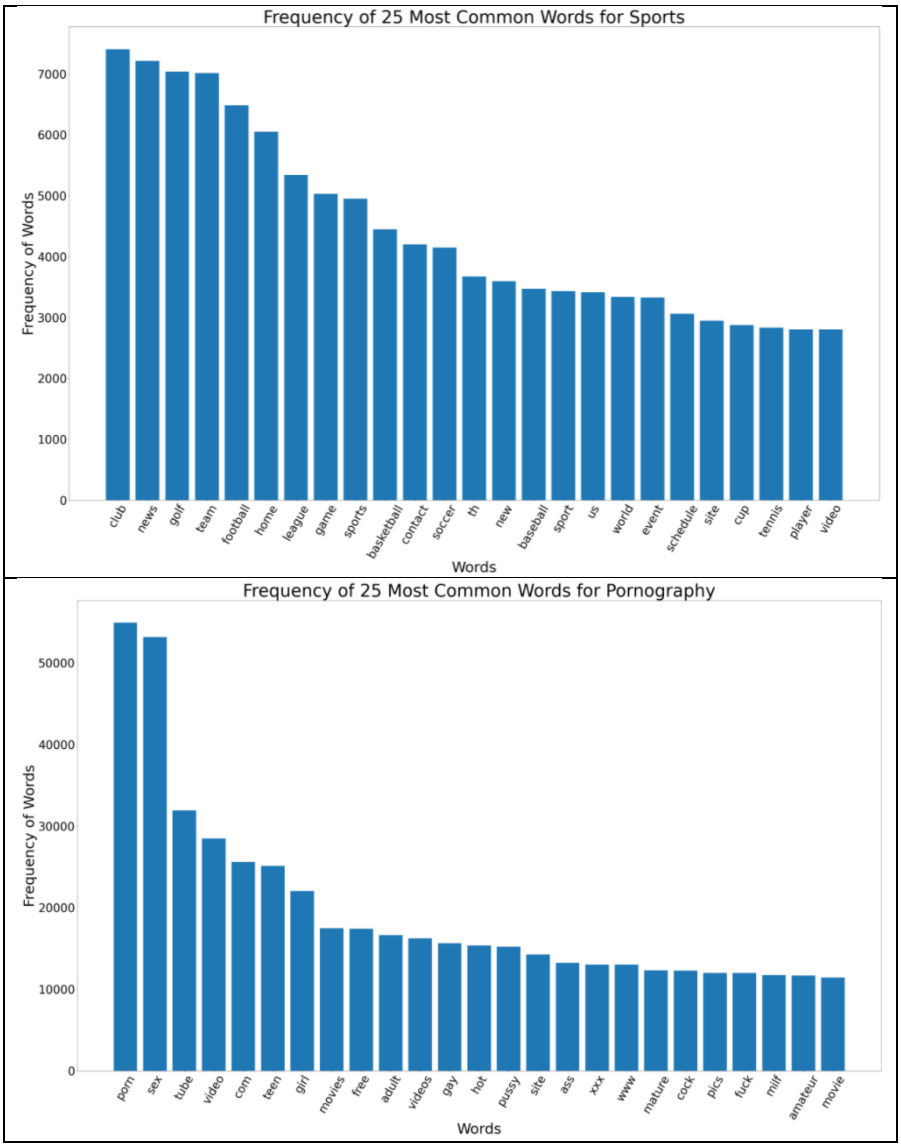


Table 4: Top 25 Occurring Terms by Category