DOI: https://doi.org/10.31341/jios.42.1.5 UDC 004.738.52
Open Access Original Scientific Paper

The Use of Support Vector Machines When Designing a User-Defined Niche Search Engine

Maria Jakovljevic *jakovm@unisa.ac.za*
School of Computing
University of South Africa, South Africa

Howard Sommerfeld *howiesommerfeld@gmail.com*
Platform45, Technical Consultant,
Johannesburg, South Africa

Alfred Coleman *colema@unisa.ac.za*
School of Computing
University of South Africa, South Africa

Abstract

This study presents the construction of a niche-search engine, whose search topic domain is to be user-defined. The specific focus of this study is the investigation of the role that a Support Vector Machine plays when classifying textual data from web pages. Furthermore, the aim is to establish whether this niche-search engine can return results that are more relevant to a user than when compared to those returned by a commercial search engine. Through the conduction of various experiments across a number of appropriate datasets, the suitability of the SVM to classify webpages has been proven to meet the needs of a niche-search engine. A subset of the most useful webpage-specific features has been discovered, with the best performing feature being a webpages' Text & Title component. The user defined niche-search engine was successfully designed and an experiment showed that it returned more relevant results than a commercial search engine.

Keywords: Support vector machine, search engine, text classification and processing, information retrieval

1. Introduction

This study relates to the formation of a niche-search engine, whose search topic domain is to be user-defined. Current commercial search engine needs an improvement in terms of search capabilities. There is a necessity to determine if a user-defined niche search engine through the use of support vector machines could return results which are more relevant than those returned by a commercial search engine.

Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane [3], [13]. The Support Vector Machine (SVM) [26] has been shown to be one of the best text classification solutions available today [2], [12], [10]. Web pages in their raw form consist of HTML code, which is essentially textual data. The amount of elements on web pages that can be used in the classification process is large, since textual data has a high-dimensional feature space and web pages have other useful elements that can indicate what the information on the page pertains to.

The authors [10], [9], [11 highlight the need for text classification, describing how the rate of growth of information on the WWW is increasing and hence, so there is the need for an efficient text classification solution. Research findings on text classification propose new methods of classification algorithms, while others propose alternative routes [1], [9], [10], [11]. The main advantage to be gained from analyzing these research findings is the highlighting of various short-falls of already existing techniques. The flaws detected can provide the necessary room for improvement that this study is aiming to achieve.

There are a number of reasons behind the rationale for this research, both theoretical and practical. With the current explosion of digital data being experienced world-round, the use of search engines plays a pivotal role in making sense of such abundant information [10], [17], [10, 11], [18], [24]. Returning only relevant information to a user who is looking to gain insight into a specific topic can achieve such sense making.

As a secondary motive, it is interesting to find out more information about the inner workings of search engines that are the fore runners in cutting edge Internet technology. In particular, it will be interesting to observe the engine's performance on a query, which pertains to, an irregular or extremely specific domain that has a very limited scope.

Based on the above discussion the first and primary aim of this study is to determine whether the use of this niche-search engine can return to the user results that are more relevant than those returned by any commercial search engine, when tested on the same query. The two-fold secondary aim of this research is to firstly

a. Ascertain the suitability of the SVM for the classification of web pages in the context of a niche search engine; and secondly to
b. Determine which subset of features contained on a web page should be extracted in order to gain a high accuracy and relevance capability of the SVM for the particular classification task at hand.

Regarding two-fold secondary aim, if an optimal subset of web page specific features can be formulated, such that the levels of accuracy and relevance of the SVM's classifications are increased significantly when compared to already existing feature selection techniques, then these aims will have been achieved. The capabilities of the SVM in terms of providing a suitable solution for classification in the context of a niche search engine will also be determined.

What follows below are a number of critical research questions which were derived in order to gain further insight into the problem at hand:

RQ1. What are the different types of features present on web pages?

RQ2. What are webpage-specific features which can be used to assist the classification task of the SVM?

RQ3. Is there a particular feature, or combination of features, which will in fact improve the standard performance of the SVM?

RQ4. What makes the particular use of an SVM suitable in the context of a niche search engine?

Based on the above discussion the following primary hypothesis is derived:

H1: The use of the user-defined search engine will return results that are more relevant than those returned by any commercial search engine.

If the relevancy of searches increased beyond current commercial leaders' levels, the process of using a search engine can be made much more effective. The next section presents the theoretical background.

2. The Theoretical Framework for a User-Defined Niche Search

2.1. Formal Definition of the SVM and Text Classification

The theoretical background is based on research findings reflecting SVMs and text classifications [2], [10] [9], [10], [11], [1], [22], [24]. The main goal of text classification is to classify documents into a "fixed number of predefined categories" [10]. This can be done with the aid of machine learning from examples. Considering that data will need to be trained before the engine is able to work properly, this is extremely relevant to the research at hand. The main reasons behind such a great suitability of the SVM to this problem stem from the fact that text documents have properties of high dimensional feature spaces, few irrelevant features and sparse instance vectors [10]. These properties make text documents good candidates for SVM inputs.

The support vector machine is categorized into a class of algorithms called kernel methods [10] which are distinguished by their dependency on data only through dot products. A particularly appropriate consequence of this notion is the ability of the SVM to "compute a dot product in some possibly high dimensional feature space" [2, p. 1], as is the case regarding a textual document. The dot product between two vectors is defined as

$$\mathrm{w}^{\mathrm{T}}\mathrm{x} = \sum \mathrm{w}_{i} \;\; \mathrm{x}_{i}. \tag{1}$$

SVMs also fall into a category of machines dubbed "*linear classifiers"* [2, p.2], and in the particular case of this study, it is an example of a "two-class" classifier. This means that the learning problems the machine deals with categorizes input into one of only two classes, most commonly a positive class (usually labelled by a '+1') or a negative class ('-1'). Since this study deals with the classification of web pages, the vectors used by the SVM are the actual documents or web pages to be classified into a group, and are denoted by **x**. Following this description, each word of a document will be a component of the vector **x**, denoted by $x_{i.}$ On a larger scale, $\mathbf{x}_i$ denotes the vector

$$\text{in } {}_{1}^{i=} \text{ dataset } \{(\mathbf{x}\ i,\ yi)\}^{n} \qquad (2)$$

where y_i *is* the label (+1, -1 or 0 if it is unclassifiable) associated with $\mathbf{x}_i$ [2, p.2]. A linear classifier is formed around a vital function called *discriminant function*, which is of the form

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b, \qquad (3)$$

where **w** is called a weight vector and *b* is a bias unit. When the bias is set to 0, "the set of points **x** such that

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} = 0 \qquad (4)$$

are all points that are perpendicular to **w** and go through the origin" [2, p. 2]. This is in general form a *hyper plane*, defined as

$$\{f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathrm{b} = 0\} \qquad (5)$$

that divides the space into two, one part for each class group, while the sign of the resulting discriminant function $f(\mathbf{x})$, per vector **x**, determines into which of the two parts the document is categorized into [2]. This boundary formed by the hyper plane that separates these two groups is named the *decision boundary* that divides the hype plain into two sets (see Figure 1)

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathrm{b} = 0 \qquad (6)$$

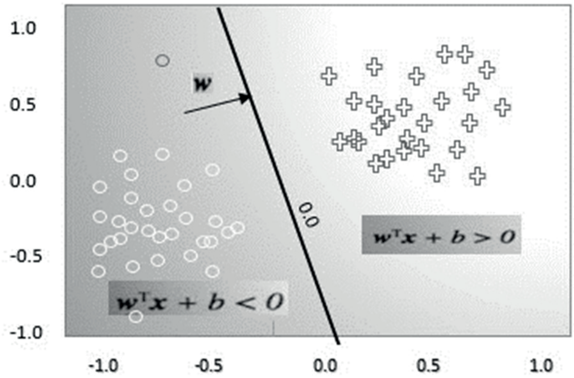

Figure 1. The decision boundary (adapted from [2], p 225)

Thus far, only linear classifiers have been explored; as this will be enough to suffice the needs of this study (see Figure 2).

The circled data points are the support vectors, the examples that are closest to the decision boundary that determine the margin with which the two classes are separated The operation of the SVM algorithm is based on finding the hyper plane that gives the largest minimum distance to the training examples. This distance receives the important name of *margin* within SVM's theory (see Figure 2).

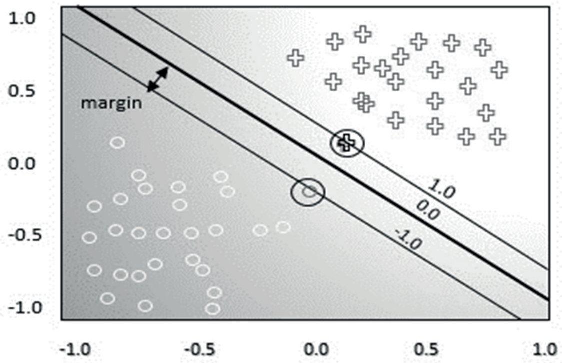

Figure 2. A linear SVM (adapted from [2], p 228)

Data that is deemed *linearly separable* is defined to mean that there exists "a linear decision boundary that separates positive from negative examples" [2, p. 4]. For the use of this study only the use of data linearly separable will be explored. Due to the nature of linearly separable data, there needs to be some point that actually divides the two classes. This is the role of a margin in the hyper plane.

Let **x**+ (**x**-) denote the closest point to the hyper plane among the positive (negative) examples, then the margin is defined to be the distance from **x**+ to the decision boundary. The point (s) **x**+ (**x**-) is defined to be the support vector, which is the point closest to the decision boundary (see Figure 3). Hence the fundamental function of the SVM is to derive a "discriminant function that maximizes the geometric margin" [2, p.6], in order to ensure accurate classification. Examples falling in-between the margins are called slack variables and represent instances of misclassified examples. A special constant term *C,* called *the soft-margin constant* is used to mathematically maximize the size of the margins on the hyper plane while simultaneously minimizing the amount of slack.

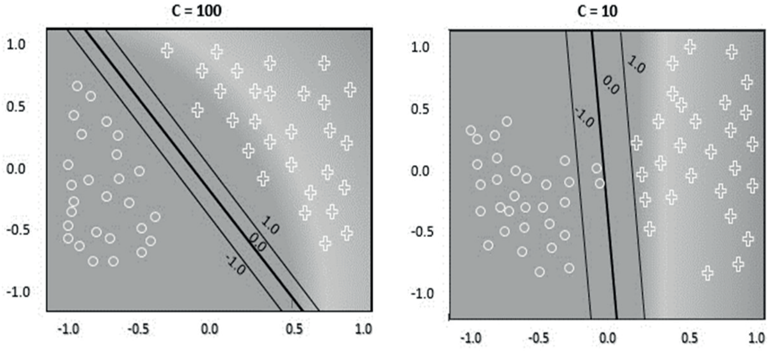

Figure 3. The effect of the soft-margin constant, C, on the decision boundary (adapted from [2], p 231)

A smaller value of C (right) allows to ignore points close to the boundary, and increases the margin. The decision boundary between negative examples (circles) and positive examples (crosses) is shown as a thick line. The lighter lines are on the margin (discriminant value equal to -1 or +1). The grayscale level represents the value of the discriminant function, dark for low values and a light shade for high values.

By the above description of the machines workings, it can be said that when training an SVM, the margin of the hyper plane should be maximized. In the case of linearly separable data, it can hence be said that the accuracy of a classifier such as an SVM is largely dependent on this soft-margin constant, *C.* Although there are some heavily mathematical mechanisms at play behind the scenes of the SVM's classification technique, the above high-level description of how the machine operates should give the reader a better insight into how it classifies documents into different classes.

2.2. Established Techniques to Enhance SVM Performance

2.2.1 Aggressive Feature Selection

The researchers [9] provide a description of one of the methods used to improve the performance of the SVM using a technique called aggressive feature selection. Using the argument of [10], the authors agree that the high feature space of text documents make the use of SVMs for classification suitable. The authors, however, disagree with claims that state all features are relevant and need to be used in classification [10] and that feature selection decreases the accuracy of SVM classification [1].

It is important to note however, that the documents that distinctly benefit from this type of feature selection are those that contain many redundant features. Such documents are described to be those which can be "told apart using a small numbers of words" [9] making the large number of other, common, words redundant. In order to achieve decent levels of SVM performance when classifying documents of this specific, redundant, nature, aggressive feature selection is required, which was not previously thought to be the case. The authors also accomplished a measure that predicts whether aggressive feature selection would be beneficial to a specific dataset or not.

2.2.2 Ambiguity Measure

Authors [11] explained the use of techniques called "Ambiguity Measure algorithm" feature selection method. The authors also agree that one of the best classifiers is an SVM; however they note, due to the works of [28], that the time is taken to train a model for SVM is longer than for any other text classification procedure.

The ambiguity measure feature selection method makes use of only of the most unambiguous features of a document, where unambiguous features are understood to be those features which, when present in a document, "indicate a high degree of confidence that the document belongs to one specific category" [11, p. 916].The authors found that when compared to current state of the art feature selection algorithms, the use of the ambiguity measure algorithm "performs statistically significantly better" [11, p. 920].

Further, it is shown that when no feature selection is used at all, a technique which has been claimed to be suitable for text classification in [10], [19], [26] and [28], the use of the ambiguity measure algorithm reduces the time to train data for an SVM "by more than 50%" [11,p.916]. More importantly when using this particular algorithm, the authors showed that the overall effectiveness of an SVM to classify text documents correctly did not decrease, at least up until a certain point.

2.3. Web Specific Techniques

The techniques which follow apply to a more focused subset of text classification problems illuminating the inner workings of the SVM as a tool for text categorization [15], [16]. These problems are specific to the needs of this particular study and so are included, as they deal with the areas of web page classification and making use of search engine queries to aid in the classification process.

2.3.1 Using Click-through Data

The author Joachims [2002] highlights the fact that training data for an SVM can be both expensive and difficult. The author proposes the use of Click-through data attained from commercial search engines in order to reduce the above-mentioned expenses behind data training. Click-through data is explained to be a log of a user's query, the ranking given to each returned result, and a list of links the user clicked after receiving the results.

A few notable features of Click-through data explain that it is freely available, in abundance and is also easily accessible through a commercial search engine. The author conducted experiments to see if a retrieval algorithm could be developed, with the aid of information from Click-through data, which would improve the relevancy of documents classified using an SVM. The results of the experiments showed that the "Ranking SVM can indeed learn a retrieval function" [12] which is based on Click-through data and also that this "learned retrieval function" did show improvements regarding relevancy when compared to retrieval functions used by commercial search engines [12, Section 5.3].

2.3.2 Exploiting Properties Specific to Webpages

Sun et al. [24] provide an interesting and very appropriate perspective for this instance of text classification using an SVM, in the particular case when web pages are concerned. These researchers aimed to determine whether using other non-text components found specifically in web pages, such as HTML tags and Hyperlinks, in the SVM classification process will increase the accuracy of returning more relevant results.

Aside from only the text components, the authors proposed that the use of "titles, anchor words and a combination of these elements" would help increase the classification abilities of a standard SVM. There are arguments both for and against the use of anchor words to aid SVM classification. Using web pages classified by only their textual components, the authors formed a baseline for their experimental analysis. The results showed that the combination of the "text, title and anchor word" elements resulted in the most efficient text classification method for web pages, with returned results having high levels of relevancy as well.

2.4. Alternative Approaches

2.4.1 Transductive Approach to Using an SVM

Deviating from the known techniques the author [11] introduces the notion of a Transductive SVM (TSVM). Unlike investigating the outcomes of using different feature selection algorithms, the TSVM is a re-designed SVM, but we do see that "TSVMs inherit most properties of SVMs" [5, Section 4], so there are some expected similarities; hence the fact that a TSVM will be well suited to text-classification problems is implicit. The regular SVM is based on inductive techniques, meaning it "induce[s] a general decision function for a learning task" [11], Abstract], while the "Transductive" [26] approach considers a particular test set of textual data to be classified and tries to "minimize misclassification of just those particular examples" [11]. According to experiments performed by Joachim [11] it shows that compared to inductive approaches, substantial improvements are gained using the TSVM, especially for small training sets.

According to experiments performed by Joachims [11] findings show that substantial improvements are gained using the TSVM, especially for small training sets. A new algorithm is both developed and empirically tested to show that using TSVMs leads to "improvements over the currently best performing method" [11] of text classification.

3. Research Methodology

3.1. Experimental Research Design

The study uses the experimental design and in particular exploratory experimentation [5], [7], [20]. An experiment must test hypothesis that is both testable and falsifiable; controlled (must have one testing variable per an experiment); reproducible (an experiment must be reproducible, and results repeatable) [25]. The study caters for factors in the hypothesis testing such as observability and measurability and determines average percentages of searching results. The exploratory experimentation applied in this study will contribute to the advancement of text classification and the field in general [5], [20].

3.2. Analysis of Data

The researcher made precise and detailed observations of experimental training and manipulation of data sets. Descriptive statistics was used that implies a simple quantitative summary in the form of frequencies, percentages, and averages, graphs of data sets that have been collected and analyzed [20] [5] [25].

Furthermore, the method of analysis and synthesis will be used, in order to explain the scientific literature and results of data manipulation through the synthesis of simple judgements and data in a more complex form. In addition, the method of compilation and logical methods will be applied, specifying scientific data processing, drawing conclusions regarding SVM and text classifications [5], [14].

3.3. Validity and the Reliability

To address the issue of reliability this study tested sufficiency - if there was enough of the data to support the findings; competency - if the data was both valid and reliable; relevancy - if it had a logical, sensible relationship to the finding it supports [20, pg. 1]. The researchers of this study were aware of uncontrolled variables (e.g. a difference in the operating systems and the data sets). The conditions (assumptions, controls, and variables) of the experiment were documented thoroughly so that the dataset can be re-created, to determine its validity. The researchers were addressing the established validation techniques in the field in order to measure the classification system that increased validity and reliability of the study. The SVM model was applied with the clear classification system and implementation procedures [21] [25].

3.4. Techniques to Evaluate and Measure the Classification System

There are a number of already existing techniques available to measure the classification system that was considered in this study: In order to validate the hypothesis, a method of testing the hypothesis is required. When evaluating a classification system, there are two main areas that need to be examined. These are:

- The degree of credibility of the classifier and
- The cost of making classifications, or more importantly, misclassifications

There are a number of already existing techniques available. To measure these classification criteria, a number of existing techniques were used as follows below:

	Positive	Negative
True	True Positive (TP)	True Negative (TN)
False	False Positive (FP)	False Negative (FN)

Figure 4. Confusion matrix for a binary classification problem [17, p. 220]

Confusion Matrix [17, p. 220]- Simple matrix, where the rows refer to the category that the classifier assigns a particular instance, and the columns refer to the category that an instance of a description belongs to; In the case of binary classification, which is the same as the case of text classification, there are only four cells (see Figure 4).

Cost function [17 p. 221] - European standards suggest the ratio of 100 false negatives for one false positive. For example, prefer to have a page classified as relevant when it is irrelevant, as opposed to classifying it as irrelevant when it is a relevant page.

Bernoulli process [17, p. 221] - A sequence of independent events whose outcome are considered either as success or as failure; this answers the question: "How close is the estimated level of accuracy to the actual accuracy?"

10-Fold cross validation technique [17, p. 221] - Solves the problem of not having enough test cases or examples.

Extreme Case - leave-one-out method [17, p. 222] - Use all except one example to train the classifier and then test the left out Example to see if it works.

Receiver operating characteristic (ROC) Curves [17, p. 222]- Plotting the True Positive rate (TPR) versus the False Positive rate (FPR); Indication of good results achieved when ROC curve is as far away from the diagonal of the plot as possible.

The primary data was collected using the evaluation techniques described above. Using testing dataset [21] populated with enough examples, the tests above can be run - thereafter, an analysis of the results will lead to validation of the hypothesis.

For the purposes of this study, the SVM library was used named *"SVM light"* (Joachims, 2008). In the next section, a high-level, overall description of what the implementation of "SVM light" as a whole is presented.

3.5. The SVM Library and Model Implementation

In order to have an insightful view of how the SVM was used within this study, as well as understand what is meant by the terms "labelled feature vector", "model" and "classification/training phase", the SVM model implementation is presented below.

During the initial training phase, the focused crawler [8] passed to the SVM the first 200 results returned from the user's query on a commercial search engine, along with a label indicating if the user found that page relevant or not. Each result was in the format of the page's HTML text without the HTML tags present, so it was essentially a string of words. The focused crawler [6] has three main components: a classifier which makes relevance judgments on pages crawled to decide on link expansion, a distiller which determines a measure of centrality of crawled pages to determine visit priorities, and a crawler with dynamically reconfigurable priority controls which is governed by the classifier and distiller [3].

The label is an integer representing a relevant page (+1) or an irrelevant page (-1). For each page, every unique word on that page was mapped to an integer value stored in a dictionary, which is referred to as a *feature*. Each feature is associated with a weighting, which is referred to as its value. This feature-value pairs, along with the label representing if the page is relevant or not, make up a *labelled feature vector*. After the user has marked 200 pages as relevant or irrelevant, the collection of these 200 labelled feature vectors was passed to the SVM for training. Using these labelled feature vectors, the SVM creates a model file specific to that user's query.

This model file is stored for each user and is used when a new page is encountered. The crawler passes a newly crawled webpage to the SVM, which loads the model file, convert the newly crawled page into a labelled feature vector, and test this new labelled feature vector against the model file, returning a decimal valued number which is either positive or negative. If the result of the classification returns a positive decimal, the page is indexed by the indexing solution and finally displayed to the user. If the result is negative, the webpage is not indexed and the crawler continues to crawl and test webpages against the model. This process encapsulates the essence of this specific implementation of the SVM, and its role in the overall solution.

3.5.1 The Procedure: SVM vs Commercial Search Engine

As the work flow of the overall solution expresses, the 200 pages were used for training the SVM, are the first 200 pages returned from a commercial search engine using the user-specified query. The user is then required to mark each of these pages as relevant or irrelevant, before the SVM classification of newly crawled websites begins to take place.

The number of pages found relevant by the user, can be used as a baseline for gaining how relevant the user finds a commercial search engine. Following the training phase, the user will then classify how many of the first 200 pages returned by the niche-search engine they found relevant or irrelevant.

To deem whether the niche-search engine returns more relevant results to a user than a commercial search engine does, these two numbers of relevant pages can be compared. If it is found that the user-defined search engine returns more relevant pages then when compared to the commercial search engine, then the hypothesis would have been confirmed. What follows below is an outline of the experiments which were conducted.

3.6. Experiments

3.6.1 Verifying the Suitability of Using the SVM for Classification

a. Experiment 1: Using Raw Text as a Baseline

This experiment attempted to verify the first part of the secondary research questions formed above. In order to do so, the SVM model was trained and tested using raw HTML text. The description “raw” pertains to the fact that the text contained stop words and the words in the text were not be stemmed. “Stop words” are considered to be some of the most common, short function words, such as “the”, “is”, “at”, “which” and “on”. The removal of these words does not deter meaning or importance from the understanding of a document.

However, stop word removal is beneficial and is used “prior to, or after, [the] processing of natural language data” [12]. A “stemmed” word is considered to be the form of a word “to which affixes can be attached” [23], for example the verb “wait” is the stem of the words “waits”, “waited” and “waiting”.

b. Experiment 2: Comparing Results - Raw Text to Stemming and Stop Word Removal

If encouraging classification accuracy can be achieved without their removal, a comparative experiment should be conducted whereby the accuracy of the SVM should be measured when these two elements (raw text to stemming and stop word) are indeed removed. Since the SVM will be processing a fewer number of words, which are also more relevant than other words such as stop words, the expectation is that the SVM's classification accuracy shall increase. Regardless of the outcome of both experiments, carrying them out will ensure the SVM's suitability for the purposes of this study, thereby confirming one part of the secondary research questions presented above.

c. Experiment 3: Finding the Best Sized Dataset for Training an SVM Model

The SVM model file is an essential element for accurate text classification. Hence it is important to find an appropriate sized dataset to use when training the model file such that the SVM's classification ability is as accurate as possible. The bigger the dataset the better, since a large dataset will contain many examples the machine can use to learn what a user considers as relevant or not [21]. However, there is a trade-off here between size, accuracy and usability. Since the user will need to train the SVM before it can begin classifying unseen documents, and in order to enhance the user's experience of using the niche-search engine, this training process should not go on for an unreasonable or indefinite amount of time.

An experiment should thus be conducted where various sizes of training datasets are used to create SVM model files. Using these different model files, the SVM's accuracy should be measured on a consistent set of verification data. The resulting optimal dataset size will therefore reflect a maximized classification accuracy of the SVM, whilst minimizing the amount of training examples required to be manually classified by a user.

d. Experiment 4: Testing Various Feature Sets for use by the SVM

The main elements of an HTML page consist of its *text, title, anchor-words and hyper-links*. There is also an element called *keywords*, however, this element was omitted. The reason for this is that on most web pages, it was found that this element contains largely spam, or inaccurate and unrelated advertising material, in order to be ranked highly, or associated with popular queries, by a commercial search engine.

In order to find the best feature, or combination of features, each of the elements' effect on the SVM's accuracy should be individually measured. The same measure of the SVM's accuracy should also be taken for various combinations of all of the above mentioned features. For each feature, or combination thereof, a number of executions should be taken and the final results averaged over the number of executions undergone. Once these results have been recorded for all possible combinations of elements, the feature, or set of features, which returned the highest accuracy results should be used by the SVM to classify web pages in the final solution.

e. Experiment 5: Accommodating for Bias Training Data

The search engine being designed in this study requires that the user train the SVM using 200 sample pages which are to initially be provided by a commercial search engine. The user is then required to mark each of these pages as relevant or irrelevant. It is owing to this fact that the event of there being a perfectly balanced split between positive and negative training examples is very unlikely.

Hence when measuring the most effective feature, or set of features, that is to be used by the SVM for training and classification purposes, this factor of bias should be taken into consideration. The experiment described above for finding the most effective feature or set of features, should hence be carried out on a balanced dataset, as well as on unbalanced datasets, which are both positively and negatively biased. The final solution, should be the most consistently best performing feature, or set of features, across all three datasets [21].

If there is no single conclusive solution across all three datasets, the results from both of the biased datasets should be taken ahead of the results using a perfectly balanced dataset, since this will be the closest representation of a realistic query scenario.All of the experiments listed above were executed and their results gathered. These results were useful in making design decisions for the SVM when incorporating it into the final niche-search engine.

3.6.2 About the Dataset

As in [24] the WebKb [4] dataset was used in the experiments described above. This dataset consists of over 4000 pages which have been manually classified into several categories. The categories used in the experiments were the pages related to “Course”, “Student”, “Project” and “Faculty” categories. This subset of the data was chosen, since these particular categories contained at least 800 pages of each respective topic. The experiments are to be conducted using 400 pages for training a model and 400 pages for testing that model, hence they require at least 800 individual pages pertaining to each topic.

It should be noted that the actual content of websites used in the training phase may affect the ability of the SVM to classify a page correctly. For example, a “Course” page’s title feature might prove to be more accurate than a title feature from a “Student” classified webpage. Hence the experiments should be conducted across all the categories individually and their results summed and averaged for the final values. The results are presented in the section that follows.

4.1 Verifying the Suitability of Using the SVM for Classification

4.1. Using Raw Text as a Baseline

This experiment was undergone using balanced, positively biased and negatively biased datasets. Only the text feature of web pages was used, and there was no removal of stop words within the text, nor was the words stemmed. Below is a graph which shows both the individual sets of results and the average accuracy achieved across all the datasets and categories (see Figure 5).

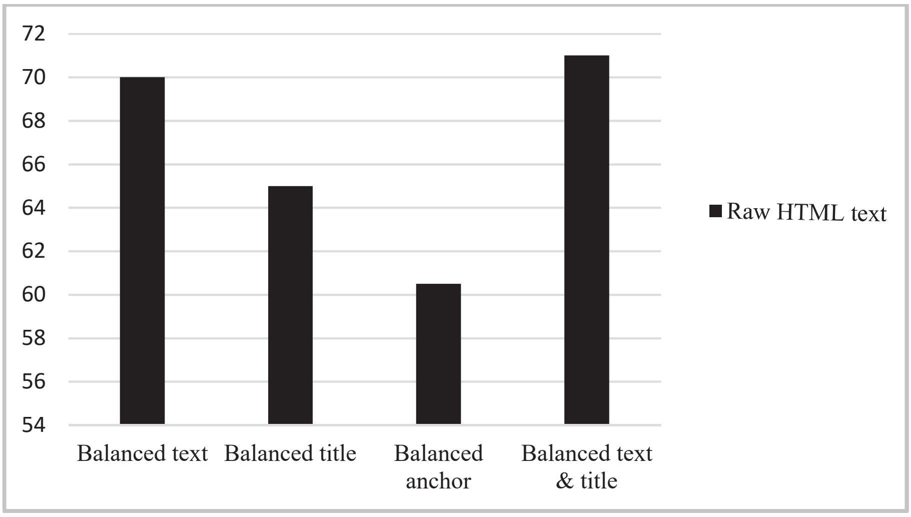

Figure 5. Verifying SVM suitability

These results show that the overall classification accuracy of the SVM using only raw HTML text, on average is 64, 23%. In an ideal scenario, when dealing with perfectly balanced training samples, this can reach up to 74, 44%.

These results are encouraging, since they are expected to improve with the removal of stop words and also having the words stemmed. Hence it can be seen that since the results lie in the range of 56-74%, which in the worst case is greater than 50%, and in some cases this margin is dramatically increased. Thus it can be said that the SVM is suitable for the task of text classification, in all realistic scenarios.

4.2. Comparing Results from Raw Text to Using Stemming and Stop Word

4.2.1 Removal

The results from using raw text as features for the SVM in the above experiment were encouraging. Hence the next experiment removed stop words from the text being classified and also stemmed each word of the text. The testing domain was broadened to include balanced, biased and unbiased data. The experiment was also conducted across all categories in the WebKb dataset [4], with the average for each feature set across all categories having been measured. The graph below illustrates the results.

Twelve out of eighteen comparisons showed an improvement when using text which has stop words removed from it and whose words were stemmed (see Figure 6). More importantly, the best performing feature sets, in particular Text & Title, benefited from this enhanced text across all variations of the balance of training data. Thus it is suggested that the HTML text to be classified by the niche-search engine should be stemmed and have its stop words removed to optimize the classification process.

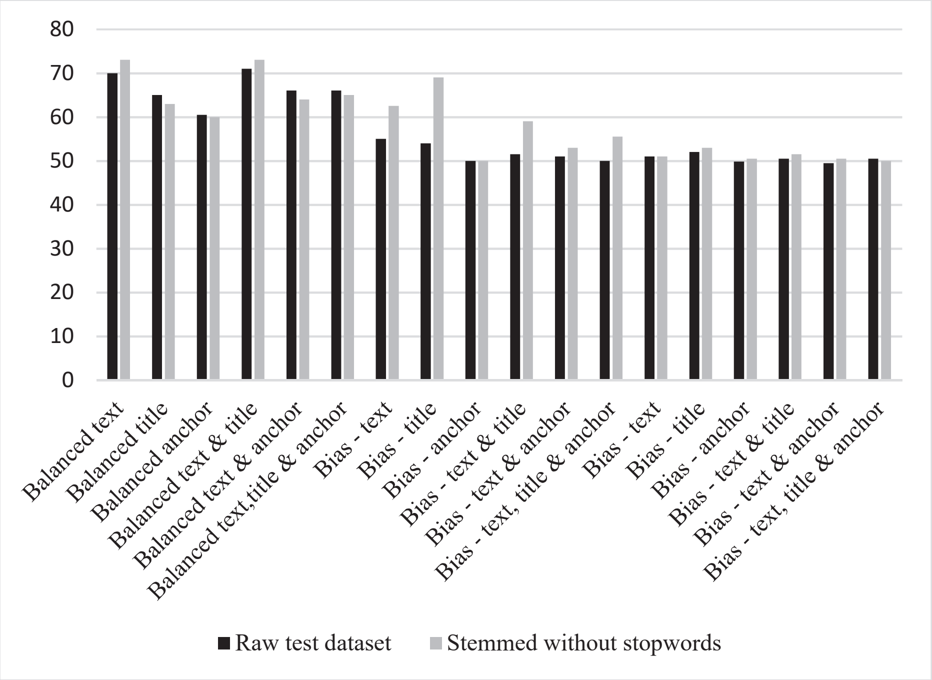

Figure 6. Comparing raw text to stemmed text without stop words

4.3. Finding the Best Sized Dataset for Training an SVM Model

The graph below indicates the resulting information gained from executing this experiment: The figure 7 shows that the accuracy derived from using 200 web pages for training is 19% more accurate than when using a dataset of 100 training examples. This accuracy rate is increased by a further 3% when the dataset size is increased to 400 examples.

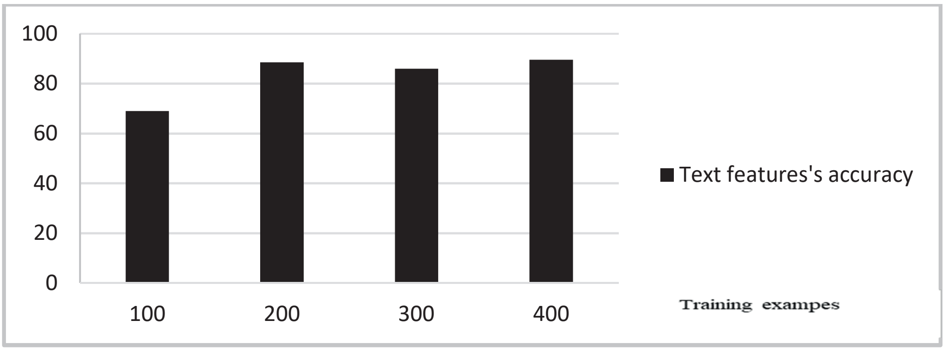

Figure 7. Choosing optimal dataset size for training

This increase in accuracy is disproportional to the doubling of the dataset size, and so in the prospect of making the solution as user friendly as possible, the trade-off point of using a dataset containing 200 training examples was adopted in this solution.

4.4. Selecting the Best Performing Feature Set for Use by the SVM

4.4.1 Testing Various Feature Sets

Below is a graph indicating the various accuracy levels attained from using various web page elements as features, as well as different combinations of the features. These results were obtained using balanced dataset and all features combinations (text, title, anchor, text & title, text & anchor, text, title & anchor) (see the following figure).

Using text and title combination contribute to highest accuracy level of 73% whilst using anchor is reflected as 60% accuracy level (see Figure 8).

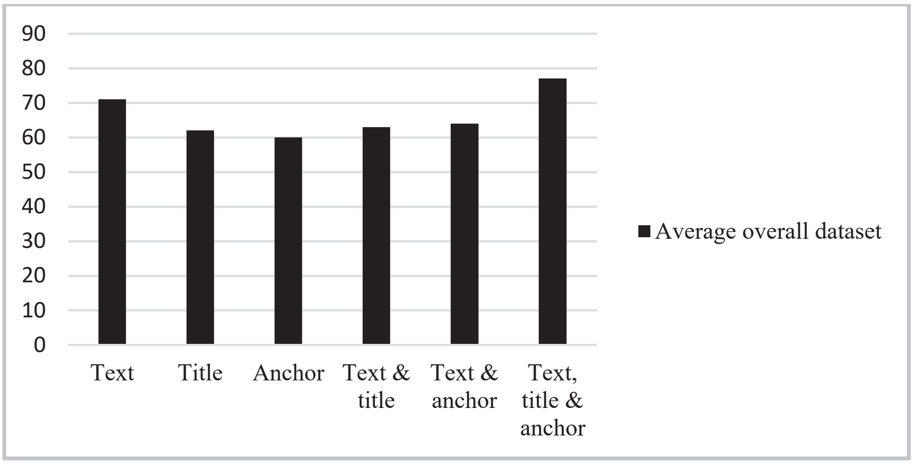

Figure 8. Testing various feature sets on balanced data

4.4.2 Accommodating for Bias Training Data

Since it has been discussed that the idea of a perfectly balanced set of training examples may not occur very frequently, the same tests were run using both positively and negatively biased training data. The graphs 9 and 10 below indicate the results of these tests.

As was mentioned in the experiment, the case where no conclusive feature is present across all three datasets has occurred. While the balanced dataset has the best results, with the feature combination of "Text & Title" elements of a web page being the best performer, it is unrealistic to assume a perfect split of training data.

Hence the best performing feature set of the two biased datasets should be selected. In this instance, the positively biased datasets indicate that the "Title Only" feature of a web page will give the highest levels of accuracy when using the SVM as a text classification solution (see Figures 9 and 10).

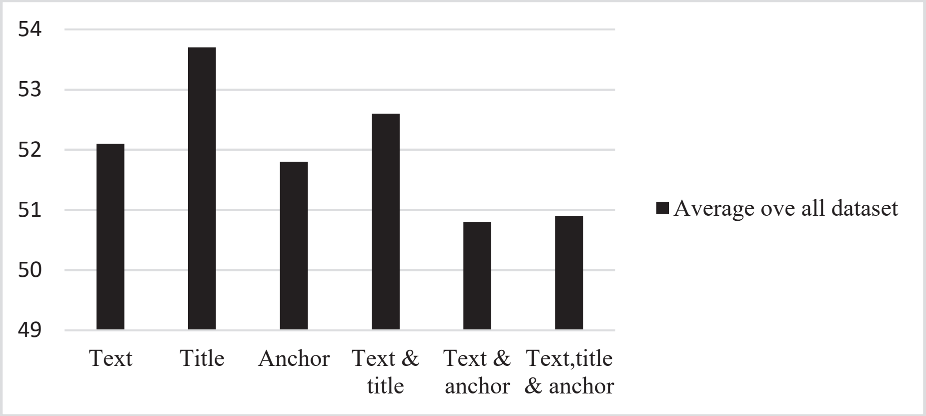

Figure 9. Testing various feature sets on positively biased data

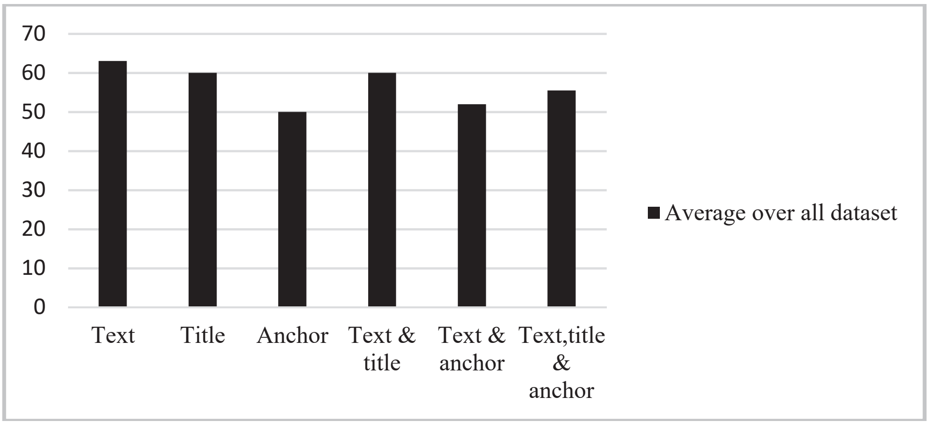

Figure 10. Testing various feature sets on negatively biased data

The use of the title element alone is not enough to give consistent classification results, as there are not enough features present when using this element. Hence the second best performing feature set, "Text & Title" is to be chosen. The negatively biased dataset best performing element is the "Text only" feature. However, when one takes the average of all three datasets, it can be seen that the use of "Text & Title" feature will provide the highest classification accuracy (see Figures 9 and 10).

It is for the reason that the choice of "Text & Title" feature combination was used in the final classification solution, since the overall accuracy of this feature combination across all datasets was 62,00%.

4.4.3. Determining Overall Relevancy Compared to a Commercial Search Engine

The results from these experiments were gathered and tallied but there were, however, certain limitations encountered. The niche-search engine took over 3.5 hours to crawl and classify as relevant only 50 pages. Due to time constraints, it was not possible to retrieve the full 200 results as was initially claimed in the experiment outline. In order to put the overall results into a broad perspective, the percentage of pages found relevant when using a commercial search engine were compared against the percentage of relevant pages returned by the niche search engine.

A query was trained with having 47 out of 200 results marked as relevant which returned from a commercial search engine. The same query was then used with the trained niche-search engine, where 36 out of 50 of the returned results were manually deemed to be relevant to the user's query. This means that 23.5% of a commercial search engines results were relevant, while 72% of the niche-search engines results were found to be relevant to the same user query. Hence it can be said that a user-defined niche search engine is able to return results which are more relevant than a commercial search engine.

The query used in the experiment was "Swiss rescue dogs"; the pages marked as relevant by the user were only those pages relating to the breed of the dog, namely a Swiss Mountain Dog, as well as information about breeding the dog and how to find breeders of the dog. The user defined the sub-domain of breeding, breeds and breeders related the Swiss Mountain Dog, as relevant and the niche-search engine returned relevant results pertaining to these topics 72% of the time.

It should also be noted that the training data at hand reflects the same scenario as negatively biased training data. Based on the experiments conducted with the SVM, this indicates an expected classification accuracy of approximately 60%. The accuracy of the niche-search engine however, is higher than 60%. This can potentially be owing to the performance of the focused crawler; crawling the most relevant pages.

4. Discussion

In this study the SVM received as input a web page in its raw form and classified its text as relevant or not, based on what the user finds relevant when searching for their query. The scope of this solution consisted of three main components, namely: a focused crawler, a Support Vector Machine (SVM) [26] and an indexing solution.

The focused crawler [6] was used to find web pages relating to a particular search query. The resulting web pages were sent to an SVM, which was used to classify their content as relevant or irrelevant. Once the relevancy of a page was determined, the site's URL address was passed onto the indexing solution for efficient retrieval by the search engine. While the focused- crawler may somewhat filter out irrelevant web pages, the majority of the workload needed to classify the documents as relevant or irrelevant falls onto the SVM.

The main elements of an HTML page consist of its *text, title, anchor-words and hyper-links*. There is also an element called *keywords*, however, it was found that this element contains largely spam (as an answer to RQ-1: *What are the different types of features present on web pages?)*

The positively biased datasets indicate that the "Title Only" feature of a web page will give the highest levels of accuracy when using the SVM as a text classification solution. The HTML text to be classified by the niche-search engine should be stemmed and have its stop words removed to optimize the classification process (as an answer to RQ-2: *What are webpage-specific features which can be used to assist the classification task of the SVM?)*

The author [9] suggested that aggressive feature selection would be beneficial for the SVM's classification capabilities. Various experiments were carried out in this study whereby aggressive feature selection was used, such as when using the Text, Title & Anchor word feature set. The results however proved that less aggressive feature selection was better, as using a single feature, or a feature set consisting of no more than two features, was better than using a feature set consisting of three features.

The research results [24] were different to those presented within this study, as the author claimed the best feature set was that of Text, Title & Anchor words, whilst this study claims that only a webpage's Text & Title elements provided the highest classification accuracy to be achieved by an SVM.

The results of the experiments also showed that there was a particular set of features which performed the best when using an SVM for text classification. It was seen that the best feature set to use for classification by the SVM were the "Text & Title" features from a webpage. (*As an answer to RQ- 3: Is there a particular feature, or combination of features, which will in fact improve the standard performance of the SVM?)*

The idea proposed in [11] suggests that the use of only the most unambiguous features when classifying data using an SVM can increase its classification abilities. This is reflected by the high accuracy rate of the Title feature in a number of experiments, since the Title of a webpage, if it is a legitimate title, would be considered an unambiguous feature.

The following are the specification for the SVM within the study:

- The text that is passed from the crawler to the SVM will have its stop words removed and its words stemmed,
- The model file to be used by the SVM will only be trained once the user manually classifies, 200 webpages, since this size was found to be the optimal one,
- The text passed to the SVM for classification will consist of an HTML page's Text, and Title elements, since these features returned the highest level of classification accuracy.

Since results indicated that the SVM is a suitable candidate for text classification, when combined with the above design decisions, the best possible performance from the SVM can be gained (as an answer to *RQ-4: What makes the particular use of an SVM suitable in the context of a niche search engine?)*

The notion of using Click-through data [12] from a commercial search engine to aid the construction of the niche-search engine was not adopted in the approach taken when designing this study. Hence the log of a user's query and the links the followed from the commercial search engine were not available to our solution.

Through the execution of the experiment, it has been shown that the user defined niche-search engine has the ability to out-perform a commercial search engine in terms of relevant results. However, our experiments cannot confirm the *hypothesis 1* in general; it can only confirm it for the particular data set used. Further investigations are necessary.

5. Some Limitations

It has been seen through the failed attempt of certain queries that the training process is very important to the overall solution. This is a difficult process as it is hard to resolve ambiguities and if these ambiguities occur in the training phase, the model reflects them by not returning accurately relevant results.

The confusion matrix presented in the research methodologies section was not employed when measuring the final experiment's results. The reason for this was that it was not possible to measure the number of false negatives returned by the classifier. The classifier only returned results that were deemed relevant by the SVM. Those which were not relevant were not shown to the user and hence could neither be confirmed nor denied by the user as irrelevant or not. The confusion matrices for testing different feature sets on the SVM were recorded in this study.

A massive downside of the solution proposed within this research study is the expensive toll of training the search engine with relevant results. The user was required to go through 200 pages, manually and more importantly, accurately, classifying them as relevant or not. The results from the niche-search engine can only be obtained thereafter.

The downside of the solution lies in the fact that the training process required for accurate results is long and tedious, whilst at the same time important to be done correctly in order to ensure high accuracy levels. It should be stated that the niche-search engine is better suited for a smaller, particular domain as opposed to a commercial search engine

The number of results was not even. The commercial search engine returned 200 examples to use as training data for the SVM, where the niche-search engine returned only 50 classified results.

6. Conclusion and Further Work

The research study involved the construction of a niche-search engine, whose search topic domain was user, defined. The purpose of this construction was to investigate the niche-search engine's ability to return results to users which are more relevant than those returned by a commercial search engine.

Thus, the search engine was successfully created and it was comprised of three main components, namely a focused crawler, a support vector machine (SVM) and an indexing solution. The research contained in this study pertains to the effects and capabilities of the SVM.

Various experiments were conducted in order to determine if research questions and Hypothesis could be answered. The hypothesis was set out stating that the niche-search engine could return results which are more relevant than those returned by a commercial search engine. The research questions aimed to discover if the SVM was firstly suitable for the needs of the niche-search engine. If the classification ability of the SVM was found to be suitable for the study, a secondary requirement of the second question was to find which subset of webpage specific features gave the best classification ability to the SVM.

Through the results of the experiments, it has been shown that the SVM has the ability to classify textual documents with a level of accuracy that is in all cases greater than 50%. In the particular case of balanced data, this accuracy increases and it can be concluded tentatively that the SVM is a suitable solution for classifying web pages in the context of a user-defined niche-search engine.

Incorporating the resulting best performing components of the SVM into a classification solution for the user defined niche-search engine was completed. The resulting effect of this allowed the niche-search engine to outperform a commercial search engine by returning results which were more relevant to a user, give a specific user defined domain. Therefore, we can tentatively conclude that these results provide a contribution to the area of text classification.

Pertaining to the main aim, the search engine with higher relevancy-levels of results returned for a particular query was created. The optimal subset of web page specific features was formulated and the levels of accuracy and relevance of the SVM's classifications were increased when compared to already existing feature selection techniques. Thus, the secondary aims were achieved. The capabilities of the SVM in terms of providing a suitable solution for classification in the context of a niche search engine were also determined.

As a future avenue of work, it should be a priority to reduce the time and effort taken to train the niche-search engine, while still maintaining the good accuracy performance. Finally, another future avenue of work in the related field can introduce a mechanism to retrain the SVM model used by the search engine after every 10 new pages it returns to the user as relevant. This will enhance the accuracy of the niche-search engine, as the model will dynamically adapt to the user's preferences. Further research is necessary to improve search algorithms and devise new techniques beyond SVM.

References

[1] R. Bekkerman, Distributional Clustering of Words for Text Categorization, Master's thesis, CS Department, Technicon Israel, Institution of Technology, 2003.

[2] A. Ben-Hur and J. Weston, " A Users Guide to Support Vector Machines", in O. Carugo, F. Eisenhaber (eds.), "Data Mining Techniques for the Life Sciences, Methods in Molecular Biology", 609, DOI 10.1007/978-1-60327-241-4_13, 2010.

[3] A. Chakrabarti, M. van der Berg and B. Dom, "Focused Crawling: a New Approach to Topic-Specific Web Resource Discovery", in Proceedings of the 8th International World-Wide Web Conference (WWW8), 1999.

[4] CMU World Wide Knowledge Base (Web->KB) project, [Online] Available: http://www.cs.cmu edu/afs/cs.cmu.edu/project/heo-11/www/wwkb/index.html (current November 2016).

[5] J.W. Creswell, Collecting Data in Mixed Methods Research. [Online], Available: http://www.sagepub.com/site/default/files/ upm-binaries/10983_Chapter_6.pdf, (Current Nov 2016).

[6] M. Diligenti, F. Coetzee, M. Lawrence, S.L. Giles, and M. Gori, “Focused Crawling Using Context Graphs” in Proceedings of the 26th VLDB Conference, Cairo, Egypt, 527-534, 2000.

[7] A. Field and G. Hole, How to Design and Report Experiments, London: Sage. 2003.

[8] T. Fu, A. Abbasi, and H. Chen, “A Focused Crawler for Dark Web Forums”, *Journal of the Association for Information Science and Technology*, vol 61, issue 6, pp 1213-1231, 2010.

[9] E. Gabrilovich and S. Markovich, “Text Categorization With Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5”, in Proceedings of the 21st International Conference on Machine Learning, ACM, pp 41, 2004.

[10] T. Joachims, “Optimizing Search Engines Using Click through Data”, in Proceedings of the eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, ACM, pp,133-142, 2002.

[11] T. Joachims, SVM Light: Support Vector Machine. [Online] Available: http://svmlight.joachims.org/ (current November 2016).

[12] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, in Proceedings of the European Conference on Machine Learning (ECML-98), Germany, vol 1398, pp 137-142, 1998.

[13] T. Joachims, “Transductive Inference for Text Classification Using Support Vector Machines”, in Proceedings of International Conference on Machine Learning (ICML), pp 200-209, 1999.

[14] R. Johnson, J. Freund and I. Miller, Probability and Statistics for Engineers, 8th Edition, Pearson, 2011.

[15] E. Leopold, and J. Kindermann, “Text Categorization with Sup- Port Vector Machines: How to Represent Texts in Input Space”, *Machine Learning*, 46, pp 423–444, 2002.

[16] D. Lewis, Y. Yang, T. Rose and F. Li, “A New Benchmark Collection for Text Categorization Research”, *JMLR*, 5, pp 361–397, 2004.

[17] H. Marmanis, and D. Babenko, Algorithms of the intelligent eb, Manning Publications Co, ISBN 9781933988 665, 2009.

[18] B. McCallum, K. Nigam, J. Rennie and K. Seymore, “Building Domain-Specific Search Engines with Machine Learning Techniques”, in

Proceedings AAAI, Spring Symposium on Intelligent Agents in Cyberspace, 1999.

[19] S. Mengle and N. Goharin, Passage Detection Using Text Classification. Institute of Technology, Chicago, [Online].Available:http://ir.cs.georgetown.edu/downloads/JASIST-PassageDetection.pdf (current November 2016).

[20] S.L. Morgan and C.G. Waring, Guidance on Testing Data Reliability. [Online] Available: http://www.auditorroles.org/ files/toolkit/role2/Tool2aAustinCityAud_GuidanceTesting Reliability.pdf (current November 2016).

[21] A. Rajaraman and J.D. Ullman, Data Mining of Massive Datasets. Doi: 10.1017/CBO9781139058452.002, pp 1–17, 2011.

[22] S. Saket, S. Mengle and N. Goharian, "Using Ambiguity Measure Feature Selection Algorithm for Support Vector Machine", *Classifier, SAC*, 08, pp 916–120, 2008.

[23] G. Sampson and M. Postal, The 'Language Instinct' Debate. Continuum International Publishing Group, 2005.

[24] A. Sun, E.P. Lim and N. Wee-Keong, "Web Classification Using Support Vector Machine", ACM Workshop on Web Information and Data Management (In Conjunction with the International Conference on Information and Knowledge Management -CIKM2002), McNeal, Virginia, USA, November, 2002.

[25] W.F. Tichy, "Should Computer Scientists Experiment More?", *IEEE Computer*, vol 3, no 5, pp 32-40, 1998.

[26] V. Vapkin, Statistical Learning Theory, Wiley, ISBN: 978-0-471-03003-4, 768 pages, 1998.

[27] W. Wenxian, C. Xingshu, Z. Yongbin, W. Haizhou, D. Wang, Zongkun, "A Focused Crawler Based on Naive Bayes Classifier", Intelligent Information Technology and Security Informatics (IITSI), Third International Symposium, 2-4 April, 2010.

[28] Y. Yang, J. Zhang, J. and B, Kisiel, "A Scalability Analysis of Classifiers in Text Categorization", in Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, 96–103, 2003.