Global Scientific JOURNALS

GSJ: Volume 9, Issue 5, May 2021, Online: ISSN 2320-9186
www.globalscientificjournal.com

DATA MINING AND TEXT CLASSIFICATION IN INFORMATION RETRIEVAL SYSTEM USING -NEAREST NEIGHBOR AND SUPPORT VECTOR MACHINE ALGORITHM

ALABI O. A[1]
Research Assistance (PhD in View)
Department of Computer Science
University of Port Harcourt

CHIADIKOBI B. C[2]
Department of Computer Science
University of Port Harcourt

ABSTRACT
Cloud has severs as store house of abundant information available in various electronic forms. In ages, the increase in the performance of computers in handling large quantity of text data led researchers to focus on reliable and optimal retrieval of information already exist in the Big and huge resources. Though the existing search engines, answering machines has succeeded in retrieving the data relative to the user query, the relevancy of the text data (data mining and classification) is not appreciable of the huge set. It is hence binding the range of resultant text data for a given user query with appreciable ranking to each document stand as a major challenge. In this paper, we propose an hybrid approach to access relevant documents for a given query finding the most appropriate boundary to related documents available on web and rank the document on the basis of query rather than customary Content based classification. The experimental results will elucidate the categorization with reference to closeness of the given query to the document.

Keywords: Data mining, information retrieval system, Text classification.

Introduction
As the volume of information is getting increased in the internet day by day there is a need for people to have the tools that find, filter the information and manage the resources. It is highly difficult for the people to maintain the huge data manually and it is very time consuming to extract the information effectively without any indexing and classification techniques [7]. Automatic text categorization is one particular tool to retrieve and make use of the text information efficiently.

Over the past two decades, the automatic management of electronic documents has been a major research field in computer science. Text documents have become the most common type of information repositories especially with the increased popularity of the internet and the World Wide Web. Internet and web documents like web pages, emails, newsgroup messages, internet news feed etc., contain million or even billion of text documents [8].

There are several applications where text categorization (classification) plays an important role like technical, professional, business and web based areas. Also the classification is considered to be an important research field used to identify the data and classify it based on several theoretical

approaches. Using automatic text categorization the stories can be categorized based on subject categories, academic papers are often classified by technical domains and sub-domains, patient reports in health care organizations etc. Automatic text categorization is efficient and cheaper when compared to manual categorization where it needs more number of people to manually label or categorize the data. Several methods can be implemented for categorizing the text that varies in their accuracy and computation efficiency [9].

Text categorization is defined as for a given set of previously unseen documents D = {d1, d2, d3….} and a set of pre-defined classes or categories C = { c1,c2,c3…. ck}, a classifier (categorizer) is a function κ that maps a document from set D to the set of all subsets of C and is shown in Figure 1.

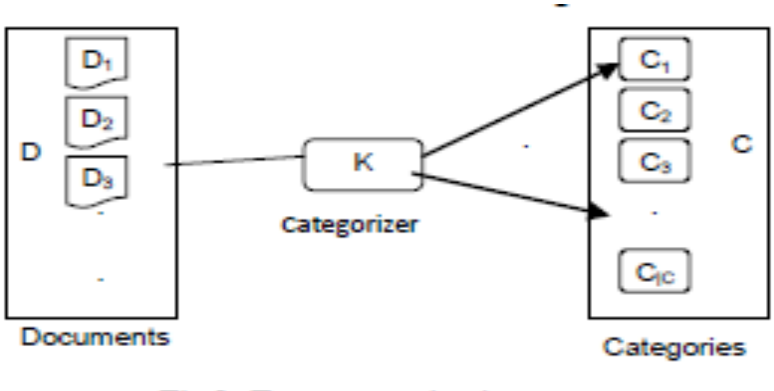

Fig 1: Text categorization process

2. TEXT CATEGORIZATION TECHNIQUES

A large number of statistical classification and machine learning techniques have been applied to text categorization, including regression models, Bayesian classifiers, and decision trees, nearest neighbor classifiers, neural networks, and support vector machines [10]. In this section, we will briefly review some of the classification techniques.

a. Decision Trees

Decision trees are most widely used classifier, consist of a set of rules which are applied in a sequential way and finally yield a decision. Their robustness to noisy data and capability to learn disjunctive expressions seem suitable for document classification. One of the most well known decision tree algorithms is ID3 and its successor C4.5 and C5. It is a top-down method which recursively constructs a decision tree classifier. A decision tree classifier is a tree in which internal nodes are labeled by attributes (words occurrences in the case of text categorization), branches departing from them are labeled by tests the weight that an attribute has in the test document, and leafs are labeled by categories [11]. Decision tree categorizes a test document by recursively testing the weights that the attributed labeling the internal nodes have in document vector, until a leaf reached.

They can be best explained by observing the training process, which starts with a comprehensive training set. It uses a divide and conquer strategy: For a training set *M* with labeled documents the word *ti* is selected, which can predict the class of the documents in the best way. Then *M* is partitioned into two subsets, the subset *M1* with the documents containing *ti*, and the subset *M2* with the documents without *ti*. This procedure is recursively applied to *M1* and *M2*. It stops if all documents in a subset belong to the same class *Lc*. It generates a tree of rules with an assignment to actual classes in the leaves. The key step is the choice of the term tk on which to operate the partition. Generally, a choice is made according to an information gain or entropy criterion. However, such a fully grown tree may be prone to over fitting, as some branches may be too

specific to the training data. Most DT learning methods thus include a method for growing the tree and one for pruning it, for removing the overly specific branches.

b. Naïve Bayesian Algorithm

A Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions. A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood. Abstractly, the probability model for a classifier is a conditional model *p* (C|F1….Fn) over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F1 through Fn. Using Bayes theorem, we write

=) ,....., (1 n F F C p) ,..... ( ) ..... , ( ) (1 , 1n n F F p C F F p C p

In plain English the above equation can be written as evidence likehiood prior posterior □ □

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features Fi are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model which can be rewritten as follows, using repeated applications of the definition of conditional probability [12].

c. Support Vector Machine

Support vector machines (SVMs) are supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts for each given input to which of two possible classes it is a member of, which makes the SVM a non-probabilistic binary linear model. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [15].

More formally, a support vector machine constructs a hyperlane set of hyper planes in a high or infinite dimensional space, which can be used for classification and regression. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Whereas the original problem may be stated in a finite dimensional space, it often happens that in that space the sets to be discriminated are not linearly separable. For this reason it was proposed that the original finite dimensional space be mapped into a much higher dimensional space, presumably making the separation easier in that space. SVM schemes use a mapping into a larger space so that cross products may be computed easily in terms of the variables in the original space, making the computational load reasonable. The cross products in the larger space are defined in terms of a kernel function K (x, y) selected to suit the problem. The hyper planes in the large space are defined as the set of points whose cross product with a vector in that space is constant [16] [17]. The vectors defining the hyper planes can be chosen to be linear combinations with parameters αi of images of feature vectors that occur in the data base. With this choice of a hyper plane the points x in the feature space that are mapped into the hyper plane are defined by the relation:

$$\Sigma \alpha_i K(x_i, x) = constant$$

Note that if K(x, y) becomes small as y grows further from x, each element in the sum measures the degree of closeness of the test point x to the corresponding data base point xi. In this way the sum of kernels above can be used to measure the relative nearness of each test point to the data

points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyper plane can be quite convoluted as a result allowing much more complex discrimination between sets far from convex in the original space.

d. K-Nearest Neighbor Classifier

The Nearest Neighbor classification is a non-parametric method and it can be shown that for large datasets. The error rate of the 1-Nearest Neighbor classifier is never larger than twice the optimal error rate. In this classifier to decide whether the document di belongs to the class Ck, the similarity Sim(di, dj) or Dissim(di , dj) to all documents dj in the set is determined. The k most similar training documents are selected. The proportion of neighbors having the same class may be taken as an estimator for the probability of that class and the class with the largest proportions assigned to the document dj. The algorithm has two parameters (k and similarity/dissimilar value) which decide the performance of the classifier and are empirically determined. However, the optimal number „k" of neighbors may be estimated from additional training data by cross validation [13] [14]. The major drawback of the classifier is the computational effort during classification, as basically the similarity of a document with respect to all other documents of a training set has to be determined.

After having a quick overview of each classifier, we need to overcome the problems faced by each classification algorithm and develop a new approach to classify the document into a reserved class. Here we adopted a latest approach of KNN classifier.

The following are challenges of clustering in information retrieval system.

1. **The identification of distance measure:** For numerical attributes, distance measures that can be use dare standard equations like Euclidian, Manhattan, and maximum distance measure. All the three are special cases of Minkowski distance. But identification of measure for categorical attributes is difficult.

2. **The number of clusters:** Identifying the number of clusters is a difficult task if the number of class labels is not known beforehand. A careful analysis of number of clusters is necessary to produce correct results. Else, it is found that heterogeneous tuple may merge or similar type tuple may be broken into many. This could be catastrophic if the approach used is hierarchical. Because in hierarchical approach if a tuples gets wrongly merged in a cluster that action cannot be undone. While there is no perfect way to determine the number of Clusters, there are some statistics that can be analyzed to help in the process.

These are the Pseudo-F statistic, the Cubic Clustering Criterion (CCC), and the
Approximate Overall R-Squared.

3. **Lack of class labels:** For real datasets (relational in nature as they have tuples and attributes) the distribution of data has to be done to understand where the class labels are.

4. **Structure of database:** Real life Data may not always contain clearly identifiable clusters. Also the order in which the tuples are arranged may affect the results when an algorithm is executed if the distance measure used is not perfect. With a structure less data (for example, having lots of missing values), even identification of appropriate number of clusters will not yield good results. For example, missing values can exist for variables, tuples and thirdly, randomly in attributes and tuples. If a record has all values missing, this is removed from dataset. If an attribute has missing values in all tuples then that attribute has to be removed described in. A dataset may also have not much missing values in which case methods have been
Suggested in. Also, three cluster-based algorithms to deal with missing values have been proposed based on the mean-and-mode method in.

5. **Types of attributes in a database**: The databases may not necessarily contain distinctively numerical or categorical attributes. They may also contain other types like nominal, ordinal, binary etc. So these attributes have to be converted to categorical type to make calculations simple.

In the forthcoming Proposed Method section of the report, we will discuss our ideas applying some researcher‟s thoughts.

3. Proposed System

Let T be a document which is represented by a scalar point within a space s. Let s1, s2,....sn be n numbered vector points each representing a training document of the dataset considered as corpora in the same space s. A customary K Nearest Neighbor method requires an input integer k which is the number of vectors retrieved that are relevant to the given point T. K-NN also need a metrics to measure the closeness of each training vector point to the test point p. The training datasets are themselves pre-labeled into predefined category. The result is the label name of maximum number of documents which are close to p of the k points. But when discussing about Information Access Systems (IAS) like Web Search Engines, Answering Machines, Blogs, Software resource providers, etc. the testing input is not in the form of document but a small query in terms of word or phrase itself. The result would be more or less acceptable when dealt with an accurate word. But when the query is in the form of phrase, it may lead to trouble the IAS. We will then segment the given l worded query into l words. For each word in the query, we will calculate the closeness to each training documents and hence by consider k-documents for each word. As a next step, we introduce a new technique− stroking where we build possible phrases among the l-words. Again K-Nearest Neighbor Algorithm is applied on the l batched k documents to further formalize the l X k documents into phrased numbered of documents. Such iteration of KNN in multi-steps is applied until a single phrase query forms k documents relevant closely to the document. Ranking plays an important role, as long as search and other information retrieval applications keep developing and growing. Here ranking to each document is a bit easy task as it can be assigned on the basis of basic metric −closeness of the resultant k documents. KNN is considered as the most tractable computationally among most of the Instance Based Classification Methods as it effectively works with huge amount of datasets. K-nearest neighbor has a major drawback when used to retrieve information alone. This is because the greater percentage of its computation usually takes place after the query has been submitted to the database engine especially if a large dataset is being considered in the query since the number of similarities required to perform the calculation is considerably reduced. Based on this, we decided to combine

K-NN with another tool, the support vector machine (SVM). The SVM solves the problem of K-NN by ensuring that the computation is done at a greater speed and lesser time so that information can be retrieved quickly.

3.1 Text Categorization overview

Text categorization is the task of automatically assigning input text to a set of categories [1]. The objective of text categorization is to assign a category to an entry from a set of predefined categories to a document. So far many methods of the text categorization are presented, such as Support Vector Machine, k nearest neighbors, neural network, bayes classier and decision tree, etc. most of them are instance based and some content based. K nearest neighbor is a simple, valid, non-parametric method among them. KNN has undergone into many changes in the present era as the traditional KNN has two fatal defects that are time of similarity computing is huge and its performance depends on training sample set. For multi document text categorization, similarity between unknown samples and also between known samples need to be calculated resulting in the high competency value. Also the test vector matrix becomes high dimensional leading to increase in time complexity. When dealing with query based document categorization, where a single document may serve for multiple categories, a mere binary

assignment is not sufficient. For this computation of similarity must be carried out as a special case.

As studied in [3], there are three approaches using which we can increase the speed of calculation for KNN:

1. By reducing the dimension of text vector,
2. By using the smaller sample set,
3. By quickening the speed of finding the k nearest neighbors.

Using these approaches, the problem regarding speed of computing can be appreciably reduced that is were the SVM came in to solve speed computation issues .

3.2 Ranking

For a given query, Ranking is one parameter which defines how good a document is better closer than the other document to fall itself under a specific category. Unfortunately the index terms identification which is considered as the most crucial part of ranking will in no way help if considered a Boolean Model. But the Statistical Model is based on similarity between the statistical properties of the text document and the query is no doubt a good and classical approach. In this context too ranking is done predominantly on the basis of three approaches - point wise approach involving classification on the basis of single documents, pair wise approach involving classification of document pairs and list wise approach involving document lists [2]. But, here the proposed KNN approach directly apply statistical model ranking to the text data felicitating the document access time and ranking time.

4. Classification of Data

In the statistically based vector-space model, a document is conceptually represented by a vector of keywords extracted from the document, with associated weights representing the importance of the keywords in the document and within the whole document collection; likewise, a query is modeled as a list of keywords with associated weights representing the importance of the keywords in the query [5]. Here weighted indexes serve as large number of word attributes enabling for a high precise classification. Once weights are derived from predefined dataset, the input for the query based model is ready to feed. Queries are usually generated from the user‟s perspective and the more the information available the more is the documents retrieved. In general, any query consists of a word or a collection of words. Web search engines retrieve the documents related to each word in comparison with the content and show the results on the interface i.e. the browser as a whole. Hence, documents are usually more in number. In Query Based classification system using KNN, the expected minimum inputs are: A query and a positive integral value to how many number of results to be retrieved. The structure is as discussed in the Figure 2.

As seen in the model in Figure 2., S is a vector space consisting of pre-classified four dataset groups labeled L1,L2,L3 and L4.each label Li (i=1,2,3,4) contains a finite vector point each representing the document of their respective class. First, for each document in the labeled class, indexes are to be created by classical formalization techniques. Then the given user query is checked for number of words contained in it. Let them be l. This can make the relevancy of document retrieval in an appreciable form.

In Figure 2, the user given query is decomposed into l word number of fragments. Then for each fragment of query, the distance between the indexes of each trained vector and query fragment is calculated.

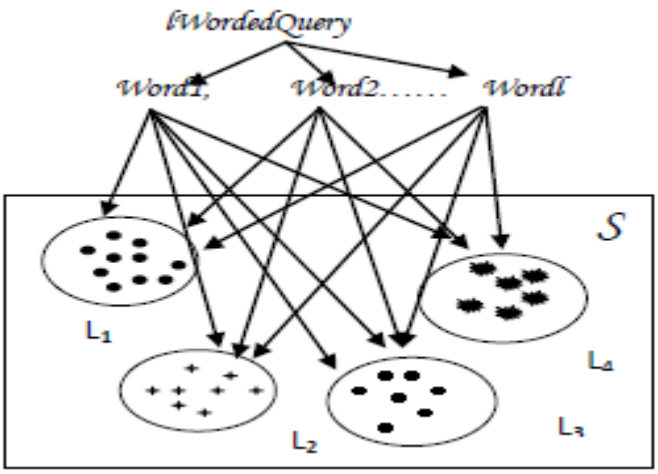

Figure 2. Query comparing between each of class

There are lots of methods available to calculate the distance. Two important methods are:

1. Euclidean distance: The Root square sum of differences between the two points is considered as distance [6].

$$d_E(l,q) = \sqrt{(l_1 - q_1)^2 + (l_2 - q_2)^2}$$

For a Cartesian coordinate system (two dimensional),

l= (l1, l2) is the position of label vector and q= (q1, q2) is the position of query word.

2. Manhattan distance: It is the sum of the lengths of the projections of the line segment between the points *l*, *q* onto the coordinate axes. It is designed as a n-dimensional vector space [6].

$$d(l,q) = \sum_{i=1}^{n} | li - qi |$$

Using any of the two methods, we can evaluate the distance between the sub word of query and indexes of each document. By constructing a confusion matrix, we can consider the statistically closer documents. The k valued documents are now obtained for each word in the query.

The query terms are then joined into two phrased words and the same above procedure is applied with the earlier retrieved documents in labels as dataset group.

The algorithm for such a query based Classification is shown in the Table 1.

Input: Query q I i words in it

A preset integral value k

N labeled classes as corpus

Algorithm:

i) Split the given query into i words

ii) For int k=1 do the steps (iii) and (iv).

iii) Calculate dist between lj where j =1,2,..j documents in Label *i* using *distance formula*

iv) Take k least distant values form label 1

v) Repeat the steps until l labels

vi) Build phrase between i and i+1th word

vii) Do repeat (i) to (vi) for all i worded until 1 phrase i.e. i worded query

5. Dataset used (20 NEWGROUPS)

The 20 Newsgroups has been a predominantly using dataset for experiments in text applications of machine learning techniques. The 20Newsgroups is a collection of approximately 20,000 newsgroup documents grouped into 20 predefined categories. The category list is provided in the Table 2. The dataset was originally collected by Ken Lang, for his Newsweeder research learning to filter net news paper.

The query based KNN and SVM approaches has different time complexities when implemented offline and in online. For a given query as shown in the Table 3, the processing time or the time complexity is shown in respective columns in offline and in online. It was observed for the same dataset, at online and at offline there is comparatively bearable time lag found and is shown in Figure 3.

Ranking of the documents retrieved is based on the least value of the distance calculated in the ultimate steps of calculation. KNN application here is done with k value preset to 100. It is thus the model is more precisive in nature.

Table 2. 20 NewsGroups DataSet

Category	**# train docs**	**# test docs**	**Total # docs**
alt.atheism	480	319	799
comp.graphics	584	389	973
comp.os.ms-windows.misc	572	394	966
Comp.sys.ibm.pc.hardware	590	392	982
Comp.sys.mac.hardware	578	385	963
Comp.windows.x	593	392	985
misc.forsale	585	390	975
rec.autos	594	395	989
rec.motorcycles	598	398	996
rec.sport.baseball	597	397	994
rec.sport.hockey	600	399	999
sci.crypt	595	396	991
sci.electronics	591	393	984
sci.med	594	396	990
sci.space	593	394	987
soc.religion.christian	598	398	996
Talk.politics.guns	545	364	909
Talk.politics.mideast	564	376	940
Talk.politics.misc	465	310	775
Talk.religion.misc	377	251	628
Total	**11293**	**7528**	**18821**

6. RESULT ANALYSIS

The resultant documents retrieved offline and online for the given queries are shown in the Table 4.

Table 4.Retrived documents

Query	**Documents/retrieved Offline and Online**
Information Retrieval	32
Data mining	48
World wide web	52

Cloud computing	15
Extraction	62
Data	70

The results have provided a better perspective view for the on KNN approach there by giving a better scope for optimality and feature based categorization too.

Effect on Different k values

As the document to be retrieved are predetermined initially (For our experiment k=100), the time complexity may vary accordingly. For huge amount of data, Centroid Based distance calculation gave best results in addition to classical above discussed forms.

We tested the performances of the KNN methods with different values of parameter k, the number of nearest neighbors selected. Notice that when k = m, KNN becomes comparatively less, where m denotes the number of training queries. Datasets with different k values in terms of different groups as k increases, the performances first increase and then decrease. More specifically,

1. When only a small number of neighbors are used, the performances of KNN are not so good due to the insufficiency of training data.
2. When the numbers of neighbors increase, the performances gradually improve, because of the use of more information.
3. However, when too many neighbors are used (approaching 1500), the performances begin to deteriorate.

7. CONCLUSIONS

Data mining and Text Categorization on text documents is proposed with KNN and SVM approach. The hybrid method has found the most relevant documents for a given query. Also it is useful for finding the most appropriate boundary to related documents available on web and rank the document on the basis of query rather than customary content based classification. Experimental results shows, this approaches of Text Categorization have provided a better perspective view, there by giving a better scope for optimality and feature based categorization. This method has significantly reduced the query response time, improving the accuracy and degree of relevancy.

8. REFERENCES

[1] Sebastiani, F.,Machine learning in automated text categorization. ACM Computing Surveys, 34(1), pp. 1–47, 2002.

[2] XiuboGeng, Tie-YanLiu, TaoQin, AndrewArnold, HangLi and Heung-YeungShum, “Query Dependent Ranking Using
K-Nearest Neighbor,” ACM, SIGIR08, July20–24,2008,Singapore.

[3] Dik L. Lee, uei Chuang, H Ent Seamons,“ Document Ranking and the Vector-Space Model”,a research theisis, March-April,1997.

[4] T.Y.Liu,Y.Yang,H.Wan,H.Zeng,Z.Chen,andW.Y.Ma,“Support Vector machines classification with a very large scale taxonomy. SIGKDD Explor. Newsl,7(1):36–43.

[5] Gongde Guo , Hui Wang , David Bell , Yaxin Bi , and Kieran Greer, “Using kNN Model-based Approach for Automatic Text Categorization”.

[6] Stavros Papadopoulos, Lixing Wang, Yin Yang, Dimitris Papadias, Panagiotis Karras, "Authenticated Multi-Step Nearest Neighbor Search"
[7] Yang, Y. & Pedersen, J.O., A comparative study on feature selection in text categorization. Proceedings of ICML-97, 14th International Conference on Machine Learning, ed.D.H. Fisher,Morgan Kaufmann Publishers, San Francisco, US: Nashville, US, pp. 412–420, 1997.
[8] Guru, D. S., Harish B. S., and Manjunath, S. 2009. "Clustering of Textual Data: A Brief Survey", In the Proceedings of International Conference on Signal and Image Processing, pp. 409 – 413.
[9] Dr. Riyad Al-Shalabi , Dr. Ghassan Kanaan and Manaf H. Gharaibeh "Arabic Text Categorization Using kNN Algorithm"
[10] K. Aas, L. Eikvil, Text Categorization: A Survey. Norwegian Computation Center, Oslo, 1999
[11] R.M. Duwairi, A Distance-based Classifier for Arabic Text Categorization, In Proceedings of the International Conference on Data Mining, Las Vegas USA, 2005.
[12] Ioan Pop "An approach of the Naive Bayes classifier for the document classification" General Mathematics Vol. 14, No. 4 (2006), 135–138
[13] Hotho, A., Nürnberger, A., and Paaß, G. 2005. A Brief Survey of Text Mining. Journal for Computational Linguistics and Language Technology. Vol. 20, pp. 19 – 62.
[14] Yang, Y., Slattery, S., and Ghani, R. 2002. A study of approaches to hypertext categorization. Journal of Intelligent Information Systems, Vol 18(2), pp. 219 – 241.
[15] Joachims, T., Text categorization with support vector machines: learning with many relevant features. Proceedings of ECML-98, 10th European Conferenceon Machine Learning, eds. C. N´edellec & C. Rouveirol, Springer Verlag, Heidelberg, DE: Chemnitz, DE, pp. 137–142, 1998. Published in the "Lecture Notes in Computer Science" series, number 1398.
[16] Joachims, T., Transductive inference for text classification using support vector machines. Proceedings of ICML-99, 16th International Conference on Machine Learning, eds. I. Bratko & S. Dzeroski, Morgan Kaufmann Publishers, San Francisco, US: Bled, SL, pp. 200–209, 1999.
[17] Drucker, H., Vapnik, V. & Wu, D., Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10(5), pp. 1048–1054, 1999.