Formal Concept Analysis Based Corrective Approach Using Query-log for Web Page Classification

Abdelbadie Belmouhcine and Mohammed Benkhalifa
Computer Science Laboratory (LRI), Computer Science Department, Faculty of Science, Mohammed V-Agdal University
Rabat, Morocco
Email: belmouhcine@gmail.com, khalifa@fsr.ac.ma

***Abstract*—Web page classification has many applications and plays a vital role in web mining and semantic web. Web pages contain much irrelevant information that does not reflect their categories or topics, and operates as noise in the process of their classification, especially when using a text classifier. Thus, the use of information from related web pages can help to overcome the problem of noisy content and to get a better result after the classification. Web pages are linked either directly by hyperlinks or indirectly by user's intuitive judgment. In this work, we suggest a post classification corrective method that uses the query-log to build an implicit neighborhood, and collectively propagate classes over web pages of that neighborhood. This collective propagation helps improving text classifier results by correcting wrongly assigned categories. Our technique operates in four steps. In the first step, it builds a weighted graph called initial graph, whose vertices are web pages and edges are implicit links. In the second step, it uses a text classifier to determine classes of all web pages represented by vertices in the initial graph. In the third step, it constructs clusters of web pages using Formal Concept Analysis. Then, it applies a first adjustment of classes called Internal Propagation of Categories (IPC). In the final step, it performs a second adjustment of classes called External Propagation of Categories (EPC). This adjustment leads to significant improvements of results provided by the text classifier. We conduct our experiments using five classifiers: SVM (Support Vector Machine), NB (Naïve Bayes), KNN (K Nearest Neighbors), ICA (Iterative classification algorithm) based on SVM and ICA based on NB, on four subsets of ODP (Open Directory Project). We also compare our approach to Classification using Linked Neighborhood (CLN) considered as the closest algorithm to EPC. Results show that: (1) when applied after SVM, NB, KNN or ICA classification, IPC followed by EPC help bringing improvements on results. (2) F1 scores provided by our approach with any of the five classifiers are significantly better than those obtained by CLN. (3) The performance provided by our proposed approach grows proportionally to the size of the query-log, and to the density of the weighted graph.**

***Index Terms*—formal concept analysis, centrality degree, semantic web, web page classification, query-log.**

I. Introduction

Nowadays, the size of the Web is growing drastically. Thus, document organization is becoming more critical task of information retrieval. Furthermore, with the growth of this size, this organization becomes more difficult. Unlike textual document, web pages link to each other either by ingoing and outgoing links or from a user's perspective. Hence, those links can help to improve web page categorization.

The content of web pages contains lot of noise due to some secondary content such as advertising, contact information… Also, some web pages lack textual content and contain images or videos. Therefore, although the content of the web page does not provide useful information about its category, the exploration of the neighborhood can help to determine that category.

Chakrabarti, Dom and Indyk [1], reported that the use of hyperlinks in the classification may decrease classification performance. They concluded that hyperlinks should be used carefully. Thus, we will try to create implicit links between pages by extracting patterns from a query-log of a search engine. The motivation behind this approach is that search engine's query-log contains lot of information about web pages' relevance. In addition, this log shows semantic relationships between pages clicked by a user in response to a query.

In this article, we propose a post classification corrective technique that proceeds in four phases: (1) Neighbors Discovery, (2) Initial classification (bootstrapping), (3) Formal Concept based Internal Propagation of categories (IPC) for the first adjustment and (4) External propagation of categories (EPC) for the second adjustment. In the first step, it uses a query-log to build an initial weighted graph. In this graph, web pages are represented by vertices and each edge connects a pair of web pages clicked together in response to a query in the query-log. Each edge is weighted by the clicking frequency (extracted from the query-log) of its pair of web pages. In the second step, it submits every web page (in the initial graph) to a classifier that exploits the page's local content to determine its "initial class." In the third step, it uses Formal Concept Analysis (FCA) [2] which consists of a set of mathematical tools aiming at analyzing and manipulating data, to construct clusters of correlated pages. Then it assigns to each cluster the category with the highest score in that cluster. The score of each category within

doi:10.4304/jetwi.6.2.200-209

a cluster is computed as the sum of category based centrality degrees of all web pages belonging to this category. In the final step, for each target web page, we assign a score to each category. This score corresponds to the sum of weights of links connecting the target web page to web pages of this category. Then, the target web page is assigned to the category with a maximum score.

We test our approach on four binary classifications, where datasets are extracted from the Open Directory Project (ODP) [3]. We also used AOL query-log [4], [5] to extract implicit links and build the initial weighted graph.

Our study shows that the clustering of web pages using implicit links and FCA helps to have groups of web pages that are related based on the users' intuitive judgments. Also, our approach ameliorates the classification using text alone due to the propagation of classes in the implicit neighborhood (labeled graph) and to the use of weights that reflect strengths of relationships between web pages.

In this paper, our principal contributions are:

1. Suggestion of a novel FCA based clustering method that leverages the query- log information. This clustering approach is link-based and does not necessitate the number of clusters as input (as K-means [6]).
2. Proposition of a weighting scheme for implicit links were weights correspond to the frequencies of clicks on two web pages together from query response as provided by the query-log.
3. Proposition of a post classification corrective approach that uses Internal Propagation of Categories followed by External Propagation of Categories in order to make categories rectifications for classification's results improvement.

The remainder of this document is structured as follows: In section 2, we review recent work on collective classification, label propagation and the use of implicit links and query-log to improve web page categorization. In section 3, we give some definitions related to Formal Concept Analysis. In section 4, we provide details of our approach. In section 5, we show the experimental setting adopted. Then, we present and discuss obtained results. Finally, we conclude our work and cite some of our future perspectives.

II. Related Work

Many works have used information provided by the neighborhood in web page categorization. This information can be labels, content or part of the content of linked web pages. Some previous works used hyperlinks to construct the neighborhood of web pages. However, there are researchers who utilized artificial links, which allow web pages to be connected even though there is no hyperlink relationship between them.

Several works have utilized implicit links built using the query-log. Shen et al. [7] introduced new links between web pages, called implicit links by leveraging query-log information. They constructed links between web pages clicked by users through the same query. They compared two types of implicit links with three types of explicit links using two links leveraging methods: Classification by Linked Neighborhood (CLN) and virtual documents based classification (VDC). In VDC, they tried both Support Vector Machine (SVM) and Naive Bayes (NB) for the classification. Their experiments showed that the use of implicit links improves both micro and macro F1 scores. Likewise, Kim et al. [8] proposed a semi supervised classification method that propagates class labels from labeled pages to unlabeled ones by leveraging click-log (query-log). They used a similarity that assumes that pages clicked by the same user's queries are similar. They compared their method to Gradient Boost Decision Tree algorithm and proved that it outperforms this latter on three datasets. Xue et al. [9] introduced an iterative reinforcement categorization (IRC) that exploits relationships between web objects to reinforce content based classification results by the propagation of the category from one object to related ones. They used web pages and queries as objects and clicks on pages through queries as relationships between them. They concluded that using this relationship improves the classification's F1 measure over content based method, virtual document based method and query meta-data based method. Dai et al. [10] decomposed pages to small semantic units called blocks such as tables, paragraphs… They used associations between blocks and queries to perform a Block Propagation Categorization (BPC) that, unlike traditional propagation approaches, propagates only useful related blocks among web pages to highlight their topics. They reported that BPC improved results over traditional approaches. Similarly, to all those works our proposed method uses query-log to build implicit links. However, in this work we suggest a weighting model for those implicit links that reflect the strength of relationships.

All collective classification approaches have a bootstrapping step in which labels are assigned to unlabeled items using a classifier. All those labels are adjusted in the inference phase. This latter can be seen as a collective correction step. Many techniques related to the collective classification problem or the graph labeling problem [11] have been proposed. The three popular collective classification methods are Iterative Classification Algorithm (ICA) [12], Gibbs Sampling Algorithm (GSA) [12] and Loopy Belief Propagation (LBP) [12]. Liu et al. [13] presented an approach that represents web pages using a weighted graph whose weights are obtained by considering contextual information of web pages and a dependent function between pages measured using mutual information, link features and link structure of two pages. In this graph, a small portion of web pages is labeled while the majority is not. They propagated labels from labeled nodes to unlabeled ones using probabilistic matrix methods and belief propagation. They showed that combining label propagation with link information can considerably help to improve classification results over Transductive SVM and Harmonic Gaussian Model. Relaxation labeling [14] is another technique that was applied in collective classification. It has shown good results in web page classification [1], [15]. Angelova and Weikum [15] proposed a relaxation labeling technique that starts by assigning

initial classes to web pages using Naïve Bayes or Support Vector Machine. Then, in each iteration, the probability of assigning a pair of labels to end points of the edge between two documents can be either smoothed using Laplace Smoothing in the case of Hard Labeling or by multiplying individual probabilities over the entire graph in the case of Soft Labeling. The process is repeated until convergence. Although our approach seems to be similar to relaxation labeling [1], [15], it does not estimate any probabilities from labels distribution in the graph. Furthermore, the relaxation labeling calculates the classification's probability of a web page p into a class c as the sum of likelihoods of all possible labels of neighbors of p with class c. However, in our approach, we tried to use a linear combination of weights and class vectors to compute scores of different categories for the target web page p based on its neighbors' class vectors.

In this paper, we suggest a post classification corrective approach. This latter is collective and iterative graph-based approach, that uses information from the query-log to build implicit links between web pages and propagates labels over the implicit neighborhood in order to adjust classes wrongly assigned by a text classifier.

III. Formal Concept Analysis

Let O and M respectively be the objects set and the attributes set and I a relation between O and M, which shows if an object $o \in O$ has an attribute $m \in M$. The triplet K=(O,M,I) is called a formal context [2] and can be seen as an object-attribute binary matrix. The Galois connection corresponds to the two mapping functions $f : 2^O \to 2^M$ and $g : 2^M \to 2^O$ such as:

$$\forall A \in 2^O, f(A) = \{m \in M / \forall a \in A, \ (a,m) \in I\}$$
$$\forall B \in 2^M, g(B) = \{o \in O / \forall b \in B, \ (o,b) \in I\}$$

Given a formal context K, a pair (A,B) is called a formal concept if $A \in 2^O$, $B \in 2^M$, f(A)=B and g(B)=A. The set of all concepts (A,B) forms a lattice whose ordering relation is inclusion. This lattice is called Galois Lattice [2].

In this paper, we will build a formal context K(V,V,E) from a graph G(V,E), where V is the vertices set that represents web pages and E is the edges set that represents links between web pages. In this case, we will have f=g, and we will represent each concept (A,B) using A. A is considered as cluster of vertices (web pages).

IV. Proposed Approach

The URLs, which are clicked by a user from responses corresponding to a query in a search engine, are connected from the user's perspective. Thus, they are related in a particular context. If many users clicked on two URLs together, the relationship between those URLs becomes stronger. Therefore, these URLs go high probably to the same topic. In this paper, we relate web pages clicked by a user through a query and give to each link a score that corresponds to the frequencies of occurrence of those pages together as clicked by a user in the query-log.

As was done in [7], we construct implicit links between web pages, those links wire web pages that appear in the result of a user query. But, our approach differs from [7] in:

1. The treatment of missing labeled neighbors. Indeed, in the CLN introduced in [7] the category of a web page is determined using labeled web pages of its neighborhood. The class of a target web page corresponds to the most represented category within its labeled neighbors while the unlabeled neighbors are simply ignored. However, labeled elements in the neighborhood of a target web page may be rare or even absent. In this work, we use a text classifier (in the second step of our approach) to assign categories to all web pages. Our motivation to do this initial classification is the exploitation of neighbors information to support classification's performance.
2. The use of implicit links weights. In [7], implicit links are not weighted. Whereas, in this paper, we assign weights to them. Those weights correspond to the number of time that two web pages appear together in the query-log. This is motivated by the use of strength of users' intuitive judgments (via implicit links extracted from query-log). This relevancy estimation helps to improve classification's results.
3. The method's principle. The CLN proposed in [7] is a classification approach while our approach is a post classification corrective approach. Our motivation is to use neighboring information to correct wrongly assigned categories.

Our method operates in four stages. In the first stage which is called Neighbors Discovery, we construct a graph called initial graph, whose vertices are web pages (URLs) and edges link web pages clicked in tandem by a user through a query. Those edges are weighted using frequencies of clicks on their both extremities together as provided by the query-log. Fig. 1 illustrates an initial graph constructed from a search engine's query-log.

In the second stage which is called Initial classification, we simply classify each web page using content based classifier. The resulted graph is called labeled graph. Although edges in this graph keep the same weights as in the initial one, vertices contain web pages labeled by their initial classes given by the content based classifier. Fig. 2 gives an example of a labeled graph constructed using the initial graph in Fig. 1.

In the third stage which is called Formal Concept

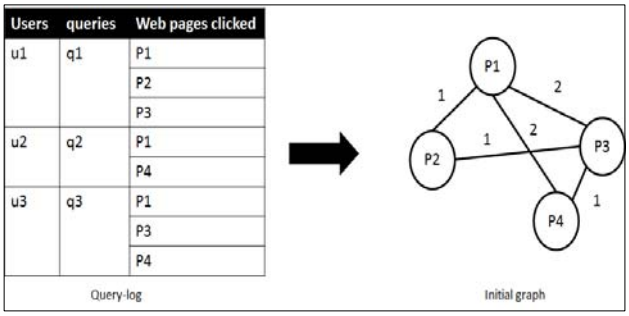


Figure 1. Example of an initial graph constructed from a query-log.

Analysis based IPC for the first adjustment, we build clusters using formal concept analysis [2] by extracting concepts from the Galois lattice. From the labeled graph, we construct a formal context whose objects and items are web pages (Vertices), and relationships are clicks (Edges). From that formal context, we extract concepts and build a Galois lattice using Formal Concept Analysis [2]. Since enumerating all concepts is computationally expensive, and concepts of the second level (the level above the bottom concept) are smaller and more concise than those of other levels, we extract all concepts of this level. We consider those concepts as the set of clusters. For each cluster in this set, we assign to all its pages the category that has a maximum score within that cluster. We define the score of a category c in a cluster C as follows: $S_C(c) = \sum_{p/class(p)=c} C_D(p,c)$ where $C_D(p,c)$ is the category based centrality degree of the web page p and the category c. This degree corresponds to the centrality degree [16], [17] of the web page p, computed using only neighbors sharing the same category with p. This is simply the number of links in the labeled graph, connecting the vertex that corresponds to the target web page, to other vertices having the same label. We can say that $S_C(c)$ corresponds to the sum of category based centrality degrees of web pages belonging to the category c in the cluster C. We call this voting process Internal Propagation of Categories (IPC). After this phase, we obtain a first updated labeled graph where classes of web pages have been updated according to the previously described process. Fig. 3 contains the formal context with its Galois lattice constructed from the graph in Fig. 1.

Finally, in the fourth stage called EPC for the second adjustment, we perform a process called External Propagation of Categories (EPC). For each target web page p in the labeled graph, we give a score to each category c. This score is the sum of weights of links wiring the target web page p to web pages belonging to the category c. Then, p is assigned to the category that has the highest score. We repeat this process iteratively until either all labels in the labeled graph remain the same or the maximum number of iterations is reached. After updating all the labels in this graph, we obtain a second updated labeled graph. The class c assigned to a web page p is given by the following model:

$$\arg\max_{c}(g(p,c))$$

Where:

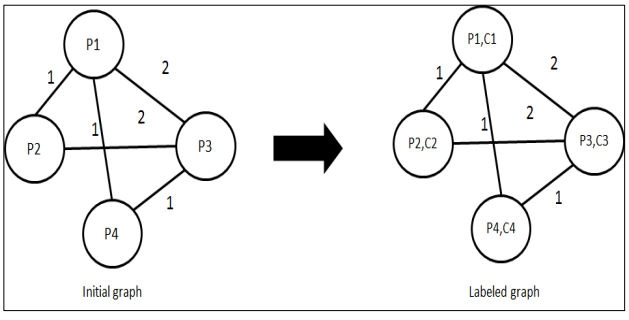


Figure 2. Labeled graph built using the initial graph of Fig. 1.

$$g(p,c) = \sum_{p' \in Neighbors(p)} f(p',c) * W(p,p')$$

$$f(p,c) = \begin{cases} 1 \text{ if class(p)=c} \\ 0 \text{ if class(p)} \neq \text{c} \end{cases}$$

$Neighbors(p)$ is the set of neighbors of p according to the labeled graph.

$W(p,p')$ is the weight of the edge linking p and p' in the labeled graph.

The labels in EPC can be updated synchronously or asynchronously. In synchronous updating, every vertex at iteration (t+1) is updated based on the labeled graph state at the end of iteration t. However, in asynchronous updating, vertices are updated sequentially. First, a vertex v_1 is updated. Then, a vertex v_2 is updated (using the new value of v_1), and so on.

Our approach converged for all classifiers employed and all datasets used, when the updating mode is asynchronous. However, for the synchronous updating mode, the approach did not converge in some cases (when the base classifier is for example SVM). Since the convergence is not guaranteed, we arbitrary set a maximum number of iterations to 10 for both updating modes.

Fig. 4 summarizes our proposed approach.

V. Experiments

In this section, we show the experiments conducted to validate the advantage of our approach. We describe the experimental design followed, i.e. preprocessing techniques used, classifiers applied, datasets chosen, evaluation metrics utilized, and results obtained using those evaluation measures with a discussion.

A. Experimental Setup

a. Pre-processing

We applied a number of preprocessing techniques to each web page in the dataset. The aim of those techniques is cleaning and normalizing the raw text contained in these web pages. In tokenization step we turn all terms to lower case, we removed some special characters, punctuation marks and numbers. Also, we removed all scripts, styles, mimes headings and HTML tags. For stemming process, we applied the well-known Porter method [18]. After the preprocessing stage, we build the dictionary which consists of words resulting from pre-processing. Thus, we consider web pages as bags of words. We rep-

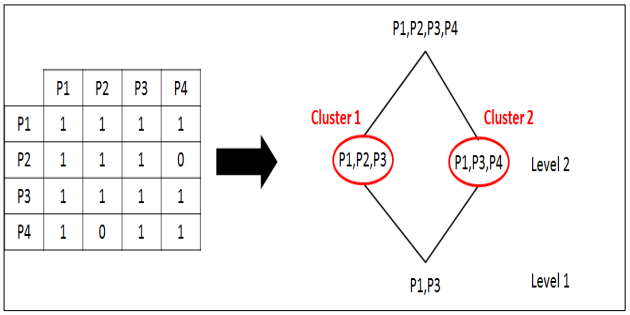


Figure 3. The formal context and the Galois lattice of the initial graph of Fig. 1.

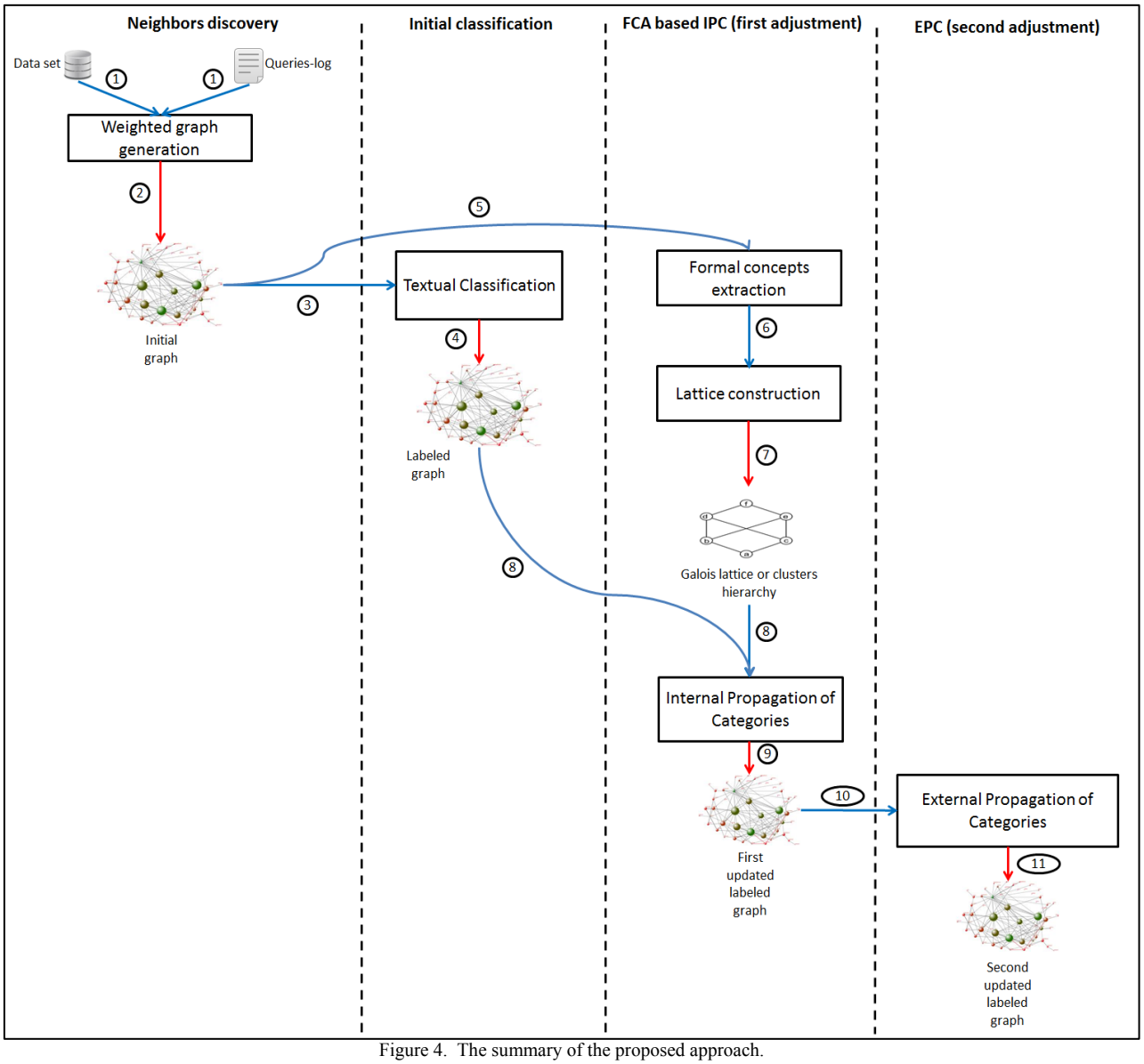


Figure 4. The summary of the proposed approach.

resent our web pages using the conventional Vector Space Model [19]. We associate each web page p to its vector $V_p = (n_{1p}, n_{2p}, ..., n_{mp})$; where n_{ip} denotes the weight of the i^{th} term in the web page p. We adopted TF-IDF [20], [21] based weighting model to obtain the weights:

$$n_{ip} = TF_{ip} \times IDF_i$$

$$IDF_i = \log(\frac{N}{df_i})$$

Where TF_{ip} is the frequency of the i^{th} term in the web page p, N is the number of target web pages, and df_i is the number of web pages where the i^{th} term appears.

b. *Classifiers used*

In order to evaluate our approach, we used four different classifiers from three different categories: Eager classifiers (Naïve Bayes and Support Vector Machine), lazy classifiers (K Nearest Neighbors) and Collective classifiers (Iterative Classification).

Support vector machine

Support Vector Machines (SVM) [22] is a performant learning algorithm that works well in text classification [23]. It is a large margin classifier that aims at minimizing the generalization error in order to avoid overfitting. From multiple versions of SVM described in [24], we used Sequential Minimal Optimization (SMO) version which was developed in [25], [26]. We use C=1 for the tolerance degree to errors. In addition, we use a linear kernel that proves to be efficient for text classification, where we have high features vector dimension [23].

Naïve Bayes

Naïve Bayes (NB) is a simple and very known classification algorithm [27], [28]. It uses the joint probabilities of attributes and classes, to estimate the probabilities of categories given a document and assumes that features are conditionally independent of each other to make the computation of joint probabilities simple.

K Nearest Neighbors

K Nearest Neighbours (KNN) is the simplest classification algorithm in the state of the art [29]. It is a lazy learner that predicts the category of an instance based on its K nearest training samples in the features space based on an inter-instance similarity. This algorithm does not generate a model from training instances but rather stores all those training examples directly and uses them to determine the class of a new instance.

Iterative classification

Iterative classification algorithm (ICA) [12] is widely used algorithm for collective classification, which refers to the categorization of interrelated documents described as vertices of a graph. In this method, the graph's vertices are classified at each iteration using currently predicted labels of their neighbors. ICA uses a local classifier to do both the classification and the inference. In this paper, we evaluated our approach with two implementations of ICA: the first uses SVM as local classifiers (ICA_{SVM}) and the second uses Naïve Bayes (ICA_{NB}). In our experiments, we use N=10 as ICA's maximum number of iterations.

c. Datasets

We test our approach using four binary problems, but our method can be generalized to multi-labels classification problems, by applying one against others classification.

Datasets used in this paper are taken from the Open Directory Project (ODP) [3] and AOL query-log [4]. ODP is a tremendous repository containing around 4.6 million web pages and is organized into 765,282 categories and subcategories [30]. We constructed four binary classification tasks as shown in Table 1: "Adult" vs. "Other" (1606 web pages), "KidsAndTeens" vs. "Other" (1591 web pages), "Health" vs. "Other" (1749 web pages), "Games" vs. "Other" (2012 web pages). AOL is a query-log that contains a collection of around 20 million web queries collected from 650000 users during three months [5].

Our approach needs some web pages to train content based classifiers and some web pages to test the method. We conduct all our experiments using 10-fold cross validation [31]. We used one fold for training and the nine others for testing so that the number of unlabeled web pages will be much greater than labeled ones.

To evaluate the effect of the size of the query-log on our approach, we divided the log into five distinct parts; each contains 20% of the entire query-log. First we use our approach with only one part and record the performance. Next, we add another part and record the performance again. We repeat that process until no part remains.

TABLE 1 :
DATASETS STATISTICS.

Dataset	Total number of web pages (n)	Total numbers of links (m)	Density $\left(\frac{2m}{n(n-1)}\right)$
Games	2012	5894	0.29%
Health	1749	4298	0.28%
Adult	1606	5768	0.45%
KidsAndTeens	1591	3786	0.30%

d. Evaluation measures

To evaluate results obtained using our approach, we use the standard metrics: recall, precision and F1, which are commonly used to evaluate the classification task. Recall is defined to be the number of correct assignments by the classifier divided by the total number of correct assignments. Precision is the number of the classifier correct assignments divided by the total number of the classifier's assignments. F1, introduced by Van Rijsbergen [32] is obtained using the following formula:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

B. Results and Discussion

In this section, we show the experiments' results of our method obtained on four binary subsets of ODP. We discuss results according to five points related to SVM, NB, KNN, ICA_{SVM} and ICA_{NB} base classifiers supported by our corrective approach. The first and second points pertain to global effects of respectively IPC and EPC on classification results of every base classifier (SVM, NB, KNN, ICA_{SVM} and ICA_{NB}). The third point discusses the effect of EPC updating mode (Synchronous/Asynchronous) on classification performances. The fourth point is a comparison of results obtained using IPC+EPC and those obtained using CLN. Finally, the fifth point is related to the effect of the query-log size and the weighted graph density on classification's performances obtained using our corrective approach.

- Table 2 shows that the use of our FCA based clustering technique to adjust web pages' classes (SVM+IPC, NB+IPC, KNN+IPC, ICA_{SVM}+IPC and ICA_{NB}+IPC), ameliorates results of SVM, NB, KNN, ICA_{SVM} and ICA_{NB} for Adults, KidsAndTeens and Games datasets. This is because each cluster groups web pages that are strongly related from users' perspectives. Also, the weighting of the vote of each web page based on its centrality degree allows pertinent web pages (within a cluster) to have a stronger influence in the cluster's category determination process. Thus, the use of the IPC helps the classifier to improve its performance. For Health dataset, the IPC decreases performances of the base classifier when using NB, SVM and ICA_{SVM}. This is caused, as shown in Table 1, by the fact that the density of the weighted graph is low for that dataset. Indeed, since our clustering is based on links, the number of links has an influence on clusters' number.
- As shown in Table 2, our proposed approach significantly ameliorates results of base classifiers for almost all datasets. This observation proves that the use of implicit links to propagate classes over web pages in order to update their categories helps improve results of text classifiers. This is because there are pages that are assigned to wrong classes by the initial classifier. However, the pages that are related implicitly to them from the user's perspective help correcting their classes and hence, im-

proving the performance. Exceptionally, for the Health dataset and when using SVM, KNN or ICA_{SVM}, EPC decreases performances obtained using content based classifiers. This is mainly caused by the fact that EPC is done after IPC, and this latter decreased the performance because of the raison explained in the previous point.

In general, EPC improves results obtained by IPC. Exceptionally, when using KNN as a base classifier, the EPC does not ameliorate IPC performances on Health dataset. This is mainly caused by the bad F1 score obtained by initial classifier. Indeed, many web pages in the graph are misclassified. Thus, EPC propagates errors and declines results rather than ameliorating them.

TABLE 2:
PERFORMANCE SUMMARY OF OUR APPROACH.

	Adults			KidsAndTeens			Health			Games		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
NB	0.892	0.803	0.845	0.67	0.739	0.703	0.901	0.89	0.895	0.818	0.849	0.833
NB+IPC	0.904	0.863	0.883	0.696	0.781	0.736	0.866	0.904	0.885	0.821	0.862	0.841
NB+IPC+EPC (Asynchronous)	0.896	0.933	0.914	0.711	0.872	0.783	0.885	0.924	0.904	0.865	0.888	0.876
NB+IPC+EPC (Synchronous)	0.896	0.935	0.915	0.71	0.875	0.784	0.878	0.923	0.9	0.86	0.887	0.873

(a) With Naïve Bayes as base classifier

	Adults			KidsAndTeens			Health			Games		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SVM	0.833	0.805	0.819	0.67	0.703	0.686	0.93	0.937	0.933	0.856	0.92	0.887
SVM+IPC	0.853	0.869	0.861	0.687	0.765	0.724	0.889	0.929	0.909	0.846	0.898	0.871
SVM+IPC+EPC (Asynchronous)	0.854	0.941	0.895	0.703	0.85	0.77	0.9	0.941	0.92	0.879	0.906	0.892
SVM+IPC+EPC (Synchronous)	0.856	0.941	0.896	0.701	0.852	0.769	0.893	0.942	0.917	0.878	0.9	0.889

(b) With SVM as base classifier

	Adults			KidsAndTeens			Health			Games		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
KNN	0.623	0.682	0.651	0.546	0.712	0.618	0.718	0.365	0.484	0.466	0.885	0.611
KNN+IPC	0.656	0.773	0.71	0.559	0.738	0.636	0.712	0.372	0.489	0.471	0.913	0.621
KNN+IPC+EPC (Asynchronous)	0.687	0.895	0.777	0.573	0.764	0.655	0.721	0.35	0.471	0.468	0.943	0.626
KNN+IPC+EPC (Synchronous)	0.688	0.9	0.78	0.572	0.766	0.655	0.718	0.348	0.469	0.465	0.943	0.623

(c) With KNN as base classifier

	Adults			KidsAndTeens			Health			Games		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
ICA_{SVM}	0.852	0.826	0.836	0.692	0.729	0.71	0.931	0.937	0.934	0.858	0.925	0.89
ICA_{SVM} +IPC	0.905	0.921	0.913	0.781	0.81	0.795	0.912	0.949	0.93	0.902	0.932	0.917
ICA_{SVM}+IPC+EPC (Asynchronous)	0.913	0.937	0.925	0.792	0.823	0.807	0.911	0.948	0.929	0.905	0.933	0.919
ICA_{SVM}+IPC+EPC (Synchronous)	0.913	0.937	0.925	0.792	0.823	0.807	0.911	0.948	0.929	0.905	0.933	0.919

(d) With ICA_{SVM} as base classifier

	Adults			KidsAndTeens			Health			Games		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
ICA_{NB}	0.901	0.86	0.88	0.697	0.766	0.73	0.902	0.885	0.893	0.83	0.861	0.845
ICA_{NB} +IPC	0.885	0.938	0.911	0.766	0.84	0.801	0.882	0.912	0.897	0.887	0.916	0.901
ICA_{NB} +IPC+EPC (Asynchronous)	0.889	0.952	0.919	0.775	0.851	0.811	0.883	0.92	0.901	0.89	0.921	0.905
ICA_{NB} +IPC+EPC (Synchronous)	0.889	0.952	0.919	0.775	0.851	0.811	0.883	0.92	0.901	0.89	0.921	0.905

(e) With ICA_{NB} as base classifier

- As shown in Table 2, both of updating modes give good results. The results obtained after performing EPC using synchronous updating and after performing EPC using asynchronous updating mode are always close to each other. Hence, updating modes have no effect on results provided by our approach.
- From Table 3, even though CLN gives better precision than our proposed approach, recalls of our approach are much better than those of CLN. Recall amelioration rates made by our approach range between 68.71% and 234.17%. The CLN uses only manually labeled neighbors to classify a target web page. Therefore, this leads to good precisions. However, many web pages could not have labeled web pages in their neighborhood. Thus, they will be unlabeled and lead to very low recalls and consequently low F1 scores.
- Based on Fig. 5, we observe that the performance improvement, brought by our approach, increases proportionally with the size of the query- log and the density of the weighted graph. This support the conclusion, drawn in [7]–[9] which states that the size of the query-log positively influences performances of approaches that are based on it. For Health dataset and especially when using KNN as a base classifier, the performance decreases with the growth of the query log size and the weighted graph's density. This is because the performance of the initial classifier (KNN) on that dataset is very low (F1=0.484). Thus, with the growth of the graph's density, more links will be added to the graph and then, errors will be propagated to more web pages.

V. Conclusion And Perspective

In this article, we introduced a new technique that helps improve results of a text classifier in web page classification. This method operates in four steps and leverages implicit links built using the query-log. The experimental results show that, within appropriate empirical setting, our scheme improves performance of SVM, NB, KNN, ICA based on SVM and ICA based on NB classifiers by two successive adjustments of categories assignments. Our main findings include:

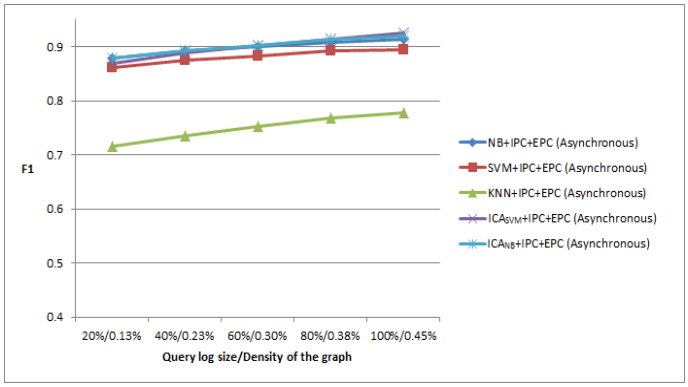
1. The FCA, based on implicit links, generates clusters whose web pages are topically related to each other. This explains the classification performance improvements brought by the IPC through correction of wrongly assigned categories. The weighting scheme, based on implicit links, helps the EPC making better categories tuning. This is shown in additional enhancement in classification performances provided by EPC.
2. Both the number of links in the graph (graph's density) and the query log size have an influence on the performance of our proposed approach. This latter performs better when the graph is dense.

We hope that our investigation will assist future researchers to see how this approach will help improving classifiers other than SVM, NB, KNN and ICA. Also, the use of a technique other than clustering, that takes profit from the implicit links can be prominent.

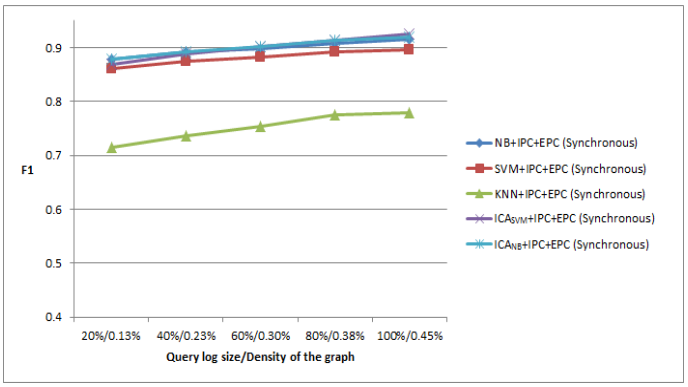
In the future, we will combine our approach with the use of other semantic information contained in web pages. We will also try to use hyperlinks instead of implicit links to see how they will contribute if used in the same way. Furthermore, we will use hyperlinks along with implicit links to see if they will improve results of the classification.

TABLE 3:
PERFORMANCE OF OUR APPROACH CONTRASTED TO CLN.

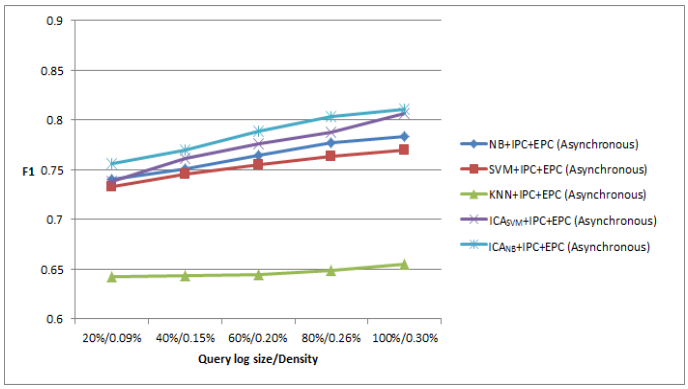
	Adults			KidsAndTeens			Health			Games		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SVM+IPC+EPC (Asynchronous)	0.854	0.941	0.895	0.703	0.85	0.769	0.899	0.941	0.92	0.879	0.906	0.892
SVM+IPC+EPC (Synchronous)	0.856	0.941	0.896	0.701	0.852	0.769	0.893	0.942	0.917	0.878	0.9	0.888
NB+IPC+EPC (Asynchronous)	0.896	0.933	0.913	0.711	0.872	0.783	0.885	0.924	0.904	0.865	0.888	0.876
NB+IPC+EPC (Synchronous)	0.896	0.935	0.914	0.71	0.875	0.783	0.878	0.923	0.9	0.86	0.887	0.873
KNN+IPC+EPC (Asynchronous)	0.687	0.895	0.777	0.573	0.764	0.655	0.721	0.35	0.471	0.468	0.943	0.626
KNN+IPC+EPC (Synchronous)	0.688	0.9	0.78	0.572	0.766	0.655	0.718	0.348	0.469	0.465	0.943	0.623
ICA_{SVM} +IPC+EPC (Asynchronous)	0.913	0.937	0.925	0.792	0.823	0.807	0.911	0.948	0.929	0.905	0.933	0.919
ICA_{SVM} +IPC+EPC (Synchronous)	0.913	0.937	0.925	0.792	0.823	0.807	0.911	0.948	0.929	0.905	0.933	0.919
ICA_{NB} +IPC+EPC (Asynchronous)	0.889	0.952	0.919	0.775	0.851	0.811	0.883	0.92	0.901	0.89	0.921	0.905
ICA_{NB} +IPC+EPC (Synchronous)	0.889	0.952	0.919	0.775	0.851	0.811	0.883	0.92	0.901	0.89	0.921	0.905
CLN	0.964	0.244	0.389	0.879	0.147	0.252	0.922	0.164	0.278	0.957	0.17	0.288



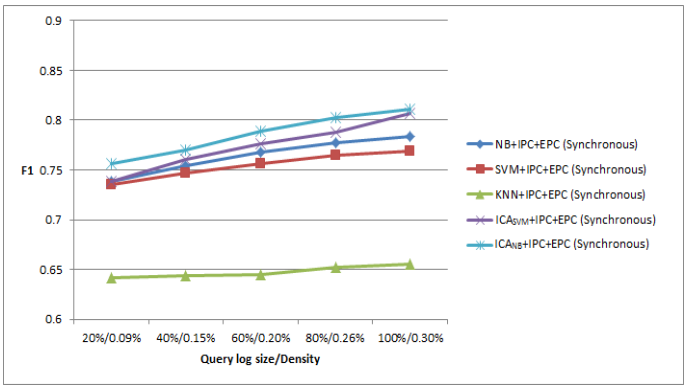
(a) Adult with asynchronous updating



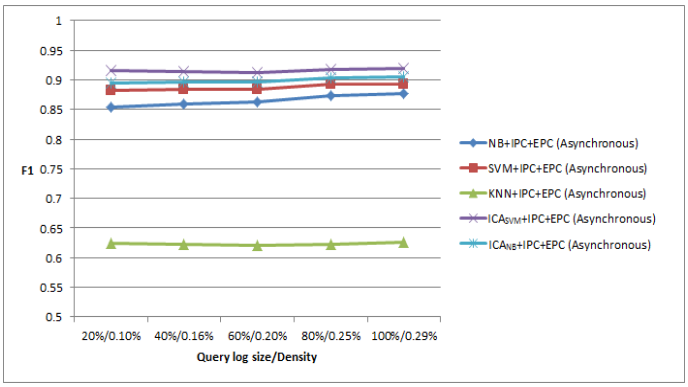
(b) Adult with synchronous updating



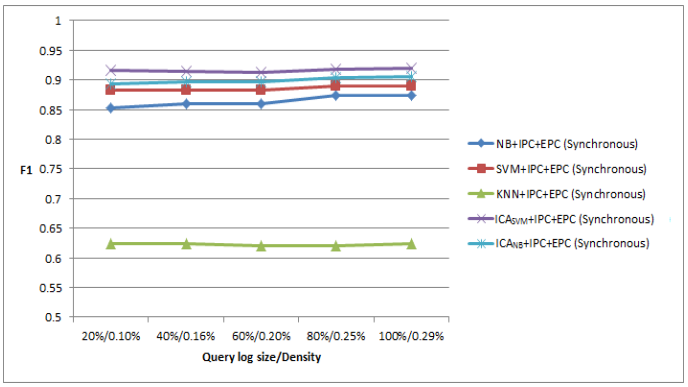
(c) KidsAndTeens with asynchronous updating



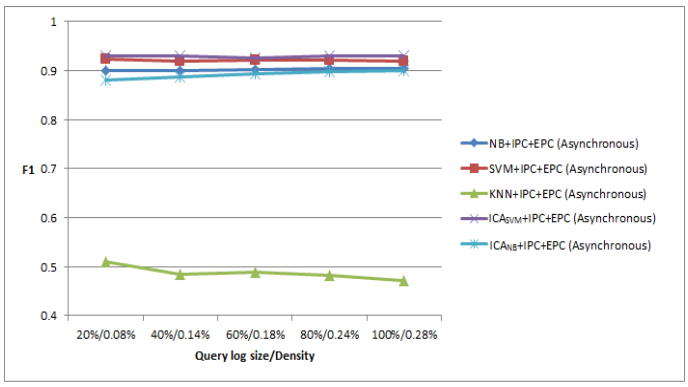
(d) KidsAndTeens with synchronous updating



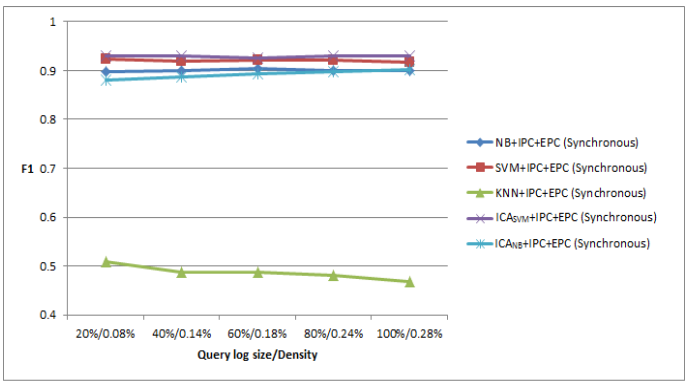
(e) Games with asynchronous updating



(f) Games with synchronous updating



(g) Health with asynchronous updating



(h) Health with synchronous updating

Figure 5: The effect of the query-log's size and the weighted graph's density on our approach.

REFERENCE

[1] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," *SIGMOD Rec.*, vol. 27, no. 2, pp. 307–318, Jun. 1998.

[2] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[3] "ODP - Open Directory Project." [Online]. Available: http://www.dmoz.org/. [Accessed: 24-Feb-2013].

[4] "AOL search data mirrors." [Online]. Available: http://gregsadetsky.com/aol-data/. [Accessed: 09-Aug-2013].

[5] "AOL Search Query Logs - RP." [Online]. Available: http://www.researchpipeline.com/mediawiki/index.php?title=AOL_Search_Query_Logs. [Accessed: 09-Aug-2013].

[6] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of carefull seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[7] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen, "A comparison of implicit and explicit links for web page classification," in *Proceedings of the 15th international conference on World Wide Web*, New York, NY, USA, 2006, pp. 643–650.

[8] S.-M. Kim, P. Pantel, L. Duan, and S. Gaffney, "Improving web page classification by label-propagation over click graphs," in *Proceedings of the 18th ACM conference on Information and knowledge management*, New York, NY, USA, 2009, pp. 1077–1086.

[9] G.-R. Xue, Y. Yu, D. Shen, Q. Yang, H.-J. Zeng, and Z. Chen, "Reinforcing Web-object Categorization Through Interrelationships," *Data Min. Knowl. Discov.*, vol. 12, no. 2–3, pp. 229–248, May 2006.

[10] W. Dai, Y. Yu, C.-L. Zhang, J. Han, and G.-R. Xue, "A novel web page categorization algorithm based on block propagation using query-log information," in *Proceedings of the 7th international conference on Advances in Web-Age Information Management*, Berlin, Heidelberg, 2006, pp. 435–446.

[11] D. Jensen, J. Neville, and B. Gallagher, "Why collective inference improves relational classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2004, pp. 593–598.

[12] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-rad, "Collective classification in network data," 2008.

[13] R. Liu, J. Zhou, and M. Liu, "Graph-based Semi-supervised Learning Algorithm for Web Page Classification," in *Sixth International Conference on Intelligent Systems Design and Applications, 2006. ISDA '06*, 2006, vol. 2, pp. 856–860.

[14] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene Labeling by Relaxation Operations," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, no. 6, pp. 420–433, 1976.

[15] R. Angelova and G. Weikum, "Graph-based text classification: learn from your neighbors," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2006, pp. 485–492.

[16] S. Wasserman and K. Faust, *Social network analysis: methods and applications*. Cambridge; New York: Cambridge University Press, 1994.

[17] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, p. 215, 1978.

[18] M. F. Porter, "Readings in information retrieval," K. Sparck Jones and P. Willett, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 313–316.

[19] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[20] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, Aug. 1988.

[21] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

[22] C. Cortes and V. Vapnik, "Support-Vector NSetworks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[23] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *Machine Learning: ECML-98*, C. Nédellec and C. Rouveirol, Eds. Springer Berlin Heidelberg, 1998, pp. 137–142.

[24] C.-J. Lin, "Asymptotic convergence of an SMO algorithm without any assumptions," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 248–250, 2002.

[25] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[26] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO Algorithm for SVM Classifier Design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, Mar. 2001.

[27] T. M. Mitchell, *Machine Learning*, 1st ed. McGraw-Hill Science/Engineering/Math, 1997.

[28] A. McCallum and K. Nigam, *A comparison of event models for Naive Bayes text classification*. 1998.

[29] D. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[30] L. Henderson, "Automated Text Classification in the DMOZ Hierarchy." 06-Nov-2009.

[31] F. Mosteller and J. W. Tukey, "Data Analysis, Including Statistics," in *Handbook of Social Psychology (G. Lindzey and E. Aronson, eds.)*, 2nd ed., vol. 2, Addison-Wesley, Reading, MA, pp. 80–203.

[32] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Butterworth-Heinemann, 1979.