Evaluation Methods for Focused Crawler: An Overview

[1]Dr. Vivek Chandra, [2]Ms. Nidhi Saxena

[1]GM & Head-IT, MPPKVVCL, Jabalpur, (M.P), INDIA
[2]Programmer, RDVV, Jabalpur (M.P), INDIA

Abstract: **The exponential growth of documents available in the World Wide Web makes it ever more difficult to discover relevant information on a specific topic. In this context, growing interest is emerging in focused crawling, a technique that dynamically browses the Internet by choosing directions that maximize the probability of discovering relevant pages, given a specific topic. Predicting the relevance of a document before seeing its contents (i.e., relying on the parent pages only) is one of the central problems in focused crawling because it can save significant bandwidth resources. This paper gives an overview of the various evaluating methods and technique that can be used for focused crawlers. The ultimate aim of these techniques is to predict the values of a discrete class attribute.**

Keywords: **DP, Focused Crawler, Classifier, Deep Web, WEB 2.0**

I. INTRODUCTION

Today we can't imagine the Web without search engines. The World Wide Web (WWW) is a popular and interactive medium of fertile information these days, this information is spread on various servers all over the world; therefore finding the relevant and personalized information on web is a tough task to carry out. Hence an intelligent document discovery mechanism is required. Data mining techniques could be used to solve this problem.

Tools for the assessment of the quality and reliability of web applications are based on the possibility of downloading the target of the analysis. The most important performance indicator for a focused crawler is its completeness, efficiency, scalability, Interpretability and robustness, measuring respectively the ability to visit the Web site entirely and without errors.

Structure of this paper as follow: In section 2 we have given a common architecture of focused crawler and its working. Why were they developed and for what reason. In section 3 we explained crawler evaluation method and measures and in section 4 the text classification learning technique has been dealt. Section 5 focuses on the problems of web crawling, in particular the deep Web and Web 2.0. Section 6 summarizes this paper and provides some conclusions about the topical web crawlers.

II. ARCHITECTURE

A focused crawler attempts to bias the crawler towards pages in certain categories in which the user is interested. Chakrabarti et al. [1] proposed a focused crawler based on a classifier. The idea is to first build a text classifier using labeled example pages from, say, the ODP[1]. Then the classifier would guide the crawler by preferentially selecting from the seed those pages that appear most likely to belong to the categories of interest, according to the classifier's prediction. Fig. 1 shows the common architecture of focused/topical crawler.

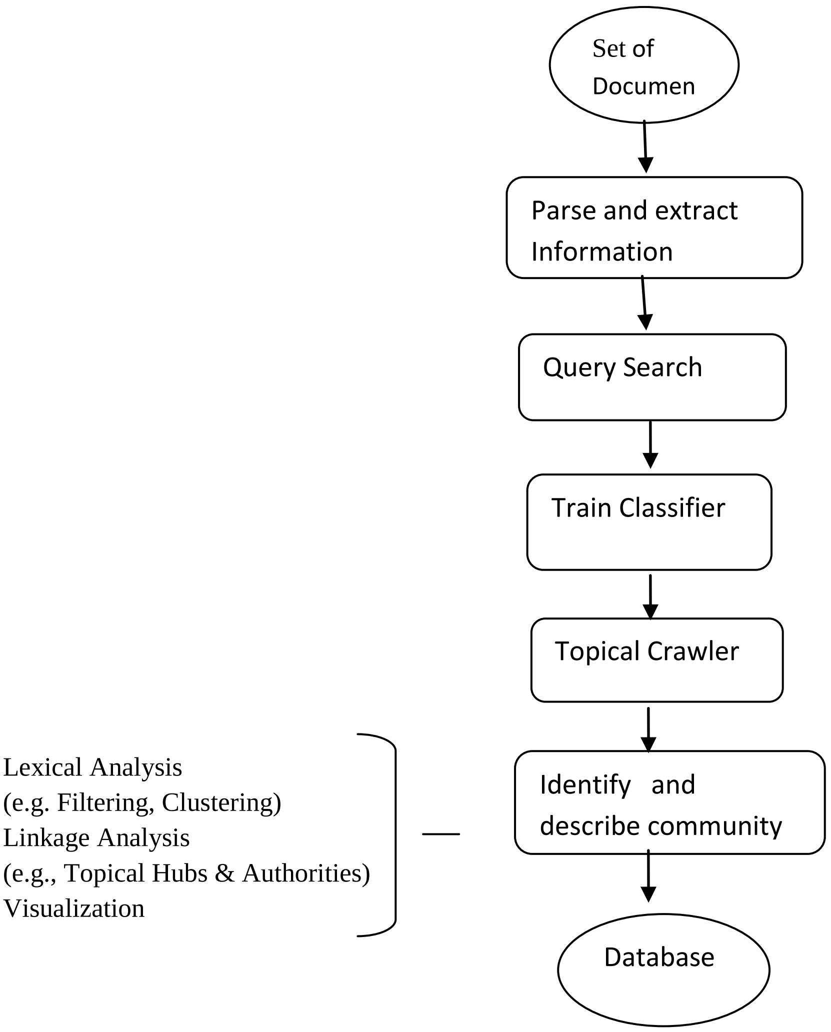
Set of Documen
Parse and extract Information
Query Search
Train Classifier
Topical Crawler
Identify and describe community
Database
Lexical Analysis
(e.g. Filtering, Clustering)
Linkage Analysis
(e.g., Topical Hubs & Authorities)
Visualization

Fig 1: COMMON ARCHITURE OF FOCUSED CRAWLER

Fig. 1 illustrates general approach. First Given a document or a set of documents (seed) from already manual created corpus , we first parse and extract information. The extracted information may include the title, author names, affiliations, abstract, keywords (if provided) and references. We then use this information to characterize the document or the set of documents. In the case where we have more than one document we may correlate the information from different documents to find common or overlapping characteristics. For example, we can identify words or phrases that are commonly used, or the references that are popularly cited in the given set. The information extracted through parsing is used to query a search engine. For example, we can use the title and the author names as individual queries to a search engine. The pages corresponding to the top results of the search engine are then treated as positive examples of the desired information (there may be a filtering step to avoid bad examples). The positive examples are used to train a classifier. The negative examples for the classifier may be picked from positive examples obtained from unrelated or random documents. Once the classifier is trained, it is used to guide a topical crawler. A topical or focused crawler is a program that follows hyperlinks to automatically retrieve pages from the Web while biasing its search towards topically relevant portions of the Web. The trained classifier will provide the crawler with the needed bias. Once a collection of Web pages has been downloaded by the crawler, we analyze them to find more structured information such as potential Web communities and their descriptions. The analysis process includes both lexical as well as link (graph) based analysis.

There are two approach for developing intelligent agent that help user find and retrieve relevant information from the web i.e. content-based approach and collaborative approach. In this content based approach, the system search for items that match based on an analysis of content using the user preferences. In the collaborative approach, the system tries to find user with similar interest to give recommendation to.

Several research issues around topical crawlers have received attention. One key question is how to identify the environmental signals to which crawlers should attend in order to determine the best links to follow. Rich cues such as the markup and lexical (text) signals within Web pages, as well as features of the link graph built from pages already seen, are all reasonable sources of evidence to exploit.

III. CRAWLING EVALUATION FRAMEWORK

A computer does not have “experiences”, like human learning from past experiences. A computer system learns from data, which represent some “past experiences” of an application domain for a target function that can be used to predict the values of a discrete class attribute. This task is commonly called: Supervised learning, classification, or inductive learning.

The major points which are required to be considered to evaluate a crawler:

- Whether to put a value in true or false classes, a decision is needed.
- How to predict class true value discriminate them from false value.

Supervised learning having two step process

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy

The learned model helps the system to perform better real world task as compared to no learning.

A. Learning Techniques for classification

Decision tree learning is one of the most widely used techniques for classification. Its classification accuracy is competitive with other methods, it is very efficient. The learned classification model is represented as a tree, called a decision tree. A decision tree can be converted to a set of rules. Each path from the root to a leaf is a rule. A Greedy divide-and-conquer algorithm is used for decision tree learning. The most popular impurity functions used for decision tree learning are information gain and information gain ratio.

There several issues in decision tree learning like Tree Pruning and Overfitting, Handling Missing Attribute Values, Handling Skewed Class Distribution etc.

B. Classifier Evaluation

After a classifier is constructed, it needs to be evaluated for accuracy. Effective evaluation is crucial because without knowing the approximate accuracy of a classifier, it cannot be used in real-world tasks. There are many ways to evaluate a classifier, and there are also many measures.

The main measure is the classification **accuracy** which is the number of correctly classified instances in the test set divided by the total number of instances in the test set. Some researchers also use the **error rate,** which is **1 – accuracy**.

Accuracy is only one measure (error = 1-accuracy). Accuracy is not suitable in some applications in text mining; we may only be interested in the documents of a particular topic, which are only a small portion of a big document collection. In classification involving skewed or highly imbalanced data, e.g., network intrusion and financial fraud detections, we are interested only in the minority class. High accuracy does not mean any intrusion is detected. e.g, 1% intrusion. Achieve 99% accuracy by doing nothing. The class of interest is commonly called the positive class, and the rest negative classes.

IV. EVALUATION METHODS

A. Holdout Set

The available data D is divided into two disjoint subsets, the **training set** D_{train} *(for learning model)* and the **test set** D_{test} (for testing model), $D = D_{train} \cup D_{test}$ and $D_{train} \cap D_{test} = \varnothing$. The test set is also called the holdout set. This method is mainly used when the data set D is large.

B. Precision, Recall, F-score and Breakeven Point

Precision and **recall** are more suitable in such applications because they measure how precise and how complete the classification is on the positive class. It is convenient to introduce these measures using a **confusion matrix** (Table 1). A confusion matrix contains information about actual and predicted results given by a classifier.

Table 1: Confusion Matrix

	Predicted Positive	Predicted Negative
Actual positive	TP	FN
Actual negative	FP	TN

Where

TP: the number of correct classifications of the positive examples (**true positive**)

FN: the number of incorrect classifications of positive examples (**false negative**)

FP: the number of incorrect classifications of negative examples (**false positive**)

TN: the number of correct classifications of negative examples (**true negative**)

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}$$

Precision *p* is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

Recall *r* is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

A single measure to compare different classifiers, the **F-score** is often used. F_1-Score is the harmonic mean of precision and recall.

$$F_1 = \frac{2pr}{p+r} \quad ; \quad F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

There is also another measure, called precision and recall breakeven point, which is used in the information retrieval community. The break even point is when the precision and the recall are equal. This measure assumes that the test cases can be ranked by the classifier based on their likelihoods of being positive. For instance, in decision tree classification, we can use the confidence of each leaf node as the value to rank test cases

C. Receiver Operating Characteristic Curve (ROC)

A receiver operating characteristic (ROC) curve is a plot of the true positive rate against the false positive rate. It is also commonly used to evaluate classification results on the positive class in two-class classification problems. The classifier needs to rank the test cases according to their likelihoods of belonging to the positive class with the most likely positive case ranked at the top. The true positive rate (TPR) is defined as the fraction of actual positive cases that are correctly classified,

$$TPR = \frac{TP}{TP + FN}. \qquad FPR = \frac{FP}{TN + FP}.$$

TPR is basically the recall of the positive class and is also called sensitivity in statistics

D. Scoring and ranking

Instead of assigning each test instance a definite class, scoring assigns a probability estimate (PE) to indicate the likelihood that the example belongs to the positive class. Scoring is related to classification.

V. LEARNING METHOD FOR TEXT CLASSIFICATION

Due to the rapid growth of online documents in organizations and on the Web, automated document classification is an important problem. Although the techniques discussed in the previous sections can be applied to text classification, it has been shown that they are not as effective as the methods presented in this section.

A. Naïve Bayesian Text Classification

A text document consists of a sequence of sentences, and each sentence consists of a sequence of words. However, due to the complexity of modeling words sequence and their relationships, several assumptions are made in the derivation of the Bayesian classifier. That is also why we call the final classification model, the *naïve Bayesian classification model*. Specifically, the naïve Bayesian classification treats each document as a "bag" of words. The generative model also makes the following words and document length based assumptions:-

- Words of a document are generated independently of their context, that is, independently of the other words in the same document given the class label. This is the familiar naïve Bayesian assumption used before.
- The probability of a word is independent of its position in the document. For example, the probability of seeing the word "student" in the first position of the document is the same as seeing it in any other position.
- Document length is independent of the document class.

With the assumptions, each document can be regarded as generated by a multinomial distribution. Each document is drawn from a multinomial distribution of words with as many independent trials as the length of the document. We compute the probability that a particular mixture component c_j generated the given document d_i. Using the Bayes rule and Equations, we get

$$\Pr(c_j \mid d_i; \hat{\Theta}) = \frac{\Pr(c_j \mid \hat{\Theta}) \Pr(d_i \mid c_j; \hat{\Theta})}{\Pr(d_i \mid \hat{\Theta})}$$

$$= \frac{\Pr(c_j \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j; \hat{\Theta})}{\sum_{r=1}^{|C|} \Pr(c_r \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r; \hat{\Theta})}$$

Where $w_{di,k}$ is the word in position k of document d_i (which is the same as using wt and N_{ti}). If the final classifier is to classify each document into a single class, the class with the highest posterior probability is selected:

$$argmax\, c_j \ \in c \Pr\left(c_{j\,|d_j}; \, \Theta \right)$$

On naïve Bayesian classifier we say it is Easy to implement, Very efficient and Good results obtained in many applications but is not suitable for Assumption: class conditional independence, therefore loss of accuracy when the assumption is seriously violated (those highly correlated data sets).

B. Support Vector Machines (SVM)

Support vector machines were invented by V. Vapnik and his co-workers in 1970s in Russia and became known to the West in 1992[2]. SVMs are linear classifiers that find a hyperplane to separate two class of data, positive and negative. Kernel functions are used for nonlinear separation. SVM not only has a rigorous theoretical foundation, but also performs classification more accurately than most other methods in applications, especially for high dimensional data. It is perhaps the best classifier for text classification [3].

Applying the Kurush-Kuhn-Tucker conditions, it is found that the training examples that influence the optimal decision boundary are the ones that are closest to it. These training points are called the support vectors and the resulting optimal hyperplane classifier a support vector machine. The optimal hyperplane thus found can be written as:

$$g(x) = w.x + w_0 = 0$$

Where x is a point on the plane, w are weights learned in the training process that determine the direction of the hyperplane, and w0 fixes the position of the hyperplane in space. Fig 2 shows the SVM looks for the separating hyperplane with the largest margin error bond. . SVM is also to detect whether a given page is blog page.

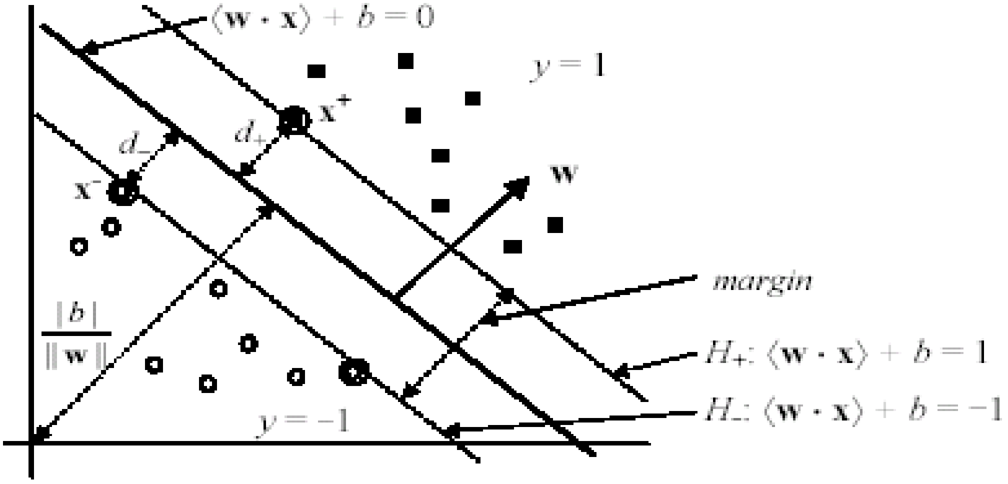
$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$
$y = 1$
$\mathbf{x}^+$
d_+
d_-
$\mathbf{x}^-$
$\mathbf{w}$
margin
$\frac{|b|}{\|\mathbf{w}\|}$
H_+: $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 1$
H_-: $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = -1$
$y = -1$

Fig 2: Maximal margin hyperplane

SVM has successful applications in many complexes, real-world problems such as text and image classification, hand-writing recognition, data mining, bioinformatics, medicine and bio-sequence analysis and even stock market.

C. Neural Networks

The word *network* in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations [4].
An ANN is typically defined by three types of parameters:

- The interconnection pattern between the different layers of neurons.
- The learning process for updating the weights of the interconnections
- The activation function that converts a neuron's weighted input to its output activation.

Neural networks are similar to biological neural networks in performing functions collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned. The term "neural network" sually refers to models employed in statistics, cognitive psychology and artificial intelligence.

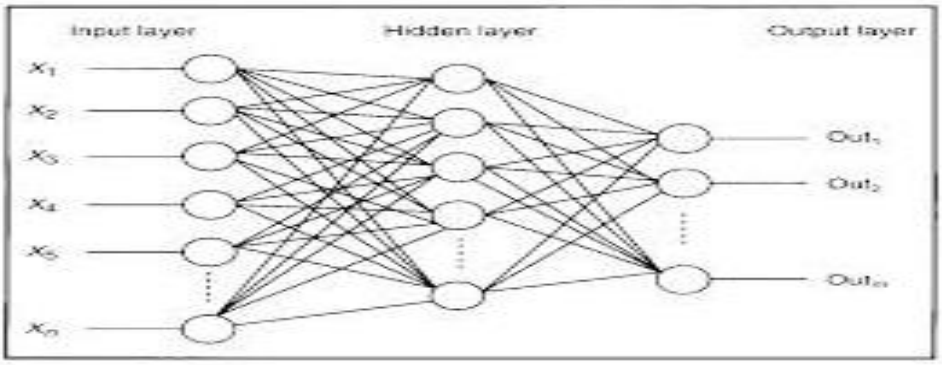

Fig 3: An artificial neural network

Neural network models which emulate the central nervous system are part of theoretical neuroscience and computational neuroscience. Fig 3 shows An artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

VI. PROBLEMS AND FUTURE PROSPECTS

A few crawling problems have already been mentioned in a previous section. Two huge current problems will be mentioned in this section: Crawling the deep Web and crawling AJAX based web contents.

A. Crawling the deep Web

The deep Web refers to WWW content, which is stored in backend databases and which dynamic web pages are built upon [5]. This content can only be accessed via submitting HTML forms which retrieve information from these hidden databases. That is the reason why the deep Web is also called the hidden Web. A deep Web crawler is able to do so and was first introduced in 2001[6]. In the following it is described how the crawler interacts with an HTML form and extracts information from a response page served from a hidden database. This ability is the major difference to a traditional web crawler. Although this crawler was invented in 2001 it demonstrates how a deep Web crawler works in general. In order to add only relevant pages crawled from the deep Web to Google's search engine index, query templates are evaluated according to the distinctness of the resulting web pages.

Julien Masanès in [8] surveys the field of Web archiving and Web archiving crawling. He discusses in particular crawling the deep Web and stresses the need of archiving the data from both the surface and deep Web for the necessity of Web preservation.

B. Crawling Web 2.0 Applications

Rich internet applications (RIAs) provides a new and excellent user experience evolved in the era of Web 2.0. Scanning or crawling such pages is a problem because of its complexity. Basically traditional crawlers fail in two ways trying to crawl RIAs: First of all browser behavior is hard to replicate and second, identifying key server side resources because these are accessed via Xml Http Request objects via JavaScript calls [7].

VII. CONCLUSION

In this paper the techniques deployed for evaluation of focused crawlers has been elaborated. The past years proved that the internet is not only growing, it is changing as well. Web 2.0 applications have emerged, invoking severe problems with web searching in general. Ways of encountering some of the problems have been discussed while some issues require further exploration, which leave this interesting field of science open for future research.

REFERENCES

[1] S. Chakrabarti, M. van der Berg, and B. Dom, Focused crawling: a new approach to topic-specific web resource discovery, Proceeding of the 8th International World Wide Web Conference (WWW8), 1999.

[2] Boser, B., I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of Fifth Annual Workshop on Computational Learning Theory, 1992.

[3] Data Mining Algorithms In R-Classification-penalized SVM - Wikibooks, open books for an open world.htm.

[4] Artificial Neural Networks Neural Network Basics - Wikibooks, open books for an open world.htm

[5] Bergman, M. K. (2001). The deep web: Surfacing hidden value. Journal of Electronic Publishing, 7(1).

[6] Raghavan, S. and Garcia-molina, H. (2001). Crawling the hidden web. In In VLDB, pages 129–138.

[7] Shah, S. (2006b). Vulnerability scanning web 2.0 client-side components. http:/ /www. Security focus. com/ infocus/ 1881.

[8] J. Masanès. Web archiving. Springer, 2006.