International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Context based document searching using Apriori Association for Mobile Devices

Ashwini Dhanne[Ȧ*], Geetanjali Kale[Ȧ] and Parag Kulkarni[Ḃ]

[Ȧ]Computer Engineering, Pune, Pune University, PICT, Pune,India.
[B]Computer Engineering, Eklat Research, Pune ,India.

Accepted 05 June 2014, Available online 20 June2014, **Vol.4, No.3 (June 2014)**

Abstract

Now a day's numbers of documents on Internet are increasing. Many traditional searching techniques may not return most accurate results in some cases. This can overcome by considering context of document i.e. content profile. In this paper we have proposed a system which tries to address this issue with novel method of content profiling. Proposed system is using Naïve Bayes text classifier and Apriori association rule to build content profile. This system accepts document from users and reply with set of similar documents based on context of document. Here we are considering document context of both input document and stored documents. This gives system where searching can also be done without using keywords.Content profile: Set of simple sentences that capture essence of document.

Keywords: *Clustering, classification, association rules, Feature extraction or construction, Information storage and retrieval, Content Analysis and Indexing, Information Search and Retrieval.*

1. Introduction

Because of mobility & advancements in network access technology of mobile device, that is widely used to access network. Many Advanced technologies are used to access documents from the network e.g. Wi-Fi, 3G, 2G, GPRS. Mobile devices have several limitations such as short battery life, less memory space and processing speed, less connection stability because of moving device. Because of these limitations it is necessary to improve document access and response process. As enormous amount of unstructured data is available on web sites. Retrieving relevant data from such a huge store is challenging task. Many of traditional document access techniques use only keywords. Which may not returns most accurate results in some cases. Because they don't consider context of document i.e. content profile.

We must look beyond the text in the document. The words that surround term help us understanding their content profile or the situation in which that specific term is being used. Many techniques are available for text data classification. We use Naïve Bayes classifier to classify the document at first level. We then find content profile by using Apriori based association rule mining to find words which occurs together. Last stage is reply document profile with rank. Ranking is based on closeness of document with query document. We are using cosine similarity formula to find closeness within document. Here we are maintaining user profile which is mapped with document profile. To give right user a right document (Goyal, P.*et al*, 2013), (Kulkarni A. R. *et al*., 2012),(F. Liu *et al*, 2004).

Our proposed system considers two cases. The first case is user inputs keyword, system reply document. The second case is user having a document and wants similar documents. User input document as a query and system gives similar set of documents as a reply. The system should select a certain amount of words as a query for searching a document. It is for two reasons, one is to reduce computational cost, and the other is that the unimportant words will degrade the similarity score. For query document, system performs indexing of query document, then selects frequent terms as a query. System refers to document context and user profile context while replying to user.

2. Related Work

2.1 Document Indexing

It is the activity of mapping document D_i into compact representation of its content. Indexing can be directly interpreted by classifier building algorithm and by classifier. In indexing, text $d_j$is represented by vector of term weights,

$d_j=\{w1_j,......wT_j\}$

Here T is dictionary i.e. set of Terms(also known as features) that occur at least ones in at least k documents and $0 \le wkj \le 1$ quantifies the importance of tk in characterizing the semantics of d_j . Typical values of k are between 1 and 5. An Indexing method is characterized by Term and the method which is used to compute Term weights. To select term in document is to identify words occurring in document (without stop words which are

*Corresponding author: **AshwiniDhanne** removed in preprocessing). Stems i.e. morphological roots can be obtained by applying stemming algorithm(Frakes*et al*, 1992). Term weights can be either binary *wkj={0,1}* or real valued w=0 ≤ *wkj* ≤ 1, depending on whether classifier and classifier building input type. When weights are binary it indicates presence/absence of term in document. When weights are real values they are calculated by either statistical or probalistic techniques (Debole*et al*,2003). Dimensionality reduction is done by feature selection techniques like mutual information (aka information gain) (Lewis D.D. *et al*,1994), chi square (Yang *et al*, 1997), or gain ration (Zobel*et al*, 1998), or feature extraction technique ,such as latent semantic indexing (Wiener E.D *et al*, 1995; Sch¨utze H *et al*, 1995) or term clustering(Lewis *et al*, 1992). Paper (Baker L.D *et al*, 1998 ; Bekkerman*et al*, 2001; Slonim*et al*, 2001) uses supervised term clustering techniques. Dynamic Inverted indexing technique is used for document indexing (Maxim Martynov*et al*, 1996)

2.2 Document Context or Document Content Profile

Context is information that characterizes event. Here document context gives abstract representation of document. The main goal of document context is to present main ideas in document/set of documents in short and readable paragraph. Lexical association is used on Bernoulli model of randomness, which gives context sensitive weight to document terms (PrathimaMadadi UNLV Theses/Dissertations/Professional Papers/Capstones). (Goyal*et al*. 2013) apriori algorithms frequent item set generation algorithm is used to find document context. (Kulkarni A. R. *et al*., 2012) Document context is generated using Naïve Bayes and Apriori association algorithm which gives result to extent.

2.3 User Profile Context

Traditional document retrieval model uses only user query relation information. It means that user information need is completely represented by user's query. But when same query is submitted by different users, search engine returns same result regardless of who submitted the query. This may not be suitable for users with different information needs. To overcome such problems, many recent works use user's profile features in order to re-rank document (Speretta S *et al*, 2004; Lewis D.D. *et al*,1994), to refine query (Goyal, P *et al*,2013) or to adapt the relevance function (J.P McGowan*et al* ,2003; T. Haveliwala*et al*,2002; F. Qiu*et al*, 2006).

Speretta, S. *et al*.,(Speretta, S. *et al*, 2004) proposed to model user's interests as weighted concept hierarchies extracted from user's search history. F. Liu *et al*., (F. Liu *et al*, 2004) user profiles are used to represent user's interests. A user profile consists of set of categories, and for each category, there is a set of weighted terms. Web Personae (J.P McGowan*et al* ,2003) is a browsing and searching assistant based on web usage mining. Recently, extensions of the Page Rank algorithm (F. Qiu*et al*, 2006;

T. Haveliwala*et al*,2002) have been proposed. This algorithm computes multiple scores, instead of just one, for each page, one for each topic listed in the Open Directory. W. NesrineZemrli*et al*., (W. NesrineZemirli*et al*, 2006) proposed the approach that integrates the user's long-term interests into a unified model of query evaluation.

2.4 Document Ranking

Luhn, H. P (Luhn, H. P. *et al*, 1957) proposed a statistical approach to search literary information. Maron and Kuhns (1960) went much further by suggesting how to actually weight terms, including some small-scale experiments in term-weighting.

2.4.1 Vector Space Model

Sample document and query can be envisioned as an *n*-dimensional vector space, where *n* corresponds to the number of unique terms in the data set. A vector matching operation, based on the cosine correlation used to measure the cosine of the angle between vectors, can then be used to compute the similarity between a document and a query, and documents can be ranked based on that similarity.

similarity $(d_j,q_k)= \frac{\sum_{i=1}^{n}(td_{ij}\times td_{ik})}{\sqrt{\sum_{i=0}^{n} td_{ij}^{2}\times\sum_{i=0}^{n} tq_{ik}^{2}}}$

where

td_{ij} = thei^{th}term in the vector for document *j*

tq_{ik}= the i^{th}term in the vector for query *k*

n = the number of unique terms in the data set

2.4.2. Probalistic Model

The major probabilistic model in use today was developed by Robertson and Sparck Jones (1976). This model is based on the premise that terms that appear in previously retrieved relevant documents for a given query should be given a higher weight than if they had not appeared in those relevant documents. In particular, they presented the following table showing the distribution of term *t* in relevant and non relevant documents for query *q*.

Document Relevance

Document Indexing

	+	-	
+	r	n - r	n
-	R - r	N-n-R+r	N- n
	R	N-R	N

N = the number of documents in the collection

R = the number of relevant documents for query *q*

n = the number of documents having term *t*

r = the number of relevant documents having term *t*

$$w^{1} = \log\frac{\left(\frac{r}{R}\right)}{\left(\frac{n}{N}\right)} \qquad w^{2} = \log\frac{\left(\frac{r}{R}\right)}{\left(\frac{n-r}{N-R}\right)}$$

$$w^{3} = \log\frac{\left(\frac{r}{R-r}\right)}{\left(\frac{n}{N-n}\right)} w^{4} = \log\frac{\left(\frac{r}{R-r}\right)}{\left(\frac{n-r}{N-n-R+r}\right)}$$

They then use this table to derive four formulas that reflect the relative distribution of terms in the relevant and non relevant documents, and propose that these formulas be used for term-weighting (the logs are related to actual use of the formulas in term-weighting).

2.5 Document Classification

To retrieve documents effectively it is necessary that document should be properly classified. For proper classification it is necessary to select effective classification method. There are many classification methods are in use they are Decision tree categorization, Rule based categorization, Neural networks, Support vector machine, Bayesian categorization are described below.

2.5.1 Decision tree categorization

Decision tree (H. Schutze*et al*, 1995) uses hierarchical structure for document classification. Each vertex of decision tree gives the attribute and edges of tree correspond to attribute condition. Leaf node assigns classification. Starting from root node, it searches for proper attribute to traverse tree till to get class. The division of the data space is performed recursively in the decision tree, until the leaf nodes contain a certain minimum number of records, or some conditions on class purity.

Most of the decision tree implementations in the text literature tend to be small variations on standard packages such as ID3 and C4.5, in order to adapt the model to text classification.

2.5.2 Rule based categorization

In rule-based classifiers (B. Liu *et al*, 1998), data space is modeled with set of rules, in which the left hand side is a condition on the underlying feature set, and the right hand side is class label. The rule set is model which is generated from training data. For given test data sets of rules are determined for which test instance satisfies the condition on left hand side of rule. Then predicted class labels are determines as function of class labels of rules which are satisfied by test instance. In its most general form, the left hand side of the rule is a Boolean condition, which is expressed in Disjunctive Normal Form (DNF).

2.5.3 Neural Network

The basic unit in a neural network is a *neuron* or *unit*. Each unit receives a set of inputs, which are denoted by the vector *Xi*, which are the term frequencies in the *i*th document. Each neuron is also associated with a set of weights *A*, which are used in order to compute a function *f*(·) of its inputs. A typical function which is often used in the neural network is the linear function as follows:

$$pi = A \cdot Xi$$

Thus, for a vector *Xi* drawn from a lexicon of *d* words, the weight vector *A* should also contain *d* elements. Now consider a binary classification problem, in which all labels are drawn from $\{+1,-1\}$. We assume that the class label of *Xi* is denoted by *yi*. In that case, the sign of the predicted function *pi* yields the class label. A number of implementations of neural network methods for text data have been studied in (I. Dagan *et al*, 1997; N. Littlestone*et al*, 1988; H. T. Ng *et al*, 1997; H. Schutze*et al*, 1995; E. Wiener *et al*, 1995).

2.5.4 Support Vector Machine

Support-vector machines were first proposed (C. Cortes *et al*, 1995; V. Vapnik*et al*, 1982) for numerical data. The main principle of SVMs is to determine separators in the search space which can best separate the different classes. It has been noted in (T. Joachims*et al*, 1997) that text data is ideally suited for SVM classification because of the sparse high-dimensional nature of text, in which few features are irrelevant, but they tend to be correlated with one another and generally organized into linearly separable categories. The SVM classifier has also been shown to be useful in large scale scenarios in which a large amount of unlabeled data and a small amount of labeled data is available (Luhn, H. P. *et al*, 1957).

2.5.5Bayesian categorization

Text document classification widely uses Naive Bayes classifier due to its ease of use and accuracy. Naïve Bayes classifier uses Bayes theorem (Kulkarni, A.R *et al*, 2012). The probability that a test document td belongs to class C is given by

$$P(C/D)= \frac{P(D/C).P(C)}{P(D)}$$

The above equation presents the Bayes Theorem which is used to classify the document.In the equation above ,

C : Classes to which the document belong
D: Set of documents $\{d_1,d_2,\ldots\ldots\ldots,d_n\}$
If P(D) is same for all classes then it is ignored
But D={X1,X2,…..,Xn} where n number of features or terms in case of document.
Thus P(D|C) is written as,
P(X1|C)*P(X2|C)*…*P(Xn|C)

Thus P(C|D)= $P(C)\prod_{i=1}^{n} P(Xi\,|\,C)$

All of these documents should be preprocessed before applying classification. It will remove all raw data from documents. For other than text documents metadata is used to retrieve contexts.

2.6 Term Association Techniques

To deriving context of document it is necessary to identify relation between terms of document. This relation can be found by using association techniques. There are two methods are available to find the association between the terms.

2.6.1Apriori Association

2.6.2 FP growth

2.6.1 Apriori Association

This technique is used to find out frequent terms and find out the relation/association between them. This technique generates candidate key, based on candidate key it performs join operation of candidate key with itself. Likewise in next it again generates next candidate key and performs join operation. This process continues till it gets final associated frequent terms. Here in this paper our proposed system using apriori association rules. This found to give accurate context.

2.6.2 FP growth

This association technique scans the database in two passes and don't generate candidate key. This requires less memory as compare to Apriori. But complexity is more as compare to Apriori Association. Here in this proposed system is considering only two dimensions i.e. sentence number and terms in sentenced. It is found that using Apriori gives satisfactory results without memory overhead problem.

3. Proposed System

The proposed system consists of two models. Query generation model which determines what words in a document might be more favorable to use in a query; and document searching model, which evaluates the similarity between a given query and each document in the target document set. The system creates context of query document and searches for similar document context stored on system.

A. Query Generation model

Step I. Document Indexing

In document indexing tf*idf is calculated. In indexing, text $d_j$is represented by as vector of term weights ,
$d_j=\{w1_j,......wT_j\}$
Here T is dictionary i.e. set of Terms(also known as features) that occur at least ones in at least k documents and $0 \leq wkj \leq 1$ quantifies the importance of tk in characterizing the semantics of d_j .

Step II. Document Classification

Naïve Bayes Classifier,
$P(C/D)= \frac{P(D/C).P(C)}{P(D)}$
C : Classes to which the document belong
D: Set of documents $\{d_1,d_2,\ldots\ldots\ldots,d_n\}$

If P(D) is same for all classes then it is ignored
But D={X1,X2,…..,Xn} where n number of features or terms in case of document.
Thus P(D|C) is written as,
P(X1|C)*P(X2|C)*…*P(Xn|C)
Thus P(C|D)= $P(C)\prod_{i=1}^{n} P(Xi\,|\,C)$
Step III.Document Context

Apriori association algorithm used to find frequent terms. Apriori association rule is used to set rule on frequent terms. Apriori association rule results in document context.
$D_m=\{t_1=W_1^m,t2=W_2^m,...,t_i^m=W_i^m,...,t_n^m=W_n^m\}$
$S_t=\{[(W_i,W_j,W_k):C_1],[(W_p,W_q,W_r):C_2],...\}$
$S_t$represents set of associated words in class which determine context of documents.

B. Document Searching Model

Here first stored documents are processed as follows:

Step I. Document Indexing

In document indexing tf*idf is calculated. In indexing, text $d_j$is represented by as vector of term weights ,
$d_j=\{w1_j,......wT_j\}$

Here T is dictionary i.e. set of Terms(also known as features) that occur at least ones in at least k documents and $0 \leq wkj \leq 1$ quantifies the importance of tk in characterizing the semantics of d_j .

Step II. Document Classification

Naïve Bayes Classifier,
$P(C/D)= \frac{P(D/C).P(C)}{P(D)}$
C : Classes to which the document belong
D: Set of documents $\{d_1,d_2,\ldots\ldots\ldots,d_n\}$
If P(D) is same for all classes then it is ignored
But D={X1,X2,…..,Xn} where n number of features or terms in case of document.
Thus P(D|C) is written as,
P(X1|C)*P(X2|C)*…*P(Xn|C)
Thus P(C|D)= $P(C)\prod_{i=1}^{n} P(Xi\,|\,C)$
Step III. Document Context

Apriori association algorithm used to find frequent terms. Apriori association rule is used to set rule on frequent terms. Apriori association rule results in document context.

$D_m=\{t_1=W_1^m,t2=W_2^m,...,t_i^m=W_i^m,...,t_n^m=W_n^m\}$
$S_t=\{[(W_i,W_j,W_k):C_1],[(W_p,W_q,W_r):C_2],...\}$
$S_t$represents set of associated words in class which determine context of documents.

Step IV. Context Matching

Here context from first model is compared with contexts from second model to find out similar document context.
Document Ranking:Query Document contexts and stored document context are stored in Vector Space model.

Cosine similarity formula is used to give set of closely similar context.

similarity (d_j,q_k)= $\frac{\sum_{i=1}^{n}(td_{ij}\times td_{ik})}{\sqrt{\sum_{i=0}^{n} td_{ij}^{2}\times\sum_{i=0}^{n} tq_{ik}^{2}}}$

where

td_{ij} = thei^{th}term in the vector for document j

tq_{ik}= the i^{th}term in the vector for query k

n = the number of unique terms in the data set

4. Architecture of Proposed System

Following figure describes proposed system.

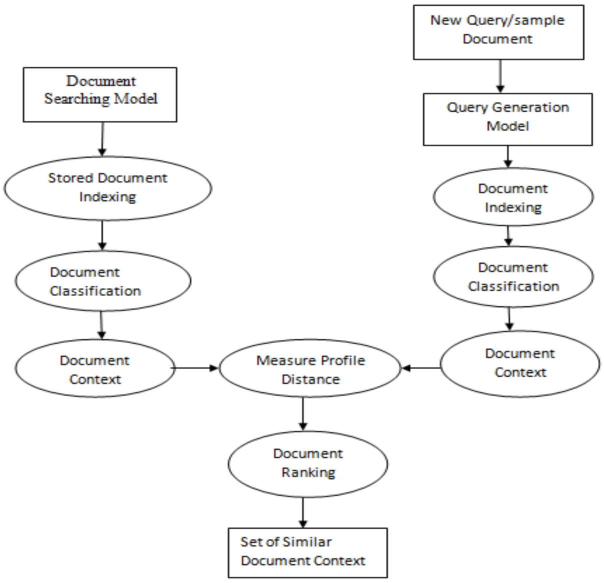

Figure 1: Proposed System

5. Results

Here we used this paper as test document to bring out its context. We selected terms from the document by excluding stop words like “the”, “for”, “it” etc. Remaining terms are used to carry out association between them, which resulting into context. Following table shows set of associated terms and the resulting context of test document.

Table1 Associated Terms with context

Associated Terms	Context
document, context, documents, set, text, query, term, used, data, model, information, classification, profile, user, pp ,indexing, vector, categorization, rule, words, acm, association, retrieval, opposed, conference, apriori, proceedings, number, content, based ,classifier	document context apriori association association rule text categorization content profile information retrieval

The above result in a table gives context of document. The context above gives essence of document. By referring to context we get knowledge about document

Conclusion

Only considering keyword does not handle the issue of document search. Finding interrelation and association between keywords brings out the context of document. This paper using Apriori association helps to find out association. Complexity is less as compared to FP Growth Association technique. This same method can be further extend to larger dataset and can use multilevel association.

References

Maxim Martynov, Boris Novikov (1996), An Indexing Algorithm for Text Retrieval, *Proceedings of the International Workshop on Advances in Databases and Information Systems (ADBIS’96).* Moscow, September 10–13.

Frakes, W.B. (1992), Stemming algorithms. Information Retrieval: Data Structures and Algorithms, *eds. W.B. Frakes& R. Baeza-Yates, Prentice Hall: Englewood Cliffs, US,* pp. 131–160.

Debole, F. &Sebastiani, F. (2003), Supervised term weighting for automated text categorization, *Proceedings of SAC-03, 18th ACM Symposium on AppliedComputing, ACM Press, New York, US: Melbourne, US*, pp. 784–78.

Zobel, J. & Moffat, A. (1998), Exploring the similarity space, *SIGIR Forum*, 32(1), pp. 18–34.

Lewis, D.D., (1992), An evaluation of phrasal and clustered representations on a text categorization task, *Proceedings of SIGIR-92,* pp. 37–50.

Lewis, D.D. &Ringuette, M.(1994.), A comparison of two learning algorithms for text categorization, *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US,* pp. 81–93.

Yang, Y. & Pedersen, J.O.(1997), A comparative study on feature selection in text categorization, *Proceedings of ICML-97, 14th International Conference onMachine Learning, ed.D.H. Fisher,Morgan Kaufmann Publishers, San Francisco, US: Nashville, US,* pp. 412–420.

Wiener E.D, Pedersen J.O, Weigend A.S, (1995) A neural network approach to topic spotting, *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, pp. 317–332.

Sch¨utze H, Hull D.A, & Pedersen, J.O (1995), A comparison of classifiers and document representations for the routing problem, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, eds. E.A. Fox, P. Ingwersen& R. Fidel, ACM Press,New York, US: Seattle, US, pp. 229–237.

Baker L.D, & McCallum A.K (1998), Distributional clustering of words for text classification, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, eds. W.B. Croft,A. Moffat, C.J. van Rijsbergen, R. Wilkinson & J. Zobel, ACM Press, NewYork, US: Melbourne, AU, pp. 96–103.

Bekkerman, R, El-Yaniv R, Tishby, N. & Winter, Y (2001), On feature distributional clustering for text categorization, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, eds. W.B. Croft, D.J. Harper, D.H. Kraft & J. Zobel, ACMPress, New York, US: New Orleans, US, pp. 146–153.

Slonim, N. &Tishby, N. (2001), The power of word clusters for text classification, *Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research*, Darmstadt, DE.

PrathimaMadadi, *University of Nevada, Las Vegas*,Text Categorization Based on Apriori Algorithm's Frequent Itemsets, UNLV Theses/Dissertations/Professional Papers/Capstones.

Goyal, P.; Behera, L.; McGinnity, T.M. (2013), A Context-Based Word Indexing Model for Document Summarization," *Knowledge and Data Engineering, IEEE Transactions on* , vol.25, no.8, pp.1693,1705, Aug.

Kulkarni, A.R.; Tokekar, V.; Kulkarni, P.(2012), Identifying context of text documents using Naïve Bayes classification and Apriori association rule mining, *Software Engineering (CONSEG), 2012 CSI Sixth International Conference on* , vol., no., pp.1,4, 5-7 Sept.

W. NesrineZemirli, Lynda Tamine-Lechani, MohandBoughanem, A Personalized Retrieval Model based on Influence Diagrams.

J.P McGowan (2003), A multiple model approach to personalized information access. *Master Thesis in computer science, Faculty of science, University College Dublin*, February

T. Haveliwala (2002), Topic-sensitive Page Rank, *International ACM World Wide Web conference*, pp 727-736

F. Liu, C. Yu, (2004) Personalized Web search for improving retrieval effectiveness, *IEEE Transactions on knowledge and data engineering*, 16(1), pages 28-40.

F. Qiu, J. Cho (2006), Automatic identification of user interest for personalized search, *International ACM World Wide Web conference*, pp 727-736

Speretta S, Gauch S, Personalizing search based user search histories, *In Proceedings of the 13th International Conference on Information Knowledge, Management, CIKM.* (2004) 238–239

I. Dagan, Y. Karov, D. Roth (1997), Mistake-driven Learning in Text Categorization, *Proceedings of EMNLP*.

C. Cortes, V. Vapnik.(1995), Support-vector networks. *Machine Learning*, 20: pp. 273–297.

T. Joachims (1997), A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *ICML Conference*.

B. Liu, W. Hsu, Y. Ma (1998), Integrating Classification and Association Rule Mining, *ACM KDD Conference*.

N. Littlestone (1988), Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm, *Machine Learning*, 2: pp. 285– 318.

H. T. Ng, W. Goh, K. Low (1997), Feature selection, perceptron learning, and a usability case study for text categorization, *ACM SIGIRConference*.

J. R. Quinlan (1986), Induction of Decision Trees, *Machine Learning*, 1(1), pp 81–106,.

H. Schutze, D. Hull, J. Pedersen (1995), A comparison of classifiers and document representations for the routing problem, *ACM SIGIRConference*,.

V. Vapnik (1982), Estimations of dependencies based on statistical data, *Springer*.

E. Wiener, J. O. Pedersen, A. S. Weigend (1995), A Neural Network Approach to Topic Spotting, *SDAIR*, pp. 317–332.

V. Sindhwani, S. S. Keerthi (2006), Large scale semi-supervised linear SVMs, *ACM SIGIR Conference*.

Luhn, H. P.(1957), A Statistical Approach to Mechanized Encoding and Searching of Literary Information, *IBM Journal of Research and Development* , vol.1, no.4, pp.309,317, Oct.