A Comparison of Implicit and Explicit Links for Web Page Classification

Dou Shen
Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong
dshen@cs.ust.hk

Jian-Tao Sun
Microsoft Research Asia
49 Zhichun Road
Beijing, P.R.China
jtsun@microsoft.com

Qiang Yang
Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong
qyang@cs.ust.hk

Zheng Chen
Microsoft Research Asia
49 Zhichun Road
Beijing, P.R.China
zhengc@microsoft.com

ABSTRACT

It is well known that Web-page classification can be enhanced by using hyperlinks that provide linkages between Web pages. However, in the Web space, hyperlinks are usually sparse, noisy and thus in many situations can only provide limited help in classification. In this paper, we extend the concept of linkages from explicit hyperlinks to implicit links built between Web pages. By observing that people who search the Web with the same queries often click on different, but related documents together, we draw implicit links between Web pages that are clicked after the same queries. Those pages are implicitly linked. We provide an approach for automatically building the implicit links between Web pages using Web query logs, together with a thorough comparison between the uses of implicit and explicit links in Web page classification. Our experimental results on a large dataset confirm that the use of the implicit links is better than using explicit links in classification performance, with an increase of more than 10.5% in terms of the Macro-F1 measurement.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous; I.5.4 [**Pattern Recognition**]: Applications—*Text processing*

General Terms

Algorithms, Experimentation, Verification.

Keywords

Web page classification, Query Log, Implicit link, Explicit link, Virtual Document

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

1. INTRODUCTION

Automatic Web-page classification by using hypertext is a major approach to categorizing large quantities of Web pages. Two major kinds of approaches have been studied for Web-page classification: content-based and context-based approaches. Typical content-based classification methods utilize words or phrases of a target document in building the classifier and can achieve only limited accuracy. This is because very often a Web page contains no obvious clues textually for its category. For example, some pages contain only images and little text information. By exploiting the hypertextual information, context-based approaches additionally exploit the relationships between the Web pages to build a classifier [14, 8, 7]. As a consequence, they are found to be more accurate than pure content based classifiers. However, the hyperlinks sometimes may not reflect true relationships in content between Web pages, and the hyperlinks themselves may be sparse. A technical Web page may link to a commercial page describing a beauty product. In such situations, it is unreliable to employ hyperlinks for classification.

We observe that hyperlinks provide only one kind of linkages between Web pages. By properly utilizing the Web query logs, we can discover other important linkages that are just as important. In this paper, we propose to extract a new kind of links known as implicit links from the Web query logs that are accumulated by search engines. These logs record the Web users' behavior when they search for information via a search engine. Over the years, query logs have become a rich resource which contains Web users' knowledge about the World Wide Web (WWW). In order to mine the latent knowledge hidden in the query logs, much research has been conducted on the query logs [17, 2]. Many applications of log analysis have been conducted in categorization and query clustering [1, 4, 20]. Consider the behavior of a Web user. After submitting a query, a Web user gets a long list of Web pages with snippets returned by a search engine. Then the user could look through some of the returned pages selectively. All the queries the users submit and the Web pages they click on constitute the query log. Although there may be some noise in query logs, the clicks still convey impor-

tant information on the similarity between Web pages and queries in a large query log. In the context of Web page classification, an implicit type of similarity information we can use is the fact that the Web pages that are clicked by users through the same query usually belong to the same categories. It is this kind of knowledge that we make use of in order to build the implicit links between Web pages.

In this paper, we define two kinds of implicit links that can be extracted from a query log and three kinds of explicit links using the traditional hyperlink concepts. We further define two different approaches in making use of these links. The first one is to classify a target page according to the labels of the neighbors of that page. We show that using this algorithm for classification, the implicit links are more reliable in classification. The best result of using implicit links is about 20.6% higher than the best result through the explicit links in terms of the Micro-F1 measure. The second classification method we use is to construct a virtual document based on different kinds of links, and then conduct classification on the virtual documents. The experiments on the Web pages crawled from the Open Directory Project (ODP)[1] and the query logs collected by MSN show that using the implicit links can achieve 10.5% improvement compared to the explicit links in terms of Macro-F1.

The main contributions of this paper could be summarized as follows:

1. We introduce a new resource – the query log – to help classify Web pages. Based on the query logs, a new kind of links – the implicit links – is introduced. Comparison between the implicit and explicit links on a large data set shows that the implicit links are more helpful for Web-page classification.

2. We define the concept of a virtual document by extracting "anchor sentence (AS)" though implicit links which corresponds to "anchor text (AT)" and "extended anchor text (EAT)" associated with hyperlinks. We show that using our proposed virtual documents, the classification performance can be improved by more than 10.5% relative to the best result obtained by explicit links in terms of the Macro-F1 measurement.

The rest of the paper is organized as follows. In Section 2, we present the related works on query logs and Web classification through hyperlinks. We give the definition of two kinds of "Implicit Links" and three kinds of "Explicit Links" in Section 3. Section 4 gives the approaches to utilize the Links. The experimental results on the ODP dataset and query logs collected by MSN as well as some discussions are shown in Section 5. Finally, we conclude our work in Section 6.

2. RELATED WORK

In the past, much work has been done on context-based Web page classification by exploiting hyperlinks among web pages. Chakrabarti et al. used predicted labels of neighboring documents to reinforce classification decisions for a given document [3]. Oh et al. proposed a practical method for exploiting hypertext structure and hyperlink information [14]. They modified the Naive Bayes algorithm to classify documents by using neighboring documents that were similar to the target document. Both the predicted labels and the text contents of the neighboring documents were used to assistant classification. The experimental results on an encyclopedia corpus that contains hyperlinks validate their algorithms. Fürnkranz also reported a significant improvement in classification accuracy when using the link-based method as opposed to the full-text alone [7] on 1,050 pages of the WebKB corpus[2], although adding the entire text of "neighbor documents" seemed to harm the ability to classify pages [3]. In [8], the authors investigated six kinds of regularities in a hypertext corpus with three classifiers. The conclusion in [8] showed that using words in Web pages alone often yields sub-optimal performance for classifiers, compared to exploiting additional sources of information beyond document content. It also showed that linked pages can be more harmful than helpful when the linked neighborhoods are highly noisy and that links have to be used in a careful manner. Eiron and McCurley found that the anchor text is typically less ambiguous than other types of texts; thus they can be used to produce a good representation for web pages [6]. Glover et al. came to the same conclusion in [9] on a subset of the WebKB dataset as Fürnkranz used [7] and a dataset crawled from Yahoo! which consists of less than ten thousand pages. Glover et al. [9] concluded that the full-text of a Web page is not good enough for representing the Web pages for classification. They created virtual documents by incorporating anchor text and extended anchor text. The experimental results demonstrated that the virtual documents, especially when constructed through extended anchor text are of great help. In this paper, we enhance the notion of virtual documents by the implicit and explicit links for classification. Our experimental results on a large dataset not only confirm the previous conclusions, but also show that the virtual documents constructed through our proposed implicit links are consistently better than those constructed through hyperlinks.

Query logs are being accumulated rapidly with the growing popularity of search engines. It is natural for people to analyze the query logs and mine the knowledge hidden in them. Silverstein et al. made a static analysis of an AltaVista query log from six weeks in 1998 consisting of 575 million nonempty queries [17]. Beitzel et al. [2] studied a query log of hundreds of millions of queries that constitute the total query traffic for an entire week of a general purpose commercial web search service which provided valuable insight for improving retrieval effectiveness and efficiency. By analyzing and leveraging the information from query log, many applications become feasible or easier. Raghavan and Sever described an elegant method of locating stored optimized queries by comparing results from a current query to the results from the optimized query [16]. Beeferman and Berger proposed an innovative query clustering method based on query log [1]. By viewing the query log as a bipartite graph, with the vertices on one side corresponding to queries and those on the other side corresponding to URLs, they applied an agglomerative clustering algorithm to the graph's vertices to identify related queries and URLs. Wen et al. incorporated click-through data to cluster users' queries [20]. They analyzed a random subset of 20,000 queries from a single month of their approximately 1-million queries-per-week traffic. Chuang and Chien proposed a technique for categorizing Web query terms from

[1]http://dmoz.org/

[2]http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/ data/

the click-through logs into a pre-defined subject taxonomy based on their popular search interests [4]. Xue et al. proposed a novel categorization algorithm named IRC (Iterative Reinforcement Categorization algorithm) to categorize the interrelated Web objects by iteratively reinforcing individual classification results via relationships across different data types extracted from query logs [21]. However, to the best of our knowledge, no research has considered building implicit links between documents through query logs for Web page classification. In the next section, we formally introduce the concept of implicit links.

3. INTRODUCING IMPLICIT LINKS

Consider a typical scenario when a Web user finds information by a search engine. A user issues a query which may be a well-formed natural language question or one or more keywords or phrases to a search engine. The search engine replies with a list of Web pages to the user together with a summary of each page. The summary, together with other clues such as the URL and title, can give the user a general idea of the contents of the page. Based on this information, the Web user clicks on some Web pages that appear to be most relevant to the query. The query log records the process, by keeping an entry in query log for every session in which the Web user submits a query and clicks on some returned pages. An example of such a quadruple is as follows:

$$Entry := < U, Q, D_q, T >$$

where U denotes the web user who submits the query. The user is often represented by the IP address from which the user accesses the Internet. Q denotes the query the Web user submits to the search engine. D_q represents the returned Web pages clicked by the user. T represents the time when such an entry occurs. In this work, we only focus on using the query logs for Web page classification. Thus, we omit T and consider an entry as $< U, Q, D_q >$. In fact, we could further simplify the entry as $< Q, D_q >$ by omitting U. However, often times when the same query is issued by two different users, it may mean different things. Such ambiguities between users make the pages that are constrained by Q alone to be more heterogeneous in content than the pages that are constrained by both U and Q.

Based on a query log, we define two kinds of implicit link relations according to two different constraints:

- L_I1: d_i and d_j appear in the same entries constrained by U and Q in the query log. That is d_i and d_j are clicked by the same user issuing the same query;
- L_I2: d_i and d_j appear in the same entries constrained by the query Q only. That is d_i and d_j are clicked according to the same query, but the query may be issued by different users. It is clear that the constraint for L_I2 is not as strict as that for L_I1. Thus, more links of L_I2 can be found than L_I1, however they may be more noisy.

Similar to the implicit links, we define three kinds of explicit links based on the hyperlinks among the Web pages according to the following three different conditions:

$$L_E(i,j) = \begin{cases} 1 & Cond_E \\ 0 & Other \end{cases}$$

- $Cond_E1$: there exist hyperlinks from d_j to d_i (In-Link to d_i from d_j);
- $Cond_E2$: there exist hyperlinks from d_i to d_j (Out-Link from d_i to d_j);
- $Cond_E3$: either $Cond_E1$ or $Cond_E2$ holds.

We denote these three types of explicit links under the above conditions as L_E1, L_E2, L_E3 respectively.

In the above definitions, we distinguish between the in-link and out-link because, given the target Web page, the in-link is the hyperlink created by other Web page editors who refer to the target Web page. In contrast, the out-link is created by the editor of the source Web page. They may be different when used to describe the target Web page.

Based on other considerations, it is possible for us to define other links. For example, we could define a kind of implicit link between two pages if they belong to two different entries and the intersection of the two clicked-page sets of the two entries is not empty. We could also require that the similarity between two pages be larger than a certain threshold to define a certain kind of explicit link. However, for the purpose of comparing explicit and implicit links, we will only focus on the above five different types of links.

From the above definitions, we observe that the implicit links give the relationships between Web pages from the view of Web users. However, the explicit links reflect the relationships among Web pages from the view of Web-page editors.

4. APPROACHES OF LEVERAGING LINKS

4.1 Classification by Linking Neighbors (CLN)

A straightforward approach to utilizing links for Web page categorization is to predict the label of the target Web page by the labels of its neighbors through majority voting. We call this method "Classification by Linking Neighbors" (CLN for simplicity). The formal definition of this method is given below:

$$\text{Category(d)} = \underset{c_i}{\text{argmax}} \left(\underset{\text{d}_\text{j} \in S_\text{L}(d)}{\text{count}} (\text{d}_\text{j} \in c_i) \right)$$

where L represents a kind of link; $S_L(d)$ is the set of Web pages which relate to d through L; $\underset{\text{d}_\text{j} \in S_\text{L}(d)}{\text{count}} (\text{d}_\text{j} \in c_i)$ denotes the number of the neighbors which belong to the class c_i. This algorithm is similar to k-Nearst Neighbor (KNN). However, k is not a constant as in KNN and it is decided by the set of the neighbors of the target page.

4.2 Virtual Document-Based Classification

The CLN method classifies a Web page only based on the labels of its neighboring pages. It does not consider the content of Web pages. In this work, we enhance classification performance through the links by constructing virtual documents. Given a document, the virtual document is constructed by borrowing some extra text from its neighbors. Although originally the concept of the virtual document is pioneered by [9], we extend the notion by including different links.

To form a virtual document, we consider two approaches. One is to adjust the non-zero weights of existing terms in the original keyword vector that represents the page. The other

is to bring in new terms from the neighbors so that the modified vector will have additional non-zero components. The second approach changes the vector more drastically. The experimental results in [14] showed that the latter method resulted in a 23.9 % decrease in F1 measure. Thus, in this paper, we adopt the former approach to construct virtual documents.

The plain text in Web pages, as well as the rich information marked by HTML, such as the title, meta-data and paragraph headings could be all used for Web-page classification. Some research works have studied the contribution of these elements to Web-page classification [15, 8]. In order to show the contribution of different links on Web page classification, we try to find an appropriate way to represent web pages by local words and then take it as a baseline to compare with the virtual document representation. In our previous study, we found that the weighted combination of Plain text, Meta-data and Title usually achieves the best classification performance. Thus, we take these as the local representation of web pages. We give these concepts below:

- Plain Text: Plain text refers to the remaining text after all the html tags are removed from web pages.
- Meta-data: Meta-data refers to the content embedded in the html tags <Meta> and </Meta>, such as "Keywords" and "Description";

Due to the different characteristics of the implicit and explicit links, we define different types of virtual documents. For explicit links, Anchor text (AT) is usually employed to form a virtual document. Since a Web page often has other pages linked to it, the accompanying anchor text often describes the target Web page. These anchor text provides a very good description of the content of the page in different contexts and by different people. In [9], Glover et al. defined a concept of extended anchor text (EAT) which incudes the set of rendered words occurring up to 25 words before and after an associated link as well as the anchor text itself. Their experimental results showed that both anchor text and extended anchor text could improve Web-page classification greatly. Therefore, we also use both methods in constructing the virtual documents to compare with our proposed virtual documents constructed through implicit links.

For the implicit links, there is no "anchor text" as defined in the case of explicit links. As a consequence, in this work, we define a corresponding concept-"anchor sentence". Since the implicit links among the Web pages are built through "queries", we could define the "anchor sentence" through a query. If an implicit link between d_i and d_j is created according to a query Q, the set of sentences in d_j which contain all the words in Q are regarded as the "anchor sentences". We then collect the anchor sentences to construct a virtual document. The query Q is preprocessed as shown in section 5.1. We require the anchor sentences include all the words in the query to guarantee that the content of the virtual document focuses on the query.

After constructing the virtual document through links, any traditional text classification algorithm could be employed to classify the web pages. In this paper, we take Naive Bayesian classifier and Support Vector Machine as the classifiers to compare the quality of different virtual documents.

Table 1: Three largest categories

Category		#Pages
First Level	Second Level	
society	religion and spirituality	78531
reference	Education	45089
computers	Software	39261

Table 2: Three smallest categories

Category		#Pages
First Level	Second Level	
Home	news and media	1
News	chats and forums	1
Home	personal organization	1

5. EXPERIMENTS

Above, we have defined a new kind of links between Web pages: implicit link. In this section, we empirically verify the merits of introducing this type of links. We introduce the data set, our evaluation metrics, and the experimental results based on those metrics with the analysis. All the classification results shown in this paper are obtained through 10-fold cross validation to reduce the uncertainty of data split between training data and test data.

5.1 Data Set

The dataset used in this work contains 1.3 million Web pages, which are crawled from ODP. All these Web pages have been manually classified into hierarchical directories, whose first level contains 17 categories. Web pages in the "Regional" category are also included in other categories. In addition, Web pages in the "World" category are not written in English. Therefore, these two categories are not considered in our experiments. Under the remaining 15 first-level categories, there are 424 second-level categories. Our experiments are conducted at the second level. Table 1 and Table 2 present the three largest and three smallest categories. Among the 424 second-level categories, there exist 76 categories each of which contains less than 50 pages. The unbalanced distribution may be the main reason for the low value of Macro-F1 measurement in the following experimental results.

A subset of the real MSN query log is collected as our experiment data set. The collected log contains about 44.7 million records of 29 days from Dec 6 2003 to Jan 3 2004. Some preprocessing steps are applied to queries and Web pages in the raw log. We processed the log into a predefined format as shown in Section 3. All queries are converted into lower-case, and are stemmed using the Porter algorithm.[3] The stop words are removed as well. After preprocessing, the log contains 7,800,543 entries, having 2,697,187 users, 308,105 pages and 1,280,952 queries. That is, among the 1.3 million ODP Web pages, 308,105 of them clicked by users in the 29 days are studied in this paper. The average query length is about 2.1 words.

5.2 Classifiers

In this paper, we would apply two classifiers including Naive Bayesian classifier (NB) and Support Vector Machines

[3]http://www.tartarus.org/∼martin/PorterStemmer/index.html

(SVM) on the virtual documents. A brief introduction of them is given below.

5.2.1 Naive Bayesian Classifier (NB)

NB is a simple but effective text classification algorithm which has been shown to perform very well in practice [12, 13]. The basic idea in NB is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. As described in [12], most researchers employ NB method by applying Bayes' rule:

$$P(c_j|d_i;\hat{\theta}) = \frac{P(c_j|\hat{\theta})\prod_{k=1}^{n} P(w_k|c_j;\hat{\theta})^{N(w_k,d_i)}}{\sum_{r=1}^{|C|} P(c_r|\hat{\theta})\prod_{k=1}^{n} P(w_k|c_r;\hat{\theta})^{N(w_k,d_i)}}$$

where $P(c_j|\hat{\theta})$ can be calculated by counting the frequency with each category c_j occurring in the training data; $|C|$ is the number of categories; $p(w_i|c_j)$ stands for the probability that word w_i occurs in class c_j. This last quantity may be small in the training data. Thus, Laplace smoothing is chosen to estimate it; $N(w_k, d_i)$ is the number of occurrences of a word w_k in d_i ; n is the number of words in the training data.

5.2.2 Support Vector Machine (SVM)

SVM is a powerful learning method introduced by Vapnik et al. [5, 19]. It is well founded in terms of computational learning theory and has been successfully applied to text categorization [10]. The SVM algorithm is based on the Structural Risk Maximization theory, which aims to minimize the generalization error instead of the empirical error on training data alone. Multiple variants of SVM have been developed [11]; in this paper we train the classifier using the SVM^{light} software package[4] due to its popularity and high performance in text categorization. For all the following experiments, we used a linear kernel for comparison purposes because of its high accuracy for text categorization [10]. The trade-off parameter C is fixed to 1 for comparison purpose and the one-against-all approach is used for the multi-class case[10].

5.3 Evaluation Measurements

We employ the standard measures to evaluate the performance of Web page classification, i.e. precision, recall and F1-measure [18], which are widely applied in the field of classification. Precision (P) is the proportion of actual positive class members returned by the system among all predicted positive class members returned by the system. Recall (R) is the proportion of predicted positive members among all actual positive class members in the data. F_1 is the harmonic average of precision and recall as shown below:

$$F_1 = \frac{2 \times P \times R}{P + R}$$

To evaluate the average performance across multiple categories, we apply two conventional methods: micro-average and macro-average. Micro-average gives equal weight to every document; while macro-average gives equal weight to every category, regardless of its frequency. In our experiments, both of them are used to evaluate the performance of classification.

[4]http://svmlight.joachims.org/

Table 3: Statistics on links of different type

Link type	Consistency	#Links	#Link/page
L_I1	0.569	162901	2.00
L_I2	0.462	484462	4.63
L_E1	0.458	1148217	4.38
L_E2	0.458	1148217	3.23
L_E3	0.437	1056208	2.18

5.4 Experimental Results

In this section, we show the statistics of different links in the data set firstly. Then the results of the two approaches of leveraging different links are presented, based on which we compare the merits of implicit links and explicit links.

5.4.1 Statistics on the "Links"

Table 3 introduces the characteristics of the five types of links in the data set. In this table, consistency refers to the percentage of links that have the two linked pages from the same category. This quantity can reflect the quality of the links, in that the higher the consistency, the better the quality of the link is. #Links refers to the number of links of each kinds of link in our dataset and #Links/page refers to the density of links. From Table 3, we can find that the consistency of L_I1 is about 23.2% higher than L_E1 and L_E2 which indicates that L_I1 is more reliable for guessing the category of a target page by the labels of its linked neighbors. One fact for explaining this observation is that very often hyperlinks are created not according the similarity of content between Web pages, but for other reasons such as commercial advertisement. We could also find that the consistency of L_I2 to be much lower than L_I1, though the number of links of L_I2 is far larger than L_I1. Such an observation shows that we could get more links by relaxing the condition, but the quality of links may become worse.

Table 3 supports the published facts that linked pages can be more harmful than helpful and that links have to be used in a careful manner [8]. In these cases, two pages linked together by L_I2 or any kind of explicit links tend to belong to different categories with a probability more than 0.5.

5.4.2 Results of CLN

In order to compare the contributions of different links for the classification task, we run the experiments of CLN using a subset of the ODP Web pages. The subset is determined by the coverage of the links. For example, given L_I1, we collect all the pages which are covered by implicit link of type 1 and then split them into 10 groups randomly to perform the 10-fold cross validation. The results listed in Figure 1 are the average of the experiment results of 10 runs. Micro-F1 and Macro-F1 represent the micro and the macro average F1 measure from the 424 categories. From Figure 1, we could see that the two kinds of implicit links both achieve better results than the three kinds of explicit links. The best result achieved by implicit links is about 20.6% in terms of Micro-F1 and 44.0% in terms of Macro-F1 higher than the best result achieved by explicit links. We can also see that the L_E1 based classification method outperforms that based on L_E2. The explanation of this observation is that the average number of In-Links per page (4.38) is much larger than the average number of Out-Links per page (3.23) which can help remove the effect of noise.

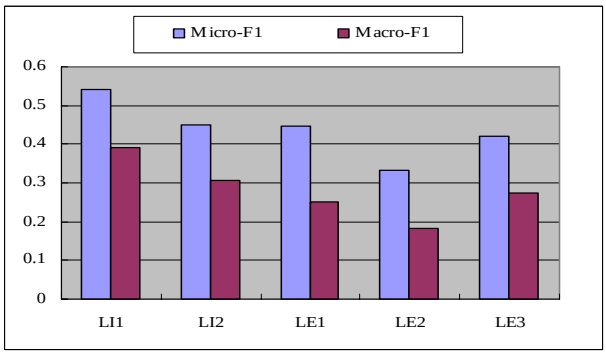

Figure 1: Results of CLN on different kinds of Links.

5.4.3 *Experiments on different virtual documents*

From our previous study, we find that the weighted combination of Plain text (PT), Meta-data and Title (MT) can achieve the best classification performance among all local representations of the Web page. Thus, in this paper, we simply adopt the weighted combination of PT and MT (by 1:2, which achieve the best result on our dataset) as the baseline (denoted as LT, Local Text) and do not present the detailed comparison between the different local representations since we focus on the comparison between implicit links and explicit links. For each kind of virtual document, it is first tokenized with a stop-word remover and Porter stemming. Then, each document is represented as a bag-of-words, in which the weight of each word is their term frequency. In this experiment, the document frequency selection (DF) [22] is applied for feature selection. The words whose DF are lower than four are removed from the feature set.

Results on different virtual documents

In Section 4.2, we gave a brief description of the construction of virtual documents. In this section, we describe the virtual documents in more detail along with an empirical comparison. Given a Web page, we have two ways to construct a virtual document: (1) through the anchor text or the extended anchor text if explicit links are considered. Alternatively, this can be done through anchor sentences if the implicit links are considered. (2) Through local text within a Web page. The local text refers to the weighted combination of Plain text, Meta-data and Title. In this part, we only consider the best links of implicit link and explicit link as shown in results of CLN (that is L_I1 and L_E1) to construct virtual documents. Together we have five combinations through which to form a virtual document, as listed in Table 4.

Table 4: Approaches to constructing virtual documents

	Explicit Link(L_E1)	Implicit Link(L_I1)
LT	ELT	ILT
AS, EAT,AT	AT, EAT	AS

In this table, ELT refers to the virtual documents constructed by combining the local text from all the pages that link to the target Web page through L_E1. ILT refers to the virtual documents constructed by combining the local text from all pages which link to the target Web page through L_I1. EAT refers to virtual documents consisting of the extended anchor text. These are the set of rendered words occurring up to 25 words before and after the anchor text. They also include the anchor text itself. In order to test the impact of implicit links and explicit links, in the following experiments, we only work on a subset of the ODP dataset that we collected in which all the pages are covered by both L_I1 and L_E1. The subset contains 56,129 pages. Table 5 shows the performance of classification on different kinds of virtual documents, where Mi-F1 refers to Micro-F1 and Ma-F1 refers to Macro-F1.

Table 5: Performance on different kinds of virtual document

(1) Classification performance by SVM

	LT	ILT	ELT	AS	EAT	AT
Mi-F1	0.607	0.652	0.629	0.591	0.519	0.403
Ma-F1	0.348	0.444	0.384	0.389	0.297	0.253

(2) Classification performance by NB

	LT	ILT	ELT	AS	EAT	AT
Mi-F1	0.551	0.583	0.515	0.556	0.464	0.361
Ma-F1	0.25	0.336	0.298	0.296	0.226	0.163

To analyze the results of different kinds of virtual documents, two factors should be considered. One is the average size of the virtual documents and the other is the consistency or purity of the content of the virtual documents. The average size of each kind of virtual document is shown in Table 6.

Table 6: Average size of different kinds of virtual document (in terms of KB)

	LT	ILT	ELT	AS	EAT	AT
Ave size	5.60	11.10	23.80	2.30	0.30	0.20

From Table 5, we observe that although it is supposed that AS, EAT and AT can reflect the content of the target pages correctly, the classification performance based on them is just as good as the baseline, or even worse than the baseline. One possible reason may lie in the fact that the average sizes of AS, EAT and AT are usually too small to obtain any satisfying classification results. On the other side, though the average size of ELT is large enough compared to the average size of LT, too much noise may be introduced when it is constructed through L_E1 and the included noise biased the classification results. We also see that ILT is much better than ELT, with a relative increase of 12.8% in terms of Micro-F1 through NB classifier. The reason may be that the content of these Web pages linked by L_E1 may be quite different. Thus, by combining them together, the diversity of the content may bias the classification result. However, the content of the Web pages linked by L_I1 tend to be similar to each other. Therefore, it is more appropriate to employ the implicit links when constructing a virtual document to improve the classification performance.

Table 7: Classification performance of different combinations between AT, EAT, AS and LT

(1) Results on SVM

	1:0	4:1	3:1	2:1	1:1	1:2	1:3	1:4	0:1	Impr*
Micro-F1										
AS	0.607	0.654	0.659	0.661	**0.662**	0.661	0.657	0.650	0.591	**9.06%**
EAT		0.616	0.627	0.636	0.638	**0.639**	0.635	0.628	0.519	5.27%
AT		0.596	0.606	0.612	0.611	0.612	**0.615**	0.610	0.403	1.30%
Macro-F1										
AS	0.348	0.423	0.420	**0.432**	0.426	0.431	0.429	0.422	0.389	**24.14%**
EAT		0.370	**0.383**	0.391	0.384	0.386	0.387	0.376	0.297	12.37%
AT		0.355	**0.365**	**0.365**	0.356	0.356	0.352	0.352	0.253	4.83%

(2) Results on NB

	1:0	4:1	3:1	2:1	1:1	1:2	1:3	1:4	0:1	Impr*
Micro-F1										
AS	0.551	0.626	0.627	**0.629**	0.622	0.618	0.613	0.612	0.556	**14.16%**
EAT		0.614	**0.620**	0.614	0.600	0.614	0.616	0.613	0.464	12.52%
AT		0.594	**0.596**	0.594	0.579	0.585	0.586	0.584	0.361	8.17%
Macro-F1										
AS	0.250	**0.353**	0.352	0.349	0.337	0.338	0.338	0.346	0.296	**41.20%**
EAT		**0.313**	0.306	0.306	0.281	0.297	0.304	0.310	0.226	25.20%
AT		**0.304**	0.296	0.291	0.275	0.281	0.287	0.285	0.163	21.60%

Impr* = the greatest improvement in each row compared to LT (that is the column "1:0").

Combination and Parameters tuning

Although the classification results of AT, EAT and AS shown in Table 5 are not very encouraging, there are special cases for which AT, EAT and AS can capture the meaning of the target Web page well. In the next experiment, we construct a virtual document by integrating the AT, or EAT, or AS to the target Web pages. The weight between AT, EAT, AS and the local text in the combination is changed from 4:1 to 1:4. The detailed result is given in Table 7. From Table 7 we could see that either AT, EAT or AS can improve the performance of classification to some extent with either classifier in terms of either Micro-F1 or Macro-F1. In particular, AS can provide greater improvement, especially with the NB classifier in terms of Macro-F1. The improvement is about 41.2% compared to the baseline, 12.8% compared to EAT and 16.1% compared to AT. Another observation from Table 7 is that the different weighting schemes do not make too much of a difference, especially in terms of Micro-F1.

In addition to the results reported above, we also combine the LT, EAT and AS together to test whether they complement each other. However, this combination makes a relatively very small improvement compared to the combination of LT and AS only.

Effect of Query Log's Quantity

We have shown that when combined with the local text, AS, EAT and AT all can improve the classification performance on the subset of ODP dataset in which all the pages are covered by both L_I1 and L_E1. AS can consistently outperform AT with either NB or SVM classifier. In this part, we test the effect of the query log's quantity. We divide the entries in the query log into ten parts according to the time when the entries are recorded. Each part contains about 10 percent of the total entries. Firstly, we use the first part to construct implicit links, and then construct AS. After that, we incorporate these AS to the subset of ODP dataset that we used in previous experiments (with the weighting schema as LT: AS = 2 : 1) and do classification on the combined dataset. Then, we use the first two parts and then the first three parts, ..., until all the parts are used. The results are shown in Figure 2. From Figure 2, we could find that with more query logs used, the classification performance improves steadily. Such an observation is very encouraging since we can expect to get better results with larger query logs. Another fact is that query log data are accumulated rapidly with the popularity of search engines. Therefore, it is safe to claim that the implicit link will play more and more important roles in Web page classification.

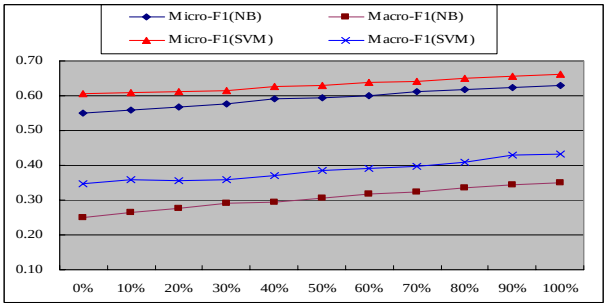

Figure 2: The effect of query log quantity

6. CONCLUSION AND FUTURE WORK

With the growing popularity of search engines, more and more query logs are being collected. We have shown here that these data can become an important resource for Web mining. In this paper, by analyzing the query log, we define a new relationship between Web pages - implicit links. In addition, we define different types of explicit links based

on the existing hyperlinks. Similar to the Anchor Text used in the traditional way, we define a new approach to construct virtual documents by extracting the Anchor Sentences. Through the two different classification approaches, the link-based method and the content-based method, we compared the contribution of the implicit links and the explicit links for Web page classification. The experimental results show that implicit links can improve the classification performance obviously as compared to the baseline method or the explicit links based methods.

In the future, we will also test whether the implicit links will help Web page clustering and summarization. In addition, we wish to explore other types of explicit and implicit links by introducing other constraints, and compare them with the links given in this paper. Besides furthering the research following the framework of this paper, we will try to exploit more knowledge from the query logs. For example, when dealing with query logs, most researchers only consider the clicked pages and assume that there exist some "similarity" relationships among these pages. They neglect the "dissimilarity" relationships among the "clicked Web pages" and "unclicked Web pages". In our future work, we will study the possibility of constructing "dissimilarity" relationships to help Web page classification and clustering.

7. REFERENCES

[1] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *KDD '00: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416, New York, NY, USA, 2000.

[2] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–328, New York, NY, USA, 2004.

[3] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 307–318, New York, NY, USA, 1998.

[4] S.-L. Chuang and L.-F. Chien. Enriching web taxonomies through subject categorization of query terms from search engine logs. *Decision Support Systems*, 35(1):113–127, 2003.

[5] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[6] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 459–460, Toronto, Canada, 2003.

[7] J. Fürnkranz. Exploiting structural information for text classification on the www. In *IDA '99: Proceedings of the 3rd Symposium on Intelligent Data Analysis*, pages 487–498, 1999.

[8] R. Ghani, S. Slattery, and Y. Yang. Hypertext categorization using hyperlink patterns and meta data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 178–185, 2001.

[9] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake. Using web structure for classifying and describing web pages. In *WWW '02: Proceedings of the 11th International Conference on World Wide Web*, pages 562–569, Honolulu, Hawaii, USA, 2002.

[10] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, 1998.

[11] T. Joachims. Learning to classify text using support vector machines. *Dissertation*, Kluwer, 2002.

[12] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[13] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[14] H.-J. Oh, S.-H. Myaeng, and M.-H. Lee. A practical hypertext categorization method using links and incrementally available class information. In *SIGIR '00: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 264–271, Athens, Greece, 2000.

[15] C. Quek. Classification of world wide web documents. *Thesis*, School of Computer Science, CMU, 1997.

[16] V. V. Raghavan and H. Sever. On the reuse of past optimal queries. In *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 344–350, Seattle, Washington, USA, 1995.

[17] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[18] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, London, 1979.

[19] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, NY, USA, 1995.

[20] J.-R. Wen, J.-Y. Nie, and H. Zhang. Clustering user queries of a search engine. In *WWW' 01: Proceedings of the Tenth International World Wide Web Conference*, pages 162–168, Hong Kong, China, 2001.

[21] G.-R. Xue, D. Shen, Q. Yang, H.-J. Zeng, Z. Chen, Y. Yu, W. Xi, and W.-Y. Ma. Irc: An iterative reinforcement categorization algorithm for interrelated web objects. In *ICDM '04: Proceedings of the 4th IEEE International Conference on Data Mining*, pages 273–280,Brighton, UK, 2004.

[22] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, Nashville, TN, USA, 1997.