Finding Named Pages via Frequent Anchor Descriptions

J. Malawong **A. Rungsawang**

Massive **I**nformation & **K**nowledge **E**ngineering
Department of Computer Engineering
Faculty of Engineering
Kasetsart University, Bangkok, Thailand

Email: {g4565043, fenganr}@ku.ac.th

Abstract

This article describes about finding documents of interest via frequent anchor descriptions that being derived from the ".gov" web collection. The main idea of our approach is that we consider frequent anchor descriptions as documents. To find out the frequent item sets, we apply the Apriori algorithm with a new scoring criterion, called the *maximum correspondence*. We likewise integrate both retrieval scores calculated from anchor descriptions and title texts of the web pages to identify the resulting named pages, and foundthat these combination scores can boost the precision performance. Concluded from our preliminary experiments , this approach yields a considerable efficiency of named page finding in the aspect that it also highly reduces the document search space.

1 Introduction

The information searching generally focuses on the quantity of information that the user obtains. On the other hand, named page finding only focuses on the most significant web page represented as the answer to a query. However, named page finding needs higher precision performance. Therefore, finding a named page is a more difficult task than other on-line document retrieval. Named pages are frequently named by anchor descriptions and those pages are usually linked to with a great number of web pages. The consequence is that those pages obtains a variety of anchor descriptions. The appropriate name of each page is able to be assumed to be the name with the most frequent occurrence. With the according reasons, document representation, anchor descriptions, or title texts, for instance, is noticeably nifty for addressing the relevant named pages.

In this paper, we propose a novel approach to enhance the retrieval performance of named page finding through using frequent item sets. We consider the anchor descriptions and title texts as the item sets. Each item set represents the name of the document. Afterward, we discover the frequent item sets using the Apriori [6]. We then apply the new scoring method, i.e. the *maximum correspondence* function, to address the

most significant frequent item set. In addition, we also study from resulting experiments on the integration of anchor descriptions and title texts to enhance the named page retrieval precision.

We organize this article in the following way—Section 2 describes about the document representations. Section 3 describes about the frequent item set discovery algorithm. Section 4 is dedicated for more details we supplement to the algorithm. Section 5 describes about the obtained experiment result. Section 6 finally concludes the paper.

Anchors	Occurrences
Kasetsart University	1,500
Kasetsart	834
Kasetsart Home Page	322
KU	301
KU City	115
Main	75
Home	24

Table 1: An example of anchor descriptions.

2 Document Representations

There are many categories of document representations; for example, contents, abstracts, title texts, anchor descriptions, links, the depth of URL addresses, and the integration of those [1, 2], each of which has a potential of improving the retrieval performance. Nonetheless, there are advantages and disadvantages of each. In this work, we select the anchor descriptions and the title texts. It is due to the fact that they offer tremendous improvement of retrieval performance with the support of relatively small size of document representation.

2.1 Anchor Descriptions

A *hyperlink* is a relationship between two documents or two fragments of the same document. In the hypertext markup language (HTML), the target is referred by the link and the link's anchors are displayed to the users with the underlined text. As soon as the user click the anchor, their browser will display the target document. We call a link anchor appearing on the browser as an *anchor description*. The anchor description always describes its target. Moreover, a target is likely to be associated by other web documents and initiates considerable anchor descriptions if its target is popular.

Previous studies [3, 4] found that anchor descriptions has a potential to enhance the retrieval performance. Therefore, we are in agreement to construct anchor documents [3] as a representative of each document in data collection for a substitution of the term. Each anchor document contains all anchor descriptions of a page's incoming links. Table 1 illustrates an example of the anchor document of the web page `www.ku.ac.th`. In this paper, we will call each entry as an *anchor description set*. Moreover, we will represent it with the notation $N \times \Lambda$, where N is the number of occurrences and Λ is the element sequence of the anchor descriptions. For example, the first entry of Table 1 can be represented with the notation $1500 \times$ Kasetsart University. It is noticeable that each anchor description is the page's name appearing on the browser. The anchor description sets consequently represent the associativity between the link's anchor and its name.

2.2 Title Texts

The titles of web pages can be used as the document representation and it yields comparable, or even superior, retrieval results than full-text retrieval. Previous study [5] extracted several tag fields from data collection and found that the `<title>` field could produce better performance comparing with full-text retrieval. Furthermore, the retrieval system takes less substantial time and space. Therefore, we extract the title fields from the data collection and amass them similarly to the anchor description sets. We nevertheless separate the extracted title texts from those.

2.3 Size of Document Representations

We provide the quantitative measurement of the full text, title texts, and anchor descriptions derived from the .gov collection in Table 2. It is noticeable that anchor descriptions and title texts are relatively smaller than the whole collection. Moreover, they also contain relationship among links and their names regardless of document contents. This facilitates searching relevant named pages in collection containing only documents' names. As a result, we will likely obtain higher recall performance.

3 Frequent Item Set Discovery

Discovery of frequent item sets is the principle spirit of a great number of data mining approaches and it has been well studied in the context of association rules [7]. We usually decompose the problem solution of association rule mining into two phases—discovery of frequent item sets and rule generation from the discovered frequent item sets.

3.1 Frequent Item Sets

Let $I = \{i_1, i_2, i_3, \ldots, i_m\}$ be set of literals, so-called *items*. Let a non-empty set of items T be called an *item set* or a *transaction*. Let database D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Each transaction has its statistical significance, so-called *support value*. The support of the item set X in the database D is defined in Equation 1.

$$\text{support}(X, D) = \frac{|\{T \in D | X \subseteq T\}|}{|D|} \quad (1)$$

Let X be an item set. A transaction T is said to contain X if and only if $X \subseteq T$. The support of an item set X is the number of the percentage of transaction in the database that contains X. X is frequent if the support of X is not less than a user-defined support threshold, called *minimum support*.

Example Let Table 3 be a transaction database D, with a set of items $I = \{a, b, c, d, e, f, g, h\}$. Let the minimum support be 20 percents; the consequent frequent item sets are listed in Table 4.

3.2 Apriori Algorithm

The algorithm called *Apriori* [6] iteratively generates all possible item sets whose supports qualify the minimum support threshold. The first iteration of the algorithm counts item occurrences in order to generate the frequent 1-item sets (each 1-item set exactly contains only one item). For each next iteration, the frequent item set L_{k-1} found in the $(k-1)$th iteration are used to generate the candidate item sets C_k, using `apriori-gen` function described in Algorithm 1.

Representation	Size	Nb. Doc	Avg. Len.
Full Text	19,455 MB	1,247,753	15.2 KB
Title Text	148 MB	893,544	0.17 KB
Anchor Description	3,302 MB	827,256	4.0 KB

Table 2: Quantitative measurement of each document representation derive from the .gov collection.

Trans. ID	Item sets
1	a, b, c, d, f
2	b, c, d, f, g, h
3	a, c, d, e, f
4	c, e, f, g

Table 3: An example of a Transaction database D

Length (l)	Frequent l-item sets
1	a, b, c, d, e, f, g
2	$ac, ad, af, bc, bd, bf, cd, ce, cf, cg, df, ef, gf$
3	$acd, acf, adf, bcd, bcf, cdf, cef, cfg$
4	$acdf, bcdf$

Table 4: Frequent item sets with 20-percent minimum support derived from transaction database D.

Afterward, the database is scanned and the support of candidates in C_k is counted. The output of the first phase of Apriori algorithm consists of a set of k-item sets (where $k = 1, 2, 3, \ldots$), whose supports qualify the specified minimum support threshold. Algorithm 1 presents a formal description of the algorithm. We assume that items in each item set are lexicographically sorted.

Algorithm 1 The Apriori Algorithm

```
Scan D to find L_1.
Let k = 2.
while L_{k-1} ≠ ∅ do
  C_k = apriori-gen(L_{k-1}).
  for all transaction t ∈ D do
    C_t = subset(C_k, t).
    for all candidate c ∈ C_t do
      c.count = c.count + 1.
    end for
  end for
  L_k = {c ∈ C_k | c.count ≥ minsup}.
end while
return ⋃_k L_k.
```

Candidate item sets C_k are generated from previously generated frequent item sets L_{k-1} through using the `apriori-gen` function. The `apriori-gen` performs two operations, as follows.

Joining step: Each large item set from L_{k-1} is joined together.

Pruning step: Each item set $c \in C_k$ such that some $(k-1)$-subset of c, in which L_{k-1} does not contain, is deleted.

4 Contributions in Frequent Item Set Discovery

In this article, we will focus on the TREC-11 named page finding mission. We separate the method into three phases, as follows.

4.1 Document Representation Extraction

In this phase, we extracted the document representation, i.e. anchor descriptions and title texts, from TREC-11 data collection suite. Current data collection (Year 2002) is a crawler's outcome concentrated on the government domain (`.gov` websites) and it consists of 1.25 million documents. We analyzed links and their anchors from each document. Afterward, we kept those in their targets' anchor document. We then constructed an anchor description collection with every document. An anchor document collection comprises of 5.5 million documents. It contains anchor documents in data collection and contains anchor documents created from links whose target does not occupy in the data collection. Moreover, we filtered out the anchor documents whom the data collection does not contain, as well. After that, we subsequently performed morphological analysis process. We filtered out the stop words and every special sign except the hyphen (-), but we did not, in contrast, perform lexicon stemming. Finally, an anchor document collection contains only 0.8 million documents. We also extracted the title texts from the data collection. We kept them in the collection analogous to the anchor documents.

4.2 Frequent Anchor Description Discovery

Let $I = \{w_1, w_2, w_3, \ldots, w_n\}$ be a set of words in an anchor document. Let T be an item set that $T \subseteq I$. For example, the set of anchor descriptions representing Kasetsart University web site is shown in Table 5.

Let the minimum support be 20 percents. We discovered the frequent anchor descriptions via the Apriori algorithm (see Algorithm 1). We provide the consequence of applying the Apriori to data in Table 5 in the Table 6. We then extract the frequent anchor descriptions from anchor documents in order to provide an availability of retrieval. For title text collection, in view of the fact that each document possesses only one item set, we will permanently judge it to be frequent.

4.3 Relevant Named Page Retrieval

For the reason that the present retrieval methodology exploiting term frequencies and their weights in the data collection does not have a capability to rank the frequent anchor descriptions for this system, we therefore develop the *maximum correspondence* function, as in Equation 2.

$$\Psi_i = \max_j \vec{q} \cdot \vec{\lambda}_{ij} \tag{2}$$

Where, Ψ_i is a maximum correspondence among the query $\vec{q}$ and each $\vec{\lambda}_{ij}$ anchor description of the document i.

The equation ranks each document with correspondence between the query with each frequent anchor description. The maximum score from every frequent anchor description is the score of the document. We likewise apply the Equation 2 with the title text collection.

5 Experimental Results

With the intention of having a manageable task, we decomposed the evaluation phase into three experiments. We evaluated the system with only the frequent anchor description approach, then the title texts, and finally the integration of both. 150 queries are automatically prepared from TREC-11 named page finding task's queries without stems. We illustrated the retrieval result in Table 7 and Figure 1. The distribution of the result is depicted in Figure 2.

From Table 7, the frequent anchor description outperforms the title texts collection, since the frequent anchor description has a capability to provide more documents' names considered as exact names. Nonetheless, some documents do not have or have inadequate anchor descriptions due to incomplete links. These problems essentially affects the retrieval performance. On the other hand, we found that web documents always have title texts that can likewise perform well in case of the mentioned problems. Therefore, it is noticeable that the integration of both yields the better performance, as shown in Figure 1 and Figure 2. The ranks of relevant named pages are boosted to the top rank and more relevant named pages can be found.

6 Conclusion and Future Works

The frequent anchor descriptions can represent the document's name with relatively small size of data. However, the incomplete link problem principally affects the reduction of performance. The title texts can elucidate this problem, but it provides inadequate information, in contrast. The integration of both can provide better exper-

Item Set ID	Anchor Description
1	Kasetsart University Page
2	Homepage of Kasetsart University
3	Kasetsart The University of Agriculture
4	Kasetsart University Page
5	Homepage of Kasetsart University
6	Homepage of Kasetsart University
7	Kasetsart University Page

Table 5: Anchor description representing Kasetsart University's web site

Length (l)	Frequent l-anchor description
1	{ Kasetsart }, { University }
2	{ Kasetsart University }, { Kasetsart of }, { Kasetsart page }, { Kasetsart Homepage }, { University Page }, { University Homepage }, { University of }
3	{ Kasetsart University of }, { Kasetsart University Page }, { Kasetsart University Homepage }, { Kasetsart Homepage of }. { University Homepage of }
4	{ Kasetsart University Homepage of }

Table 6: Frequent anchor descriptions of Kasetsart University's web site

Ranks	Titles	Anchors	Integration
Top 1	20	56	64
Top 5	24	19	20
Top 10	9	8	10
Top 25	13	9	13
Top 50	5	4	4
Not Found	79	54	39
MRR	0.406	0.588	0.680

Table 7: Named page finding retrieval result

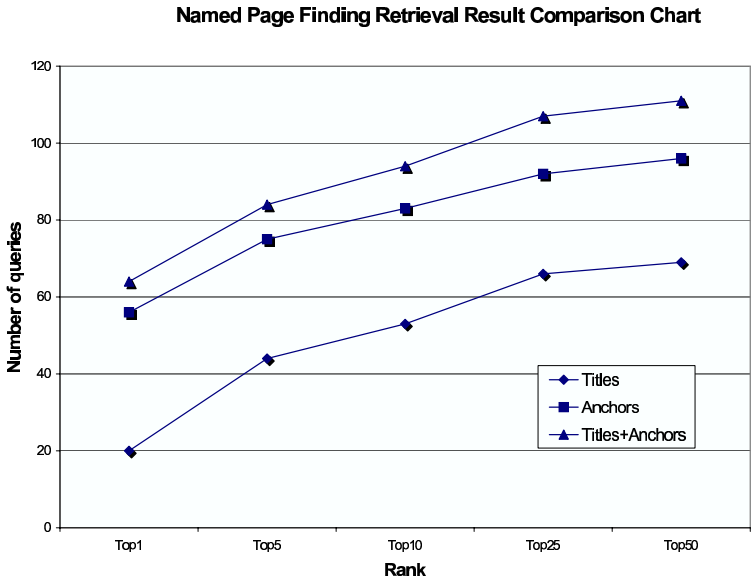

Figure 1: Named page finding retrieval result comparison chart

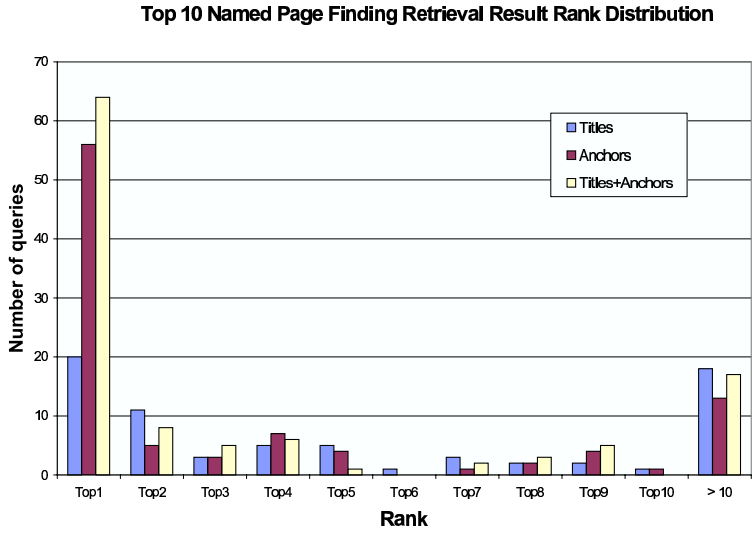

Figure 2: Name paged finding retrieval result distribution

imental result due to the management ability of incomplete links and relatively small size of data comparing with full text data. We concluded that the frequent anchor description provides a valuable source of information for named page finding task.

The frequent anchor description can enhance the retrieval performance for commercial search engines, but the maximum correspondence function operates very lingeringly on simple frequent anchor description file structure. We consider this our future work. The association rule discovery from frequent anchor description can provide confidence value for the scrutiny of strong frequent anchor description. Moreover, it can reduce the size of frequent anchor description collection.

Acknowledgement

We would like to thank all MIKE staffs for their comments and working spirit. We are much obligned to Prachya Boonkwan for his kindness in reviewing this paper.

References

[1] J.A. Shaw and E.A. Fox, *Combination of multiple searches*, in Proceedings of the 3rd Text REtrieval Conference (TREC-3), pp. 105-115. Gaitherburg, MD: National Institute of standards and Technology, 1995.

[2] C.C. Vogt and G.W. Cottrell, *Predicting the performance of linearly combined IR systems*, in Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp. 190-196, New York: ACM, 1998.

[3] N. Craswell, D. Hawking and S. Robertson, *Effective Site Finding using Link Anchor Information*, in Proceedings of 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 250-257, 2001.

[4] Sergey Brin and Lawrence Page, *The anatomy of a Large-scale hypertextual web search engine*, in Proceedings of WWW7, 1998.

[5] W. Xi and E.A. Fox, *Machine Learning Approach for Homepage Finding Task*, in Proceedings of the 10th Text REtrieval Conference (TREC-10), pp. 686-697, Gaithersburg, MD: National Institute of standards and Technology, 2001.

[6] R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*, in Proceedings of VLDB'94, pp. 487-499, Santiago, Chile, 1994.

[7] R. Agrawal, T. Imielinski and A. Swami, *Mining Association Rules Between Sets if Items in Large Database*, in Proceeding of ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington DC, USA, 1993.