Chapter 1

Introduction

As the rapid growth of on-line documents and electric publications, the size of document databases has become very large and the number of document databases has quickly increased. When an user searches very large databases, the original query of the user is often a naive query that retrieves too many documents to browse exhaustively. Facing to large result set of the original query, the user has to refine the original query. However, the task of refining an effective query is difficult in the sense that it requires users, without any knowledge about the collection of documents. Therefore, it is highly desirable that a system can support query refinement by displaying information about the collection of documents to users.

Query refinement is mainly concerned to information retrieval field in which a number of researches have been developed such as query expansion, relevance feedback, etc. In the other hand, data mining([Fayy96]), also known as knowledge discovery in databases, has been recognized as a new area for database research. Query refinement also relates to data mining and knowledge discovery in databases because query refinement is not only the process of retrieval but also the process of discovering knowledge that can be displayed to users as a guide for improving query.

In this paper, a method of query refinement support for keyword retrieval of document databases is proposed by mining Association Rules (ARs) among keywords being extracted from a document database.

1.1 Document Retrieval by Keyword

Keywords have been widely used as queries in document databases, web search engines, etc. In document databases, a document includes a set of keywords and a keyword associates with a set of documents. A query is described as a disjunctive normal form of keywords. A conjunction of keywords retrieves the documents each of which contains all the atom keywords in the conjunction. The query retrieves a set of documents which is the union of all the results of each conjunction. The relationship between the retrieval result and the query is described as follows. The more the number of keywords in a conjunction is, the less the number of documents retrieved by the conjunction is. The more the number of conjunctions in a query is, the more the number of documents retrieved

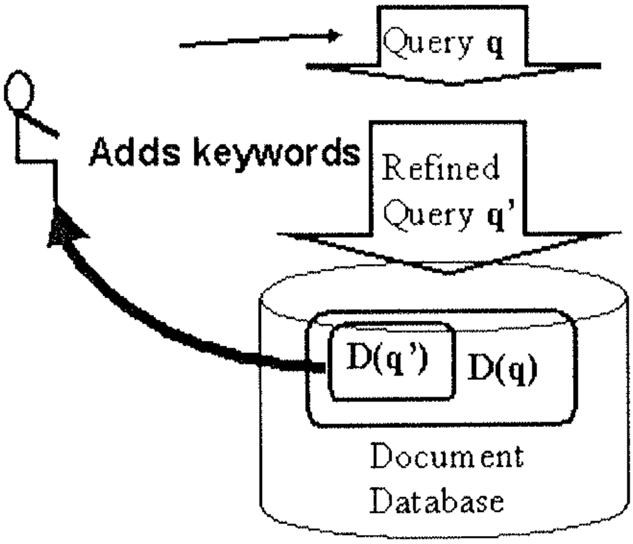


Figure 1.1: Query refinement

by the query is. Adding keywords to conjunction form can reduce the size of result set and adding keywords to the disjunction form can increase the size of result set.

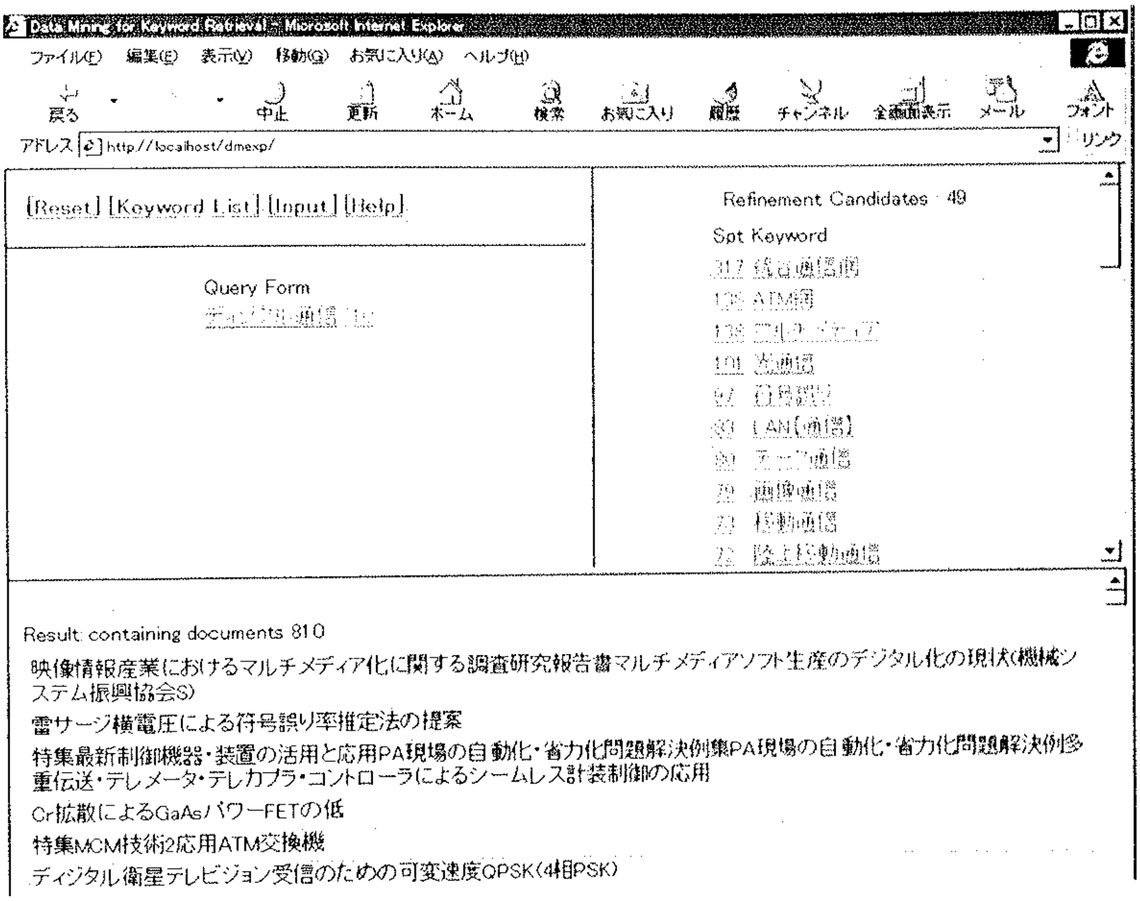
The original query submitted by user is not often an effective query that retrieves a large result set or small result set. In order to formulate an effective query, it is important that each of conjunction in the query has a suitable size of result set. This research discusses how suitable conjunctions are formulated.

1.2 Query Refinement Support

If the original query submitted by an user retrieves a very large result set the user has to refine her/his original query by conjunctively adding appropriate keywords to the original query or choosing a new keyword as a new query. Figure 1.1 shows the process of query refinement. An user's original query q retrieves the result set $D(q)$ that has a lot of documents. In order to get a suitable size of result set, the user refines the original query q to a new query q'. The new query q' retrieves the result set $D(q')$ that satisfies user' need.

Query refinement is the process of transforming a query into a new query that more accurately reflects the user's retrieval need. However, it is difficult for users, without any knowledge about the collection of documents, to predict the keywords that will appear in the documents the users want to have. Therefore, it is highly desirable that a system can provide information about the collection of documents for helping users to refine their queries.

Figure 1.2 shows an example of query refinement support. First, the system offers a set of the refinement candidates with respect to the user's original query. Then, user references the refinement candidates and adds appropriate keywords selected from the refinement candidates to the original query. In the example, the original query submitted by an user is "digital communication" that has a large result set of 810 documents. The refinement candidates with respect to "digital communication" are displayed on the right


Data Mining for Keyword Retrieval - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
戻る
中止
更新
ホーム
検索
お気に入り
履歴
チャンネル
全画面表示
メール
フォント
アドレス http://localhost/dmexp/
リンク

[Reset] [Keyword List] [Input] [Help]

Query Form

ディジタル通信 [illegible]

Refinement Candidates 49

Spt Keyword

317 統合通信網
135 ATM網
138 マルチメディア
101 光通信
97 符号誤り
83 LAN(通信)
80 データ通信
79 画像通信
73 移動通信
72 陸上移動通信

Result containing documents 810

映像情報産業におけるマルチメディア化に関する調査研究報告書マルチメディアソフト生産のデジタル化の現状(機械システム振興協会S)

雷サージ横電圧による符号誤り率推定法の提案

特集最新制御機器・装置の活用と応用PA現場の自動化・省力化問題解決例集PA現場の自動化・省力化問題解決例多重伝送・テレメータ・テレカプラ・コントローラによるシームレス計装制御の応用

Cr拡散によるGaAsパワーFETの低

特集MCM技術2応用ATM交換機

ディジタル衛星テレビジョン受信のための可変速度QPSK(4相PSK)

Figure 1.2: An example of query refinement

frame of screen. Associating to each candidate, the number on the right side of refinement candidate shows the size of result set returned by conjunctively adding the keyword to the original query "digital communication". If the user chooses refinement candidate keyword "picture communication" and adds conjunctively to the original query, the query {"digital communication", "picture communication"} will retrieve 79 documents each of which contains both "digital communication" and "picture communication". This process will be carried out repeatedly until the user successfully refines the query.

Here, the refinement candidates are extracted from document database by statistics methods based on co-occurrence between keywords. Co-occurrence between keywords means that pairs of words occur frequently together in documents. In the example above, the relationship between keywords "digital communication" and "picture communication" is called co-occurrence because these two keywords appear together in more than one documents. Keyword "picture communication" is an relevant keyword of "digital communication" and vice versa. In very large databases, a keyword always has a lot of relevant keywords. Therefore it is hard for users to browse all co-occurrence keywords. In such databases, query refinement support systems choose only the part of co-occurrence keywords to be displayed to users. The method of choosing refinement candidates from co-occurrence keywords will be discussed in this research.

When document databases are very large, the computation of co-occurrence keywords might spend a lot of time. The computation of co-occurrence keywords need to also be taken into consideration in query refinement support system.

1.3 Related Researches

Our research mainly relates to query expansion, reference feedback and query refinement by similarity.

1.3.1 Query Expansion

Query expansion is a traditional method of query modification. In brief, the technique of query expansion is based on an association hypothesis which states: if an index term is good at discriminating relevant from non-relevant documents then any closely associated index term is also likely to be good at this ([Van79]). In prior works, synonyms and variant spellings of the original query are automatically added by means of thesauri and controlled vocabularies. More recent works on query expansion ([Peat91, Qiu93, Voor94, Buc95b, Srin96, Xu96]) have been based on global and local analysis etc. These researches add automatically related keywords to the original query according to keywords extracted from the matched document set. These researches use statistics techniques to identify keywords that are similar to query and that should be added to the query. Calculation of the degree of similarity between pairs of keywords is usually based on similarity coefficients, such as the cosine, Dice or Tanimoto coefficient [Peat91] which are defined in the following.

Given two keywords X and Y occurring in $F(X)$ and $F(Y)$ document, respectively, these coefficients are defined to be

$$COSINE(X,Y) = \frac{F(X,Y)}{\sqrt{F(X) \times F(Y)}}$$

$$DICE(X,Y) = \frac{2 \times F(X,Y)}{F(X) + F(Y)}$$

and

$$DICE(X,Y) = \frac{F(X,Y)}{F(X) + F(Y) - F(X,Y)}$$

where $F(X,Y)$ is the number of documents in which X and Y co-occur. It will be noticed that these coefficients are symmetric .

The extracted keywords have high similarity to the keywords used in the original query. However, these approaches have the difficulties to differentiate between relevant and non-relevant documents, because the keywords with higher similarities intend to include the similar documents. It is also difficult for the system to accurately reflect the user's retrieval request.

1.3.2 Relevance Feedback

The relevance feedback in [Salt90, Efth93, Spin94] is the method of interactive query modification. It request the user to browse the result set of documents and choose relevant documents the keywords in which are added to the original query. Relevance feedback is based on probabilistic model to rank the result set of documents, in response to a query, in order of decreasing probabilities relevance ([Koll90, Aalb94, Coop94, Pers94]). That is, if X, a vector of binary weights of keywords, is the description of a document, then that document's rank is determined by the conditional probability $P(relevance|X)$. The process of document rank is as follows. First, numerical weights are assigned to individual keywords in documents(and possibility in queries). Then, the individual weight are combined to obtain an overall measure of significance of the document to the compound query. Finally, these significance values are used to produce ranking of documents. But, in order to choose refinement keywords, users have to browse a lot of documents.

1.3.3 Query Refinement by Similarity

Query refinement is also an interactive method of query modification. Query refinement ranks documents with respect to user's original query and chooses keywords from ranked documents as suggested candidates ([Véle97]). The suggested candidates are added to the original query by user. When an user gives a query, query refinement system first finds all documents matching the user's query. The system then combines and ranks the keywords in documents, and finally displays the highest ranked keywords as suggested candidates. The user can get information about the collection by browsing the list of suggested candidates.

Query refinement also chooses the keywords with higher similarities as suggested candidates. This is similar to query expansion. The usage of symmetric similarity leads to bad effectiveness of screening and dynamic computation of similarity delays the query response.

1.3.4 Summary

The researches above are based on co-occurrence between keywords to refine queries. This seems to be a common principle of query refinement. The researches above use symmetric similarity to estimate relationship between keywords and choose the keywords with higher similarities as refinement candidates. This leads to bad effectiveness of screening and has the difficulties to differentiate between relevant and non-relevant documents. Query expansion is a automatic method of query modification. But it is also difficult for the system to accurately reflect the user's retrieval request. Relevance feedback is an interactive method query modification. But, in order to choose refinement keywords, users have to browse a lot of documents. Query refinement offers a set of the suggested candidates for user to choose. But the method of generating refinement candidates in query refinement have some weak point such as effectiveness of screening, dynamic computation of similarity etc.

We will discuss the detail in the next chapter.

The rest of this thesis is organized as follows. Chapter 2 introduces the overview of this research. In Chapter 3, the preliminaries of this study is given and the stem rules and the stepwise refinement will be discussed. Chapter 4 defines the coverage and a refinement query space based on the coverage. Chapter 5 gives an outline of our prototype system and addresses experimental results. Chapter 6 concludes our discussion.