Keyword Search Based Information Retrieval System in Relational Database

Ei Ngwe Sin
Computer University, Mandalay
eingwesin.ucsm.edu@gmail.com

Abstract

Keyword search technology has been the most widely used querying method nowadays. The amount of data in relational databases is growing rapidly. The information discovery from these databases is necessary for users. Free-form keyword search over relational database management system (RDBMS) has been attracted. The information discovery from the commercial RDBMS requires the knowledge of query language and the database structure .This paper presents the medicine information searching system that is implemented on the free-form keyword search strategies. This system allows users to search medicine information by entering keyword only. The algorithm of join expression generator is used to generate the join expressions of non-free tuples. As a ranking function, the "state-of-the-art IR definition" ranking function is used. The resulted answers are ranked according to the score from the ranking function. Thus, the system can give the answers that are more related to the input sentence.

Keyword Relational Database, Keyword Search

1. Introduction

Information retrieval (IR) is a field that has been developing in parallel with database systems for many years. As the increasing usage of computer systems, it became possible to store large amounts of information; and finding useful information from such collections became a necessity. Thus, structured databases (in internet or in enterprise or in personal desktops) usually also contain a large amount of text data. The need for ordinary users to find information from text in these databases is more increasing.

The objective of this paper is to provide effective search of text information in relational database. The free-form keyword search is implemented on one relational database that keeps the information of medicine. The resulted answers are ranked according to the ranking function called "state-of-the-art IR definition".

The rest of the paper is structured as follows: Section 2 discusses the related work. Then, Section 3 describes the keyword search in relational database. The algorithm of join expression generator is in Section 3.1 and Section 3.2 describes the ranking function of the system. Section 4 describes the system design. Section 5 explains the implementation of the system. The evaluation result of the system is shown in Section 5.1. Section 6 describes the conclusion and future work of the system.

2. Related work

There are many different approaches for keyword search in RDBMS. The problem of free-form keyword search over structured and semi-structured data has been addressed in [2]. Discover system exploits the RDBMSs schema graph information to return qualified joining trees of tuples as results [3]. IR-style keyword search proposed to use information retrieval (IR) ranking technologies for keyword search in relational databases to get results that are more effective. They also proposed some efficient query-processing algorithms to obtain Top-K results [4]. The comparison of IR system and relational database has been proposed in [6]. They showed that relational database management systems are insufficient for full text searches. Usage of %LIKE% operator is too primitive compared to information retrieval.

3. Keyword search in relational database

In relational databases, information is stored in the form of columns, tables and primary key to foreign key relationships. The logical unit of answers needed by users is not limited to an individual column value or even an individual tuple; it may be multiple tuples joined together. It is necessary to assign a single ranking score for each

tuple tree (joining expression), which may consist of multiple tuples with text columns, in order to rank the answers effectively.

3.1 Generating of join expression

In this paper, to generate join expression, the concept of 'candidate network generator algorithm' is used from [3]. The algorithm is shown in Figure 1:

Algorithm of Join Expression Generator

1. Traverse tuple set graph breadth first
2. Q ← tuple sets containing keyword
3. For each join expression n of tuple sets in Q
   do
   3.1. if n satisfies the pruning condition then ignores n
   3.2. else if n satisfies the acceptance condition then output n
   3.3. else expand n by one tuple set to all possible directions in tuple set graph and insert expansions to Q

Figure 1.Algorithm of join expression generator

The conditions in the algorithm are defined as follows. The pruning condition is the condition that the join expression contains the same non-free tuple more than once. The condition that the join expression of tuple sets J contains at least two keywords, i.e, keywords (J) = $\{k_1,k_2\}$, is the acceptance condition. The definition of the expansion rule is that if the size of join expression of tuple sets does not exceed the constant (T), the more tuples can be added to the join expression. For example T=3 or 4 etc…based on the database schema.

3.2 Ranking function

The various ranking methods are used in [3],[4],[1].In this paper, the "state-of-the-art IR Definition " ranking function is used [4]. This ranking function is developed exactly to improve the document-ranking quality for free-form keyword queries. As the modern RDBMSs already include full text search capabilities over individual text attributes, they use these capabilities to improve the quality of results over RDBMS.

A state-of-the-art IR definition for a single-attribute scoring function Score is as follows in equation 1:

$$Score(a_i, Q) = \sum_{w \in Q \cap a_i} \frac{1+\ln(1+\ln(tf))}{(1-s)+s\frac{dl}{avdl}} . \ln \frac{N+1}{df} \quad (1)$$

where for a word w in keyword query Q and attribute a_i, tf is the frequency of w in a_i, df is the number of tuples in a_i 's relation with word w in this attribute, dl is the size of a_i in characters, avdl is the average attribute-value size, N is the total number of tuples in a_i's relation and s is a constant (usually 0.2).

The final score for join expression T, The Score function is as follows in equation 2,

$$Score(T,Q)=Combine\ (Score(A,Q),size(T)) \quad (2)$$

where,$Score(A,Q)=<Score(a_i,Q),\ldots,Score(a_n,Q)>$

The definition for Combine is in equation 3,

$$Combine\ (Score(A,Q),size(T))= \frac{\sum_{a_i \in A} Score(a_i,Q)}{size(T)} \quad (3)$$

The definition of size(T) is based on the application.

4. System design

4.1. System flow diagram

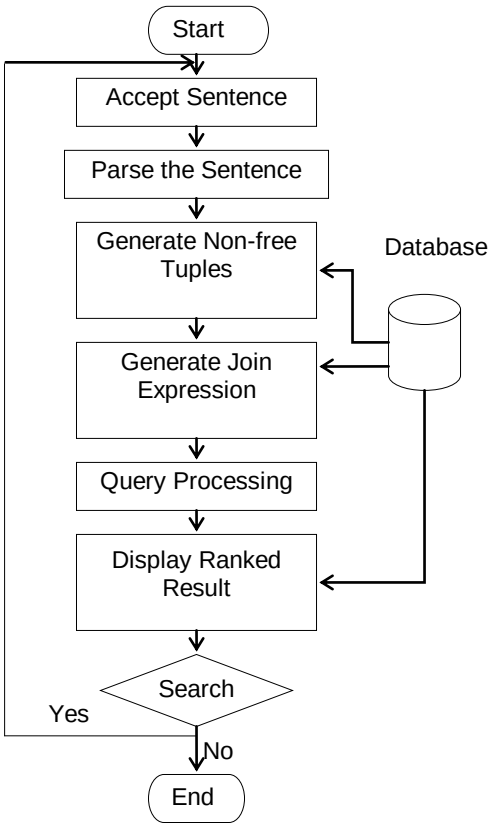

Figure 2 System Flow Diagram

Figure 2 is the system flow diagram of keyword search process. In keyword search process, the input from user is a sentence. This sentence may contain keywords such as the brand name of medicine, the symptoms of diseases, the pharmaceutical classification of the medicine, etc. In other words, the part of information that he wants to know is input sentence. In parse the sentence stage, the input sentence is parsed into keywords. Then the process generates non-free tuples that contain at least one input keywords from the database. In the next stage, the join expression of tuples is generated based on the database structure. For each join expression, the process generates score by using "State-of-the-art IR Definition" ranking function. Finally, the ranked results are displayed to the user.

5. Implementation

This system is implemented on Intel Pentinum 4 Processor, 512 RAM memory, running on the Microsoft Windows XP. The underlying database is Microsoft SQL Server 2005.

There are three main tables in the medicine database of the system. They are "Product Table", "Group Table", "Manufacture Table". The "Product Table" stores the product name and indication as shown in Table 1. The "Group Table" stores ingredient and dosage as shown in Table 3. The "Manufacture Table" stores company name and country as shown in Table 2.

Table 1.Product table

PId	Pname	Indication	GId	MId
p1	Biogesic	Either pain of fever alone,or where both pain and fever exist. wide variety of painful conditions including headache,dysmenorrhea,myalgias,neuralgias,rheumatic fever, common colds, flu and otherupper respiratory tract infections.	g1	m1
p2	Calpol	Treatment of mild to moderate pain andas antipyretic, symptomatic relief of headache, migraine, neuralgia, toothache and teething pains,sore throat, rheumatic aches and pains, influenza, fever and feverish colds.	g2	m1
p3	Konidin	Cough due to allergy, flu, common cold or symptoms of bronchitis.	g4	m3
P4	Mixagrip	To relieve influenza symptoms	g3	m4

Table 2. Manufacture Table

MId	Company	Country
m1	Ranbaxy	India
m2	Konimex	Malaysia
m3	Galaxosmithkline	Thailand
m4	United Lab	Phillipines

Table 3. Group Table

GId	Ingredient	Dosage
g1	Paracetamol	Mild cases: Adults 1 tab older Children ½ tab.
g2	Paracetamol 500 mg tablet suspension 120 mg/5ml	Caplet Adults 1-2 caplets 3-4 time/day .
g3	Paracetamol, Phenylpropanol amine HCl	Caplet Adults 1-2 caplets 3-4 time/day child :6-12 years:1/2 caplet 3 times
g4	Chlorphenirami ne maleate, dextromethorph an HBR.	Adults and child above 12 years: 1-2 tablets 3 times daily

5.1 Evaluation result

Assume the user input sentence is "medicine for headache and fever produced by India". The system parses the sentence into keywords. The resulted keywords are "headache", "fever" and "India". The non-free tuples that contains at least one keyword are generated. They are "p1" and "p2" from Product table and "m1" from Manufacture table.

Then join expressions are generated by using the candidate network generator algorithm. Based on the database structure, the value of T is 2 for the maximum number of joins. Table 4 is the generation of join expressions for "p1". The resulted join expression "p1<>Manufacture<>m1" is marked with "*".

Table 4. Generation of Join Expression

#	Queue/From
1a	p1
2a	p1<> Group{}
b	p1<> Manufacture{}
3a	p1<>Group{}<>Product{}/2a

b	p1<>Group{}<> p1/2a
c	p1<>Group{}<>p2/2a
d	p1<>Manufacture{}<>Product{}/2b
*e	p1<>Manufacture{}<>m1/2b
f	p1<>Manufacture{}<>p1/2b
g	p1<>Manufacture{}<>p2/2b

The resulted join expressions from "p2" and "m1" are "m1<>Product<>p2", "m1<>Product<>p1", "p2<>Manufacture<>m1" respectively.

5.1.1. Calculation of score for keyword "headache", "fever" and "India". The keyword "headache" and "fever" is found in "Indication" attribute of tuples, "p1" and "p2" of Product table. Thus, for keyword query "Q", the score for them in each tuple are as follows.

$$\text{Score (p1.Indication,Q)}= \frac{1+\ln(1+\ln(1))}{(1-0.2)+0.2\frac{220}{112}}.\ln\frac{4+1}{2} + \frac{1+\ln(1+\ln(3))}{(1-0.2)+0.2\frac{220}{112}}.\ln\frac{4+1}{2} = 2.1057$$

$$\text{Score (p2.Indication,Q)}= \frac{1+\ln(1+\ln(1))}{(1-0.2)+0.2\frac{149}{112}}.\ln\frac{4+1}{2} + \frac{1+\ln(1+\ln(1))}{(1-0.2)+0.2\frac{149}{112}}.\ln\frac{4+1}{2} = 1.719$$

The keyword "India" is found in "Country" attribute of tuple,"m1" of Manufacture table.

$$\text{Score(m1.Country,Q)}= \frac{1+\ln(1+\ln(1))}{(1-0.2)+0.2\frac{5}{6.4}}.\ln\frac{4+1}{1} = 1.6831$$

5.1.2 Calculation of combine score for each join expression. In "Combine" function, the size(T) is defined as 1 for this system because the system uses one relational database. The combine score for resulted join expressions, "p1<>Manufacture<>m1" is as follows:

$$\text{Combine (Score (A,Q) ,size(T))}= (2.1057+1.6831)/1 = 3.7888$$

For "m1<>Product<>p2", "m1<>Product<>p1" and "p2<>Manufacture<>m1", the scores are 3.4021, 3.7888, 3.4021, respectively.

5.1.3. Displaying the ranked result. The join expressions are executed in query processing. If the results are duplicated, the system choose one result. The output is the information of medicine "Biogesic" and "Calpol". The first result is "Biogesic" and the second is "Calpol" according to the score. The result is shown in Table5.

Table 5. Result Table

Product	Indication	Ingredient	Dosage	Company	Country
Biogesic	Either pain or fever alone, or where both pain and fever exist. Wide variety of painful conditions including headache, dysmenorrheal,myalgias, neuralgias, rheumatic fever, common colds, flu and other upper respiratory tract infections	Paracetamol	Mild case sAdult1 tab olderChild ½ tab	Ranbaxy	India
Calpol	Treatment of mild to moderate pain and as antipyretic. Symptomatic relief of headache, migraine, neuralgia, toothache and teething pains, sore throat, rheumatic aches and pains, influenza, fever and feverish colds.	Paracetamol 500mg tablet suspension 120 mg/5 ml	Caplet Adult 1-2 caplets 3-4 time a day	Ranbaxy	India

6. Conclusion and future work

Nowadays, many strategies for keyword search in relational database are proposed. This paper uses the algorithm of join expression generator based on the concepts of candidate network generator algorithm. As the usage of the state-of-the-art IR definition, the information result that contains the user's input

keywords most is output as a top-1 result. If the two results contain the same number of the same input keywords, the result with little text is output first.

Thus, when user search with chemical ingredient of the medicine as input, the system can give the medicine information that contains only the user input chemical ingredient first, rather than the medicine information with the compound ingredient.The future work is that the more complex situation such as keyword search in multi-database with semantically related information will be considered.

7. References

[1] A. Sanjay, C. Surajit, D. Gautam, “DBXplorer:A System for Keyword-Based Search over Relational Databases".

[2] C. Yi, W. Wei, L. Ziyang, L. Xuemin, “Keyword Search on Structured and Semi-Structured Data”, SIGMOND'09, Providence, Rhode Island, USA., June 29-July 2, 2009, pp. 1-10.

[3] H. Vagelis, P. Yannis, “DISCOVER: Keyword Search in Relational Databases", Proceedings of the 28th *VLDB* Conference, Hong Kong, China,2002.

[4] H. Vagelis, G. Luis, P. Yannis, “Efficient IR-Style Keyword Search over Relational Databases", Proceedings of the 29th VLDB Conference, Berlin,,Germany, China,2003.

[5] L.Fang, Y. Clement, M. Weiyl, C. Abdur, “Effective Keyword Search in Relational Database",SIGMOd 2006,Chicago,Illinois,USA,June 27-29,2006.

[6] Y. Ozgur, Y. Burcu, Y. Raris, A. Arslanl, “Relational Databases Versus Information Retrieval Systems: A Case Study".