Novel Query Expansion Technique using Apriori Algorithm

A. Rungsawang, A. Tangpong, P. Laohawee, T. Khampachua
{fenganr,g4165239,g4165221,g4165248}@ku.ac.th

Massive Information & Knowledge Engineering
Department of Computer Engineering
Faculty of Engineering
Kasetsart University, Bangkok, Thailand

Abstract. One problem in query reformulation process is to find an optimal set of terms to add to the old query. In our TREC experiments this year, we propose to use the association rule discovery (especially apriori algorithm) to find good candidate terms to enhance the query. These candidate terms are automatically derived from collection, added to the original query to build a new one. Experiments conducted on a subset of TREC collections gives quite promising results. We achieve a 19% improvement with old TREC7 adhoc queries.

1 Introduction

Enriching a user's query with synonyms or related terms can improve search performance in a text retrieval system. There are at least two methods to reformulate a search query. The first one is to use relevance feedback where related terms come from the contents of user-identified relevant documents, or pseudo-relevance feedback where expanded terms come from the top k retrieved documents which are assumed to be relevant [4]. The second one is to include terms from an online thesaurus [6] or manually selected terms [3] to the old query. However, these two methods (except the pseudo-relevance feedback) involve with the presence of the user or an additional knowledge source.

We concentrate in our TREC experiments this year with a novel query enhancement technique. Additional terms appended to the original query are obtained from applying apriori algorithm, an association rule discovery used in data mining to extract some useful rules from a large database, to a subset of TREC collections. We have not obtained any promising result with the new adhoc TREC8 query set yet, but achieved 19% improvement with the old adhoc TREC7 queries using our DSIR retrieval system [5].

Our report has been organized in the following way. Section 2 introduces breifly the apriori algorithm, and shows how we apply it to a subset of TREC collections. Section 3 gives detail how we set up our experiments and provides preliminary results. Finally, section 4 concludes this report.

2 Applying Apriori algorithm to TREC Collection

The apriori algorihtm, introduced by Agrawal [1], has been widely used to mine useful knowledge in large transaction databases. Typically, a transaction is a list of items (or goods) purchased by a customer during a visit in a store or supermarket. Knowledge are derived in terms of a list of association rules such as "95% of customers who purchase tries and auto accessories also later get automotive services done", or a formal rule like "85% of customers who buy product A and B also buy product C and D. Discovering all such customer buying patterns is valuable for cross-marketing and attached mailing applications. Other applications include catalog design, product placement, customer segmentation, etc., based on their buying patterns [2].

The problem description of mining association rules can be given as follows. The *support* of a set of items (called later *itemset*) in a transaction database is the fraction of all transactions containing the itemset. An itemset is called *frequent* if its support is greater or equal to a user-specified *support threshold*. An *association rule* is an implication of the form $X \Rightarrow Y$ where X and Y are disjoint itemsets. The support of this rule is the support of $X \cup Y$. The *confidence* of this rule is the fraction of all transactions containing X that also contain Y (i.e. the support of $X \cup Y$ divided by the support of X). From the second example above, the "85%" is the confidence of the rule {A,B} ⇒ {C,D}. Given a set of transactions, the problem of mining association rules is to find all association rules that have support and confidence greater than the user-specified minimum support, and minimum confidence respectively [2].

To apply association rule mining to our query reformulation problem, we assume that each document can be seen as a transaction while each separate word inside can also be seen as item or product bought buy a customer. Applying apriori algorithm to a TREC collection with specified minimum and maximum support, and confidence will produce a set of association rules in form of $X \Rightarrow Y$, where X and Y are disjoint set of related words (or *wordset*). An enhanced query is then reformulated by adding all words in wordset Y if the wordset X appears in the original query.

3 Preliminary Experiments and Results

We still use DSIR [5] retrieval system in our experiments. Since each pass of association rule discovery takes several hours on a Pentium class machine, we then choose to conduct the experiments using only the FT (Financial Times) collection. FT is first preprocessed in form of transaction database before applying apriori algorithm. All derived rules are employed in the query expansion process. The minimum and maximum support, and confidence are then the additional indexing parameters. Table 1 below gives some results of our experiments.

The first row in Table 1 gives average precision concluded from a set of experiments using original TREC7 queries to search in FT collection with varying document-query weighting combinations. We choose this row as the baseline. The other rows gives results when our new query expansion technique has been

Run	lnc.ntc	lnc.atc	lnc.anc	lnc.ltc
no expansion	0.0722	0.0606	0.0458	0.0835
10-40-10	0.0857 (+18.70%)	0.0658 (+8.58%)	0.0333 (-27.29%)	0.0737 (-11.74%)
10-40-30	0.0861 (+19.25%)	0.0631 (+4.13%)	0.0428 (-6.55%)	0.0725 (-13.17%
10-40-40	0.0736 (+1.94%)	0.0624 (+2.97%)	0.0425 (-7.21%)	0.0893 (+6.95%)
10-40-50	0.0727 (+0.69%)	0.0610 (+0.66%)	0.0429 (-6.33%)	0.0835 (0%)

Table 1. Query expansion results.

applied. The series of numbers, for example "10-40-10", are the percentage of minimum and maximum support, and the percentage of confidence parameters used in apriori algorithm.

4 Conclusion

Table 1 illustrates very promising results. We can achieve 19% improvement with this query expansion technique, without using any additional knowledge source (i.e. thesaurus, or relevance data) or any user intervention. Since, in our TREC experiments this year, we are quite shortage of time and computing resource, we then unfortunately do not finish our test and has any result on the TREC8 query set.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Database. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington D.C., 1993.
2. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the* 20th *VLDB Conference*, Satiago, Chile, 1994.
3. P.G. Anick, J.D. Brennan, R.A. Flynn, D.R. Hanssen, B. Alvey, and J.M. Robbins. A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query. In Vidick J.L., editor, *Proceedings of the* 13th *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Bruxelles, Belgium, September 1990.
4. C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic Query Expansion Using SMART: TREC 3. In D. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*. NIST Special Publication 500-225, 1995.
5. A. Rungsawang. DSIR: The First TREC-7 Attempt. In E. Voorhees and D.K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, USA, November 1988. National Institute of Standards and Technology, NIST Special publication.
6. E.M. Voorhees. On Expanding Query Vectors with Lexically Related Words. In D. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*. NIST Special Publication 500-215, 1994.