질의 재구성 알고리즘의 검색성능을 측정하기 위한 새로운 평가 방법의 개발

김 남 호† · James C. French†† · Donald E. Brown†††

요 약

정보 검색에서 대부분의 질의 재구성 알고리즘들은 초기 입력 문서나 피드백 문서를 이용하여 질의를 재구성하므로, 질의 재구성 알고리즘의 검색 성능은 입력되는 문서들의 질에 따라 달라진다. 본 연구에서는 질의 재구성 알고리즘의 입력 문서에 대한 성능 감도를 새로운 검색성능 평가방법을 개발하여 분석하였다. 또한 CIRA라고 불리는 새로운 평가기준을 개발하여 질의 재구성 사이의 성능 변화추이를 분석하였다. 세가지의 질의 재구성 알고리즘 (질의나무 (query tree), DNF 방법, Dillon 방법)의 감도와 성능변화를 테스트 세트인 CACM, CISI, Medlars 상에서 분석하였다. 세 실험에서 질의나무가 가장 작은 CIRA를 취득했으며, 감도 분석에서는 비록 다른 알고리즘과 차이는 적으나 가장 높은 감도를 나타냈다.

Development of New Retrieval Performance Measures for Query Reformulation Algorithms

Nam-Ho Kim† · James C. French†† · Donald E. Brown†††

ABSTRACT

In information retrieval, query reformulation algorithms construct queries from a set of initial input and feedback documents, and retrieval performance can be varied by different sets of input documents. In this study, we developed a criterion for measuring the performance sensitivity of query reformulation algorithms to input sets. In addition, we also propose a way of measuring the changes in retrieved area (CIRA) during query reformulation. We compared CIRAs of query reformulation algorithms (i.e., query tree, DNF method, and Dillon's method) using three test sets: the CACM, CISI, and Medlars. In the experiments, the query tree showed the highest decreasing CIRAs during reformulations, which means the fastest convergence rate to an output set. For sensitivity analysis, the query tree scored the highest sensitivity to different input sets even though its differences to the other algorithms are very small.

1. Introduction

One of the fundamental questions in system evaluation is why, what, and how to evaluate a system [18]. In most cases, the objective of the system evaluation is the quality of the system, which can be measured by its performance. The purpose of the system evaluation can change for different circumstances. A system analyst might want to observe how well the system performs,

† 정 회 원: 선문대학교 산업공학과
†† 정 회 원: Computer Science Dept., U. of Virginia
††† 정 회 원: Systems Engr. Dept., U. of Virginia
논문접수: 1996년 7월 4일, 심사완료: 1996년 12월 10일

or whether it meets design specifications. In this research, performance of the system to be analyzed is Boolean-based information retrieval systems, especially the module of Boolean query reformulation.

Information search is an iterative process (in other words, a reformulation process). A query is usually reformulated by a user or a system using relevance feedback gained from the previously retrieved information. Researchers have found this technique to be very effective in improving retrieval effectiveness. More on relevance feedback, please refer to [8]. The changes of retrieval performance of a query during reformulations can be a major interest to a system analyst.

A new performance measure, called changes in retrieved area (CIRA) is developed and implemented to compare the performance changes of three query reformulation algorithms, i.e., the query tree [9, 10], the DNF method [16], and the Dillon's method [3, 4]. In addition, these algorithms use initial or feedback documents as an input, and their retrieval performance can be varied widely depending on different input sets. To measure the changes in performance (in other words, the sensitivity of an algorithm's performance to different input sets), a new sensitivity measure is also developed in this study.

In the next section, current performance measures in information retrieval are reviewed and methodologies for measuring effectiveness are described. There has been little research done in analyzing reformulation performance of Boolean query reformulation algorithms. Section 3 discusses new measures for measuring the changes in retrieved sets during reformulations and the sensitivity of an algorithm to different input sets. The last section reports the experimental results of the CIRA and sensitivity of the three algorithms on three test sets: the CACM, CISI, and Medlars.

2. Current Measures of Performance

There are two main characteristics in information retrieval (IR) system performance: efficiency and effectiveness [15]. Efficiency is concerned with operational aspects of the system performance such as response time, user efforts, information coverage, etc. On the other hand, effectiveness is based on the quality of the output; that is how well the system can retrieve the relevant information and reject the nonrelevant information.

Two most widely-used measures for effectiveness are recall and precision. Recall is defined as the ratio of the number of relevant documents retrieved to the total number of relevant documents in the collection. Precision is defined as the ratio of the number of relevant documents retrieved to the total number of documents retrieved from the collection. Although recall and precision are the most popular measures, they have been often criticized due to the difficulties of getting precise values. In fact, it is very hard to estimate the correct number of relevant documents in a database, especially if the size of the collection is very large (sometimes, a variation of recall called relative recall, is used which is based on a collected set of relevant documents from several searches on the same topic). Besides, these measures are based on user's relevance judgments and are rather subjective which brings controversies.

One of the difficulties using recall and precision is in comparing performance of several systems. Let r and p denote recall and precision, respectively. Suppose that $(r1, p1)$ and $(r2, p2)$ are the pairs of recall and precision from System 1 and System 2, respectively. The problem arises when $r1 > r2$ but $p1 < p2$ or vice versa. It is hard to tell which system performs better in overall. But the recall and precision are still the most widely-used measures in information retrieval. It is known from experiments [11] that there exists an inverse relationship between recall and precision. This means as recall increases, precision decreases and vice versa. To increase recall, it is necessary to bring more documents to the retrieved set, but at the same time the new nonrelevant documents retrieved is likely to decrease precision.

Since two measures with inverse relationship are hard to compare and evaluate the effectiveness of systems, single measures of performance have been proposed. The E-measure [18] is a composite measure which is a combination of recall and precision in a single expression. That is,!

$$E = 1 - \frac{1}{\alpha \cdot \frac{1}{p} + (1-\alpha)\frac{1}{r}}$$

where α is a weighting factor which can be adjusted according to the relative importance of recall and precision. The basic idea of E-measure is as follows. Suppose regions A and B represent the relevant and the retrieved sets, respectively. Let $(A \Delta B)$ denote the area which the two sets do not match, i. e., $(A \cup B\text{-}A \cap B)$. It is equivalent to $(A\text{-}B) \cup (B\text{-}A)$. The set $(A\text{-}B)$ represents the relevant documents not retrieved while $(B\text{-}A)$ represents the nonrelevant documents retrieved. We want to minimize these unwanted areas. The proportion of these areas $(A \Delta B)$ to the total areas defines the E-measure, that is, $E = |A \Delta B| / (|A| + |B|)$ where $0 \leq E \leq 1$. Therefore, the lower the E-measure, the better it is. This definition is equivalent to the one defined in terms of recall and precision with $\alpha = 0.5$. For details on E-measure, see [18]

Another single measure is $\sqrt{r \cdot p}$. Let D be a database and q be a query. Then, a vector G is defined as $q(D) = G$ where $G = [g1, g2, g3, ..., gn]$. It represents the status of documents being retrieved by the query q in the database D. That is, if the element $g_i = 1$, then the i^{th} document is retrieved; otherwise, it is not retrieved. Let R be also a vector representing the relevance judgment in D where $R = [r1, r2, r3, .., rn]$. That is, if the element $r_i = 1$, then the i^{th} document is relevant; otherwise, it is not relevant. The quality of the query q can be measured by how close these two vectors G and R are. The cosine of the angle between the two vector can be used to measure the proximity. Frants and et al. [6, 7] have shown that $\cos(\theta) = \sqrt{r \cdot p}$.

There are two distinct types of retrieved sets in information retrieval: ranked and unranked. The vector space system whose retrieval is based on the similarity between a query and documents (i.e., partial matching) is capable of ranking the retrieved documents. The number of documents retrieved can be controlled by the similarity measure (e.g., retrieve documents with its similarity greater than 0.5) or by the user (e.g., the user wants the top 50 of the retrieved set). On the other hand, documents retrieved by a Boolean query (i.e., exact matching) are assumed to have equal importance, and have no ranks.

Two of the most common ways of evaluating retrieval effectiveness are getting averages of single measures (e.g., E-measure) and constructing the recall-precision graph. The recall-precision graph [15, 18] is a widely-used technique for analyzing ranked outputs. It is a set of recall and precision scores plotted on the graph on fixed recall values (e.g., 0.1, 0.2,..., 1). For details on how to compute pairs of recall and precision from the ranked set, see [15]. Advantages of the recall-precision graph are: it is easy to compare the performance of different systems from a graph, and reveals the changes of the recall and precision as the size of the retrieved set increases. One thing about the recall-precision graph is when it is applied to an unranked output, it is sensitive to the ordering of the retrieved set. If major interests are in the overall performance rather than in the ranks of the retrieved outputs, computing averages of a single measure might be a better choice.

3. Measures of Reformulation Effectiveness

During query reformulations, a system analyst might be interested in behavior of an algorithm. Suppose that there are no improvements in performance after two reformulations. A nice feature for the algorithm to have would be a capability of detecting a rate of changes in retrieved sets during reformulations.

The measure called CIRA, which measures the rate of changes in retrieved sets is proposed in the next section. It is also an analyst's interests how to know algorithms react to different initial inputs. The sensitivity measure is described in Section 3.2.

3.1 Changes in Retrieved Area

One of interests an evaluator may have is the changes in the retrieved areas during reformulations. Let q_i(D) denote a set of documents retrieved by query q_i in D. Suppose that there are queries q_1 and q_2 obtained from reformulation 1 and 2, respectively. The changes in retrieved area (CIRA) from q_1 to q_2 can be measured as follows. Let $q_1(D)\Delta q_2(D)$ denote the areas which do not match (the same idea as in the E-measure). The proportion of mismatched area to the total area is defined as

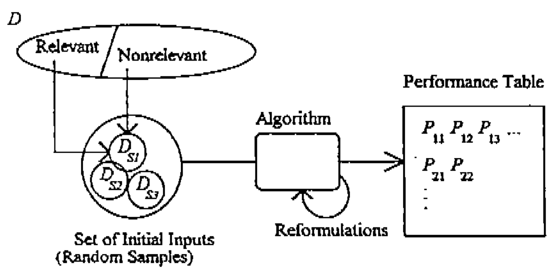
$$\mathrm{CIRA}(q_1,q_2)=\frac{|q_1(D)\Delta q_2(D)|}{|q_1(D)\cup q_2(D)|}$$

Sometimes, a performance analyst might want to observe the effects of the changes in retrieval regions, i.e., the newly retrieved (i.e., $|q_2(D)-q_1(D)|$) and the lost regions by the new query (i.e., $|q_1(D)-q_2(D)|$). These types of change cannot be measured directly from pairs of recall and precision. For instance, two totally different retrieval areas can produce the same recall and precision. It can be a valuable measure to check the behavior of query reformulation algorithms. Unlike other measures which are based on recall and precision, CIRA can be used as a stopping criteria during reformulations. The system can inform the user if there are no changes (or only small changes) in retrieval area. For example, the system can advise the user to stop reformulations when there are no changes in CIRA after a reformulation.

3.2 Sensitivity Analysis of Inputs

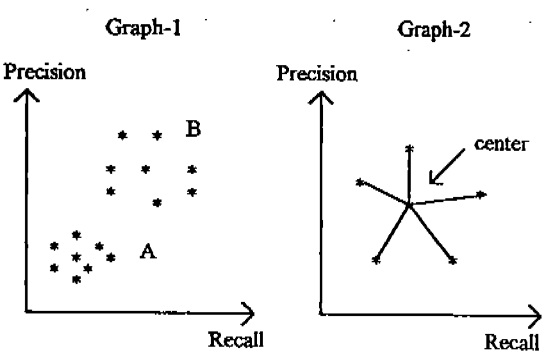
Query reformulation algorithms construct queries from a set of initial input or feedback documents. If the documents provided are poor examples of the user's information need, queries would also be poor reflections of the user information request. Suppose that q_1 is formulated from a poor initial set and q_2 is formulated from a subset of documents retrieved by q_1. The query q_2 will probably not perform well either since the query q_1 is not likely to bring new relevant documents to the set which is used by q_2. It is like a chain reaction that the following queries (e.g., q_3, q_4,...) will be negatively influenced. In consequence, there would be little performance improvement during reformulations.

It is of much interest for an analyst to investigate how the algorithms react to different input sets at the first query formulation (denote it as reformulation 0) and during a course of reformulations. The reactions at reformulation 0 would show a sensitivity of an algorithm to different input sets. That is, high variability in performance implies high sensitivity. For the following reformulations (i.e., 1, 2,...), the convergence behavior of an algorithm can be observed, that is, whether or not the algorithm settles down to a single performance level as reformulations progress. In this section, a measure for the algorithm sensitivity to different input sets is proposed.



(Fig. 1) The Process of Sensitivity Analysis of an Algorithm

In a real situation, initial input documents to the algorithms can be obtained either from the user or from the documents retrieved by the user's initial query (when the initial input is in a query form). For evaluation purposes, they can be randomly selected from the database since our intention is to investigate reactions of the algorithms to different initial input sets. Fig. 1 depicts the process of sensitivity analysis of an algorithm.) Once the numbers of relevant and nonrelevant documents for the initial input are determined (e.g., 3 relevant and 2 nonrelevant), each sample is randomly taken from the database D, and collected in an input set, $D_I=\{D_{S1}, D_{S2}, D_{S3},\ldots\}$. The algorithm uses the first sample D_{S1} to generate a query. Then, the formulated query retrieves the documents from the database D and its retrieval effectiveness (e.g., recall and precision) is computed. The result is kept in P_{11} of the performance table. The algorithm reformulates the query and stores the performance measure of the reformulated query in P_{12} and so on. Suppose that $|P_I|=n$ and the number of reformations is m. Then the size of performance table becomes n by m. The element P_{ij} corresponds to the performance measure of the i^{th} input sample for the j^{th} reformulation. The elements in the performance table can be pairs of recall and precision or any other single measures.



(Fig. 2) Densities for Sets of Recall and Precision

Suppose that each element in the performance table consists of a pair of recall and precision. Let $P(\mathrm{A}, D_I)$ denote a performance table (size of n by m) generated by algorithm A from a set of initial input samples D_I. Let $P_k(A, D_I)$ denote the k^{th} column of $P(\mathrm{A}, D_I)$, i.e., performance measures from the k^{th} reformulation. If each element of $P_k(A, D_I)$ is plotted on a recall-precision graph, it may look like one of regions in Fig. 2. Suppose that regions A and B in Graph-1 are plotted from the measures generated by algorithms A_1 and A_2, respectively. The graph shows that there are smaller differences among the pairs of recall and precision in region A than region B. This implies that algorithm A_1 is less sensitive to the initial inputs than algorithm A_2.

The sensitivity of $P_k(A, D_I)$ is defined as follows. Let $P_k(A, D_{Si})$ be a performance measure of algorithm A obtained from the i^{th} initial input sample at the k^{th} reformulation. Let $\overline{P_k}(A, D_I)$ be an average of performance measures of the k^{th} reformulation by algorithm A over the initial input set D_I. That is,

$$\overline{P_k}(A, D_I)=\frac{1}{n}\sum_{i=1}^{n} P_k(A, D_{Si})$$

Then the sensitivity of $P_k(A, D_I)$ is defined as

$$S_k(A, D_I)=\frac{1}{n}\sum_{i=1}^{n} Z[\overline{P_k}(A, D_I), P_k(A, D_{Si})]$$

where $Z(P, P')$ measures the distance between P and P'. From a pair of recall and precision as a performance measure,

$$Z[(r1, p1), (r2, p2)]=\sqrt{(r2-r1)^2+(p2-p1)^2}$$

The function $S_k(A, D_I)$ measures an average distance between the center of the region formed by a set of performance measures and each member in the region (Graph-2 in Fig. 2). The range of the function is $0\leq S_k(A, D_I)\leq\sqrt{2}/2$. The best case is when $Z(P, P')=0$, i.e., every pair of recall and precision has the same value. The worst case is when pairs are the furthest such as (0, 0) and (1, 1) where the average distance from the center is $\sqrt{2}/2$. When there is more than one queries to the algorithm, an average of $S_k(A, D_I)$ can be computed over a set of queries. let Q be a set of queries. Then

$$\overline{S_k}(A, D_I)=\frac{1}{|Q|}\sum_{q\in Q} S_k(A, D_I)$$

The sensitivity of an algorithm with a single measure based on recall and precision (e.g., E-measure) can be less accurate than with a pair of recall and precision. For example, suppose that two pairs of recall and precision, (0.4, 0.6) and (0.6, 0.4) are computed from different initial input samples. E-measures of these pairs are same if the weighting factor alpha = 0.5. But there are 0.2 difference in each recall and precision. In this research, pairs of recall and precision are used for sensitivity analysis of algorithms. As in the case of CIRA in the previous section, it just reveals the performance sensitivities to the initial inputs and does not necessarily imply that a low sensitivity is better. For example, whatever the initial condition might be, the sensitivity can be low if performance is low in general. Therefore, the sensitivity of the retrieval performance to an initial input and the CIRA should be analyzed along with retrieval performance measures such as E-measure, or recall and precision.

One of the observations that an analyst might be interested in is how different initial inputs have effects on the later reformulation performances of an algorithm. In such a case, the proposed sensitivity measure can be utilized. A desirable behavior would be to settle down (in other words, converge) to a single point as the reformulations proceed whatever the initial inputs may be.

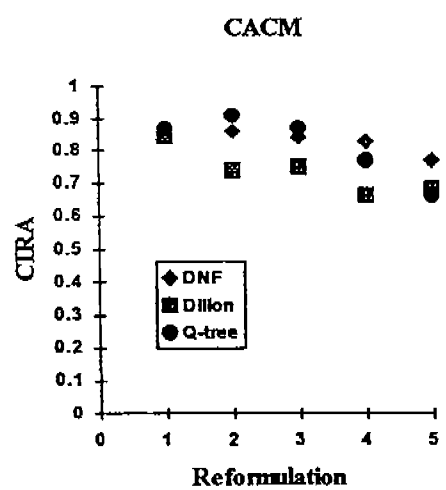
4. Experimental Results

To measure and analyze the sensitivity to different input documents and CIRA of the three query reformulation algorithms, experiments were performed in three test sets (CACM, CISI, and Medlars). Statistics of the test sets are listed in Table 1. These test sets consist of a set of documents, a set of queries and their relevance judgments. The test sets were obtained from the Virginia Disc series of CD-ROMs [5]. For the initial input documents, 3 relevant and 2 nonrelevant documents were used. In these experiments, the queries with a small number of relevant documents which was set to be less than 6 were discarded since the initial input contains 3 relevant ones. The number of feedback documents used during each reformulation was set to be 10.

〈Table 1〉 Test Set Statistics

Description	CACM (Computer Sci.)	CISI (Information Sci.)	Medlars (Medicine)
1. Number of documents	3204	1460	1033
2. Number of index terms (after the stemming)	6166	5593	7276
3. Number of queries	64	111	30
4. Number of queries with more than 5 relevant documents	41	73	30

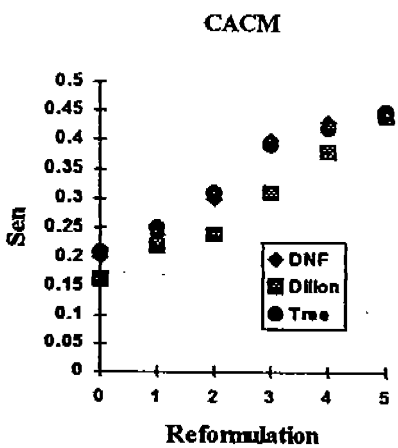
The changes in retrieved area (CIRA) of the algorithms on the test set CACM are illustrated in Fig. 3 (Q-tree stands for query tree). Reformulation i on the x-axis corresponds to the changes in retrieved area from reformulation i-1 to i. Overall, each algorithm's CIRA drops slightly as a reformulation continues. This implies that there are a smaller number of new documents retrieved at each reformulation. The Dillon's method achieved the low CIRAs compared



(Fig. 3) CIRA vs. Reformulation in CACM

with the DNF and query tree methods at most reformulations. This result means that as reformulations continue, the Dillon's method does not retrieve new documents as much as the query tree and the DNF method do. The desirable shape of the line in CIRA graph is the lower CIRAs as reformulations continue. That is a query converges to the set of relevant documents to a user. The query tree and the method have the most desirable shape among the three algorithms.

For the sensitivity to different inputs, each algorithm's sensitivity gets higher as reformulations continues. It can be conjectured that in CACM the initial inputs to the algorithms have effects on performance of the later query reformulations.

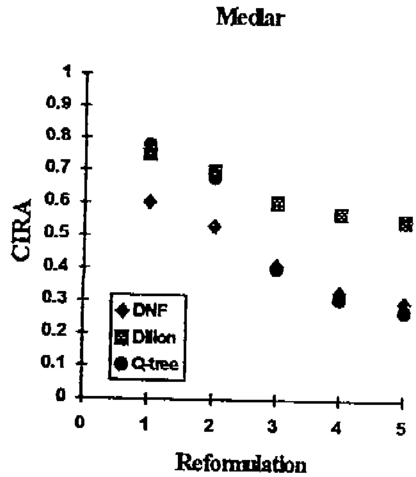


(Fig. 4) Sensitivity vs. Reformulation in CACM

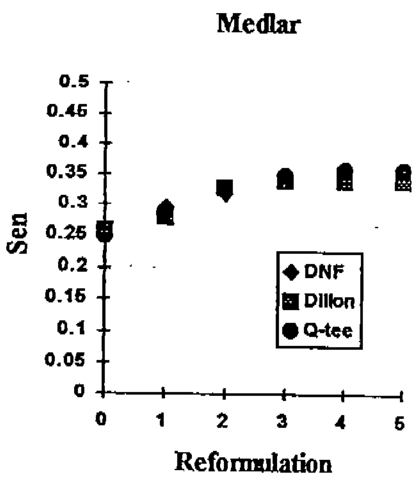
Unlike in the CACM test set, the DNF method and the query tree in Medlars (Fig. 5) have sharply decreasing CIRAs during reformulations; that is, the retrieval area converges steadily as a reformulation continues. The Dillon's method showed a similar behavior as in CACM, but has a slower convergence rate than the others.

The sensitivities of the algorithms in Medlars are less sensitive than those of the CACM (see Fig. 6). It may be from the high performances at reformulation 0. (High performance at an early stage tends to keep the same retrieved set, which means low sensitivity.)

The measures of sensitivity of all the algorithms have very similar shapes, and there is very little differences among them. The measures of sensitivity get lower and become constant as reformulations continue (e.g., reformulation 3, 4 and 5) when compared with the results from CACM

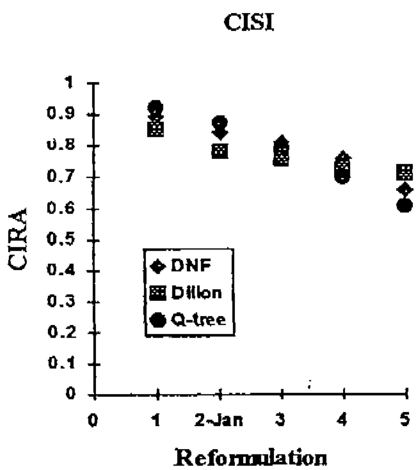


(Fig. 5) CIRA vs. Reformulation in Medlars

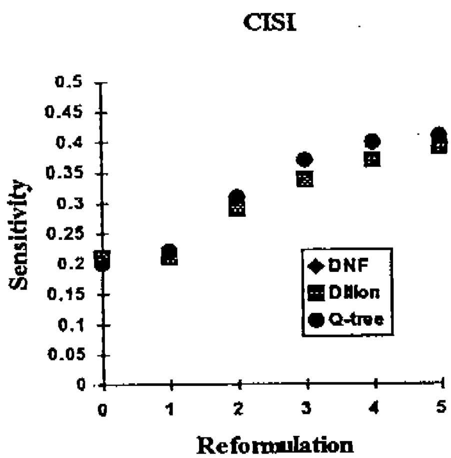


(Fig. 6) Sensitivity vs. Reformulation in Medlars

In the CISI test set (as shown in Fig. 7), three of the algorithms behave very similarly. Examining the graph closely, the query tree achieved the highest decrease ratio among the three algorithms, and the Dillon's method the lowest. For the sensitivity analysis, they also look very alike, but the query tree is most sensitive to the input sets even though the differences with other algorithms are very small.



(Fig. 7) CIRA vs. Reformulation in CISI



(Fig. 8) Sensitivity vs. Reformulation in CISI

⟨Table 2⟩ The E-measures (alpha = 0.5) and standard deviations in the test set CACM

Reformulation	0	1	2	3	4	5
Query tree	0.84/0.17	0.83/0.13	0.79/0.20	0.71/0.19	0.62/0.21	0.59/0.19
DNF	0.77/0.12	0.81/0.13	0.76/0.13	0.79/0.15	0.76/0.12	0.78/0.13
Dillon	0.90/0.14	0.87/0.13	0.88/0.10	0.88/0.08	0.90/0.05	0.90/0.07

As mentioned in the previous section, when analyzing the sensitivity measure, we need to examine the retrieval performance, e.g., recall and precision, of each algorithm at each reformulation as well. The E-measures of the three algorithms from the test set CACM are shown in Table 2 (The outcomes of the other test sets are very similar. For details, please refer to [9, 10]). The Dillon's method shows the highest E-measure among the algorithms which means the lowest performance. In Fig. 4, 6, and 8, the sensitivity measures of the Dillon's method are lowest even though their differences are not large. It can be conjectured that the low sensitivity of the Dillon's method is due to its low retrieval performance.

5. Conclusions

In this study, a new criterion of evaluating query reformulation algorithm's performance called the changes in retrieved area (CIRA) is developed and analyzed using the three test sets. This measure reveals how a query reformulation algorithm changes its retrieval area in the database during reformulations. With recall and precision alone, it is impossible to get such performance information. Another performance measure developed in this research is the sensitivity of a query reformulation algorithm to different initial inputs. Since a high sensitivity of an algorithm means that the retrieval performance heavily depends of an initial input, a low sensitivity should be one of the factors a good algorithm should have. A care should be taken when a low sensitivity observed because it can be due to the low retrieval performance.

Overall, the query tree scored the lowest CIRA between the 4^{th} and 5^{th} reformulation and showed the highest decrease in convergence ratio in the three test sets. This implies that the query tree converges to an output set faster than any other algorithms. It is also found that the three algorithms are somewhat sensitive to the input sets as shown in Fig. 4, 6, and 8.

References

[1] Breiman, L. et al, 'Classification And Regression Tree', Wadsworth Inc., 1984.

[2] Buckland, M. and Gey, F., "The relationship between Recall and Precision," Journal of American Society for Information Science, vol. 45, no. 1., pp. 12-19, 1994.

[3] Dillon, M. and Desper, J., "The Use of Automatic Relevance Feedback in Boolean Retrieval Systems," J. of Documentation, vol. 36, no. 3, pp. 197-208, 1980.

[4] Dillon, M., Ulmshneider, J. and Desper, J., "A Prevalence Formula for Automatic Relevance Feedback in Boolean Systems," Information Processing & Management, vol. 19, no. 1, pp. 27-36, 1983.

[5] Fox, E. A. and Winett, S. G., "Using Vector and Extended Boolean Matching in an Expert System for Selecting Foster Homes," Journal of American Society for Information Science, vol. 41, no. 1, pp. 10-26, 1990.

[6] Frants, V. I., and Shapiro, J., "Control and Feedback in a Documentary Information Retrieval Systems," J. of American Society for Information Science, vol. 42, no. 9, pp. 623-634, 1991.

[7] Frants, V. I., and Shapiro, J., "Algorithm for Automatic Construction of Query Formulations in Boolean Form," J. of American Society for Information Science, vol. 42, no. 1, pp. 16-26, 1991.

[8] Harman D., "Relevance Feedback Revisited," the 15th Annual International SIGIR Conference, Denmark, pp. 1-10, 1992.

[9] Kim, N., 'A Classification Approach to the Automatic Reformulation of the Boolean Queries in Information Retrieval,' Ph. D. Dissertation, University of Virginia, 1993.

[10] Kim, N., Brown, D. E., and French, J. C., "Boolean Query Reformulation with the Query Tree Classifier," the 4th ASIS SIG/IR, Columbus, Ohio, 1993.

[11] Salton, G., "Another Look at Automatic Text-Retrieval Systems," Comm. of the ACM, vol. 29, no. 7, pp. 648-656, 1986.

[12] Salton, G., 'Automatic Text Processing: the transformation, analysis, retrieval of computer, Addison-Wesley Co., 1988.

[13] Salton, G., Buckley, C. and Fox, E. A., "Automatic Query Formulations in Information Retrieval," J. of Amer. Soc. Info. Sci., vol. 34, no. 4, pp. 262-280, 1983.

[14] Salton, G., Fox, E. A. and Voorhee E., "Advanced Feedback Methods in Information Retrieval," J. of Amer. Soc. Info. Sci., vol. 36, no. 3, pp. 200-210, 1985.

[15] Salton, G. and McGill, M. J., 'Introduction to Modern Information Retrieval', McGraw-Hill, Inc., 1983.

[16] Salton, G., Voorhees, E. and Fox, E. A., "A Comparison of Two Methods for Boolean Query Relevancy Feedback," Information Processing & Management, vol. 20, no. 5/6, pp. 637-651, 1984.

[17] Tague-Sutcliffe, J. M., "Some Perspectives on the Evaluation of Information Retrieval Systems," Journal of American Society for Information Science, vol. 47, no. 1, pp. 1-3, 1996.

[18] Van Rijsbergen, 'Information Retrieval', 2nd Ed., Butterworths, London., 1979.

김 남 호

1985년 University of Pittsburgh 전산학과(학사)
1986년 Rensselaer Polytechnic Institute 대학원 수학과(이학석사)
1989년 University of Virginia 대학원 시스템공학과 (공학석사)
1993년 University of Virginia 대학원 시스템공학과 (공학박사)
1993년~1994년 한국전산원 표준본부 선임연구원
1995년~현재 선문대학교 산업공학과 교수
관심분야: 시스템최적화, 정보검색, 기계학습

James C. French

1973 University of Virginia, B.A., Mathematics.
1979 University of Virginia, M.S., Computer Science.
1982 University of Virginia. Ph.D., Computer Science.
1984~1987 Vice-President, Software Development, Information Research Corporation.
1987~1990 Senior Scientist, Institute for Parallel Computation University of Virginia.
1990~present Research Assistant Professor, Dept. of Computer Science University of Virginia.
Research: Information retrieval, distributed systems, scientific databases.

Donald E. Brown

1973 United States Military Academy, West Point, B.S.
1979 University of California, Berkeley, M.Eng. and M.S. in Operations Research
1985 University of Michigan, Ann Arbor, Ph.D. in Operations Research
1973~1982 Officer, U.S. Army, Honorable Discharge at the rank of Captain
1982~1984 Systems Analyst, Vector Research, Inc., Ann Arbor, Mi.
1985~1990 Assistant Professor of Systems Engineering, University of Virginia
1987~present Associate Director, Institute for Parallel Computation, University of Virginia
1990~1996 Associate Professor of Systems Engineering, University of Virginia
1996~present Professor and Chairman of Systems Engineering, University of Virginia.
Research: Data fusion, Forecasting and Prediction Theory, Data Analysis and Inductive Modeling.