The Effect of Adding Relevance Information in a Relevance Feedback Environment

Chris Buckley[*], Gerard Salton, James Allan

Abstract

The effects of adding information from relevant documents are examined in the TREC routing environment. A modified Rocchio relevance feedback approach is used, with a varying number of relevant documents retrieved by an initial SMART search, and a varying number of terms from those relevant documents used to expand the initial query. Recall-precision evaluation reveals that as the amount of expansion of the query due to adding terms from relevant documents increases, so does the effectiveness. It is observed for this particular experiment that there seems to be a linear relationship between the log of the number of terms added and the recall-precision effectiveness. There also seems to be a linear relationship between the log of the number of known relevant documents and the recall-precision effectiveness.

1 Introduction

Relevance feedback is a commonly accepted method of improving interactive retrieval effectiveness.[1, 2] An initial search is made by the system with a user-supplied query, returning a small number of documents to the user. The user indicates which of the returned documents are useful (relevant). The system then automatically reformulates the original query based upon those user relevance judgements. The new "feedback query" is then compared to the collection of documents, returning an improved set of documents to the user. The process can continue to iterate until the user's information need is satisfied.

The same relevance feedback techniques can be used in the routing environment, in which a user may have a long-lived information need and is interested in any new documents that match the need. In this case the user's query can be constantly updated by the system as it receives relevance information about new documents seen by the user. Over the life of the query, thousands of documents could conceivably be returned to the user for relevance judgements.

The question of how much relevance information can be effectively used by a relevance feedback process has only cursorily been studied. Until very recently, there has not been a good test bed for examination of this question. The standard small information retrieval test collections do not have enough relevant documents to be able to effectively alter the amount of learning information while keeping a constant environment for testing the effects of the added information. The TREC[3, 4] routing environment solves that problem. It is possible to gather information from a learning set consisting of hundreds of thousands of documents, with an average of hundreds of relevant documents per query. The learned information can be used to alter the original query, which can then be evaluated on a separate set of hundreds of thousands of documents, with again an average of hundreds of relevant documents per query.

We study the effects of varying two sources of information from relevant documents. The first is the number of known relevant documents used for feedback. The second is the number of terms occurring in the relevant documents that are to be added to the original query.

One of the criticisms of the TREC experimental setup is that the learning set information is gathered from the combined runs of many, very different, retrieval systems. While this is good for the completeness of the relevance judgements, it is not the environment that will be encountered in general. In practice, the relevance judgements will necessarily be gathered from one retrieval system, and thus any biases induced by

[*] Department of Computer Science, Cornell University, Ithaca, NY 14853-7501. This study was supported in part by the National Science Foundation under grant IRI 93-00124.

Set Name	Type	Size (bytes)	Size(records)
D_{12}	documents	2.3 Gbytes	742709
D_3	documents	1.3 Gbytes	336314
Q_2	queries	52380	50
JQ_2D_{12}	judgements	–	16394
JQ_2D_3	judgements	–	10489

Table 1: TREC Collection Sets Used

that system will affect the set of relevant documents to be used for feedback. To avoid that problem, the sets of relevant documents that we use for the learning information are the top documents retrieved by the SMART system itself. However, further tests here using the global set of learning information contributed by many systems show that this is not an important concern.

2 Methodology

The TREC environment offers several sets of documents, queries, and relevance judgements. The ones used for this experiment are described in Table 1.

An initial vector space run is done with queries Q_2, running on the "learning document set" D_{12}. The queries are weighted with the "ltc" weighting scheme.[5, 6] Each query term, q_k, receives a weight that increases as the number of times q_k occurs in the query increases (term frequency contribution), but decreases as the proportion of documents the term occurs in increases (the idf or inverse document frequency contribution.) The entire query vector is normalized using the cosine length of the vector.

$$q_{ik} = \frac{(\log(f_{ik}) + 1.0) * \log(N_D/n_k)}{\sqrt{\sum_{j=1}^{t}[(\log(f_{ij}) + 1.0) * \log(N_D/n_j)]^2}}$$

where f_{ik} is the number of times term k occurs in Q_i, N_D is the collection size, and n_k the number of documents with term k assigned.

The documents are weighted with the "lnc" weighting scheme which is the same as the "ltc" query scheme, except no idf factor is used. This "lnc" document weight, "ltc" query weight combination proved very effective in TREC 1 and TREC 2.

N documents are retrieved for each query, where N varies from 5 to 5000. The relevance judgements JQ_2D_{12} are used to determine which of the retrieved documents are relevant. All other documents, both retrieved and non-retrieved, are considered non-relevant. The query is then expanded by adding the X terms that occur in the largest number of relevant documents. Ties are broken by preferring those terms with the highest average weight in the relevant documents. In the experiments below, X varies from 0 to 4000.

The expanded query is weighted using the modified Rocchio formula used by Cornell in TREC 2[2, 6, 7]:

$$\begin{aligned} Q_i^{\text{new}} &= \alpha\, Q_i^{\text{old}} \\ &+ \beta \frac{1}{|\text{rel docs}|} \sum_{\text{rel docs}} wt_i \\ &- \gamma \frac{1}{|\text{nonrel docs}|} \sum_{\text{nonrel docs}} wt_i \end{aligned}$$

Query terms with overall negative weights are dropped.

This reweighting differs from the original Rocchio formula only in that the original formula used

$$-\gamma \frac{1}{|\text{known nonrel docs}|} \sum_{\text{known nonrel docs}} wt_i$$

That is, the only documents considered non-relevant were those that had been seen by the user and judged non-relevant. The modified formula makes the assumption that all unseen documents are non-relevant, an assumption which is undoubtedly false, but perhaps better than the assumption that the known non-relevant documents are representative of non-relevant documents in general.

One crucial property of the modified Rocchio approach is that the weight of a term in a query generally decreases as the amount of information known about the term decreases. A term's weight is averaged over all relevant and non-relevant documents, not just the documents in which it occurs. Thus a term that occurs comparatively infrequently in both the collection and the relevant documents will have a low value for both the contribution from the relevant documents, and the contribution from the non-relevant documents. The net contribution of the term is the (weighted) difference between those two values, which is guaranteed to be low. This property is important since random chance dictates much of the occurrence characteristics of low-frequency terms.

This can be compared with the probabilistic approaches[8, 9] in which the final contribution of a term is a ratio between a value due to relevant documents and a value due to non-relevant documents (or more precisely, the log of a ratio). The values are lower for the more infrequent terms, but since a ratio is being computed, if the relevant value and non-relevant value decrease by the same factor, the magnitude of the final contribution will not decrease. For example, with binary-weighted documents and the modified Rocchio approach, a term which occurs in 1/10 of the relevant documents and 1/100 of the non-relevant will have a weight of .09, while a term occurring in 1/100 of the relevant documents and 1/1000 of the non-relevant documents will have a weight of .009. The same two terms in the classical probabilistic model will have nearly equal weights (log (11.0) vs log (10.1)). Thus it is easy for the random effects of the infrequent terms to overwhelm the positive effects of the good terms.

The constants α, β, γ in the Rocchio formula are set to 8,16,4 respectively: the same values that were determined reasonable in TREC 2. This combination emphasizes occurrences in the relevant documents as opposed to non-relevant documents.

One subtle issue in using the Rocchio formula is that weights from documents and queries are being added together, which implies they must be based on the same factors. Thus instead of using "lnc" weights for the documents in the Rocchio formula (i.e., no idf factor for the document weights), the same "ltc" weighting scheme used by the queries is used for the documents.

Once an expanded, weighted query is constructed, it is run against D_3 (the "test document set"). D_3 again is weighted with "lnc" document weights. The relevance judgements JQ_2D_3 are used to evaluate the retrieval output. The evaluation values presented here are average recall-precision: the average precision after each relevant document is retrieved.[4] Graphically, this is the area under the well-known recall-precision graphs. A complete *relevant-rank* evaluation is done (i.e., the ranks of all relevant documents are used, not just a retrieved subset.)

Since the evaluated test runs are made on a different document collection than the learning runs, there is no need for rank freezing or residual collection evaluation.[10, 2] This is one reason why this experiment could not have been done in the past – the small collections did not have enough relevance information to be able to split collections and to both form and fairly evaluate feedback queries.

3 Results.

Table 2 shows the average recall-precision for the base case run of the original unmodified query being run against D_3 without any relevant information.

Run	Average recall-precision
"ltc" original query	0.2927

Table 2: Base-case Performance

Table 3 then gives the average recall-precision for a selected subset of values of N (number of retrieved documents to gather relevance information from) and X (number of terms from relevant documents to be

N (Number Initially Retrieved)	X (Target Number of Expansion Terms)						
	0	5	10	50	200	500	4000
5	0.3083	0.3231	0.3291	0.3386	0.3458	0.3475	0.3475
15	0.3140	0.3285	0.3364	0.3495	0.3588	0.3618	0.3623
30	0.3161	0.3331	0.3380	0.3551	0.3657	0.3692	0.3698
50	0.3213	0.3373	0.3436	0.3635	0.3750	0.3805	0.3819
100	0.3203	0.3377	0.3455	0.3660	0.3789	0.3833	0.3855
200	0.3217	0.3419	0.3478	0.3690	0.3823	0.3878	0.3893
500	0.3237	0.3408	0.3488	0.3668	0.3827	0.3892	0.3917
1000	0.3274	0.3445	0.3553	0.3733	0.3904	0.3968	0.3993
2500	0.3311	0.3488	0.3577	0.3794	0.3993	0.4045	0.4064
5000	0.3316	0.3478	0.3576	0.3812	0.4018	0.4052	0.4062
all-1/4	0.3313	0.3468	0.3560	0.3805	0.3989	0.4046	0.4033
all	0.3314	0.3459	0.3566	0.3829	0.4026	0.4068	0.4062

Table 3: Average Recall-Precision For Selected Runs

Number Docs Seen	Avg Number Rel Docs	Number Expanded At Target 500	Number Expanded At Target 4000
5	3.1	257.9	287.8
15	8.8	376.8	573.3
30	16.9	435.1	925.0
50	27.3	473.9	1252.4
100	47.8	489.3	1766.8
200	80.2	497.9	2424.1
500	137.7	497.8	3041.7
1000	188.8	498.4	3385.9
2500	248.9	498.8	3658.0
5000	280.1	499.0	3736.5
all	327.9	499.5	3770.2

Table 4: Statistics For Selected Runs of Table 3

added to query). Space prohibits showing the entire 10x13 table, but all values (except X=0) are plotted in Graph 1 with a logscale for the value of X.

Two additional runs were made using relevance information gathered from many retrieval systems instead of just the SMART initial run. The run **all** uses all relevant documents found in judgement set JQ_2D_{12}. The run **all-1/4** randomly takes approximately 1/4 of all the relevant documents of JQ_2D_{12}.

Table 4 gives the average number of relevant documents (N') retrieved for each value of N, the number of initially retrieved documents. As is to be expected, the relationship is not linear: doubling the number of retrieved documents does not double the number of relevant retrieved documents. Also given in Table 4 is the average number of terms (X') by which each query is expanded for the 500 and 4000 targets. Obviously 3.1 relevant documents will not often contain 4000 unique terms that could possibly be added to the query! There were also a very few terms dropped because they would have had a negative weight.

Let X' be defined (as above) as the average number of terms added for a run, and N' be the average number of relevant documents retrieved for a run. Graph 1 plots X' on a log scale versus the resultant recall-precision for each of the possible values of N'. Graph 2 plots N' on a log scale versus the resultant recall-precision for each of the possible values of X.

A	B	C	R-Squared	SquaredError	F(2,77)	prob (F)
0.008118	0.01015	0.292022	0.9572	0.0041	861.1116	0.0000

Table 5: Regression Analysis for RP = A * log(N') + B * log(X') + C

N	A	B	R-Squared	SquaredError	F(1,6)	prob (F)
5	0.006139	0.315116	0.9838	0.0012	363.6741	0.0000
15	0.007559	0.319546	0.9757	0.0019	240.4660	0.0000
30	0.008408	0.320761	0.9823	0.0018	332.9829	0.0000
50	0.009793	0.322903	0.9854	0.0020	404.7223	0.0000
100	0.01018	0.323824	0.9792	0.0025	283.1230	0.0000
200	0.01037	0.326296	0.9865	0.0020	439.8625	0.0000
500	0.01064	0.325219	0.9907	0.0017	638.5675	0.0000
1000	0.01145	0.328259	0.9924	0.0017	780.6660	0.0000
2500	0.01272	0.329226	0.9897	0.0022	576.3812	0.0000
5000	0.01324	0.328358	0.9806	0.0031	303.5382	0.0000

Table 6: Regression Analysis for RP(N) = A * log(X') + B

4 Statistical Analysis.

The output of the Graph 1 consists of near-straight lines that appear to follow a linear relationship of the form

$$\text{Recall-precision} = A * log(N') + B * log(X') + C$$

A linear regression analysis is presented in Table 5 using log(N') and log(X') up to 500 expansion terms which shows that a reasonably close linear relationship exists.

Separate sets of linear regressions are shown in Tables 6 and 7 for each line of Graph 1 and Graph 2 (i.e., for each possible value of N in Graph 1 and each possible value for X $\leq$ 500 in Graph 2)

5 Discussion of Results.

As expected from our earlier TREC 2 experiments, the more terms added from relevant documents, the better the recall-precision, up to a steady-state value. Massive query expansion works well for Rocchio feedback even if the relevance information is minimal. Expanding by 500 terms after looking at 5 documents

X	A	B	R-Squared	SquaredError	F(1,8)	prob (F)
0	0.004735	0.303078	0.9447	0.0019	136.5866	0.0000
5	0.005335	0.317537	0.9674	0.0016	237.7434	0.0000
10	0.006168	0.321918	0.9714	0.0017	271.6812	0.0000
20	0.007403	0.326019	0.9735	0.0020	294.1245	0.0000
30	0.007753	0.328406	0.9592	0.0026	188.2034	0.0000
50	0.008518	0.331013	0.9511	0.0031	155.7192	0.0000
100	0.009772	0.333380	0.9567	0.0034	176.7990	0.0000
200	0.01134	0.333833	0.9643	0.0035	216.0912	0.0000
500	0.01207	0.335513	0.9756	0.0031	319.9049	0.0000

Table 7: Regression Analysis for RP(X) = A * log(N') + B

improves recall-precision by 19% as compared to the base case. This is even more impressive if one considers that six queries retrieve no relevant documents in the top 5 retrieved documents, and thus are unaltered by the feedback process.

The improvement is correspondingly stronger if much more relevance information is available. Adding 500 terms after looking at 2500 documents yields a 38% improvement. While nobody will judge 2500 documents in a relevance feedback environment, it is much more reasonable in a routing environment. 2500 documents is an average of 10 documents per working day for a year which for some long-lived routing queries may be desirable.

For all levels of relevant document information, the effectiveness increase starts to level off after 500 terms have been added. Except for minor random fluctuations after the effectiveness has reached its maximum value, there are no decreases in effectiveness due to adding more terms. Table 4 shows that only a small number of relevant documents are necessary in order to substantially expand the query. Thus the desired amount of query expansion does not depend strongly on the amount of relevant information.

The straight lines of Graph 1 and the statistical analysis of Tables 5 and 6 suggest a direct relationship between the log of the number of terms added to a Rocchio feedback query, and the effectiveness of the query. This means that doubling the number of new terms will add a constant to the recall-precision measure. For this particular set of queries and documents, with relevance information based on 100 retrieved documents, doubling the number of added terms (within the range 5 to 500) will yield about a 2.4% improvement in recall-precision over the base case. (0.01018 * log (2) / .2927).

There seems to be a similar relationship between the log of the number of relevant documents used and recall-precision effectiveness. The R-squared correlation measures of Table 7 are smaller, indicating the straight line fits are not as good as in Table 6, but they still seem to be valid. For this particular collection in the ranges of values considered, doubling the number of known relevant documents has a slightly smaller effect than doubling the number of expansion terms.

As Table 6 (or Table 7) shows, the two sources of added relevance information (X' = number of added terms, and N' = number of relevant documents used) are not truly independent. As N' grows, the slope of the lines gradually increase, as can be seen from the increasing values of column A. Thus the improvement due to doubling X' gradually increases from 1.5% at N = 5 (0.006139 * log(2) / .2927) to 3.1% at N = 5000 (0.01324 * log(2) / .2927). The relationship between N' and X' should be statistically explored further (e.g., adding a log(N')*log(X') factor into the linear regression equation improves the fit) when the current effects are better understood.

The two runs using relevance information from many systems (**all** and **all-1/4**) show that that multi-system relevant documents perform better than an equal number of single-system documents, but after a certain percentage of total relevant documents has been retrieved, the source of those documents is immaterial. This is not surprising: every system has its own weaknesses and strengths and thus some types of relevant documents will be hard to retrieve initially. If those documents can be found to be relevant using other systems, then the feedback query on the original system will be able to retrieve other documents of that type.

However, the improvement is not substantial. The **all-1/4** runs do about 3-4% better than the **N=200** run in Table 3, which uses an equal number of relevant documents. The reason for performing this comparison is that one of the criticisms of the TREC experimental setup has been that the learning set relevance judgements were derived from a pool of many different retrieval systems. This setup is unrealistic since in practice any relevance judgements available would come from one system. While 3-4% is noticeable, it is not enough improvement to warrant changes to the TREC experimental design. The extremely small differences between the **all-1/4** runs and **all** runs is also welcome news for TREC participants, suggesting that any lack of completeness of the TREC relevance judgements should have little effect upon the formation of routing queries (as long as there are not weaknesses shared by all participating systems).

6 Efficiency of massive query expansion.

The efficiency of massive query expansion depends on the environment. For interactive relevance feedback, the time needed for massive expansion in a normal vector-space model is prohibitive. Well over half the inverted file is being used for some of the runs above, with search time being slowed down by a factor of 40.

The searcher or system administrator will have to judge the effectiveness/efficiency tradeoff suitable for them.

For routing though, reasonably massive expansion is feasible. As an extremely rough estimate, SMART handles about 100,000 extra query terms in about the same time as it takes to index a single document. Thus expanding 200 routing profiles by 500 terms apiece will at most double the amount of time needed to handle an incoming document. SMART indexes documents at 300 Mbytes/hour, so routing (in this example) could proceed at around 100-150 Mbytes/hour, which should be sufficient for many purposes.

7 Future work.

The conclusions reached here are based upon one collection with one set of queries using one relevance feedback approach. The general validity of the relationships proposed here remains to be proved. The most important immediate step is replication of the experiment in other environments. Earlier work with the small information retrieval test collections also reached the conclusion that as the number of added terms increases, so does the effectiveness, but nothing more detailed could be concluded. This work needs to be repeated in a large collection environment.

One very important question, of course, is why recall-precision effectiveness appears to be linear with the log of the number of added terms. A theoretical explanation would seem to be somewhat difficult given the large number of system-dependent decisions that take place between the initial simple concept of "adding more terms" and the final recall-precision output. A number of those decisions are crucial to even obtaining the result that adding more terms is beneficial. What exactly is being captured by this mathematical relationship (if it exists in general)?

8 Conclusions.

A large number of relevance feedback/routing runs are presented in the TREC routing environment. A modified Rocchio relevance feedback approach is used, with a varying amount of relevance information obtained from an initial SMART search.

For this particular experiment recall-precision effectiveness seems to vary linearly with the log of the number of terms added to the query from the relevant documents. Recall-precision also seems to vary linearly with the log of the number of known relevant documents. The implications of this mathematical relationship are not currently well-developed.

The overall effectiveness increase due to relevance feedback is quite impressive, ranging from 19% to 38% depending on the number of known relevant documents.

Finally, a comparison of feedback based upon an equal number of relevant documents obtained from either a single system (SMART), or a general pool of many systems, reveals a modest improvement of about 4% for using relevant documents from many sources.

References

[1] Gerard Salton. *Automatic Text Processing—the Transformation, Analysis and Retrieval of Information by Computer.* Addison-Wesley Publishing Co., Reading, MA, 1989.

[2] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[3] D. K. Harman. Overview of the first Text REtrieval Conference (TREC-1). In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 1–20. NIST Special Publication 500-207, March 1993.

[4] D. K. Harman. Overview of the second Text REtrieval Conference (TREC-2). In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*. NIST Special Publication (in preparation).

[5] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[6] Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART : TREC 2. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*. NIST Special Publication (in preparation).

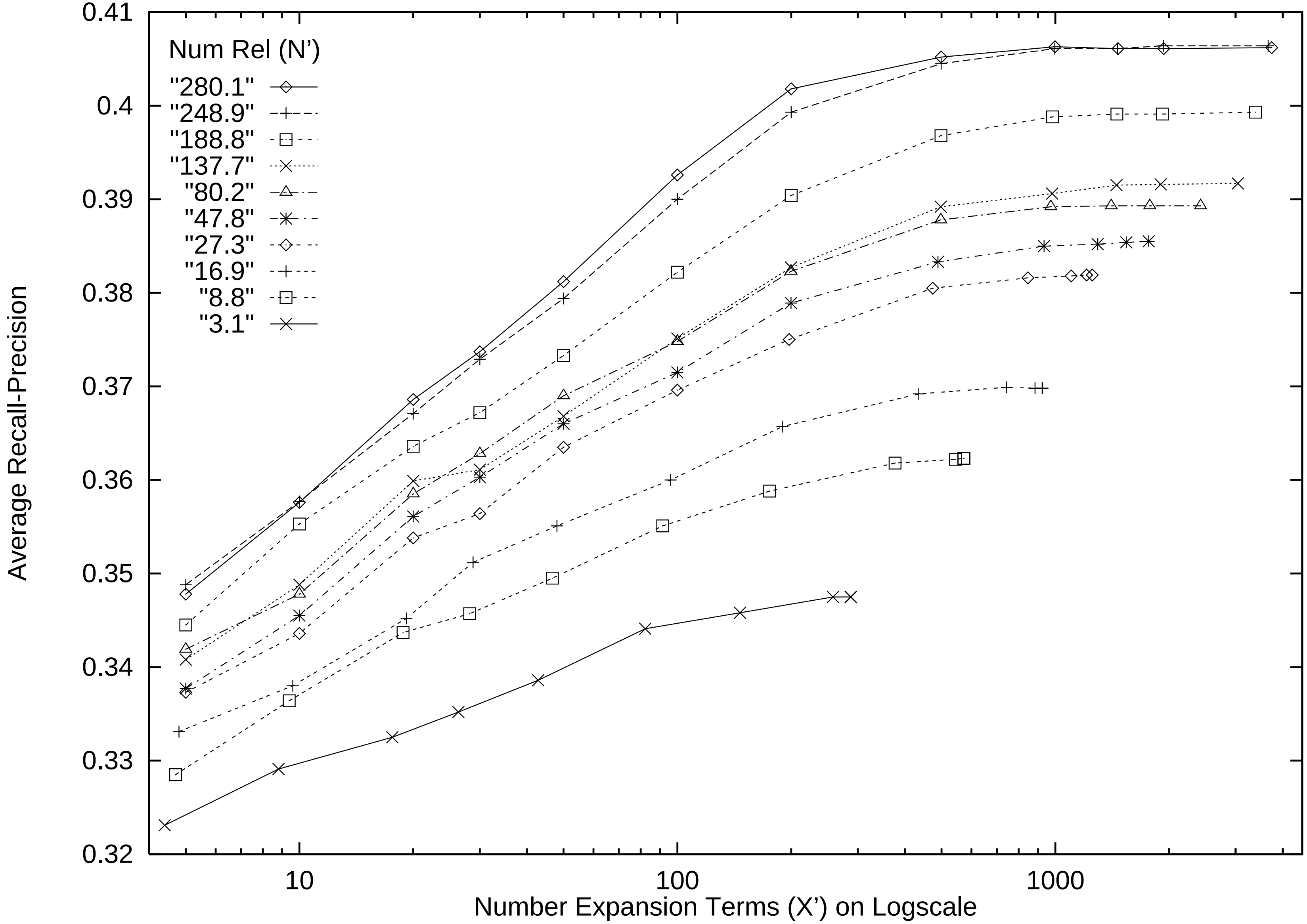
[7] J.J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Englewood Cliffs, NJ, 1971.

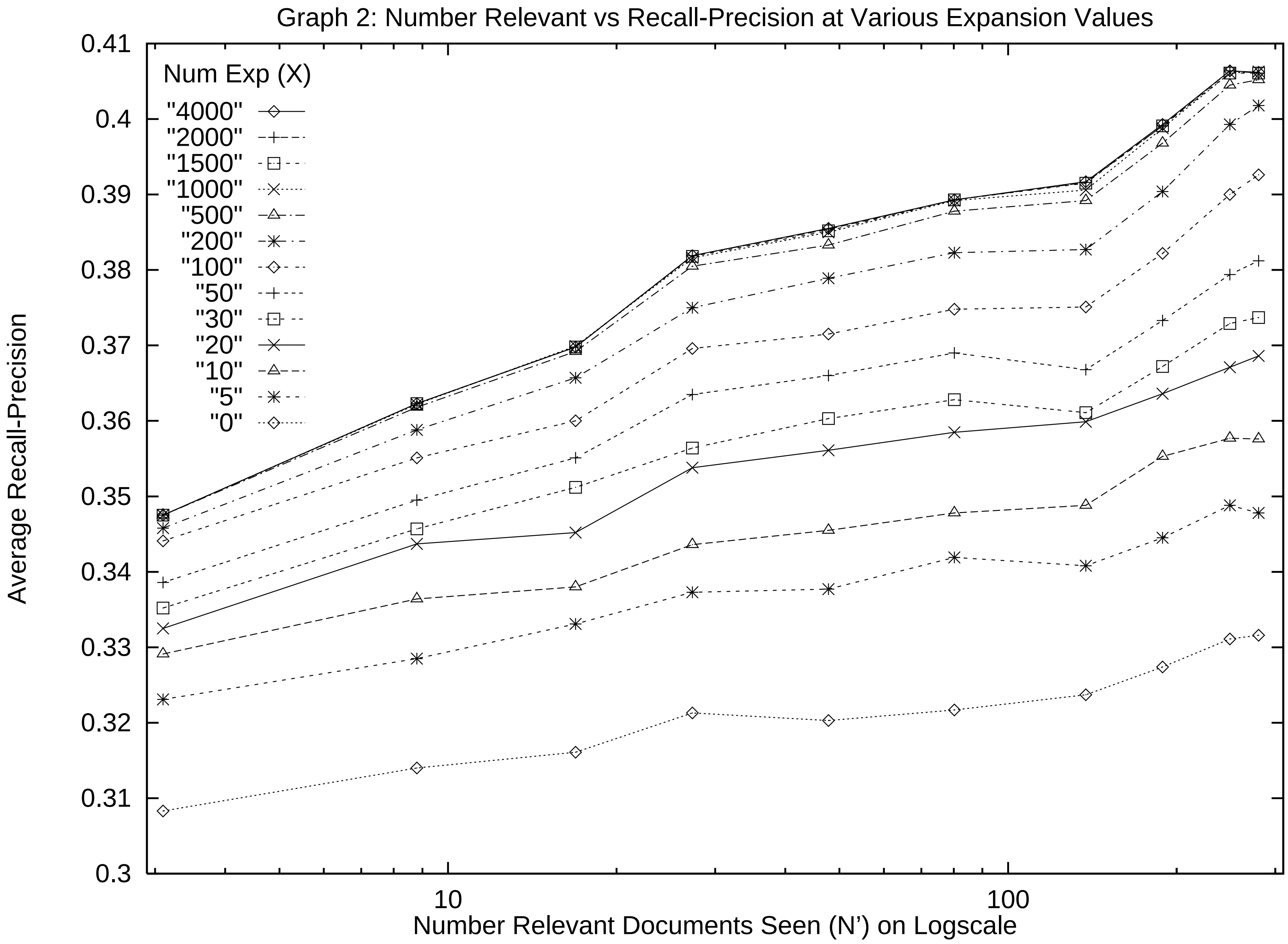
[8] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, May-June 1976.

[9] S.E. Robertson, C.J. van Rijsbergen, and M.F. Porter. Probabilistic models of indexing and searching. In R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen, and P.W. Williams, editors, *Information Retrieval Research*. Butterworths, London, 1981.

[10] Y.K. Chang, C. Cirillo, and J. Razon. Evaluation of feedback retrieval using modified freezing, residual collection, and test and control groups. In Gerard Salton, editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*, chapter 17, pages 355–370. Prentice Hall, Englewood Cliffs, NJ, 1971.

Graph 1: Expansion vs Recall-Precision at Various Number Relevant




Graph 2: Number Relevant vs Recall-Precision at Various Expansion Values
Average Recall-Precision
Number Relevant Documents Seen (N') on Logscale
0.41
0.4
0.39
0.38
0.37
0.36
0.35
0.34
0.33
0.32
0.31
0.3
10
100
Num Exp (X)
"4000"
"2000"
"1500"
"1000"
"500"
"200"
"100"
"50"
"30"
"20"
"10"
"5"
"0"