Query Modification and Expansion in a Network with Adaptive Architecture

K.L. Kwok

Computer Science Department, Queens College, CUNY,
Flushing, NY 11367

Abstract

This paper shows how a newtwork view of probabilistic information indexing and retrieval with components may implement query expansion and modification (based on user relevance feedback) by growing new edges and adapting weights between queries and terms of relevant documents. Experimental results with two collections and partial feedback confirm that the process can lead to much improved performance. Learning from irrelvant documents however was not effective.

1. Introduction

In automatic Information Retrieval (IR) one tries to retrieve relevant natural language documents from a large collection when given a statement or query of what one wants, also in free text. Many models of IR exist, for a review consult [SaMc83,Salt89]. We have been using the probabilistic indexing and retrieval (PIR) model [MaKu60, BoSw75,RoSp76,YuSa76,vanR77] because it satisfies the Probability Ranking Principle [Robe77] and also incorporates the concept of feedback naturally. Recently we have extended PIR with the concept of document components [KwKu88,Kwok90b] which leads to improvements on the original method. Furthermore, we have also been able to re-formulate our theory using an artificial neural network (ANN) approach, with learning rules that show how the net may evolve from its initial state to one with user relevance feedback information [Kwok89,90a]. This paper describes two enhancements: 1) to include adaptation of our network architecture, a process equivalent to automatic query expansion in IR; and 2) to allow for learning from irrelevant documents. Some results based on partial relevance feedback are reported in Section 4. Section 2 reviews our ANN approach to PIR, Section 3 describes the extensions, and Section 5 contains the conclusion.

2. Background

2.1 Probabilistic Indexing and Retrieval with Components

We view documents and queries as non-monolithic, but constituted of many independent components (such as phrases, single terms, etc.), and work in a component universe. This can overcome certain shortcomings of the original PIR theory [Kwok90b] such as: 1) providing probabilistic weights to each term even before relevance feedback, based on a principle of document self-recovery; 2) making use of within-item term frequencies effectively; and 3) accounting for query-focused and document-focused indexing and retrieval strategies cooperatively in one symmetric formula. Retrieval can be viewed as query-focused: i.e. rank documents with respect to q_a, with $W_{i/q}$ being the retrieval status value (RSV) for document d_i and corresponds to probabilistic retrieval [RoSp76, RoMa82]; or as document-focused: i.e. rank queries with respect to d_i, with $W_{i/d}$ being the RSV for d_i and corresponds to probabilistic indexing [MaKu60,RoMa82]. With single content terms as approximations to document components, these RSV's become:

$$W_{i/q} = \Sigma_k \, d_{ik}/L_i * w_{ak},$$

$$W_{i/d} = \Sigma_k \, q_{ak}/L_a * w_{ik}; \qquad (1,2)$$

and a symmetric sum formula that accounts cooperatively for both:

$$W_i = W_{i/q} + W_{i/d} \qquad (3)$$

Our notation is to use subscripts i,j for documents, a,b for queries and k,l for terms. d_{ik} (q_{ak}) is the frequency of term k in d_i (q_a) and $L_i = \Sigma_k d_{ik}$ ($L_a = \Sigma_k q_{ak}$) is the length of the respective item, interpreted as the number of components. w_{ik} (w_{ak}) are weights for each independent term k that occurs in d_i or q_a, and given as:

$$w_{ik} = \ln[r_{ik}/(1-r_{ik})*(1-s_{ik})/s_{ik}],$$

$$w_{ak} = \ln[r_{ak}/(1-r_{ak})*(1-s_{ak})/s_{ak}]. \qquad (4,5)$$

r_{ik} (r_{ak}) and s_{ik} (s_{ak}) are the probabilities that given relevance to item d_i or q_a (i.e. $+R_i$ or $+R_a$) or non-relevance ($-R_i$ or $-R_a$), that term k will be present in a component:

$$r_{ik} = P(t_k \text{ present}|+R_i), \quad r_{ak} = P(t_k \text{ present}|+R_a)$$

$$s_{ik} = P(t_k \text{ present}|-R_i), \quad s_{ak} = P(t_k \text{ present}|-R_a) \qquad (6)$$

The r's may be estimated from a sample of relevant items and their components. However, even before feedback we can employ the fact that the set of terms must be self-relevant to each item, and approximate them as: $r_{ik} = d_{ik}/L_i$, $r_{ak} = q_{ak}/L_a$. To estimate the probabilities associated with the irrelevants we use the statistics of the whole collection, tacitly assuming that all are irrelevant to the item and which is not unreasonable at this stage, thus: $s_{ak} = s_{ik} = F_k/N_w$. $F_k=\Sigma_i d_{ik}$ is the collection frequency of term k for all documents, and N_w is the total number of terms used in the collection. For example, if d_i has length L_i=40 terms (including repeats) and the stem 'retriev' occurs d_{ik}=3 times, r_{ik} is estimated as 3/40 because all terms are self-relevant to d_i. If 'retriev' also occurs totally F_k=200 times in the collection which has N_w=70000 terms, then s_{ik}=2/700. Better estimates result if more relevant items are available. Experiments have confirmed [Kwok90b] that this component theory can provide effectiveness better than Croft's extension to probabilistic retrieval [Crof83], and comparable to those available from the vector IR model [SaBu88], as well as the probabilistic model with default beliefs for CACM [Turt90].

2.2 An Artificial Neural Network Approach to PIR

Fig.1-3 shows a 3-layer ANN architecture for simulating the PIR formulation of Section 2.1. The D layer of nodes denote documents d_i ($1<=i<=n_d$, no. of documents) and the Q layer denote queries q_a ($1<=a<=n_q$, no. of queries.) These are bi-directionally connected with unsymmetrical weights to the middle T layer of content terms t_k (1<=k<=m, no. of unique terms). Existence of connections is defined by the presence or absence of a term in an item. Nodes of the D and Q layer serve both as input and output. For example, activity may originate from q_a, spread to the T and D layer synchronously, with appropriate signal functions (we used a linear function at the D and Q layer, and a ramp function with a threshold at the origin at the T layer). If connection weights are set correctly, activation deposited on a d_i node would correspond to the RSV of Eqn.2. By ordering all d_i nodes based on activations, we simulate document-focused ranking. If activity originates from d_i, spread to a q_a via the T layer, we obtain the query-focused RSV, Eqn.1. Adding the two results in our symmetric formula Eqn.3 for retrieval ranking. QTD or DTQ direction operations are independent and the net is **not** meant to be recurrent. The following paragraphs review the learning stages discussed previously [Kwok89, 90a]. Section 3 describes additional mechanisms to be considered in this investigation.

At the **initial** stage, connections w_{ki} (w_{ka}) from d_i (q_a) to a term t_k, Fig.1, are initialized as d_{ik}/L_i (q_{ak}/L_a) based on the manifestation of the terms within an item. Connections w_{ik} (w_{ak}) from a term t_k to d_i (q_a) are given exactly as in Eqn.4,5, with r_{ik} (r_{ak}) set to a heuristically assigned constant of 1/40, and $s_{ik} = s_{ak} = F_k/N_w$, as discussed in Section 2.1. With these values, operation of the net corresponds to the Inverse Collection Term Frequency (ICTF) strategy introduced in [Kwok89,90b], and has been shown to be much more effective than the Inverse Document Frequency (IDF) indexing [Spar72,CrHa79] that is popular for IR experiments.

After the initial stage, we can perform **self-learning**, Fig.2, since items must be self-relevant. Each q_a (d_i) in turn is clamped to activation of 1; it spreads to the T and then back to the Q layer (the D layer). Each term t_k associated with q_a will have activation $x_k = w_{ka} = q_{ak}/L_a$ (or $x_k = w_{ki} = d_{ik}/L_i$). Since the originating item q_a (d_i) is also the item under attention, it is the only node to receive a teaching signal T_a (T_i) of 1, and we postulate that the connections w_{ak} (from t_k to q_a) or w_{ik} get modified according to the following learning algorithm (equations for w_{ik} are not shown):

$$w_{ak}^{vQ+1} = w_{ak}^{vQ} + \Delta w_{ak}^{vQ}$$

$$\Delta w_{ak}^{vQ} = \Delta r_{ak}^{vQ}/[r_{ak}^{vQ} * (1-r_{ak}^{vQ})] \tag{7}$$

with

$$\Delta r_{ak}^{vQ} = \eta_Q * (x_k - r_{ak}^{vQ}) \tag{8}$$

Here $0 < \eta_Q < 1$ is a query-focused learning rate when given relevant items, and v_Q enumerates the iterations. Document-focused learning employs a different schedule (η_D,v_D). Eqn.7 shows that w_{ak} adapts because of a change in r_{ak}. r_{ak} is the probability of Eqn.6, and is the current activation x_k on term node k, since q_a induces the current relevant set and their activations. In Eqn.7 the factor s_{ak} within w_{ak} is regarded as a constant equal to F_k/N_w assigned in the initial stage. r_{ak}^0 has value 1/40, and its value during iteration is obtained as:

$$r_{ak}^{vQ} = \exp(w_{ak}^{vQ}-C_s)/[1+\exp(w_{ak}^{vQ}-C_s)], \tag{9}$$

if s_{ak} is stored so that

$$C_s = \ln [(1-s_{ak})/s_{ak}], \text{ a constant.}$$

Alternately [Kwok89,90a], we can use a separate reverse edge from q_a to t_k to store this dynamic value r_{ak}^{vQ}. As v_Q->

∞, r_{ak}^{vQ} asymptoptically approaches the value $w_{ka}=q_{ak}/L_a$, and therefore:

$$w_{ak} \rightarrow \ln [q_{ak}/(L_a-q_{ak}) * (N_w-F_k)/F_k], \quad (10)$$

The learning rule of Eqn.8 is similar to that used for competitive learning with decay [RuMc86]. In ANN, competitive learning is an unsupervised learning paradigm whereby a pool of output nodes compete to respond to a set of environmental feature vectors, and classify them into clusters. The winning node learns the expected feature vector of that cluster. In our case, it learns under supervision since the query (winning node) is known for the cluster of relevant components. The self-learn weighting formula Eqn.10 corresponds to that of indexing based on a principle of document self-recovery [Kwok86, 88], and has effectiveness comparable to that of the best non-Boolean term weighting methods available [Kwok90b].

User **relevance feedback learning** can be performed next if it is available. Referring to Fig.3 if a set of n_r documents $\{d_j\}$ are known relevant to the query q_a, we will have query-focused DTQ direction learning. Items in $\{d_j\}$ will have activations clamped to 1, and after one time step, the average activity deposited at term t_k will be $x_k = (\Sigma_j d_{jk}/L_j)/n_r$ and is the current estimate of the probability r_{ak}. Thus, r_{ak} and w_{ak} would learn as in Eqn.7,8, but the estimate is better since the sample of relevant components is bigger. The initial value of r_{ak}^0 for this stage is the self-learn value. Eqn.8 provides continuity from this self-learn value as well as allowing for a varying degree of influence from the self-relevant on the whole relevant set, controlled by the learning rate η_Q and the number of iterations v_Q. By re-sorting the feedback file, we can also provide document-focused QTD direction learning (i.e. set of queries $\{q_b\}$ known relevant to d_i), using a schedule η_D, v_D and leading to improved estimates for $W_{i/d}$. It is found that results are sensitive to these learning schedules.

3. Further Developments

The following subsections describe how the net architecture may adapt during relevance feedback, as well as how known irrelevant items may be employed. Both correspond to the well-known procedures of query expansion and modification studied by IR workers [IdSa71,Rocc71, Smva83], and has been shown to deliver good performance [SaBu90].

3.1 Adaptive Architecture for Query Expansion

Although queries and documents are viewed as items of similar category in our theory, in practice queries are usually short, terse statements of need, while documents generally are longer and more diverse in content. Our probabilistic component approach, being based on statistics of term usage, would be expected to perform better if queries are longer, providing more clues to what a user wants. One method is to expand queries automatically based on their feedback set of relevant items. Since documents are permanent records, we do not intend to expand document representations. Thus, during learning of q_a from relevant documents, a large number of terms may be activated. A K-winner-take-all network [MaEr89] may be assumed to be inherent among the nodes of the T layer that chooses relatively quickly the term nodes that contain the K largest activation levels. New connections may now grow between these highly activated nodes and the query q_a under consideration (Fig.3), with weights w_{ka} and w_{ak} (via r_{ak}) assigned values based on the activities x_k as follows:

$$w_{ka} = \alpha * x_k \qquad (11)$$

$$r_{ak} = \beta * \eta_Q * x_k \qquad (12)$$

Once r_{ak} is known in Eqn.12, w_{ak} is defined as in Eqn.5 using stored values of s_{ak}. These are one-step Hebbian learning [Hebb49] equations. Experimentally we find that α=1 and β=0.7 lead to good results. Different node expansion K values were tried, and it seems a small value of 15 to 30 can account for most of the effects.

3.2 Learning from Irrelevant Documents

In the previous sections feedback learning is limited to relevant items only. In general a set of the highest ranked irrelevant documents would also be known, and these may be employed to adapt the factors s_{ak} in the weights w_{ak}, with analogues of Eqn.7,8 for learning irrelevant items, thus:

$$w_{ak}^{uQ+1} = w_{ak}^{uQ} + \Delta w_{ak}^{uQ}$$

$$\Delta w_{ak}^{uQ} = -\Delta s_{ak}^{uQ}/[s_{ak}^{uQ} * (1-s_{ak}^{uQ})] \qquad (13)$$

with

$$\Delta s_{ak}^{uQ} = \varepsilon_Q * (x_k - s_{ak}^{uQ}) \qquad (14)$$

In practice, relevant and irrelevant learning will be independent and separate, and the schedule (ε_Q,u_Q) is expected to be quite different from (η_Q,v_Q) and has to be empirically determined. The expectation is that this might lead to decrease in $W_{i/q}$ values for, or rank demoting of, items similar to the irrelevant ones.

4. Results and Discussion

Two evaluated databases were used for our investigation, namely, CACM(52q,3204d) with 52 queries and 3204 computer science abstracts and/or titles, and CISI(76q, 1460d) of information science literature. For each query an initial run based on IDF strategy is made and the first 10 high-ranked documents are used for partial relevance feedback learning. These 10 are then removed, and recall and precision are evaluated using the residual collection, so that generalization capability from training can be ascertained. After identifying these 10 documents, some queries have no more relevant items left, while others do not have any relevant documents in the first 10. These queries are not used, so that CACM has 42 and CISI 60 queries remaining. Results are tabulated in Tables I (CACM) & II (CISI).

In each table, the first two columns present the IDF and SL (self-learn) precison values. The next two columns IDFr, SLr show the same methods using the residual collection. These do not involve relevance feedback, and both will serve as basis for comparison with feedback retrievals. Average precisions Av3 at 3 recall points (.25, .5, .75), and Av10 at 10 recall points are also given. Percentages of increase (decrease) for each method versus the basis are calculated using the Av10. The columns labelled PL'K' show results of partial learning using K expanded terms. Because some of the K highly activated terms may already exist in a query, usually less than K terms are added. The feedback learning procedure is done as follows: the relevant document set first deposits an average activity on each connected term node in the DTQ direction. The connection weights w_{ak} from t_k to q_a are trained using the schedule $(v_Q,\eta_Q) = (20,0.2)$ according to Eqn.7,8. The K highest activated term nodes are then identified, and edges are grown to and from q_a according to Eqn.11,12. QTD direction learning is then performed using the expanded queries and the schedule $(v_D,\eta_D) = (10,0.1)$. Because there are in general fewer queries relevant to a document (average 1.2) than vice versa (average 2.8), QTD learning schedule is 'softer' than DTQ, and is confirmed by experiments to give better results.

Results generally reflect what has been found in [SaBu90]. Partial relevance feedback without query expansion PL0 is effective (+92% better than IDFr for both collections). Further gains are obtained with query expansion of K terms, and most of the gains can be accounted for with K=30 (CACM +113%, CISI +103% versus IDFr). At K=60, CACM collection precision still rises, while CISI appears to decline. Query lengths are shorter in CACM (average about 13) compared to CISI (about 33), and gains with expansion are larger for short queries, as noted in [SaBu90]. Expansion based on term occurrence frequency ordering rather than activation does not perform as well. An example is given as PL15f, i.e. partial learning with expansion using 15 highest frequency terms. Our results are based on the symmetric formula W_i of Eqn.3. We find, as in [Kwok90b], that cooperative effects between $W_{i/q}$ and $W_{i/d}$ exist in all partial learning; an example is given for PL30. The fairly unsophisticated IDF is used as an initial strategy to pick the first 10 documents for feedback because we like to see how very few relevant documents (average less than 3) may improve precisions. Using IDFr as basis may make our gains look large, and another comparison is provided using SLr as the basis. All PL'K' methods reduce to SLr if feedback is not used, and the 10 IDF feedback documents are independent of them. It simulates the situation where a user poses a query and comes equiped with some known relevant items. Compared to using SLr, substantial improvements of 25% (PL0) to 39% (PL30) are recorded for CACM, and 13% (PL0) to 20% (PL30) for CISI. Comparison of PL30 versus PL0 shows that query expansion is useful: +11% for CACM and 6% for CISI. The last row also shows the number of edges added to the net at different expansion K values.

Our investigation of learning from irrelevants shows that it is much less useful. We looked at Eqns. 13,14 for DTQ learning by queries and found that we have to use very 'soft' learning schedules such as $(v_Q,\eta_Q)=(1, .05$ to $0.2)$ or performance decreases, which means results do not change much from those not using irrelevants. One can have many options to test, such as: modification only or modification with expansion for both relevant or irrelevant item learning, varying the number of irrelevant items from the feedback set, as well as the learning schedule. In a number of cases we investigated, average performance is not much changed, but precision values improve slightly at the low recall end. Examples from each collection for the case of using all irrelevants from the feedback set, employing the schedule (1,0.05), and with K=30 expansion are as follows: CACM (.519, .464, .395, .336, .283, .183, .144, .101, .070, .054; Av3, Av10 = .279, .257) and CISI (.563, .431, .336, .292, .236, .197, .147, .103, .067, .045; Av3, Av10 = .245, .242). These are to be compared with the PL30 Sym columns. Using fewer irrelevants, expanding queries with terms from irrelevants, or using much larger learning rates usually decrease performance. It seems that the global strategy of assuming the whole collection as irrelevant for estimating the probabilities s_{ik}, s_{ak} works quite well. Moreover, every document that is not about the query topic(s) is irrelevant, which means that the irrelevant set do not necessarily have a focal topic(s). The content terms used in the few irrelevant documents therefore can be quite random, and expansion or learning their probabilities is not useful. It appears that for our model, the most cost-effective gains lie in using the relevants only.

Direct comparison with [SaBu90] results is not attempted because the number of queries and feedback documents, and the initial strategy used are different. Qualitatively they are very similar, and show that feedback using our component theory of PIR is effective.

5. Conclusion

We have extended our artificial neural network for the component theory of probabilistic information retrieval to include growing of new edges between existing terms and query nodes. This implements the well-known procedure of query expansion and modification in an ANN. Learning rules for the new edges are given. Results with net architecture adaptation generally reflect those of [SaBu90]: that query edge growing is effective. About 30 terms seems to account for most of the gains. Expansion based on term occurrence frequency is not as good as that based on node activation, which has the interpretation as an average probability that a term occurs in the relevant set. Learning from irrelevant documents however does not lead to better results.

Acknowledgments

This work is partially supported by a PSC grant (PSC-CUNY 668292). Simulation experiments have been performed on the Cornell National Supercomputer Facility, a resource of the Center for Theory and Simulation in Science and Engineering at Cornell University, which is funded in part by NSF, New York State, and the IBM Corp. and members of the Coporate Research Institute.

References

[BoSw75] Bookstein, A. & Swanson, D.R. (1975). A decision theoretic foundation for indexing. J. of ASIS. 26:45-50.

[CrHa79] Croft, W.B & Harper D.J (1979). Using probabilistic models of document retrieval without relevance information. J. of Doc. 35: 285-295.

[Crof83] Croft, W. B (1983). Experiments with representation in a document retrieval system. Info. Tech.: R&D. 2 (1983), 1-21.

[Hebb49] Hebb, D.O (1949). The organization of behavior. N.Y.: Wiley.

[IdSa71] Ide, E & Salton, G (1971). Interactive search strategies and dynamic file organization in information retrieval. In: The Smart system - experiments in automatic document processing. Salton, G. ed. N.J.: Prentice Hall. pp. 373-393.

[Kwok90a] Kwok, K.L (1990). Application of neural network to information retrieval. In: Proc. Intl. Joint Conf. on Neural Networks. (Washington D.C. Jan. 1990) Caudill, M., ed. N.J.: Erlbaum. pp. II 623-626.

[Kwok90b] Kwok, K.L (1990). Experiments with a component theory of probabilistic information retrieval based on single terms as document components. ACM

Trans. on Info. Sys. 8:363-386.

[Kwok89] Kwok, K.L (1989). A neural network for probabilistic information retrieval. In: Proc. ACM SIGIR 12th Ann. Intl. Conf. on R&D in IR. Belkin, N.J. and van Rijsbergen, C.J., ed. N.Y.: ACM. pp.21-30.

[KwKu88] Kwok, K.L. & Kuan, W (1988). Experiments with document components for indexing and retrieval Info. Mngmt. Proc., 24, 405-417.

[Kwok86] Kwok, K.L (1986). An interpretation of index term weighting schemes based on document components. In: Proc. of 1986 ACM Conf. on R&D in IR. Rabitti, F. ed. N.Y.: ACM. pp.275-283.

[MaEr89] Majani, E; Erlanson, R & Abu-Mostafa, Y (1989). On the K-winner-take- all network. In: Advances in Neural Information Processing Systems I. Touretzky, D.S. ed. CA: Kaufmann. pp. 634-642.

[MaKu60] Maron,M.E. & Kuhns,J. L (1960). On relevance, probabilistic indexing and information retrieval. J. ACM. 7, 216-244.

[Robe77] Robertson, S.E. (1977). The probability ranking principle in IR. J. Doc. 33 (1977), 294-304.

[RoMa82] Robertson, S. E.; Maron, M.E. & Cooper, W.S. (1982). Probability of relevance: a unification of two competing models for document retrieval." Info. Tech.:R&D 1:1-21.

[RoSp76] Robertson, S.E. & Sparck Jones, K (1976). Relevance weighting of search terms." J. of ASIS. 27: 129-146.

[Rocc71] Rocchio, J.J.Jr (1971). Relevance feedback in information retrieval. In: The Smart system - experiments in automatic document processing. Salton, G. ed. N.J.: Prentice Hall. pp.313-323.

[RuMc86] Rumelhart, D.E & McClleland, J.L (1986). Parallel and Distributed Processing. Vol.1. Cambridge, MA : MIT Press.

[SaBu90] Salton, G. & Buckley, C (1990). Improving retrieval performance by relevance feedback. J. of ASIS. 41:288-297.

[SaBu88] Salton, G. & Buckley, C (1988). Term weighting approaches in automatic text retrieval. Info. Proc. Mgmnt. 24, 513-523.

[SaMc83] Salton, G. & McGill, M (1983). Introduction to Modern Information Retrieval. N.Y.: McGraw Hill

[Salt89] Salton, G. (1989). Automatic Text Processing. N.Y.: Addison-Wesley.

[Smva83] Smeaton, A.F & van Rijsbergen, C.J (1983). The retrieval effects of query expansion on a feedback document retrieval system. Computer J. 26, 239-246.

[Spar72] Sparck Jones, K. (1972). A statistical interprestation of term specificity and its application in retrieval. J of Doc. 8:11-21.

[Turt90] Turtle, H.R (1990). Inference networks for document retrieval. COINS T.R. 90-92, Univ. of Massachusetts at Amherst.

[vanR77] van Rijsbergen, C.J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. J. of Doc. 33: 106-119.

[YuSa76] Yu, C. T. & Salton, G. Precision weighting - an effective automatic indexing method. J. of ACM. 23 (1976), 76-86.

TABLE I: CACM Precision Values

	(52q,3204d)		Residual Collection (42q,3194d)								
	No Feedback				Partial (10 Doc) Feedback Learning						
							PL30				
Recall	**IDF**	**SL**	**IDFr**	**SLr**	**PL0**	**PL15**	**QTD**	**DTQ**	**Sym**	**PL60**	**PL15f**
0.1	.473	.650	.255	.426	.487	.515	.491	.476	.516	.527	.504
0.2	.370	.518	.210	.337	.413	.429	.427	.419	.448	.459	.398
0.3	.299	.437	.175	.283	.354	.372	.346	.370	.397	.406	.356
0.4	.258	.360	.156	.232	.292	.324	.295	.319	.336	.357	.312
0.5	.210	.293	.139	.202	.260	.284	.242	.259	.287	.287	.267
0.6	.178	.253	.085	.139	.172	.184	.152	.197	.187	.185	.165
0.7	.110	.200	.068	.098	.130	.143	.122	.133	.145	.147	.122
0.8	.086	.172	.050	.063	.098	.104	.087	.102	.103	.103	.088
0.9	.066	.117	.033	.033	.056	.075	.067	.065	.073	.073	.053
1.0	.053	.105	.025	.025	.042	.054	.051	.045	.056	.056	.038
Av3	.213	.324	.128	.198	.248	.268	.243	.253	.275	.281	.246
Av10	.210	.310	.120	.184	.230	.248	.228	.238	.255	.260	.230
% chng (Av10):			0[1]	+53	+92	+107			+113	+117	+92
% chng (Av10):				0	+25	+35			+39	+41	+25
% chng (Av10):					0	+8			+11	+13	+0
# edges inc[2]:					0	556			1195	2339	626

Note 1: 0 denotes basis from which increases or % changes are calculated.
2: Increases in edges are calculated for 45 queries, 42 plus 3 that have all relevant documents within the 10 feedback documents.

TABLE II: CISI Precision Values

	(76q,1460d)		Residual Collection (60q,1450d)								
	No Feedback				Partial (10 Doc) Feedback Learning						
							PL30				
Recall	**IDF**	**SL**	**IDFr**	**SLr**	**PL0**	**PL15**	**QTD**	**DTQ**	**Sym**	**PL60**	**PL15f**
0.1	.301	.454	.269	.508	.531	.545	.541	.461	.564	.555	.534
0.2	.225	.353	.194	.369	.407	.418	.398	.354	.426	.424	.406
0.3	.163	.264	.151	.268	.315	.331	.325	.301	.334	.336	.311
0.4	.135	.219	.130	.229	.270	.280	.284	.255	.289	.283	.276
0.5	.115	.189	.116	.183	.229	.229	.220	.207	.237	.235	.227
0.6	.098	.150	.093	.148	.184	.191	.176	.155	.197	.195	.189
0.7	.081	.113	.079	.117	.142	.146	.139	.122	.149	.145	.144
0.8	.065	.085	.065	.090	.100	.102	.101	.089	.103	.101	.102
0.9	.054	.060	.054	.058	.062	.066	.068	.055	.067	.068	.064
1.0	.039	.042	.038	.040	.040	.045	.044	.039	.045	.045	.042
Av3	.130	.201	.119	.199	.236	.241	.231	.217	.243	.246	.237
Av10	.128	.193	.119	.201	.228	.235	.230	.204	.241	.239	.229
% chng (Av10):			0[1]	+67	+92	+97			+103	+101	+92
% chng (Av10):				0	+13	+17			+20	+19	+14
% chng (Av10):					0	+3			+6	+5	+0
# edges inc[2]:					0	695			1516	3135	745

Note 1: 0 denotes basis from which increases or % changes are calculated.
2: Increases in edges are calculated for 61 queries, 60 plus 1 that have all relevant documents within the 10 feedback documents.

-----> QTD Direction DTQ Direction <-----

Q-Layer T-Layer D-Layer

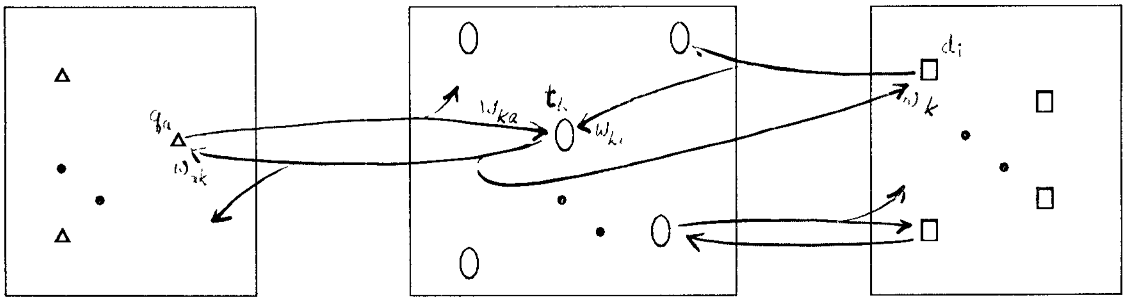

Fig.1: Three Layer Network for PIR
(not all connections are shown)

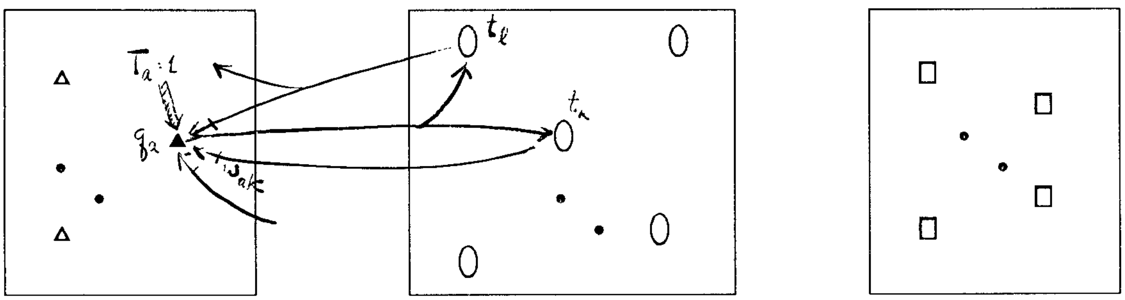

Fig.2: Self-Learning by q_a - relevant set (q_a only) clamped to activity 1; teaching signal T_a=1 (only node that learns); all connection weights w_{ak} of q_a adapt according to Eqn.7.

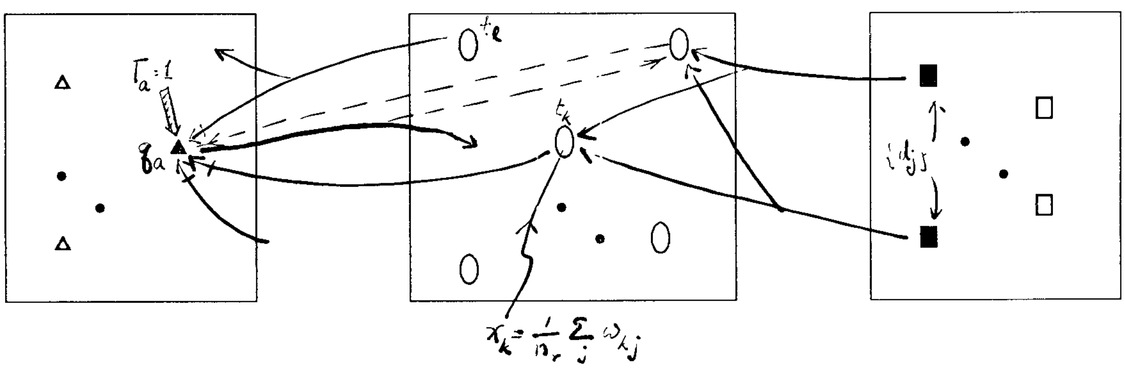

Fig.3: Relevance Feedback DTQ Learning by q_a - relevant set $\{d_j\}$ clamped to activity 1; T_a=1; w_{ak} learn according to Eqn.7; x_k is activity on term t_k at iteration ; dashed lines denote new grown edges.