Mercure at trec7

M. Boughanem , T. Dkaki, J. Mothe & C. Soulé-Dupuy

IRIT/SIG
Campus Univ. Toulouse III
118, Route de Narbonne
F-31062 Toulouse Cedex 4
Email : {bougha, dkaki, mothe, soule}@irit.fr

1 Introduction

The tests we performed for TREC-7 were focused on automatic ad hoc and filtering tasks. With regard to the automatic ad hoc task we assessed two query modification strategies. Both were based on blind relevance feedback processes. The first one carried on with the TREC6 tests: new parameter values of the relevance backpropagation formulas have been tuned. On the other hand, we proposed a new query modification strategy that uses a text mining approach. Three runs were sent. We sent two runs for the relevance backpropagation strategy: one used long topics (titles, descriptions and narratives) and the other one used titles and descriptions. We sent one run for the text mining strategy using long topics. With regard to the filtering task, we sent runs in batch filtering and routing using both relevance backpropagation and gradient neural backpropagation.

2 Mercure model

Mercure is an information retrieval system based on a connectionist approach and modeled by a three-layered network (as shown in the figure 1). The network is composed of a query layer (set of query terms), a term layer representing the indexing terms and a document layer. Mercure includes the implementation of a retrieval process based on spreading activation forward and backward through the weighted links. Queries and documents can be used either as inputs or outputs. The links between two layers are symmetric and their weights are based on tf.idf measure inspired from the OKAPI and SMART term weightings.

Let be :

w_{ij} : the weight of the link between the term neuron N_{t_i} and the document neuron N_{D_j},
tf_{ij} : the term frequency of t_i in the document D_j,
N : the number of documents in the collection,
T : the total number of indexing terms,
n_i : the number of documents containing the term t_i,
$doclen_j$: document length in words (without stop words),
avg_doclen : average document length.

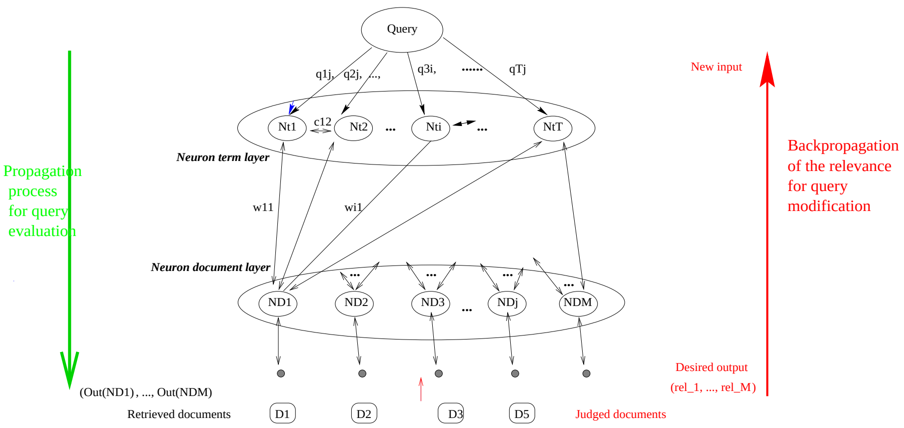

Figure 1: *The Mercure Model.*

- The query-term (at stage s) links are weighted as follows:

$$q^s_{ik} = \frac{(1+log(tf_{ik}))*(log(N/n_i)}{\sqrt{\sum_{j=1}^{T}(1+log(tf_{jk}))*(log(N/n_j)^2}} \text{ if } tf_{jk} \neq 0$$

$q^s_{ik} = 0$ otherwise

- The term-document link weights are expressed by:

$$w_{ij} = \frac{(1 + log(tf_{ij})) * (h_1 + h_2 * log(\frac{N}{n_i}))}{h_3 + h_4 * \frac{doclen_j}{avg\_doclen}}$$

The query evaluation is based on spreading activation. Each neural node computes an input and spreads an output signal:

1. The query k is the input of the network:

$$Input_k = 1$$

Then, each neuron from the term layer computes an input value from this initial query:

$$In(N_{t_i}) = Input_k * q^s_{ki}$$

and then an output value :

$$Out(N_{t_i}) = g(In(N_{t_i})$$

where g is the identity function.

2. These signals are propagated forwards through the network form the term layer to the document layer. Each neuron computes an input and an output value :

$$In(N_{D_j}) = \sum_{i=1}^{T} Out(N_{t_i}) * w_{ij}$$

then,

$$Out(N_{D_j}) = g(In(N_{D_j}))$$

The system output is :

$$Output_k(Out(N_{D_1}), Out(N_{D_2}), ..., Out(N_{D_N}))$$

These output values are then ranked to build the corresponding retrieved document list.

3 Ad hoc experiment and results

3.1 Ad hoc methodology

Our work in the TREC7 automatic ad hoc has been undertaken in two directions. The first one concerns the improvement of the automatic query modification based on document relevance. We tested new weighting formulas and we better tuned different parameters. The second proposition concerns a new investigation in query expansion based on a text mining approach. Both approaches have been performed using blind relevance feedback.

Query modification based on relevance backpropagation

The query modification is based on relevance back-propagation. It consists in spreading backward the document relevance from the document layer to the query layer [2].
Each document computes its input and output according to its relevance.

$$In(N_{D_j}) = rel_j$$

$$Out(N_{D_j}) = g(In(N_{D_j})$$

Where,

$rel_j = \frac{Coef\_Rel}{Nb\_rel}$ for relevant document
$rel_j = \frac{Coef\_NRel}{Nb\_Nrel}$ for non-relevant document

and

$Coef\_Rel$, $Coef\_NRel$: relevance coefficient of the documents (positive for relevant and negative for non-relevant documents),
Nb_rel, Nb_Nrel : number of relevant and non-relevant documents respectively.

Notice that the relevance is automatically assigned to the documents: only the top 12 retrieved documents are considered as relevant.

Finally, this activation is backpropagated to the term layer according to the formulas:

$$In(N_{t_i}) = \sum_{j=1}^{N}(Out(N_{D_j}) * w_{ij})$$

$$Out(N_{t_i}) = g(In(N_{t_i}))$$

New query-term link weights are then computed according to this formula :

$$q_{ik}^{s+1} = M_a * q_{ik}^s + M_b * Out(N_{t_i})$$

The new query is evaluated the same way the initial query is.

The parameters we have chosen for the TREC7 experiments are : $h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$.

The other parameters used in the relevance backpropagation are : $Coef\_Rel = 1$, $Coef\_NRel = -.75$, $Nb\_rel = 12$, $M_a = 2$, $M_b = .75$. Both MerAdRbtnd and MerAdRbtd runs used this technique.

Query modification based on text mining approach

The text mining approach consists in analyzing a sub-set of the retrieved documents in order to expand the initial query. A blind process as been chosen for this experiments as well and the top 12 documents constitute the set of documents to mine for each query. This number has been chosen according to previous experiments in TREC. The mining was performed using the Tétralogie system [5]. This system includes advanced information extraction functionalities. It implements document reduction (under the form of contingency tables) and factorial analysis mining functions. Given a set of items expressed in a n-dimensional space, the factorial analysis methods reduce the data dimensionality into a space which is the most important to characterize the items [1].
According to the experiments performed for TREC7, the document set mining was used to chose the terms to be added to the initial query as follows :

1. The most frequent terms have been extracted from the analyzed set of documents,
2. The co-occurrence value of these extracted terms and the query terms have been computed,
3. This crossing table was used as an input to a Correspondence Factorial Analysis (CFA),
4. Then, the terms that have the highest weight according to the AFC have been kept.

This technique was intended to determine the terms which are the most characteristic of the query terms according to the analyzed document set.

3.2 Ad hoc results and discussion

Three automatic runs have been submitted : MerAdRbtnd (long topic : title, description and narrative) and MerAdRbtd (title and description) based on backpropagation and MerTetAdtnd (long topic) based on text mining. These runs were based on a completely automatic processing of TREC queries and automatic query expansion based on "blind" feedback . Table 1 compares our runs against the published median runs.

TREC results			
Run	Best	$\geq$median	$< median$
MerAdRbtnd	0	38	12
MerTetAdtnd	0	34	16
MerAdRbtd	0	30	20

Table 1: Comparative automatic ad hoc results at average precision

Most of the runs are above the median. These results show that we obtained better results on the long topics than using the titles and descriptions only.

The average results obtained using the blind relevance feedback were less good than the one using no query reformulation. In fact, this observation is not uniform as some query results were improved by the reformulation. A deep analysis of these results could lead to a better understanding.

Type	Run	average precision	Exact precision
Long topics	basic search	.2290	.2760
"	MerAdRbtnd	.2278 (-.5%)	.2746 (-.5%)
"	MerTetAdtnd	.2237 (-2.3%)	.2617 (-.5%)
Titles-Descriptions	basic td	.1918	.2380
"	MerAdRbtd topic	.1918 (0%)	.2380 (0%)

Table 2: Ad hoc component results - 50 queries

4 Batch Filtering and Routing Experiment

The batch filtering and routing experiments were performed using Mercure system as described above.

Experiment :

The technique we used to build the batch and the routing queries is based on the relevance backpropagation process presented above. The AP88 documents were used as training data. The filtering algorithm starts with an initial query, built from all the parts of the topic, and its AP88 relevance judgments (positive and negative). Relevant and non relevant documents computes a relevance value that is backpropagated to the query. The query-term links are then modified and a query evaluation process is done through the new links. This process is repeated and a new learned query is built at each iteration.

A pool of queries was then selected. For the routing task, the queries showing having the best average precision in the training data were selected as routing queries. For the batch filtering, the TREC standard output file of each query was analyzed to build an output file containing:

$$< topic >< func >< value >< thresh >< rank >< prec >< recall >$$

as it has been done in [7]. The document activation weights that maximize each function F1 and F3 were then found and selected as thresholds. Then, the queries corresponding to these thresholds were selected and tested against the test data.

Results and discussion :

Routing task : The top 1000 retrieved documents were submitted for each routing query. One run was submitted (MerRou). The table 3 shows the comparative routing results at average precision.

TREC Routing			
Run	Best	≥median	$< median$
MerRou	5	13	37

Table 3: Comparative routing results at average precision

Batch Filtering : Two runs were submitted. One based on utility-[$F1$], it is labeled MerBF1. The other one based on the utility-[$F3$] and labeled (MerBF3). The table 4 lists the comparative batch results.

TREC batch filtering			
Run	Best	$\geq$median	$< median$
MerBF1	8	22	28
MerBF3	8	18	32

Table 4: Comparative batch filtering results for F1 and F3 functions

5 Conclusion

Our main goal this year was to perform completely automatic runs. We assessed two query modification techniques. The first one aimed at tuning the parameter values of the backpropagation formulas used in TREC6. Whereas the second one trained a text mining approach using Tétralogie system. It is difficult to draw any firm conclusion with respect to the results obtained this year. In average, the basic searches performed using Mercure have led to better results than the two query reformulation methods we have experimented. Fortunately, some reformulated queries performed better than initial queries. More attention will be paid next year to identify these queries and to better tune the two techniques. In addition to that, we plan to deeply investigate query reformulation using Genetic Algorithms. In fact, our first investigations in GA on small databases are encouraging [3].

References

[1] J.P. Benzecri, *L'analyse de données, Tome 1 et 2.* Edition Dunod, 1973.

[2] M. Boughanem & C. Soule-Dupuy, *Query modification based on relevance backpropagation*, Proceedings of the 5th International Conference on computer-assisted information searching on Internet (RIAO'97), Montreal (Canada), June 1997.

[3] M. Boughanem & L. Tamine, *Reformulation automatique de requête basée sur l'algorithmique génétique*, Proceedings of the 15th National Conference INFORSID'97, Toulouse (France), June 1997.

[4] C. Buckley & al, *Query zoning : TREC'5*, Proceedings of the 5th International Conference on Text REtrieval TREC5, Harman D.K. (Ed.), NIST SP 500-236, Nov. 1996.

[5] T. Dkaki, B. Dousset & J. Mothe, *Mining information in order to extract hidden and strategical information*, Proceedings of the 5th International Conference on computer-assisted information searching on Internet (RIAO'97), Montreal, June 1997.

[6] J.Hertz, A. Krogh & R. G. Palmer, *Introduction to the theory of neural computation* Addison Wesley, March 1992.

[7] S. Robertson and al, *Okapi at TREC-6*, Proceedings of the 6th International Conference on Text REtrieval TREC6, Harman D.K. (Ed.), NIST SP 500-236, Nov. 1997.