Mercure at trec6

M. Boughanem [1 2] **C. Soulé-Dupuy** [2 3]

[1] MSI
Université de Limoges
123, Av. Albert Thomas
F-87060 Limoges

[2] IRIT/SIG
Campus Univ. Toulouse III
118, Route de Narbonne
F-31062 Toulouse

[3] CERISS
Université Toulouse I
Manufacture des Tabacs
F-31000 Toulouse

Email : {bougha, soule}@irit.fr

1 Introduction

We continue our work in trec performing runs in adhoc, routing and part of the cross language track. The major investigations this year are the weight schemes modification to take into account the document length. We also experiment the high precision procedure in automatic adhoc environment by tuning the term weight parameters.

2 Mercure model

Mercure is an information retrieval system based on a connexionist approach and modelled by a network (as shown in the figure 1) containing an input representing the query, a term layer representing the indexing terms, a document layer representing the documents and an output representing the retrieved documents. The term nodes (or neurons) are connected to the document nodes (or neurons) by weighted indexing links. Mercure includes the implementation of two main components : the query evaluation based on spreading activation from the input to the output through the indexing links and the automatic query modification based on backpropagation of the document relevance.

2.1 Query evaluation based on spreading activation

The query evaluation is performed as follows :

1. Build the input $Input_k = (q_{1k}, q_{2k}, ..., q_{Tk})$,
2. Apply this input to the term layer. Each term neuron computes an input value :

$$In(N_{t_i}) = q_{ik}$$

   and then an output value : $Out(N_{t_i}) = g(In(N_{t_i}))$

3. These signals are propagated forwards through the network. Each neuron computes an input and an output value :

$$In(N_{D_i}) = \sum_{j=1}^{T} Out(N_j) * w_{ij}$$

   then, $Out(N_{D_i}) = g(In(N_{D_i}))$

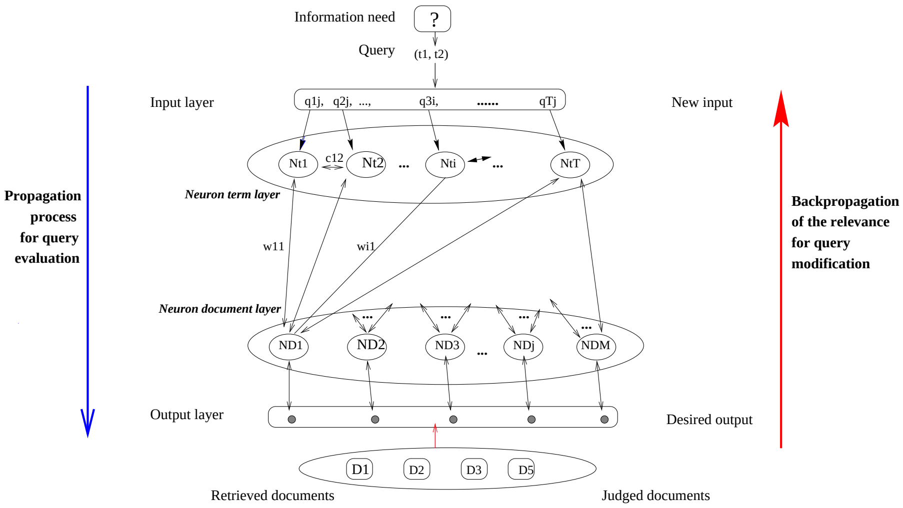

Figure 1: *The Mercure Model.*

The output vector is : $Output_k(Out(N_{D_1}), Out(N_{D_2}), ..., Out(N_{D_M}))$

These output values computed by the document neurons are used to rank the list of retrieved documents.

2.2 Query modification based on relevance backpropagation

The automatic query modification is based on spreading the document relevance values backwards the network. The retrieved documents are used to build the *DesiredOutput*. To each judged document is assigned a *relevance value*. A positive relevance value is assigned to relevant documents, a negative value to non-relevant documents. The desired output is represented by the vector of the form : $DesiredOutput = (rel_1, ..., rel_i, ..., rel_M)$.

This strategy consists in backpropagating the relevance values from the output layer to the input layer, and it is performed as follows :

1. Build the desired output : $DesiredOutput = (rel_1, ..., rel_i, ..., rel_M)$,
2. Apply this output to the neuron document layer. Each neuron computes an input value :
$$In(N_{D_i}) = rel_i$$
and then an ouput signal : $Out(N_{D_i}) = g(In(N_{D_i}))$
3. The output signals are backpropagated to the term neuron layer. Each neuron term computes an input value :
$$In(N_{t_i}) = \sum_{j=1}^{M}(w_{ij} * Out(N_{D_j}))$$
and then an output signal : $Out(N_{t_i}) = g(In(N_{t_i}))$
4. A new input is then computed according to this formula :
$$NewInput_k = \alpha * Input_k + \beta * Out(N_t)$$

This new input is applied to the term neuron layer and a new query evaluation is then done.

Several formulations can be used to construct the desired output. For this experimentation we have chosen the following formula :

- $for\ relevant\ document\ :\ rel_i = \frac{Coef\_Rel}{Nb\_rel}$
- $for\ nonrelevant\ document\ :\ rel_i = \frac{Coef\_NRel}{Nb\_Nrel}$

Where :
$Coef\_Rel$, $Coef\_NRel$: 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,

3 General Investigations

Our first investigation is to modify the indexing weight to take into account the document length. Our formula is inspired by Okapi and Smart term weight functions. It is expressed by :

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

The query term weight in the input is expressed by :

$$q_{ik} = \frac{(1 + log(tf_{ik})) * (log(N/n_i)}{\sqrt{\sum_{j=1}^{T} (1 + log(tf_{jk})) * (log(N/n_j)^2}}$$

Where :
w_{ij} : the weight of the link between the term t_i and the document D_j,
tf_{ij} : the frequency of the term t_i in the document D_j,
T : the number of documents in the collection,
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, computed for each database.

4 Adhoc experiment and results

4.1 adhoc methodology

Our investigation is to improve the query expansion in automatic adhoc environment. The "blind" relevance feedback was performed by assuming the top retrieved documents as relevant and the low retrieved as non relevant. Some efforts have been undertaken to improve the precision in the small top ranked documents. The basic goal is to produce the "High precision" by "trading" the recall for the precision, [4] [5] (e.g we can loose some relevant documents if we are sure that the remaining ones are relevant).

A way to produce a high precision could be by using "good" query term and document term weights. Our strategy in adhoc trec-6 is to weight the indexing links in order to maximize the precision at small ranked top documents and then a "normal" weight scheme (weight performing a best precision at 1000 top ranked documents) will be used in the relevance backpropagation process and in the new input spreading. The weight schemes we used in trec-6 are obtained by tuning the h_1, h_2, h_3, h_4 parameters.

Series of experiments have been undertaken on TREC-5 database and queries. The parameters we have chosen to use in TREC-6 experiment are : $h_1 = 1$, $h_2 = 0$, $h_3 = .8$, $h_4 = .2$ for the high precision and $h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$ for what we called a "normal" weight. The remaining parameters used in the relevance backpropagation are : $Coef\_Rel = 1$, $Coef\_NRel = -.75$, $\alpha = 2$, $\beta = .5$, $Nb\_rel = 12$, $Nb\_Nrel = 500$ (from 501 to 1000).

4.2 Adhoc results and discussion

Preliminary investigations

The first result we underline concerns the term weight functions. The table 1 shows the average precision of basic run obtained by some IR systems in TREC-5. We can notice that the weight schemes we used are quite good ($h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$).

TREC-5 results	
system	average precision in initial search
Mercure :	0.1578
Okapi :	0.1520
Smart :	0.1484
Inquery :	0.1442

Table 1: Comparative basic search trec-5 results

Automatic adhoc results

Three automatic runs were submitted : Mercure2 (description only), Mercure1 (long topic : title, description and narrative) and Mercure3 (title only). These runs were based on completely automatic processing of TREC queries and automatic query expansion, the high precision concept was also used. Table 2 compares our runs against the published median runs. We notice that most of the runs are above the median.

TREC results			
Run	Best	$\geq$median	$< median$
Mercure2 (description)	1	40	10
Mercure3 (title)	1	29	18
Mercure1 (long topic)	5	44	6

Table 2: Comparative automatic adhoc results at average precision

We unfortunatelly noticed an error in the script that has been used to perform the adhoc description run (the other runs are right). The weight scheme (i.e. the h_i parameters) used to produce the high precision has also been used by mistake in the relevance backpropagation process instead of the "normal" h_i values. The table 3 shows the official and the corrected runs).

We actually notice a difference between the description runs, the other runs seem good.

The table 4 the average precisions of the basic run using the high precision and the run after query expansion on the three corrected runs. The query expansion is done by using the following values of Mercure parameters : $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relevant documents and the number

Run	Official results		corrected results	
	Average precision	R. Precision	Average precision	R. Precision
Mercure2 (description)	0.1640	0.2065	0.1720	0.2108
Mercure3 (title)	0.2316	0.2689	0.2316	0.2689
Mercure1 (long topic)	0.2305	0.2700	0.2305	0.2700

Table 3: Automatic adhoc results - 50 queries

pf term added to the query is 16. We notice that the automatic query expansion is still effective in the adhoc environment.

Run	average precision
Mercure3 : title only	
basic search using h_i producing the high precision	0.2041
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.2316 (+13.47 %)
(Mercure2.C) description only	
basic search using h_i producing the high precision	0.1549
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.1710 (+10.39 %)
(Mercure1) long topic	
basic search using h_i producing the high precision	0.2128
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.2305 (+8.32 %)

Table 4: Adhoc component results - 50 queries

However we notice that the way used to improve the precision at top ranked documents did not have a positive effect as in the trec-5 adhoc . Indeed, the table 5 shows the results in the description run (Mercure2.C.N) when using the "normal" h_i values. We observe a slight different in favour of the Mercure2.C.N run. We do not yet analyze the results of the title and long topics runs.

Run	average precision
Mercure2.C.N: description only	
basic search using "normal" h_i	0.1693
Exp. $Nb\_rel = 12$, $Nb\_Nrel = 500$, 501-100 non-rel docs	0.1772

Table 5: Adhoc component results - 50 queries

5 Routing experiment and results

All trec-6 training data were used (relevant and non relevant documents). The queries are initially built automatically from all the fields of the topics and then expanded by using the 30 top terms resulting from the relevance backpropagation procedure.

Each query was evaluated by varying the different Mercure parameters, h_i and α, β, etc. The queries performing the best average precision in the training data were selected. Moreover, a slight modification has been performed in the relevance value formula, it concerns the positive relevance value. Indeed, we decided to take into account the fact that a relevant document is or not among the 1000 retrieved documents in the initial search.

The relevance value assigned to each relevant document becomes :

- $rel_i = \frac{coef\_R}{Nb\_rel} * BOOT \begin{cases} BOOT = 1 \ if \ the \ relevant \ document \ is \ not \ in \ the \ 1000 \ documents \\ BOOT < 1 \ if \ relevant \ document \ is \ retrieved \ (BOOT = .9 \ for \ routing \ trec6) \end{cases}$
- no modification if a document is nonrelevant

As the retrieved relevant documents are already close to the initial query, we give to the terms occurring in the non retrieved relevant documents more effect in the final query building.

The table 6 compares our routing runs against the medians published runs, more than 60% of queries are above the median.

TREC routing results			
Run	Best	$\geq$median	$< median$
Mercure4	1	29	18

Table 6: Comparative TREC Results at average precision

The table 7 shows the difference between the run based on the initial queries and the one based on the routing queries. We ha have no time to analyze these results

TREC routing results			
Run	average precision	R precision	Total Rel retrieved
Mercure4	0.3061	0.3400	4774

Table 7: Comparative TREC Results at average precision

Run	average precision
basic search (with the initial queries)	0.2676
Official run	0.3061

Table 8: Routing component results 47-queries

6 Cross language track : french to french

Two runs french to french were submitted in CLIR track. The indexing and search methodologies are the same than the adhoc trec6 except the stemming algorithm where a cutoff stemming method (7 characters) has been used. This stemming method has been implemented in all of our operational information retrieval systems dealing with french documents and french queries. The results obtained untill now lead us to go on the experiments with this stemming method.

Moreover, for the adhoc task the high precision procedure has not been used because there is no relevance information to tune the weight scheme. The same parameters were used for the indexing weight $h_1 = .8$, $h_2 = .2$, $h_3 = .8$, $h_4 = .2$.

The table 9 compares our runs against the published median runs. Most of the queries are above the median.

TREC-6 cross language french to french			
Run	Best	≥median	$< median$
MercureFFs (description)	0	18	3
MercureFFl (long topic)	4	17	4

Table 9: Comparative TREC cross language at average precision

The table 10 shows that the average precision and the R-precision for the different runs are quite good.

Run	Average precision	R. Precision	Total Rel Retrieved
MercureFFs (description)	0.3619	0.3848	1023
MercureFFl (long topic)	0.3778	0.4015	1033

Table 10: cross language (french to french) results - 21 queries

The important point we discuss concerns the automatic query expansion. Indeed, the table 11 shows the improvment obtained between the basic run and the run with an automatic query expansion using the following values of Mercure parameters : $Nb\_rel = 15$, $Nb\_Nrel = 500$, 501-1000 non-relevant docs and the number of added terms is 16. In both, MercureFFs and MercureFFl the improvement about 10%.

Run	average precision
description only	
basic search	0.3262
Expansion $Nb\_rel = 15$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.3619 (11%)
long topic	
basic search	0.3479
Expansion $Nb\_rel = 15$, $Nb\_Nrel = 500$, 501-1000 non-relev docs	0.3778 (8.6 %)

Table 11: Adhoc cross language component results - 21 queries

7 Conclusion

Last year, we participated in trec-5 in the adhoc and routing tasks in category B. Our main effort this year has been to participate in trec-6 in category A. We performed completely automatic runs in adhoc, routing and a part of the cross language tasks.

At first we planed to try, the passage retrieval, the data mining techniques [7] and the genetic algorithms [1] to automatically expand the queries. But finally, our investigations were the improvement of the term weighting and the automatic query modification. We spent much time on these experiments and decided to difer the planed experiments until the next year.

However, the results we obtained for the main tasks are still encouraging this year. Our participation to the CLIR track was limited to a french to french experimentation to train our french language processing. Our goal now is to go on with a real cross language experiment.

References

[1] L. Tamine Reformulation de requêtes basée sur l'algorithmique génétique Proceedings of INFORSID'97 Toulouse Juin 1997.

[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, June 1997.

[3] M. Boughanem & C. Soule-Dupuy, *Mercure : adhoc and routing tasks*, 5th International Conference on Text REtrieval TREC2, Harman D.K. (Ed.), NIST SP 500-236, 1996.

[4] C. Buckley & al, *Query zoning : TREC'5*, 5th International Conference on Text REtrieval TREC2, Harman D.K. (Ed.), NIST SP 500-236, 1996.

[5] B. croft, & al, INQUERY at TREC-5. 5th International Conference on Text REtrieval TREC5, Harman D.K. (Ed.), 1996.

[6] S. Robertson and al, *Okapi at TREC-5* . 5th International Conference on Text REtrieval TREC2, Harman D.K. (Ed.), NIST SP 500-236, 1996.

[7] T. Dkaki, B. Dousset & M. 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.