DATABASE MERGING STRATEGY BASED ON LOGISTIC REGRESSION

Anne Le Calvé, Jacques Savoy

Institut interfacultaire d'informatique
Université de Neuchâtel (Switzerland)

e-mail: {Anne.Lecalve, Jacques.Savoy}@seco.unine.ch

Abstract With the development of network technology, the users looking for information may send a request to various selected databases and then inspect multiple result lists. To overcome such multiple inspections, the database merging strategy involves the merging of the retrieval results produced by separate, autonomous servers into an effective, single ranked list. To achieve this merging, this study is concerned with a particular situation within which only the rank of the retrieved records is available as a key to combine different result lists. Based on this rather limited information, this paper describes the theoretical foundation and retrieval performance of our database merging approach based on the logistic regression.

Keywords: Database merging, collection fusion, logistic regression methodology.

1. INTRODUCTION

Actually, it becomes more and more difficult to store and manage a growing documents collection within a single computer. Recent advances in network technology allows us to disseminate information sources by partitioning a single huge corpus (or to distribute heterogeneous sub-collections) into a local-area network (INTRANET). Moreover, the INTERNET paradigm also permits to search for information across wide-area networks. Therefore, to answer to a need of information expressed by a query, some tools called metasearch send simultaneously the request to several separate search engines. To achieve this purpose, metasearch system are working in three principal steps (Dreilinger & Howe, 1997). First, the dispatch mechanism selects the appropriate search engines to which the query will be sent. Second, the interface agents convert the request into a format readable by the selected search servers (for example, based on the Z90.50 protocol for inter-system retrieval). Third, the display mechanism have to select,

sort and present a unique result list. Some examples of such automatic metasearch are, MetaCrawler, SavySearch and ProFusion system. Using manual metasearch, like All-In-One, InterNIC and again ProFusion system, the users are completely free to select the appropriate search engines.

In the rest of this paper, we shall assume that the dispatch mechanism has already selected the appropriate information servers. We will concentrate on the third step of architecture: the display mechanism. This part, known as collection fusion or database merging strategy, combines the results provided by separate search engines into a single final ranked list. Various strategies as described in Section 2, have been proposed to resolve the collection fusion or database merging problem. To treat the problem when different retrieval schemes interrogate different information collections, we present in Section 3 a new model based on logistic regression. Section 4 depicts evaluations and comparisons of most of these strategies.

2. RELATED WORK ON DATABASE MERGING

Recent works have suggested some solutions to the merging of separate answer lists obtained from distributed information services. As a first approach, we might assume that each database contains approximately the same number of pertinent items and that the distribution of the relevant documents is the same across the servers answers. Based only on the rank of the retrieved records, we may interleave the results in a round-robin fashion. According to previous studies (Voorhees *et al.* , 1995 April; Callan *et al.*, 1995 June), the retrieval effectiveness of such interleaving scheme is around 40% below the performance achieved by a single retrieval scheme working with a single huge collection representing the entire set of documents.

Voorhees *et al.* (1995 July; 1996 October) demonstrate that we may improve this ranking scheme based on the estimated expected relevance of each sub-collection to the current request. Thus, instead of extracting an equal amount of items from each sub-collection, the suggested scheme retrieves, for each information server, a number of documents related to the previous performance of the underlying sub-collection. Depending on the underlying learning schemes, the overall performance is 20% to 30% below the average precision produced by a single huge collection.

However, many search models return not only the rank of the retrieved items but also a numeric score (e.g., a retrieval status value (RSV) or a document score) indicating the similarity strength between the retrieved document and the request.

To take account for this additional information, we might formulate the hypothesis that each information server applies the same or a very similar search strategy and that the similarity values are therefore directly comparable (Kwok *et al.*, 1995 April), (Moffat & Zobel, 1995 April). Such a strategy, called raw-score merging, produces a final list sorted by the retrieval status value computed by each separate search engine. However, as demonstrated by Dumais (1994 March), collection-dependent statistics in document or query weights may vary widely among sub-collections, and therefore, this phenomenon may invalidate the raw-score merging hypothesis.

Finally, Callan *et al*. (1995 June) suggest a merging strategy based on the score achieved by both sub-collection and document. The first score is computed according to the probability that the sub-collection respond appropriately to the current request, and the second is the usual retrieved status value. The evaluation of this approach shows a performance similar to a run treating the entire set of documents as a single collection.

When only the rank is available to merge different retrieval schemes, we deal with isolated database merging problem and to resolve this, we suggest another approach based on logistic regression. When additional information is available such as document score, we will face with integrated database merging strategies. Table 1 presents a classification and examples of these four database merging situations.

	Same retrieval schemes	Different retrieval schemes
only the rank isolated DB merging	round-robin (Voorhees et al. 1996 October)	round-robin our approach
rank, score, ... integrated DB merging	raw score merging (Callan et al. 1995 June)	normalized raw score merging

Table 1: Classification and examples of database merging strategies

3. LOGISTIC REGRESSION

We suggest using the logistic regression (Cox & Snell, 1989; Hosmer & Lemeshow, 1989) as a methodology for combining multiple sources of evidence regarding the relevance of a given document. Of course, this statistical approach has been already applied in related domains such as informetrics (Bookstein *et al.*, 1992) or as a retrieval model (Gey, 1994), (Fuhr & Pfeifer, 1994). In our context, we will use the logistic regression as a theoretical methodology and as a practical mean to combine different retrieval schemes which can be based on various and very different search models such as the vector space model, the probabilistic approach, etc.

3.1. Modeling

The logistic regression is a statistical methodology to predict the probability of a binary outcome variable according to a set of independent explanatory variables. In our approach, we use a logistic regression to predict the probability of relevance of documents retrieved by different retrieval schemes. In such circumstances, explanatory variable could be the rank, the retrieval status value, or other information like the publication date. The estimated probabilities can be used to select and sort the retrieved records obtained from separate information servers in order to obtain a single ranked list.

As a first specification, we have thought using only the rank of the retrieved items as explanatory variable. In this case, the probability of relevance changes systematically with the rank order (or the serial order). However, using this ordinal variable without any transformation, assumes regular differences between retrieved documents position which is not realistic. A difference of 10 in rank seems to be more significant between the ranks 20 and 30 than between 990 and 1000. These last ranks contain a so little number of relevant documents that it could be appropriate to ignore the difference between 990 and 1000. To take into account for these irregularities, we suggest using the logarithm of the rank instead of the rank. The purpose of this transformation is to increase differences between first ranks and the rest of the rank distribution. Various experiments confirmed empirically such a logarithm transformation. Therefore, we define our model according the following equation:

$$\text{Prob}\left[D_i \text{is Rel} \mid x_i\right] = \pi(x_i) = \frac{e^{\alpha+\beta\cdot x_i}}{1+e^{\alpha+\beta\cdot x_i}} \qquad (1)$$

within which x_i is the natural logarithm (noted ln) of the rank for a retrieved document.

In this equation, the coefficients α and β are unknown parameters which fit the S-curve shown in Figure 1. The estimation of the value of these coefficients are noted $\hat{\alpha}$ and $\hat{\beta}$, and are calculated according the principle of maximum likelihood (the required computations have been done with the SAS package).

The logistic regression methodology may of course take account of multiple independent variables. However in our context, we want to resolve the isolated database merging problem within which only the rank is available as a key to select and merge the various retrieved records.

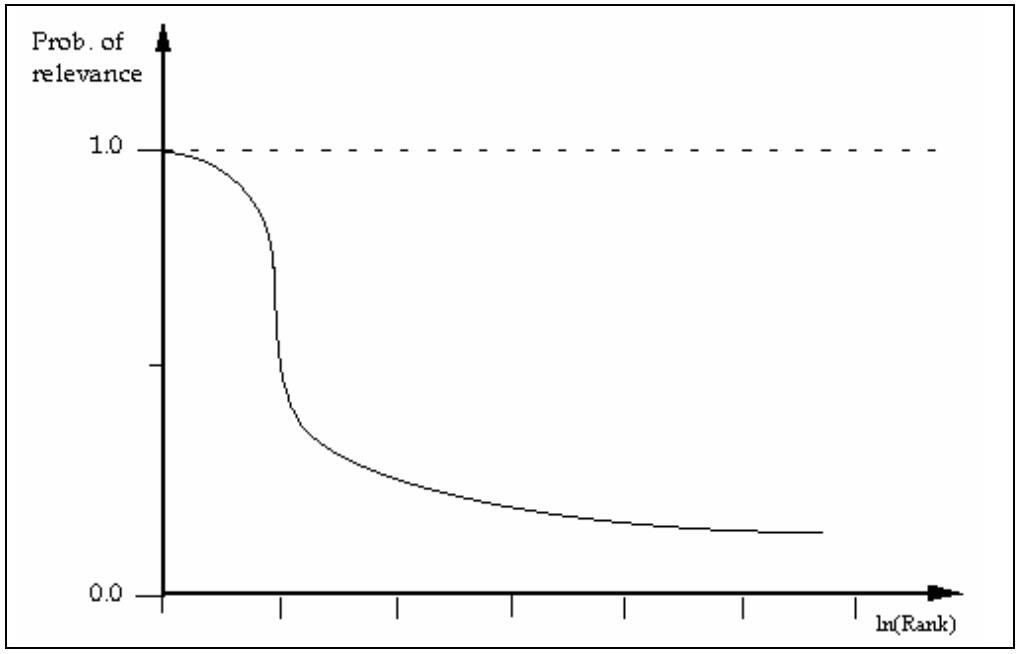

Figure 1: Example of logistic function with $\beta < 0$

3.2. Our model

To verify the validity of this model, we have chosen the "Wall Street Journal" (WSJ) collection (74,520 documents, 186 queries) extracted for the TREC conference corpora. To simulate various information servers working with different databases,

the WSJ test-collection has been divided in three separate sub-collections according to the publication year. The resulting sub-collections called WSJ90, WSJ91 and WSJ92 do not contain a similar amount of texts (ranging from 35 Mb to 146 Mb) nor a similar number of relevant articles per query. Various statistics about these collections can be found in the Appendix 1.

Moreover, a different retrieval scheme has been applied on these three sub-collections as follow:

- the first search engine is based on the OKAPI probabilistic model working with WSJ90 collection (Robertson *et al*., 1995);
- the second server stores the WSJ91 corpus and the retrieval system is based on the vector-processing LNU - LTC (Buckley *et al*., 1996 October);
- the third retrieval scheme is based on the vector-space LNC - LTC model and works with the WSJ92 collection.

Table 2 presents the coefficient values of our logistic model applied on each retrieval schemes together with related statistics.

	WSJ90 OKAPI		WSJ91 LNU		WSJ92 LNC	
	Intercept α	ln(rank) β	Intercept α	ln(rank) β	Intercept α	ln(rank) β
values	0.3218	-0.9492	0.6341	-0.9016	-0.3099	-0.9758
standard error	0.0627	0.0139	0.0555	0.0117	0.0842	0.0197
Wald test on each parameter	p=0.0001	p=0.0001	p=0.0001	p=0.0001	p=0.0001	p=0.0001
Wald test on regression	p=0.0001		p=0.0001		p=0.0001	

Table 2: Logistic regression coefficients for our model

To examine the adequacy of fit, we have used a single overall statistic of goodness of fit (e.g., Wald test for our logistic model depicted in the last raw of Table 2). All the logistic models are significant and the null hypothesis that the values of all coefficients are equal to zero, is always rejected.

More precisely, for WSJ91 LNU-LTC, the Wald test indicates that the regression model is significant (p=0.0001) with the point estimate $\hat{\alpha} = 0.6341$ (associated standard error 0.0555) et $\hat{\beta} = -0.9016$ (standard error 0.0117). The values of both coefficients may be considered as significantly different from 0 (Wald test, p=0.0001).

The confidence interval estimation (95%) for the intercept $\hat{\alpha}$ is 0.6341 ± 1.96 · 0.0555 = [0.5253; 0.7429], and for the coefficient $\hat{\beta}$ is -0.9016 ± 1.96 · 0.0117 = [-0.9245; -0.8787].

Once the three logistic regressions have be computed separately for the three sub-collections, the second part of our database merging strategy consists in merging the three independent lists of records extracted from our three information servers into a single final list. Based on the values of the coefficients depicted in Table 2, Table 3 shows the estimated probabilities associated with each sub-collection together with the combined final ranked list.

WSJ90 OKAPI		WSJ91 LNU-LTC		WSJ92 LNC-LTC	
rank	prob.	rank	prob.	rank	prob.
1	0.57976	1	0.65342	1	0.42314
2	0.41675	2	0.50229	2	0.27165
3	0.32717	3	0.41183	3	0.20070
4	0.27011	4	0.35074	4	0.15941
5	0.23043	5	0.30641	5	0.13234
6	0.20118	6	0.27262	6	0.11322
...		...		...	

combined list		
rank	prob.	from
1	0.65342	LNU-LTC (1st)
2	0.57976	OKAPI (1st)
3	0.50229	LNU-LTC(2nd)
4	0.42314	LNC-LTC (1st)
5	0.41675	OKAPI(2nd)
6	0.41183	LNU-LTC(3rd)
7	0.35074	LNU-LTC(4th)
8	0.32717	OKAPI(3rd)
9	0.30641	LNU-LTC(5th)
10	0.27262	LNU-LTC(6th)
...	...	

Table 3: Example of merging based on coefficient values shown in Table 2

From the example shown in Table 3, we can see that in the ten-best ranked records appearing in the final list, 6 are coming from the LNU-LTC scheme (WSJ91), 3 from OKAPI probabilistic scheme (WSJ90), and only one record from LNC-LTC search model (WSJ92). Looking back at the statistics of the three sub-collections depicted in Appendix 1, we can see that the WSJ91 collection contains twice more documents

(and relevant items) than WSJ90 and that WSJ92 owns only 12.8% of the relevant records (831 over 6,468).

3.3. Model interpretation

A first interpretation of logistic regression results is to analyze the odds and the probabilities. The odds of making response 1 (or "relevant") instead of response 0 (or "not relevant") is defined as:

$$odds(\pi(x_i)) = \left[\frac{\pi(x_i)}{1-\pi(x_i)} \right] = e^{\alpha+\beta \cdot x_i} \qquad (2)$$

For example, if the odds value is one, the document has equal chance to be relevant or non-relevant, and with an odds = 0.5, the document owns twice more chance to be non-relevant than relevant.

Based on the retrieval scheme LNU-LTC (WSJ91 sub-collection), Table 4 presents some estimated probabilities and the corresponding odds.

Rank i	$\pi(x_i)$	$odds(\pi(x_i))$
1	0,6534	1.8853
2	0,5023	1.0092
3	0,4118	0.7002
4	0,3507	0.5402
5	0,3064	0.4418
6	0,2726	0.3748
7	0,2459	0.3262
8	0,2243	0.2892
9	0,2064	0.2600
10	0,1912	0.2365
...		
100	0,0288	0.0297
...		
200	0,0156	0.0159

Table 4: Example of probabilities and odds on WSJ91 LNU-LTC

odds($\pi(x_i)$)	$\pi(x_i)$	ln(rank) = [-ln(1/odds)- α]/β	rank
1.5	0.6	0.253	1.29
1	0.5	0.703	2.02
0.5	0.33	1.472	4.36

Table 5: Odds and corresponding ranks for WSJ91 LNU-LTC

Moreover, we can be interested to know the corresponding ranks of the odds values 1, 1.5 and 0.5 as depicted in Table 5. This table shows that just two-best ranks are considered to have more chance to be relevant. Immediately after the rank 2, the probability to be non-relevant becomes higher than the relevant one. These results are interesting because we may directly related the rank with the probability of relevance. In our evaluation, there is a few number of relevant documents in the retrieved documents (a mean proportion of 1.6% per query when 1,000 items are retrieved by request). Our model indicates that the first two ranks may "certainly" contains relevant items. Such information is useful when facing with high precision searches.

Another interpretation of the logistic regression is the meaning of the coefficient β. In classical linear regression, this slope coefficient indicates the increase of the outcome variable for every unit of change in the independent variable. In the context of logistic regression, the interpretation of β can be done according to the following equation:

$$\beta = g(x+1) - g(x) = \ln(\psi(1)) \tag{3}$$

where g(x) is the logit defined as:

$$g(x) = \ln\left[\frac{\pi(x_i)}{1-\pi(x_i)}\right] = \alpha + \beta \cdot x_i$$

Equation 3 is therefore the logit difference, or the log of the odds ratio.

The effect of a change of *c* units in the independent variable is given by the following equation:

$$\psi(c) = e^{c\beta} \tag{4}$$

For example, in the sub-collection WSJ90 (OKAPI retrieval scheme), an increase of 10 in ranks implies that the chance of relevance increase $\psi(\ln(10))= 0.112$ times which means decrease of $1/0.112 = 8.9$ times. For the retrieval scheme LNU-LTC (sub-collection WSJ91), it implies a decrease of 7.97 times and the LNC-LTC on WSJ92 a decrease of 9.46.

4. EVALUATION

To evaluate our suggested scheme, we have used the "Wall Street Journal" collection from TREC corpora (see Appendix 1). The indexing procedures applied for both the documents and the queries are described in Appendix 2. Moreover, we must mention that, in order to deal with a more realistic situation, the topics are indexed using only the Descriptive section.

As an evaluation measure, we have used the non-interpolated average precision at eleven recall values provided by the TREC2_EVAL software based on 1,000 retrieved items per request. To decide whether a search strategy is better than another, the following rule of thumb is used: a difference of at least 5% in average precision is generally considered significant and, a 10% difference is considered very significant (Sparck Jones & Bates, 1977).

A first evaluation of the retrieval schemes performance is presented in the first sub-section while the second shows the results of database merging using the same retrieval schemes for each sub-collection. Our logistic regression model is evaluated and compared to other database merging strategies in the last sub-section.

4.1. Preliminary evaluations

The result of our first experiment is shown in Table 6 within which we have considered the probabilistic model OKAPI and various vector-processing schemes (from LNU-LTC to NNN-NNN). These retrieval strategies are evaluated on our three sub-collections (WSJ90, WSJ91 and WSJ92) and also on the whole WSJ corpus treated as a single collection. The conclusion that can be drawn for Table 6 is clear: the probabilistic model OKAPI and the vector-processing scheme LNU-LTC present a similar average precision which is usually significantly better than other search strategies.

	Precision (% change)							
collection	WSJ90		WSJ91		WSJ92		WSJ	
	170 queries		171 queries		140 queries		186 queries	
model	2074 rel. doc.		3563 rel. doc.		831 rel. doc.		6468 rel. doc.	
OKAPI - NPN	24.13		22.74		28.16		20.32	
LNU - LTC	24.31	(+0.7)	22.88	(+0.6)	27.95	(-0.7)	20.27	(-0.2)
LTN - NTC	23.89	(-1.0)	22.49	(-1.1)	24.13	(-14.3)	19.16	(-5.7)
HTN - BNN	23.21	(-3.8)	21.46	(-5.6)	26.00	(-7.7)	19.09	(-6.1)
ATN - NTC	22.27	(-7.7)	20.50	(-9.9)	24.11	(-14.4)	18.85	(-7.2)
ANN - NTC	21.03	(-12.8)	18.81	(-17.3)	22.20	(-21.2)	16.78	(-17.4)
LNC - LTC	18.39	(-23.8)	17.72	(-22.1)	22.52	(-20.0)	15.76	(-22.4)
LTC - LTC	17.55	(-27.3)	17.98	(-20.9)	22.94	(-18.5)	15.06	(-25.9)
ANC - LTC	14.12	(-41.5)	14.02	(-38.3)	19.06	(-32.3)	11.94	(-41.2)
ANN - ANN	14.70	(-39.1)	11.43	(-49.7)	15.53	(-44.9)	10.76	(-47.0)
LNC - LNC	11.89	(-50.7)	10.69	(-53.0)	14.23	(-49.5)	9.69	(-52.3)
BNN - BNN	9.37	(-61.2)	5.84	(-74.3)	7.82	(-72.2)	4.85	(-76.1)
NNN - NNN	6.36	(-73.6)	5.80	(-74.5)	7.94	(-71.8)	4.53	(-77.7)

Table 6: Evaluation of various retrieval schemes (Topic = <desc>)

4.2. Merging based on similar retrieval schemes

Using the performance achieved by the WSJ test-collection as baseline, we want to compare the relative performance of various database merging strategies when the search on each sub-collection is based on the same retrieval scheme. Thus, Table 7 compares round-robin and raw-score merging strategies when the same retrieval schemes operates for each sub-collection.

	Precision (% change)					
collection model	WSJ baseline 186 queries	WSJ90- WSJ91- WSJ92 round-robin 186 queries		raw-score 186 queries		raw-score & optimal selection 186 queries	
OKAPI - NPN	20.32	17.34	(-14.7)	20.37	(+0.2)	21.93	(+7.9)
LNU - LTC	20.27	17.28	(-14.8)	20.09	(-0.9)	22.10	(+9.0)
LTN - NTC	19.16	17.46	(-8.9)	18.76	(-2.1)	21.53	(+12.4)
HTN - BNN	19.09	16.55	(-13.3)	18.79	(-1.6)	20.76	(+8.7)
ATN - NTC	18.85	16.24	(-13.8)	18.13	(-3.8)	20.31	(+7.7)
ANN - NTC	16.78	14.80	(-11.8)	16.62	(-1.0)	18.74	(+11.7)
LNC - LTC	15.76	12.90	(-18.1)	15.60	(-1.0)	16.82	(+6.7)
LTC - LTC	15.06	12.87	(-14.5)	14.91	(-1.0)	16.04	(+6.5)
ANC - LTC	11.94	9.92	(-16.9)	11.77	(-1.4)	12.95	(+8.5)
ANN - ANN	10.76	9.17	(-14.8)	10.81	(+0.5)	11.51	(+7.0)
LNC - LNC	9.69	8.24	(-15.0)	9.76	(+0.7)	10.46	(+7.9)
BNN - BNN	4.85	5.58	(+15.1)	4.87	(+0.4)	6.18	(+27.4)
NNN - NNN	4.53	3.97	(-12.4)	4.53	(0.0)	4.92	(+8.6)

Table 7: Evaluation of database merging strategies (Topic = <desc>)

The round-robin strategy, a naive approach for resolving the database merging problem, shows a degradation of around 14%. This poor result confirms previous studies (Callan *et al.*, 1995 June) indicating a depreciation of around 50%. However, the difference between these percentages seems to be due to the underlying characteristics of the WSJ collection. In analyzing the raw-score merging, Callan *et al.* (1995 June) demonstrate that such a technique may decrease the retrieval effectiveness by around 10% when working with heterogeneous corpora. This is not confirmed in this study because the WSJ collection tends to form a more homogenous set of documents having a very similar idf measure among its sub-collections. Thus, the raw-score merging seems to be valid as a simple first approach when a huge collection of similar documents is distributed across a local-area network and operated within the same retrieval scheme.

So far, we have taken into account the result list provided by all information servers. However, as mentioned in Appendix 1, each sub-collection does not always contain a relevant item for each request. Therefore, we might figure out a collection selection procedure which may choose only sub-collections containing at least one

pertinent document for the current query (dispatch mecanism). To evaluate the impact of such a selection procedure, Table 7 indicates, in its last column, the performance achieved by such an optimal selection, ignoring sub-collections having no relevant information for a given query. The resulting data indicates that an optimal collection selection procedure may significantly enhance the retrieval effectiveness. Moreover, applying such a selection procedure before evaluating a request is economically attractive. Similar conclusion can be draw when indexing topics according both the Descriptive and Narrative section (see Appendix 3).

4.3. Merging based on different retrieval schemes

The results depicted in Table 7 are based on the assumption that each information server applies the same retrieval strategy. Such a hypothesis is clearly unrealistic, especially in a wide-area network. More pragmatically and in order to evaluate our logistic model in a more realistic context, we will face with the following scenario. Our queries will be submitted to three independent servers working with: (1) the probabilistic model OKAPI on WSJ90, (2) the vector-processing schemes LNU-LTC on the WSJ91 sub-collection, and (3) the LNC-LTC scheme on the WSJ92 corpus.

Table 8 illustrates the retrieval performance of each sub-collection together with various statistics about the underlying retrieval status values. As a baseline, we have evaluated the round-robin strategy, achieving a mean precision of 16.96.

	Precision (% change)		
Collection	WSJ90 170 queries 2074 rel. doc.	WSJ91 171 queries 3563 rel. doc.	WSJ92 140 queries 831 rel. doc.
Model	OKAPI	LNU - LTC	LNC - LTC
Average precision	24.13	22.88	22.52
# of relevant documents	2074	3563	831
# of relevant doc. retrieved	1765	2811	745
RSV min.	1.152	0.002	0.011
RSV max.	47.176	0.0065	0.0555
RSV mean	6.013	0.027	0.435
RSV standard error	2.741	0.0026	0.0268
	WSJ 186 queries		
Isolated DB merging: Round-robin Logistic ln (RANK(D_i))	 16.96 18.40 (+8.49)		
Integrated DB merging: Raw-score merging Norm. raw-score merging	 8.75 (-48.41) 15.42 (-9.08)		

Table 8: Evaluation of database merging strategies (Topic = <desc>)

When analyzing the collection fusion problem in such circumstances, the raw-score merging strategy is clearly ineffective. All the retrieved documents are extracted form the WSJ90 sub-collection because the OKAPI model retrieval status values are always greater than those of the LNU-LTC or LNC-LTC search schemes. If we normalize the retrieval status value (RSV) within each sub-collection by dividing them by the maximum RSV of each result list, the retrieval performance is always significantly worse that the round-robin strategy.

When only rank can be used as explanatory variable, our logistic model exhibits an interesting performance, always significantly better than all other merging strategies. This finding is confirmed when using longer requests (see Appendix 3).

However, the evaluation reported in Table 8 can be questionable because we have used the same set of queries to estimate the value of the coefficients of our logistic model and to evaluate our merging strategy (retrospective evaluation). To check the validity of the logistic regression coefficients on the one hand, and on the

other of our evaluations, we have divided the request set in ten disjoint sub-samples (ten-fold cross-validation) (Stone, 1974).

Figure 2 and 3 present the 95% confidence intervals of each of these estimations (coefficients of the WSJ91 LNU-LTC retrieval strategy). In these figures, the value of the parameters shown in Table 2 are indicated by a solid line. A similar picture can be drawn from the other two retrieval schemes. Thus, we may this infer that the coefficient value shown in Table 2 are trustful.

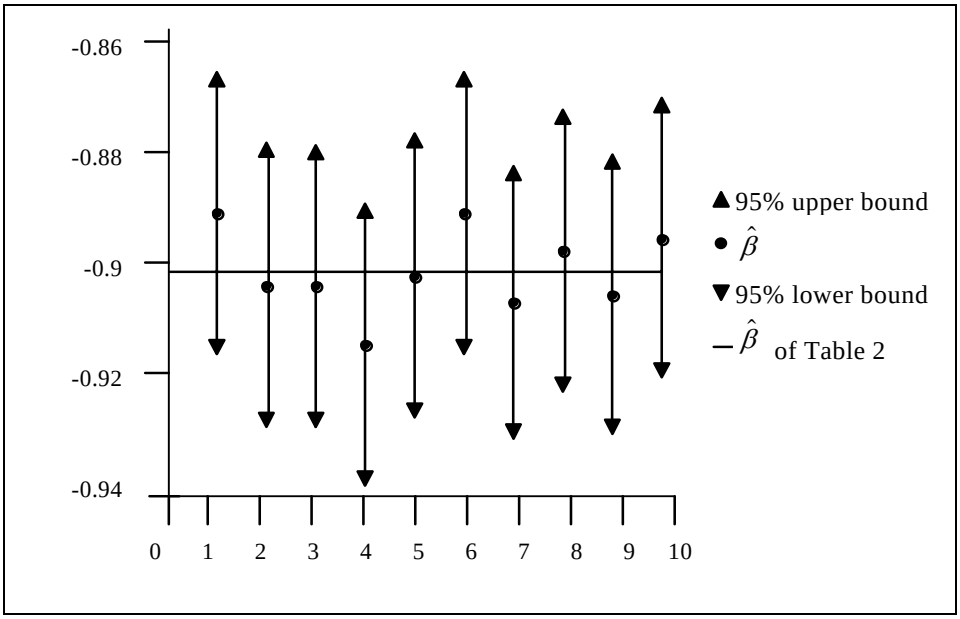

Figure 2: 95% confidence interval estimate for $\hat{\beta}$ (10-fold cross-validation) for WSJ91 LNU-LTC

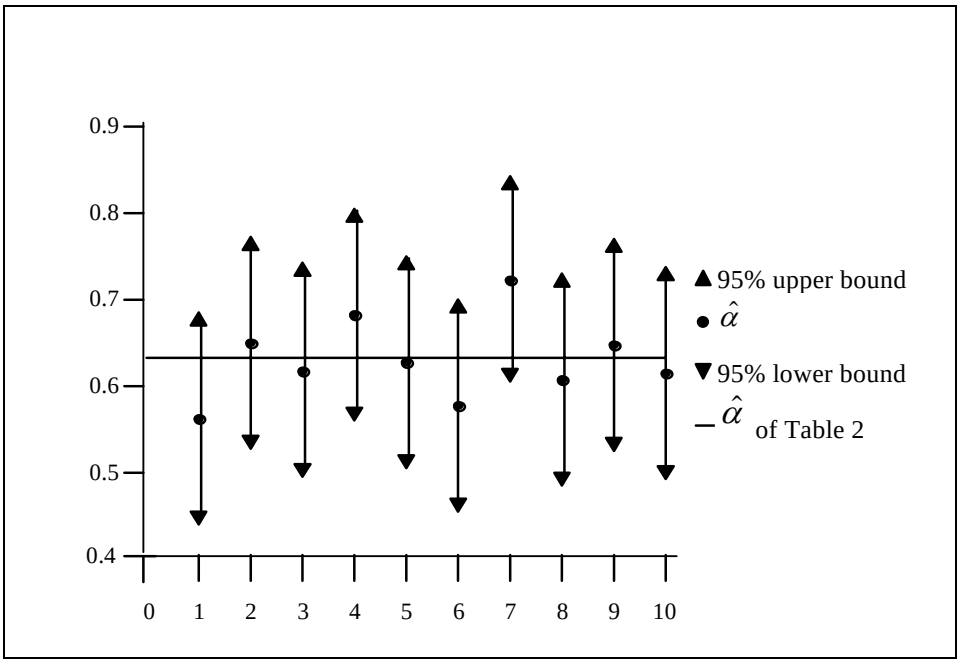

Figure 3: 95% confidence interval estimate for $\hat{\alpha}$ (10-fold cross-validation) for WSJ91 LNU-LTC

The evaluation of the 10-fold cross-validation indicates an average precision at eleven recall points of 18.40. Thus, there is no real difference in term of precision between the 10-fold cross-validation and the retrospective evaluation shown in Table 8.

5. CONCLUSION

This paper presents a new approach to combine multiple sources of evidence in database merging. This model based on logistic regression takes place in the case where only ranks are available as a key to merge different ranked list obtained by various retrieval schemes. Compared to other database merging strategies such as round-robin, raw-score and normalized raw-score, our model gives a significantly better retrieval effectiveness.

The retrospective evaluation and the ten-fold cross-validation return similar retrieval effectiveness, tending to prove that logistic regression coefficients are trustful and stable, when computed by the classical retrospective approach.

An open question to address by future work is to know how the system may take into account of incremental learning, beginning with no prior knowledge about the relative performance of the different information servers.

Finally, in this study, we never take known relevance documents or pseudo-relevance information into account (Buckley *et al.*, 1996 October) in order to improve retrieval effectiveness. Although we do not reject this attractive proposition, our objective is to evaluate the effectiveness of the initial search. Relevance feedback can therefore be used after this first search in order to enhance the retrieval performance.

Acknowledgments The authors would like to thank C. Buckley from SABIR Research, for giving us the opportunity to use the SMART system, without which this study could not have been conducted. This research was supported by the SNSF (Swiss National Science Foundation) under grants 20-43'217.95 and 20-50'78.97.

REFERENCES

Bookstein, A., O'Neil, E., Dillon, M., & Stephens, D. (1992). Applications of loglinear models for informetric phenomena. *Information Processing & Management*, 28(1), 75-88.

Buckley, C., Singhal, A., Mitra, M., & Salton, G. (1996, October). *New retrieval approaches using SMART*. Proceedings of the TREC'4, Gaithersburg, MD, NIST publication 500-236, 25-48.

Callan, J. P., Lu, Z., & Croft, W. B. (1995, June). *Searching distributed collections with inference networks*. Proceedings of the 18th International Conference of the ACM-SIGIR'95, Seattle, WA, 21-28.

Cox, D. R., & Snell, E. J. (1989). *Analysis of binary data*. 2nd ed., London, UK: Chapman and Hall.

Dreilinger, D. & Howe, A. E. (1997). Experiences with selecting search engines using metasearch. *ACM Transactions on Information Systems*, 15(3), 195-222.

Dumais, S. T. (1994, March). *Latent semantic indexing (LSI) and TREC-2*. Proceedings of TREC'2, Gaithersburg, MD, NIST Publication #500-215, 105-115.

Fuhr, N., & Pfeifer, U. (1994). *Probabilistic information retrieval as a combination of abstraction, inductive learning, and probabilistic assumptions*. ACM Transactions on Information Systems, 12(1), 92-115.

Gey, F. C. (1994, July). *Inferring probability of relevance using the method of logistic regression*. Proceedings of the 17th International Conference of the ACM-SIGIR'94, Dublin, Ireland, 222-231.

Hosmer, D., & Lemeshow, S. (1989). *Applied logistic regression*. New-York, NY: John Wiley & Sons.

Kwok, K. L., Grunfeld L., & Lewis, D. D. (1995, April). *TREC-3 ad-hoc, routing retrieval and thresholding experiments using PIRCS.* Proceedings of TREC'3, Gaithersburg, MD, NIST Publication #500-225, 247-255.

Moffat, A., & Zobel, J. (1995, April). *Information retrieval systems for large document collections.* Proceedings of TREC'3, Gaithersburg, MD, NIST Publication #500-225, 85-93.

Robertson, S. E., Walker, S., & Hancock-Beaulieu, M. M. (1995). Large test collection experiments on an operational, interactive system: OKAPI at TREC. *Information Processing & Management*, 31(3), 345-360.

Sparck Jones, K., & Bates, R. G. (1977). *Research on automatic indexing 1974-1976*. Technical Report, Computer Laboratory, University of Cambridge (UK).

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B,* 36(2),111-147.

Voorhees, E. M., Gupta, N. K., & Johnson-Laird, B. (1995, April). *The collection fusion problem*. Proceedings of TREC'3, Gaithersburg, MD, NIST Publication #500-225, 95-104.

Voorhees, E. M., Gupta, N. K., & Johnson-Laird, B. (1995, July). *Learning collection fusion strategies*. Proceedings of the 18th International Conference of the ACM-SIGIR'95, Seattle, WA, 172-179.

Voorhees, E. M. (1996, October). *Siemens TREC-4 report: Further experiments with database merging.* Proceedings of TREC'4, Gaithersburg, MD, NIST Publication 500-236, 121-130.

Appendix 1: Collection Statistics

Collection	WSJ90	WSJ91	WSJ92	WSJ
Size	73 Mb	146 Mb	35 Mb	254 Mb
# of documents	21,705	42,652	10,163	74,520
# of topics	170	171	140	186
# relevant doc.	2,074	3,563	831	6,468
Based on indexing terms				
mean	142.8	138.8	134.2	139.37
standard error	123.0	119.4	117.0	120.86
median	90	88	83	88
maximum	1532	1738	1425	1738
minimum	3	5	5	3
Based on tf_{ij}				
mean	246.6	239.6	232.2	240.64
standard error	236.5	230.4	228.0	231.90
median	145	141	134	141
maximum	4924	5791	4343	5791
minimum	3	5	5	3

Table A.1: Various statistics associated with each collection

In the WSJ90 collection, the queries {65, 80, 81, 101, 102, 104, 139, 140, 146, 201, 213, 214, 221, 225, 227, 232, 234, 236, 252, 260, 263, 272, 277, 278, 279, 280, 281, 292, 295, 296} do not have any relevant document, while for the WSJ91 corpus, the topics {66, 69, 80, 81, 103, 104, 105, 121, 141, 144, 146, 201, 210, 213, 214, 220, 231, 232, 236, 253, 260, 262, 263, 267, 276, 278, 279, 281, 296} are removed from the evaluation. For the WSJ92 collection, the queries {54, 58, 63, 64, 69, 70, 77, 80, 81, 91, 93, 99, 101, 103, 104, 105, 121, 126, 127, 129, 130, 131, 133, 139, 140, 144, 146, 201, 210, 213, 214, 217, 220, 229, 232, 236, 238, 239, 252, 253, 256, 258, 262, 263, 265, 266, 267, 268, 271, 275, 276, 278, 279, 280, 281, 288, 293, 295, 296, 300} are removed for the same reason. Finally, for the whole WSJ collection, the queries {80, 81, 104, 146, 201, 213, 214, 232, 236, 263, 278, 279, 281, 296} can be ignored.

Appendix 2: Weighting Schemes

In this paper, the indexing procedure done by the SMART system is fully automatic and based on a single term only. The representation of each topic is based on the content of its Descriptive (<desc>) section or its Descriptive and Narrative (<narr>) sections. For each document, the Text (<text>) section as well as the Subtitle (<st>), Headline (<hl>), and Summary (<lp>) sections were used to build the document surrogate. All other subsections were removed, and, in particular, the title and the concept section of each topic (see Table A.2).

Collection	Section
WSJ	<desc>, <text>, <st>, <hl>, <lp>
Query	<desc> or <desc> & <narr>

Table A.2: Selected sections used to represent documents and queries

To assign an indexing weight w_{ij} reflecting the importance of each single-term T_j, j = 1, 2, ..., t, in a document D_i, we may use one of the equations shown in Table A.3. In this table, tf_{ij} depicts the frequency of the term T_j in the document D_i (or in the request), n represents the number of documents D_i in the collection, df_j the number of documents in which T_j occurs, and idf_j the inverse document frequency ($\log [n/df_j]$). Moreover, the document length of D_i (the number of indexing terms) is noted by nt_i, and mean($nt_.$) indicates the average of the document length. The constant c is fixed to 0.2 and C is computed as $0.5 + 1.5 \cdot [nt_i / \text{mean}(nt_.)]$. Finally, the computation of the retrieval status value is based on the inner product.

BNN	$w_{ij} = 1$	NNN	$w_{ij} = tf_{ij}$
ANN	$w_{ij} = 0.5 + 0.5 \cdot \frac{tf_{ij}}{\max tf_{i.}}$	ATN	$w_{ij} = \left[0.5 + 0.5 \cdot \frac{tf_{ij}}{\max tf_{i.}}\right] \cdot idf_j$
NPN	$w_{ij} = \left[0.5 + 0.5 \cdot \frac{tf_{ij}}{\max tf_{i.}}\right] \cdot idf_j$	LTN	$w_{ij} = \left[\log(tf_{ij}) + 1\right] \cdot idf_j$
HTN	$w_{ij} = \frac{\log(tf_{ij} + 1) \cdot idf_j}{\log(nt_i)}$	OKAPI	$w_{ij} = \sum_{k=1}^{t} \frac{2 \cdot tf_{ik}}{C + tf_{ik}}$
LNC	$w_{ij} = \frac{\log(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} (\log(tf_{ik}) + 1)^2}}$	NTC	$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} (tf_{ik} \cdot idf_k)^2}}$

ANC	$w_{ij} = \frac{0.5 + 0.5 \cdot \frac{tf_{ij}}{\max tf_{i.}}}{\sqrt{\sum_{k=1}^{t} \left[0.5 + 0.5 \cdot \frac{tf_{ik}}{\max tf_{i.}} \right]^2}}$	LTC	$w_{ij} = \frac{\left[\log(tf_{ij}) + 1\right] \cdot idf_j}{\sqrt{\sum_{k=1}^{t} \left(\left[\log(tf_{ik}) + 1\right] \cdot idf_k \right)^2}}$
LNU	$w_{ij} = \frac{\frac{1 + \log(tf_{ij})}{1 + \log(mean(tf_{i.}))}}{(1 - c) \cdot mean(nt_{.}) + c \cdot nt_i}$		

Table A.3: Weighting schemes

Appendix 3: Retrieval Results Based on Longer Queries

collection model	Precision (% change)					
	WSJ baseline	WSJ90- WSJ91- WSJ92				
		round-robin		raw-score		raw-score & optimal selection	
OKAPI - NPN	28.06	23.98	(-14.54)	27.87	(-0.68)	29.25	(+4.24)
LNU - LTC	26.25	23.36	(-11.01)	26.03	(-0.84)	27.83	(+6.02)
LTN - NTC	24.03	21.23	(-11.65)	23.67	(-1.50)	25.28	(+5.20)
ATN - NTC	24.00	20.85	(-13.13)	23.50	(-2.08)	25.37	(+5.71)
LNC - LTC	23.86	20.14	(-15.59)	23.73	(-0.54)	24.89	(+4.32)
LTC - LTC	21.80	19.00	(-12.84)	21.76	(-0.18)	22.82	(+4.68)
ANN - NTC	21.79	18.80	(-13.72)	21.51	(-1.28)	23.22	(+6.56)
ANC - LTC	19.65	16.91	(-13.94)	19.32	(-1.68)	20.59	(+4.78)
HTN - BNN	17.86	16.13	(-9.69)	17.81	(-0.28)	19.35	(+8.34)
LNC - LNC	16.97	14.23	(-16.15)	17.05	(+0.47)	17.95	(+5.78)
ANN - ANN	10.66	9.10	(-14.63)	10.73	(+0.66)	11.24	(+5.44)
NNN - NNN	6.63	5.55	(-16.29)	6.63	(0.00)	6.85	(+3.32)
BNN - BNN	5.18	5.46	(+5.41)	5.18	(0.00)	6.07	(+17.18)

Table A.4: Evaluation of database merging strategies (Topic = <desc> & <narr>)

	Precision (% change)		
Collection	WSJ90	WSJ91	WSJ92
Model	OKAPI	LNU - LTC	LNC - LTC
	132 queries	131 queries	112 queries
Average precision	33.45	28.67	30.73
# of relevant documents	1758	2961	685
# of relevant doc. retrieved	1686	2604	660
RSV min.	1.151	0.002	0.0137
RSV max.	184.117	0.0327	0.4455
RSV mean	16.926	0.0082	0.0590
RSV standard error	12.367	0.0030	0.0267
Database merging	WSJ 141 queries		
Isolated DB merging:			
Round-robin	23.27		
Logistic $\ln(\text{RANK}(D_i))$	25.52 (+9.67)		
Integrated DB merging:			
Raw-score merging	11.87 (-48.99)		
Norm. raw-score merging	22.09 (-5.07)		

Table A.5: Evaluation of database merging strategies (Topic = <desc> & <narr>)