More Reflections on “Aboutness” TREC-2001 Evaluation Experiments at Justsystem

Sumio FUJITA
JUSTSYSTEM Corporation
Brains Park, Tokushima, 771-0189 JAPAN
+81-88-666-1000
Sumio_Fujita@justsystem.co.jp

ABSTRACT

The TREC-2001 Web track evaluation experiments at the Justsystem site are described with a focus on the “aboutness” based approach in text retrieval.

In the web ad hoc task, our TREC-9 approach is adopted again, combining both pseudo-relevance feedback and reference database feedback but the setting is calibrated for an early precision preferred search.

For the entry page finding task, we combined techniques such as search against partitioned collection with result fusion, and attribute-value basis re-ranking.

As post-submission experiments, distributed retrieval against WT10G is examined and two different database partitioning and three database selection algorithms are combined and evaluated.

Keywords

Aboutness, pseudo-relevance feedback, reference database, fusion, attribute-value basis re-ranking, distributed retrieval, collection partitioning, database selection.

1. INTRODUCTION

When a web page gives information to readers, readers have already understood what the information is about. Information can be about anything, but should be about something in order to be “information”.

A subject concept comprehended by explicit/implicit pacts between authors and readers is the main instance of the objective position of such “about” phrases.

In the case of artistic writings, they do not necessarily give any information but give some emotional feelings.

Even an informative document does not necessarily regard a subject concept. A curriculum vitae, for example, gives some information about someone but does not regard any subject concept. Such functional documents work as information carriers according to the complex social/institutional protocols. A curriculum vitae gives information about the professional history of someone. The problem of information access here is split into 1) whose curriculum vitae it is and 2) if it is a curriculum vitae or not. A curriculum vitae of someone can be compared with that of someone else’s but also with the medical examination report of this person. Thus information is located in the lattice of syntagmatic and paradigmatic relations of semantics.

Information access problems against entity topics are in general the identification of the type and the entity in question, given the source of information as well as information needs.

In the topic relevance search, aboutness is comprehended as representation of subject concepts, while in the entry page finding task, aboutness is split into “entriness”(entry page or not) and entity correctness (which entity it is about?). Neither “entriness” nor entity correctness can be processed as the bag of word representation.

2. SYSTEM DESCRIPTION

For the TREC-2001 Web track experiments, we utilized the engine of Justsystem ConceptBase Search™ version 2.0 as the base system.

A dual Pentium III™ server (670MHz) running Windows NT™ server 4.0 with 1024MB memory and 136GB hard disk was used for experiments.

The document collections are indexed wholly automatically, and converted to inverted index files of terms.

2.1 Term Extraction

In order to compose possible noun phrases, queries and documents in target databases are analyzed by the same module that decomposes an input text stream into a word stream and parses it using simple linguistic rules.

Extracted units are single word nouns as well as simple linguistic noun phrases that consist of a sequence of nouns or nouns preceded by adjectives.

2.2 Vector Space Retrieval

Each document is represented as a vector of weighted terms by tf*idf in inverted index files and the query is converted in similar ways.

Similarity between vectors representing a query and documents are computed using the dot-product measure, and documents are ranked according to decreasing order of RSV.

OKAPI BM25 function is utilized as the TF part of weighting function [7] so that the retrieval process can be considered as probabilistic ranking.

2.3 Passage Retrieval

Since some pages are extremely long in the wt2g data set, we became aware that using passages rather than whole pages as the indexing unit is appropriate for the sake of retrieval effectiveness.

Passage delimiting is done such that each passage becomes a similar length rather than looking for thematic/discourse boundaries.

2.4 Phrasal Indexing and Weighting

Our approach consists of utilizing noun phrases extracted by linguistic processing as supplementary indexing terms in addition to single word terms contained in phrases. Phrases and constituent single terms are treated in the same way, both as independent terms, where the frequency of each term is counted independently based on its occurrence.

2.5 Pseudo-Relevance Feedback and Reference Database Feedback

Automatic feedback strategy using pseudo-relevant documents was adopted for automatic query expansion.

The system submits the first query generated automatically from topic descriptions against the target or reference databases, and considers the top n documents from the ranked list as relevant.

The term selection module extracts salient terms from these pseudo-relevant documents and adds them to the query vector.

Then the expanded query vector is submitted against the target databases again and the final relevance ranking is obtained.

The whole retrieval procedure is as follows:

1) Automatic initial query construction from the topic description
2) 1st pilot search submitted against the reference database
3) Term extraction from pseudo-relevant documents and feedback
4) 2nd pilot search submitted against the target database
5) Term extraction from pseudo-relevant documents and feedback
6) Final search to obtain the final results

2.6 Term Selection

Each term in the example documents is scored by some term frequency and document frequency based heuristics measures described in [4].

The terms thus scored are sorted in decreasing order of each score and cut off at a threshold determined empirically.

In effect, the following parameters in feedback procedures should be decided:

1) How many documents to be used for feedback?

2) Where to cut off ranked terms?

3) How to weight these additional terms?

These parameters are carefully adjusted using TREC-9 queries (topic 451-500), wt10g data set and the relevance judgement file provided by NIST. Parameter sets for official runs are calibrated so that the early precision rather than average precision is maximized.

2.7 Spell Variation

When the system finds non stop-word terms from the “title” field text of topic description, it is clear that no document is returned. In such a case, the initial queries are expanded automatically by generated spell variations.

The procedure consists of looking for similar words in the word lists extracted from the database. Spelling similarity is measured by a combination of uni-gram, bi-gram and tri-gram matching scores.

This query expansion was adopted originally for the TREC-9 Web track runs where the “title” field contained some spell errors.

2.8 Another source of “aboutness”: Anchor Text of Hyperlinks

When we are asking what a page is talking about, sometimes anchor texts (or link texts, the texts on which a hyperlink is set) indicate an exact and very short answer.

The anchor text is typically an explanation or denotation of the page that it is linked to. Some commercial-based search engines utilize such information for advanced searches [1][2]. We treat anchor texts literally as the part of the linked document.

In total, 6,077,878 anchor texts are added to 1,173,189 linked pages out of 1,692,096 pages in the wt10g data set. So 69% of document pages in the data set are attributed to anchor text information on top of the original page information.

2.9 Link Structure Analysis

There seems to be a misunderstanding about the usage of pagerank[2] like popularity-based ranking that utilizes indirect-link information propagating rank values through hyper-link networks.

Such a ranking would not help the information seeking activities of individuals unless the individuals' information needs are strongly correlated with the popularity or the collection is heavily polluted by spam pages. The situation in navigation-oriented search seems to be the same as in the subject-oriented search. In order to show the effectiveness of popularity based ranking, information needs should be arranged according to the popularity.

Instead of the popularity-based ranking, we apply adequate link analysis according to the nature of the information seeking tasks behind the evaluation model.

2.10 Attribute-Value Basis Re-ranking

Our approach to the entry page finding task consists of combining the scoring results from different analysis procedures of pages. This is intended to rank the pages according to the following aspects:

"Entriness": the likelihood that the page is the entry point of a site.

Entity correctness: the likelihood that the page is about the entity indicated by the information need.

The following four types of analyses are processed:

-*Bag of words analysis*

This is mainly intended to gather candidate pages to be examined precisely hereafter.

The following three analyses are intended for rating both "entriness" and entity correctness.

-*Link analysis*

This examines the number of inter-server linked, inner-server linked and inner—server linker to rate "entriness" of the page.

-*URL analysis*

This examines URL form, length and names to rate both "entriness" and entity correctness.

-*Text analysis*

This examines title, inter/inner-server anchor texts and other page extracts to rate mainly entity correctness but also "entriness" by scored pattern matching.

Run tag	Index	RefTerms	Avg. Prec	R-Prec
jscbtawtl1	N	Strong	0.1890	0.2020
jscbtawtl2	N	Weak	0.1954	0.2150
jscbtawtl3	NVA	Strong	0.2003	0.2226
jscbtawtl4	NVA	Weak	0.2060	0.2308

Table 1: Performance of official runs

Through the experiments, we confirmed our expectation that only a small portion of each page is enough to be indexed for the entry page finding task. In fact, only 500 bytes of plain text including the title, the URL, anchor texts and beginning part of the page are indexed in view of the bag of word analysis.

3. WEB AD HOC EXPERIMENTS

We submitted four title-only automatic runs as follows:

jscbtawtl1: title only, link run with noun phrase indexing, more weight on reference terms

jscbtawtl2: title only, link run with noun phrase indexing, less weight on reference terms

jscbtawtl3: title only, link run with noun phrase, adjective and verb indexing, more weight on reference terms

jscbtawtl4: title only, link run with noun phrase, adjective

Run tag	words avg.	words min	words max	phrase avg.	phrase min	phrase max
jscbt9wcs1 Initial	2.1	0	5	0.7	0	3
jscbt9wcs1 Final	44.1	0	138	31.0	0	176
jscbtawtl1-2 Initial	2.38	0	5	0.56	0	2
jscbtawtl1-2 Final	80.4	0	184	34.86	0	114
jscbtawtl3-4 Initial	2.72	0	5	0.60	0	3
jscbtawtl3-4 Final	84.86	0	160	37.76	0	114

Table 2: Length of queries measured by number of single word terms and phrasal terms (without spell variation expansion)

and verb indexing, less weight on reference terms

As for the link usage, we adopted the "anchor text" of the hyperlink information as we did in TREC-9 [5].

Table 1 shows the performance of official runs and Table 2 shows the length of the queries utilized in each run.

Initial queries are very short (in average, 2.38-2.72 single word terms and 0.56-0.60 phrasal terms, maximum 5 single word terms and 3 phrasal terms , minimum 0 single word terms and 0 phrasal terms) and they do not contain enough terms.

Table 3 shows the performance comparison combining pseudo-relevance feedback and reference database feedback as well as with/without phrasal terms on the basis of jscbtawtl2 and jscbtawtl4 settings.

The automatic feedback procedure contributes to 16.1% to 18.3 % of consistent improvements in average precision in all cases.

The final queries contain 80.4-84.86 single word terms and 34.86-37.76 phrasal terms in average (maximum 184 single word terms and 114 phrasal terms, minimum 0 single word terms and 0 phrasal terms). Note that we added many more terms in the final queries than we did in TREC-9.

The improvement gained by the combination of pseudo-relevance feedback and reference database feedback is 21.4% for N index run and 20.9% for NAV index run. It is natural that N index runs where initial queries are shorter gained more from the feedback process. The improvement gain from combined feedback is larger than our TREC-9 experiments(17% in link runs). This is mainly caused by our approach to have taken more terms from feedback and promote some terms to the foreground.

In TREC-9, we explained our approach utilizing "foreground vs background" metaphor. In other words, foreground terms denote directly the subject concept of the information need while background terms connote the subject topic. If the weighting balance is changed in the query, the information need is also shifted.

In order to promote some terms to the foreground, we adopted a simple voting from two sources of feedback; one is the target collection and the other is the reference collection.

Doing such calibration, we intended to make the runs be early precision preferred rather than MAP preferred as our TREC-9 runs. Despite this, the official result showed that our system was still MAP and recall preferred in comparison with other systems.

Supplemental phrasal indexing runs perform better in average precision as well as in R-precision both with/without pseudo-relevance feedback and with/without reference database feedback. The situation observed here is consistent with our experience in TREC-9 web track experiments, but in this case, the effectiveness of phrasal indexing seems to be more stable.

Run description	Ref	PFB	AvgPrec	R-Prec
N index / SW + phrases (jscbtawtl2)	Yes	Yes	0.1954	0.2150
N index / SW + phrases	Yes	No	0.1730	0.2013
N index / SW + phrases	No	Yes	0.1903	0.2074
N index / SW + phrases	No	No	0.1609	0.1898
N index / Single words only	Yes	Yes	0.1854	0.2051
N index / Single words only	Yes	No	0.1685	0.1915
N index / Single words only	No	Yes	0.1837	0.2078
N index / Single words only	No	No	0.1537	0.1841
NVA index / SW + phrases (jscbtawtl4)	Yes	Yes	0.2060	0.2308
NVA index / SW + phrases	Yes	No	0.1824	0.2106
NVA index / SW + phrases	No	Yes	0.1979	0.2417
NVA index / SW + phrases	No	No	0.1704	0.2149
NVA index / Single words only	Yes	Yes	0.1997	0.2357
NVA index / Single words only	Yes	No	0.1745	0.2083
NVA index / Single words only	No	Yes	0.1894	0.2217
NVA index / Single words only	No	No	0.1641	0.2062

Table 3: Performance comparison (Title only, jscbtawtl2-4 parameter set)

4. ENTRY PAGE FINDING EXPERIMENTS

As table 4 shows, we submitted four entry page search runs: jscbtawep1, jscbtawep2, jscbtawep3 and jscbtawep4.

These four runs adopt essentially the same configuration but differ in two parameters of final scoring.

The full phrase match bonus weights and the bag of word analysis weights are changed as shown in table 4.

Run tag	full-match	bow wght	MRR	Top10 %	NF%
jscbtawep1	moder	low	0.754	83.4	9.0
jscbtawep2	moder	med	0.769	83.4	9.0
jscbtawep3	high	med	0.752	83.4	9.0
jscbtawep4	moder	high	0.746	83.4	8.3

Table 4: Performance of official runs of the Entry Page Finding Task

4.1 The Server Database and the Whole Database

The server database contains 11680 server pages in the wt10g collection.

In fact, this covers 78% of 100 pre-test queries, i.e., this database contains at least one answer page against each of 78 queries out of 100 queries. It also covers 66.2% of 145 test queries.

Ten pages from the server database and 1000 pages from the whole database are merged in the manner that the 10 pages from the server database come to the top of the rank.

Thus far, we applied normal retrieval processing, utilizing bag of word queries.

MRR of the 10 page ranked lists against the server database accounts for 0.6409 and that of the 1000 page ranked lists against the whole database accounts for 0.4176.

Merging them together makes MRR of 0.6462.

4.2 Attribute-value Basis Re-ranking

Thus obtained ranked page lists of 1010 pages are cut off at the top 200 pages and re-ranked by the attribute-value basis analysis modules.

4.3 Basic Text Matching and Scoring in view of "Entity Correctness"

Text fields are scored by the matching procedure that accumulates each word matching point and adjacency point.

Such analysis is much more powerful than bag of word analysis and is equivalent to the full sub-phrase indexing against all long phrases.

4.4 Augmented Text Matching and Scoring in view of "Entity Correctness"

It is likely that the URL text contains the entity name as the part of the server name or the directory names.

But it is sometimes the case that the constituent words are agglutinated. The matching is augmented in order to treat such agglutinated names.

4.5 Supplemented Text Matching and Scoring in view of "Entriness"

Some field are matched against pre-coded patterns as follows:

The "InterServerAnchorText" field is intended to be matched with anchor texts like "go to the homepage of XXX".

The "InnerServerAnchorText" field is expected to be matched with "back to the home(of XXX)".

The "Title" field is something like "Welcome to the homepage of XXX".

4.6 Link Analysis

The number of interserver linked, normalized by maximum number of interserver linked, among the candidate pages simply indicates the "entriness" of the page.

The entry page is also very likely to have at least one linker to the inner server pages unless his/her/their/its web site consists of only one page.

4.7 Score Composition

The final score is computed as the sum of the weighted scores from each analysis. Each analysis weight is calibrated by the 100 pre-test topics.

$$\begin{aligned} PageScore = {} & \mathrm{w}_1 S(BOW) + \mathrm{w}_2 S(URLType) + \mathrm{w}_3 S(URLText) \\ & + \mathrm{w}_4 S(InterServerAnchor) + \mathrm{w}_5 S(InnerServerAnchor) + \mathrm{w}_6 S(Title) \\ & + \mathrm{w}_7 S(L\arg eFonts) + \mathrm{w}_8 S(FullMatch) + \mathrm{w}_9 S(InterServerLinked) \\ & + \mathrm{w}_{10} S(InnerServerLin\ker) \end{aligned} \quad (1)$$

The full phrase match bonus is added only when all the constituent words of the entity name matches and prevents inclining to partial matching in many fields rather than full matching in one field.

After such re-ranking processes, the final results of MRR 0.746 to 0.769 are obtained.

5. DISTRIBUTED RETRIEVAL AGAINST WT10g

In view of the trade-offs between efficiency and effectiveness, there might be two possibilities for large collection retrieval.

1)Centralized Multi-stage Search

All the units are indexed in a system and first some important parts of each document like title and large font text parts are searched. If the user is not satisfied with the first results or he/she requests an exhaustive search through the collection, the second search looks through all the text part of the documents.

2)Distributed Selective Search

The collection is partitioned by some criteria like publication date order, author’s name order, original document location or content basis classification, etc., and stored into separate databases. The search process consists of 1) selecting databases to be searched, 2) distributed search in all the databases selected, 3)fusion of the result lists from the selected databases, and 4) if the user requests it, the search result lists from all the databases are presented.

Many studies on distributed retrieval have been done by researchers of the IR society, but so far, web commercial search engines tend to be implemented as centralized search systems. The problem in distributed IR is the database selection; failing to properly select the target databases causes severe degradation in effectiveness. However, some studies claim that the effectiveness of a distributed search is even better than a centralized search when an adequate selection algorithm is applied[6].

5.1 Collection Partitioning

WT10G collection is partitioned in two ways.

5.1.1 104 Pre-defined directory Partitioning

Each of 104 directories(WTX001 – WTX104) of distribution CD-R is utilized as a single database.

Each database is almost the same size. Each database contains about 10,000 to 20,000 pages and these sizes account for 60 to 80MB in text file.

5.1.2 326 Category Partitioning

Content basis classification has been done using 326 categories derived from the Yahoo US categories.

The highest two level categories of Yahoo US[8] directories were adopted and Web pages linked from them were downloaded in March 2001. These pages (142MB, 19048pages) are stored in the classifier database and each page in WT10G is submitted as a query against this classifier database. Scores of the best 15 ranked (Yahoo linked) pages are voted for the category from which the (yahoo linked) page is linked. Thus, for each page in WT10G, the category is decided and the WT10G pages are stored in partitioned databases.

Average	5190.479
Standard Error	836.2663
Median	492
Standard deviation	15099.18
Distribution	2.28E+08
Kurt	49.18961
Skew	6.136924
Range	162594
Min	1
Max	162595
Sum	1692096
Number of Samples	326

Table 5: Basic Statistics of number of pages in each database of 326 category partitioning

In this case, the database size is diverse, ranging from as small as only one page to the maximum 162595 pages (9.6% of the whole collection). Basic statistics measures of the number of pages in each database are shown in table 5.

5.2 Database Selection

The following three algorithms for selecting databases are examined.

5.2.1 CORI

The formula proposed in [3] is adopted.

$$\mathrm{T} = \mathrm{d\_t} + (1 - \mathrm{d\_t})\frac{\mathrm{df}}{\mathrm{df} + \mathrm{K}} \quad (2.1)$$

$$\mathrm{K} = \mathrm{k}((1 - \mathrm{b}) + \mathrm{b}\frac{\mathrm{cw}}{\mathrm{mean(cw)}} \quad (2.2)$$

$$I = \frac{\log(\frac{|C| + 0.5}{CF})}{\log(|C| + 1.0)} \quad (2.3)$$

$$\mathrm{p(t \mid c)} = \mathrm{d\_b} + (1 - \mathrm{d\_b}) * \mathrm{T} * \mathrm{I} \quad (2.4)$$

d_t, d_b : 0.4
cw : number of words in a database
df : document frequency of the term t in the collection c
CF : number of collections where the term t appears
| C | : number of collections

p(t|c) is the weight of the term t against the collection c and the each database is ranked by the sum of this weight over all query terms. We utilized the setting of k=200 and b=0.8.

*5.2.2 Simple DF*ICF*

This is simplest version of DF*ICF, an essential part of the CORI method.

$$\mathrm{DF} = \mathrm{d\_t} + (1 - \mathrm{d\_t})\frac{\mathrm{df}}{\mathrm{MAX\_df}} \quad (3.1)$$

$$ICF = \log(\frac{|C|}{CF}) \quad (3.2)$$

$$\mathrm{p(t \mid c)} = \mathrm{DF} * \mathrm{ICF} \quad (3.3)$$

d_t : 0.5
df : document frequency of the term t in the collection c
MAX_df : maximum df of the term through collections
CF : number of collections where the term t appears
| C | : number of collections

*5.2.3 DF*AVG-IDF*

DF*AVG-IDF is similar to DF*ICF but instead of ICF, average IDF of the term over all the databases is utilized.

5.3 Experiments

We compared two collection partitioning and three database selection methods. Figure 1 in Appendix A. shows a comparison of the combination of two partitioning and three database selection methods by using MAP, R-precision, precision at 20 docs (PREC@20) and the number of relevant documents retrieved (REL_RET). For each of the six combinations, we examined 10 runs, decreasing the number of databases to be searched from 100% down to 10% by 10% of the whole collection. For each of topic 501 to 550, the databases were selected from the top n % of the ranked database list utilizing one out of three methods. No feedback is applied in these experiments. Once the databases to be searched are decided, statistics from each database selected are gathered so that a centralized search against the whole selected database is simulated. Consequently, the problem of result fusion is excluded in these experiments.

Because of the essential similarity of the method, CORI and simple DF*ICF perform very similarly even though CORI seems to perform better in MAP and REL_RET.

After examining each database selected, we noticed that CORI tends to select larger databases than other methods. In fact, when selecting 10% databases, CORI searched 39% of the pages while TF*ICF searched 20% of pages (See Figure 2 in Appendix A.).

Content basis partitioned databases perform clearly better, especially when the portion of the collection to be searched is reduced. The most notable thing is that using a combination of content basis partitioned databases and CORI or DF*ICF, the early precision(PREC@20) is even getting better when reducing the number of databases.

DF*ICF especially marked the best PREC@20 when searching only 41% of pages out of the whole collection(20% by database numbers).

6. CONCLUSIONS

TREC-2001 Web track evaluation experiments at Justsystem group are described.

The following conclusions are drawn from these experiments:

1)We modified our TREC-9 approach, i.e., longer vectors with background down-weighting and promoting some terms to the foreground, seem to perform well.

2)A three stage approach, i.e., bag of word analyses, result fusion and attribute-value basis re-ranking, is successfully applied to the entry page finding task.

3)A distributed selective search performs better than a centralized search in early precision when an adequate database selection method and collection partitioning are applied.

4)A simple DF*ICF database selection method performs as well as the CORI method.

5)A distributed selective search performs better with content basis category partitioning of the collection than (near) random partitioning.

6)Distributed selective search is possibly a good option in early precision preferred retrieval tasks against very large collections.

In future work, we will examine better partitioning methods by equalizing the number of pages in each database of content basis category partitioning.

REFERENCES

[1] Altavista: http://doc.altavista.com/adv_search/ast_ma_clickhere.html

[2] Brin, S. and Page, L. , The Anatomy of a Large-Scale Hypertextual Web Search Engine, in Proceedings of the Seventh International World Wide Web Conference, 1998, 107-118.

[3] Callan, J.P., Lu, Z. and Croft, W.B., Searching Distributed Collections With Inference Networks, in Proceedings of the 18th Annual International ACM SIGIR Conference, Seattle Washington, 1995, 21-28.

[4] Evans, D.A. and Lefferts, R.G., Grefenstette, G., Handerson, S.K., Hersh, W.R., and Archbold, A.A., CLARIT TREC Design, Experiments and Results, in Proceedings of the First Text REtrieval Conference(TREC-1), NIST Special Publication 500-207, Washington D.C., 1993, 494-501.

[5] Fujita, S. , Reflections on “Aboutness”—TREC-9 Evaluaton Experiments at Justsystem ,in the Notebook version of the Ninth Text REtrieval Conference(TREC-9), Gaithersburg MD, 2000.

[6] Powell, A.L., French, J.C., Callan, J., Connell, M., and Viles, C.L., The impact of Database Selection on Distributed Searching, in Proceedings of the 23rd Annual International ACM SIGIR Conference, Athens Greece, 2000, 232-239.

[7] Robertson, S.E., Walker S., Jones S., Hancock-Beaulieu, M.M., Gatford, M. Okapi at TREC-3, in Proceedings of the Third Text REtrieval Conference(TREC-3), NIST Special Publication 500-225, Washington D.C., 1995, 109-126.

[8] Yahoo: http://www.yahoo.com/

Appendix A. Database Selection Experiments

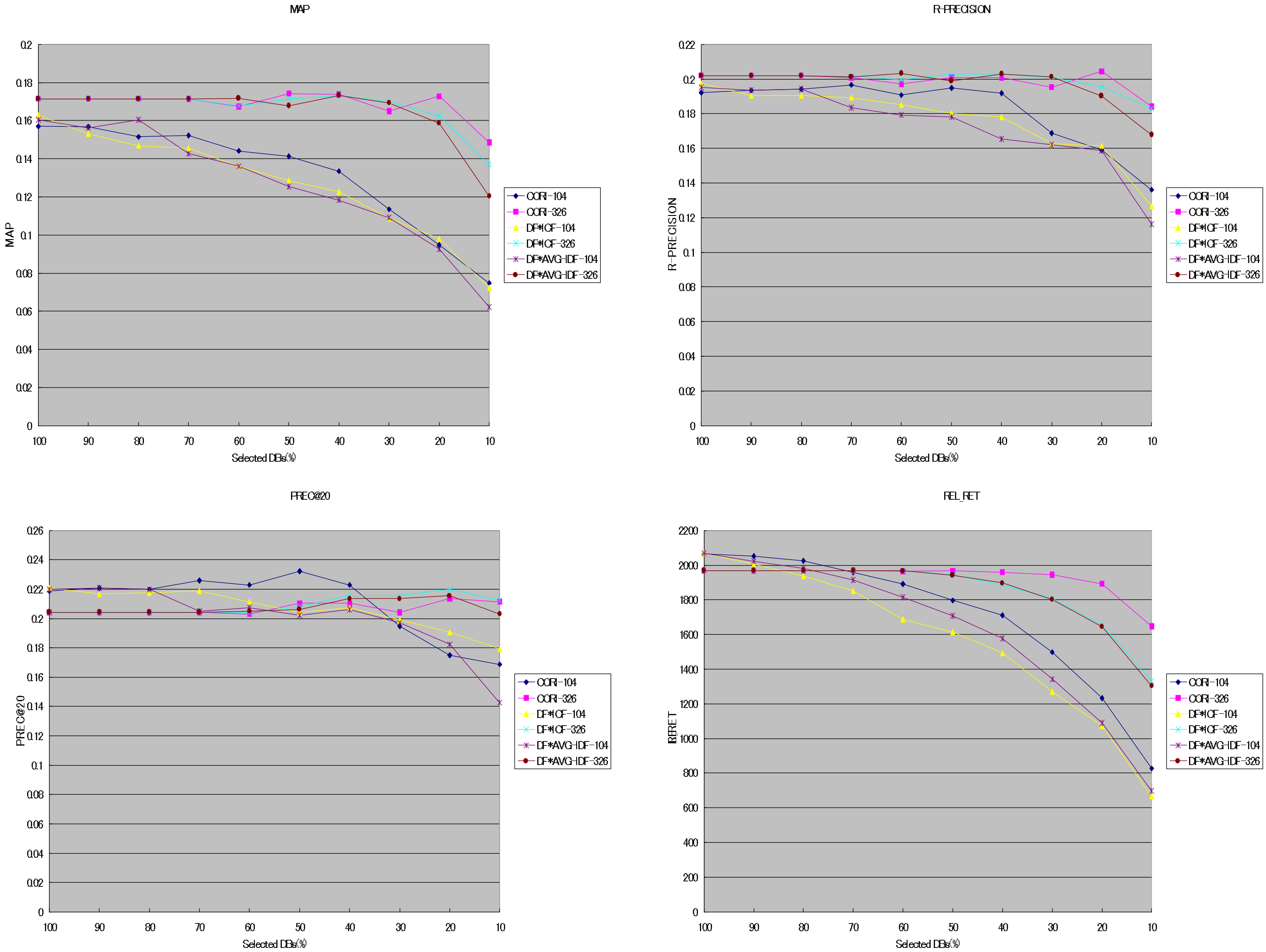

Figure1: Performance of DB Selection Runs with topic 501-550

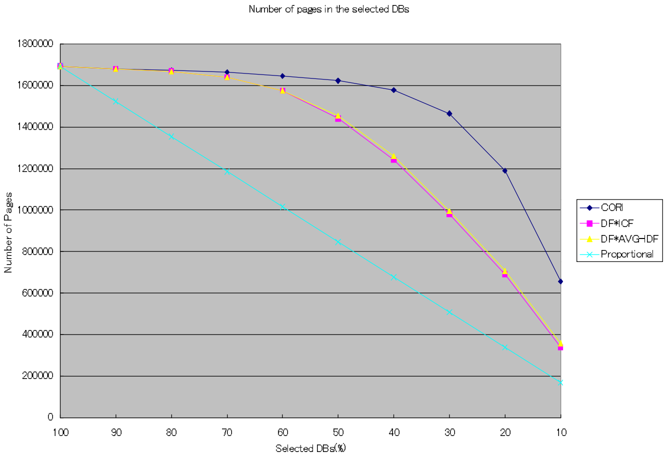

Figure2: Page numbers in the selected DBs of 326 partitioning runs