Combining the Ranked Output from Fulltext and Summary Indexes

Tetsuya Sakai
Knowledge Media Laboratory, Toshiba Corporate R&D Center, Japan
tetsuya.sakai@toshiba.co.jp

ABSTRACT

This paper demonstrates the advantage of combining a ranked output from a fulltext index and another from an index generated using generic summary texts. Our main findings are as follows: (1) The combination of a fulltext-based run and a summary-based run can significantly outperform either alone, especially when the fulltext evidence is emphasized. (2) The primary role of the summary index in the above combination is to promote some of the relevant documents already retrieved by the fulltext index, rather than to discover new relevant documents. (3) Moreover, though the summary quality can make a difference for this combination approach, even simple extracts such as 5% "lead" may give significant improvements.

1. INTRODUCTION

As discussed in [10, 11], summarization can be used in *preprocessing* or *postprocessing* of information retrieval (IR). Postprocessing involves the generation of "disposable" *query-focused* summaries either as the source of *query expansion terms* or for relevance preview by the user, and is probably more popular than preprocessing (e.g. [13, 14]). However, preprocessing, which involves the generation of "reusable" *generic* summaries as the source of *document index terms*, deserves investigation as well, since it may help improve retrieval effectiveness for some tasks, not to mention efficiency. Thus, recent work by Sakai and Sparck Jones [10, 11] suggests that this *summary indexing* approach can be useful in some IR contexts. They explored several summary-based search strategies, including the following:

Summary runs without pseudo-relevance feedback A summary index is searched with the initial queries. No *pseudo-relevance feedback* (PRF) (e.g. [9, 12]) is performed.

Summary-summary runs A summary index is used for initial search, PRF and final search.

Summary-fulltext runs A summary index is used for initial search and PRF, but the fulltext index is used for final search.

Their main conclusions were as follows:

1. For *precision-oriented search of highly relevant documents*, some summary runs without PRF are comparable to the fulltext run without PRF, and some summary-summary runs are comparable to the *fulltext-fulltext* run (i.e. traditional PRF run).
2. The *summary-fulltext* strategy may be effective for *precision-oriented search, regardless of relevance levels*. It is generally more effective than the *summary-summary* strategy.

This paper explores an alternative way to utilize generic summaries for IR, namely, to *combine* a ranked output from the fulltext index and another from a summary index, both representing the same document collection. Our motivation was that *combination of evidence* has been found to be effective for IR: previous work has shown that (i) different query/document representations and different ranking algorithms retrieve different documents; (ii) combining these methods often outperform the component methods, especially when the components are independent of each other [3]. While the summary-fulltext strategy used in the earlier tests was also a kind of combination of evidence from fulltexts and summaries, our new ranked output combination is achieved by taking a weighted average of the individual document scores, and is clearly a different approach. Thus in our new tests, the previous summary-fulltext strategy was used as a *component* in some of our combinations. Our new results are clearly better than those reported in [10, 11], in that there are many statistically significant gains over the fulltext baselines, whether measured by a precision-oriented document cutoff or by the standard TREC average precision measure. Moreover, our new combined runs are advantageous not only for retrieval of *highly* relevant documents but also for retrieval of *all* (i.e. highly or partially) relevant documents [10, 11].

The remainder of this paper is organized as follows: Section 2 distinguishes this paper from previous work. Section 3 describes our combination experiments. Section 4 provides the results and discusses their implications. Finally, Section 5 summarizes our findings.

2. COMPARISONS WITH PREVIOUS WORK

2.1 Summary Indexing

A discussion of the difference between the use of generic summaries for indexing and that of query-focused summaries after retrieval can be found in [10, 11]. The experiments reported there were more extensive than others on summary indexing (e.g. [1]) in that (1) a larger test collection was used; (2) pseudo-relevance feedback was considered; (3) three automatic *extract* types, each with different *Compression Ratios* [10, 11], and also manual *abstracts*, were examined as different summary types. This paper inherits these features, as we use the same test collection and indexes. Moreover, as mentioned in Section 1, this paper achieves better results than those in [10, 11] by combining a ranked output from a fulltext index and another from a summary index. Section 2.2 provides a brief overview on the combination of evidence for IR, and shows how we combine the document scores.

2.2 Combination of Evidence

A good review of the combination of evidence for IR is given in [3]. Combination can take place at various stages of retrieval: at one end of the spectrum, there are systems that explicitly allow different query/document representations, for example, by using structured queries that utilize nested *query operators* [3] or by using *document structure* vectors [8]. At the other end of the spectrum, we have the approach of combining the final ranked output from several separate subsystems. In the latter, although several combination methods have been proposed, it appears that simple methods, such as summing/averaging the individual document scores, work well for searching a single document collection [4, 7].

The objective of this paper is to demonstrate that the fulltext/summary ranked output combination works reasonably well, rather than to explore the best combination methods. Thus, we simply combine the document scores as follows:

$$score_{combined}(d) = c * score_{fulltext}(d) + (1-c) * score_{summary}(d) \quad (1)$$

where

$score_{fulltext}(d)$:	document score for d obtained by a fulltext-based search
$score_{summary}(d)$:	document score for d obtained by a summary-based search

Although this method does not involve any kind of *score normalization* [5, 6, 15], even an *unweighted* version of this approach (i.e. with $c = 0.5$) has been applied successfully to cross-language information retrieval, where document scores based on search request translation and those based on document translation were combined [7]. As in [15], we assume $score_{fulltext}(d)$ or $score_{summary}(d)$ is zero if d is not retrieved.

If we disregard the particular stages of retrieval in which combination of evidence takes place, the previous work most closely related to the experiments reported here is that combining a fulltext index with a *controlled vocabulary* index, and that combining fulltext evidence with *passage* evidence [2, 3]. The difference is that we are combining the fulltext evidence with that from a range of human-readable *generic summaries*, which may be manual abstracts, lead extracts or *tf-idf*-type extracts. We have also investigated the fulltext/summary combination both before and after PRF has been applied to one or both of the component searches, and have made comparisons across the combinations.

3. EXPERIMENTS

3.1 Test Collection and Indexes

The test collection used in our experiment is that used in [10, 11]. Thus it consists of the following:

- 25 search requests from early TREC "Topics";
- relevance assessments with two levels: relevant and *highly* relevant;
- 38,884 Ziff-Davis technical articles on computing.

We use a total of 10 indexes, again taken from [10, 11]:

1. the *fulltext* index f;
2. the manual *abstract* index a;
3. the *lead extract* indexes ld(5), ld(10), ld(AL) and ld(30);
4. the *tf-idf extract* indexes tf(5), tf(10), tf(AL) and tf(30).

While the manual abstracts represent a sophisticated type of summaries, the extracts are very crude types of summaries based on simple sentence selection. For each extract type (i.e. lead and *tf-idf*), four different Compression Ratios (CRs) were used: for example, ld(5) is a lead extract index with 5% CR. Furthermore, ld(AL) and tf(AL) were generated using the *Abstract-Length* extracts, which have exactly the same number of sentences as the corresponding abstracts. This is to separate the effect of CR from that of abstraction/extraction [10, 11]. Retrieval performance based on our *abstract* index can be taken as defining an upperbound for summary-based approaches, while that based on our *extract* indexes can be taken as defining a lowerbound. Thus, although we do not have a sophisticated automatic summarizer at hand, if we did have one we would expect its performance to lie somewhere between these two bounds.

3.2 Combining the Ranked Output

In our combination experiments, we take the runs described in [10, 11] as components. We follow the conventions adopted there in denoting the component runs: the runs without PRF are represented simply by the index name, e.g. f, a and ld(5). The fulltext-fulltext run is denoted by f+f, and the summary-summary runs are represented by a+a, ld+ld(5) etc., to reflect the fact that the same index is searched twice (i.e. before and after query expansion). This paper reports on the results of the following index combinations:

Combination 1 The fulltext run without PRF is combined with a summary run without PRF. This is denoted by f⊕a, f⊕ld(5), etc, and is the simplest fulltext/summary combination in our experiments.

Combination 2 The fulltext-fulltext run is combined with a summary run without PRF. This is denoted by f+f⊕a, f+f⊕ld(5), etc.

Table 1: **Combining a fulltext run without PRF and a summary run without PRF.**

(a) **AveP/ALL rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f	—	—	—	—	0.256
manual abstracts	f⊕a	—	—	**0.275**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.8)$	—	—
automatic extracts	f⊕ld	**0.266**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)$	**0.264**$^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	**0.265**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	**0.266**$^{\heartsuit\heartsuit}$ $(c = 0.8)^{\diamondsuit\diamondsuit}$	—
	f⊕tf	**0.261**$^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit\diamondsuit}$	**0.262**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	**0.270**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.8)$	**0.264**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)$	—
(b) **AveP/HIGHLY rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f	—	—	—	—	0.249
manual abstracts	f⊕a (a: 0.231)	—	—	**0.300**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.7)$	—	—
automatic extracts	f⊕ld (ld(AL): 0.186)	**0.282**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.8)$	**0.277**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	**0.303**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.7)$	**0.285**$^{\spadesuit\spadesuit\heartsuit}$ $(c = 0.8)^{\diamondsuit\diamondsuit}$	—
	f⊕tf (tf(AL): 0.198)	**0.262**$^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.8)^{\diamondsuit\diamondsuit}$	**0.261**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit\diamondsuit}$	**0.280**$^{\spadesuit\spadesuit\heartsuit}$ $(c = 0.8)^{\diamondsuit\diamondsuit}$	**0.265**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.8)^{\diamondsuit\diamondsuit}$	—
(c) **PDoc10/ALL rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f	—	—	—	—	0.360
manual abstracts	f⊕a	—	—	**0.404** $(c = 0.6)$	—	—
automatic extracts	f⊕ld	**0.396** $(c = 0.8)$	**0.384** $(c = 0.7)$	**0.400**$^{\heartsuit}$ $(c = 0.7)$	**0.376** $(c = 0.6)$	—
	f⊕tf	**0.372** $(c = 0.9)$	**0.380**$^{\heartsuit}$ $(c = 0.9)$	**0.388** $(c = 0.8)$	**0.372** $(c = 0.6)$	—
(d) **PDoc10/HIGHLY rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f	—	—	—	—	0.252
manual abstracts	f⊕a	—	—	**0.312**$^{\spadesuit}$ $(c = 0.6)$	—	—
automatic extracts	f⊕ld	**0.340**$^{\spadesuit\spadesuit\heartsuit}$ $(c = 0.6)$	**0.312**$^{\spadesuit\spadesuit}$ $(c = 0.7)$	**0.340**$^{\spadesuit\spadesuit\heartsuit}$ $(c = 0.5)$	**0.296**$^{\spadesuit\heartsuit}$ $(c = 0.6)$	—
	f⊕tf	**0.292**$^{\heartsuit}$ $(c = 0.6)$	**0.276**$^{\spadesuit}$ $(c = 0.9)$	**0.288** $(c = 0.7)$	**0.288**$^{\heartsuit}$ $(c = 0.6)$	—

Combination 3 The fulltext-fulltext run is combined with a summary-summary run. This is denoted by f+f⊕a+a, f+f⊕ld+ld(5), etc.

Some other combinations were also examined, but they were not as successful as the above three. Combining the *abstract* run without PRF and an *extract* run without PRF, and combining two *extract* runs without PRF did not give any significant gains over the fulltext baseline f. Thus, as previous work suggests [3, 15], it seems important to include at least one very good run as a component of the combination, which in our case is the fulltext run f or f+f.

Another combination that has been examined extensively (but omitted here) is that of the fulltext-fulltext run with a *summary-fulltext* run (See Section 1). Prior to the experiments, we thought this combination might outperform **Combination 3**, because many summary-fulltext runs significantly outperformed the corresponding summary-summary runs in [10, 11]. However, it turned out that this combination was clearly less successful, in that there were few significant gains over the fulltext baseline f+f. This seems to be because summary-fulltext strategies and the fulltext-fulltext strategy retrieve relatively similar document sets, as we shall see later in Section 4.7.

4. RESULTS AND DISCUSSIONS

4.1 Results Overview

Tables 1-3 summarize our results based on the aforementioned **Combinations 1-3**, respectively. As in [10, 11], retrieval performance is measured with both TREC Average Precision (AveP) and Precision at Document Cutoff 10 (PDoc10), for all relevant documents and for highly relevant documents only. Each table column represents a CR, with "AL" representing abstracts and Abstract-Length extracts (See Section 3.1). Since all runs in Table 1 use f (fulltext run *without* PRF) as the fulltext component, f is used as their fulltext baseline. Since all runs in Tables 2 and 3 use f+f (fulltext run *with* PRF) as the fulltext component, f+f is used as their fulltext baseline.

For each cell in each table, we tried different values of c, which balances the fulltext and summary evidence in Equation 1 (See Section 2.2), and the results here are the best ones in terms of statistical significance and average performance. Ideally, parameters such as c should be tuned using some separate training data, but this was not possible given our limited experimental material. We would like to repeat

Table 2: **Combining a fulltext-fulltext run and a summary run without PRF.**

(a) **AveP/ALL rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.278
manual abstracts	f+f$\oplus$a	—	—	**0.282**$^{\spadesuit\spadesuit\heartsuit}$ ($c = 0.9$)	—	—
automatic extracts	f+f$\oplus$ld	**0.283**$^{\heartsuit\heartsuit}$ ($c = 0.8$)	**0.281**$^{\heartsuit\heartsuit}$ ($c = 0.9$)$^{\diamondsuit}$	**0.282**$^{\heartsuit\heartsuit}$ ($c = 0.7$)	**0.281**$^{\heartsuit}$ ($c = 0.9$)$^{\diamondsuit}$	—
	f+f$\oplus$tf	**0.280**$^{\spadesuit\heartsuit\heartsuit}$ ($c = 0.9$)$^{\diamondsuit}$	**0.279**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ ($c = 0.9$)$^{\diamondsuit\diamondsuit}$	**0.281**$^{\spadesuit\spadesuit\heartsuit}$ ($c = 0.9$)	**0.281**$^{\spadesuit\heartsuit}$ ($c = 0.9$)	—
(b) **AveP/HIGHLY rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.276
manual abstracts	f+f$\oplus$a (a: 0.231)	—	—	**0.295**$^{\spadesuit\spadesuit}$ ($c = 0.7$)	—	—
automatic extracts	f+f$\oplus$ld (ld(AL): 0.186)	**0.287**$^{\spadesuit\heartsuit\heartsuit}$ ($c = 0.8$)	**0.281**$^{\spadesuit\heartsuit\heartsuit}$ ($c = 0.9$)$^{\diamondsuit\diamondsuit}$	**0.292**$^{\spadesuit\heartsuit}$ ($c = 0.7$)	**0.281**$^{\spadesuit\spadesuit\heartsuit}$ ($c = 0.9$)$^{\diamondsuit\diamondsuit}$	—
	f+f$\oplus$tf (tf(AL): 0.198)	**0.288**$^{\spadesuit\heartsuit\heartsuit}$ ($c = 0.7$)$^{\diamondsuit}$	**0.290**$^{\spadesuit\heartsuit\heartsuit}$ ($c = 0.7$)	**0.295**$^{\spadesuit\spadesuit\heartsuit\heartsuit}$ ($c = 0.7$)	**0.285**$^{\spadesuit\spadesuit\heartsuit}$ ($c = 0.8$)	—
(c) **PDoc10/ALL rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.360
manual abstracts	f+f$\oplus$a	—	—	**0.404**$^{\spadesuit}$ ($c = 0.5$)	—	—
automatic extracts	f+f$\oplus$ld	**0.396**$^{\heartsuit}$ ($c = 0.4$)	**0.396** ($c = 0.4$)	**0.404** ($c = 0.4$)	**0.368**$^{\heartsuit}$ ($c = 0.9$)$^{\diamondsuit}$	—
	f+f$\oplus$tf	**0.376**$^{\heartsuit}$ ($c = 0.4$)	**0.380**$^{\heartsuit\heartsuit}$ ($c = 0.4$)	**0.388** ($c = 0.6$)	**0.372** ($c = 0.4$)	—
(d) **PDoc10/HIGHLY rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.260
manual abstracts	f+f$\oplus$a	—	—	**0.312**$^{\spadesuit\spadesuit}$ ($c = 0.5$)	—	—
automatic extracts	f+f$\oplus$ld	**0.300**$^{\spadesuit}$ ($c = 0.6$)	**0.332**$^{\spadesuit\spadesuit}$ ($c = 0.4$)	**0.340**$^{\spadesuit\spadesuit\heartsuit}$ ($c = 0.4$)	**0.280**$^{\heartsuit}$ ($c = 0.4$)$^{\diamondsuit}$	—
	f+f$\oplus$tf	**0.296**$^{\heartsuit}$ ($c = 0.4$)	**0.296** ($c = 0.4$)	**0.296**$^{\spadesuit\spadesuit}$ ($c = 0.6$)	**0.288**$^{\spadesuit}$ ($c = 0.6$)	—

our experiments in a training/testing setting in the future.

The sign test is used for testing statistical significance throughout this paper. Tables 1-3 contain three kinds of tags, indicating the following statistical significance:

- A combined run which significantly outperforms its *fulltext* component run (i.e. the baseline) is indicated by "$\spadesuit$" ($\alpha = 0.05$) and "$\spadesuit\spadesuit$" ($\alpha = 0.01$).
- A combined run which significantly outperforms its *summary* component run is indicated by "$\heartsuit$" ($\alpha = 0.05$) or "$\heartsuit\heartsuit$" ($\alpha = 0.01$).
- If an *abstract-based* combined run significantly outperforms a corresponding *extract-based* combined run, this is indicated by "$\diamondsuit$" ($\alpha = 0.05$) or "$\diamondsuit\diamondsuit$" ($\alpha = 0.01$) attached to the extract-based performance value.

For example, the 5% column of Table 1(a) shows that the combination f$\oplus$tf(5) is significantly better than either of its components (i.e. f and tf(5)) with AveP for all relevant documents, but that f$\oplus$a outperforms f$\oplus$tf(5) significantly.

Other significant differences are summarized in Table 4 in the same way as in [10].

In terms of average performance, our combined runs generally outperform not only the *fulltext* component runs but also the *summary* component runs. As an illustration, the "strategy" column in Part (b) of each table shows the AveP performance values for the *abstract* and *Abstract-Length extract* component runs, for highly relevant documents. (All other component performance values are given in [10, 11].) For example, Table 1(b) shows that, for highly relevant documents, the AveP of the combination f$\oplus$a is 0.300 while those of its components f and a are 0.249 and 0.231, respectively. Thus, on average, f$\oplus$a is (0.300/0.249=)20% better than f and (0.300/0.231=)30% better than a. In the same way, with AveP for highly relevant documents, Table 2(b) shows that f+f$\oplus$a is 7% better than f+f and 28% better than a, and Table 3(b) shows that f+f$\oplus$a+a is 12% better than f+f and 31% better than a+a. As for lead extracts, with AveP for highly relevant documents, Table 1(b) shows that f$\oplus$ld(AL) is 22% better than f and 63% better than ld(AL); Table 2(b) shows that f+f$\oplus$ld(AL) is 6% better than f+f and 57% better than ld(AL); Table 3(b) shows that f+f$\oplus$ld+ld(AL) is 9% better than f+f and 54% better than

Table 3: **Combining a fulltext-fulltext run and a summary-summary run.**

(a) **AveP/ALL rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.278
manual abstracts	f+f⊕a+a	—	—	$\mathbf{0.288}^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)$	—	—
automatic extracts	f+f⊕ld+ld	$\mathbf{0.286}^{\heartsuit\heartsuit}$ $(c = 0.8)$	$\mathbf{0.282}^{\heartsuit\heartsuit}$ $(c = 0.9)$	$\mathbf{0.281}^{\heartsuit\heartsuit}$ $(c = 0.9)$	$\mathbf{0.281}^{\heartsuit\heartsuit}$ $(c = 0.9)$	—
	f+f⊕tf+tf	$\mathbf{0.284}^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	$\mathbf{0.285}^{\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	$\mathbf{0.287}^{\heartsuit\heartsuit}$ $(c = 0.9)$	$\mathbf{0.282}^{\heartsuit\heartsuit}$ $(c = 0.9)^{\diamondsuit\diamondsuit}$	—
(b) **AveP/HIGHLY rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.276
manual abstracts	f+f⊕a+a (a+a: 0.235)	—	—	$\mathbf{0.308}^{\spadesuit\spadesuit\heartsuit}$ $(c = 0.7)$	—	—
automatic extracts	f+f⊕ld+ld (ld+ld(AL): 0.196)	$\mathbf{0.296}^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.8)$	$\mathbf{0.288}^{\spadesuit\heartsuit}$ $(c = 0.9)^{\diamondsuit}$	$\mathbf{0.302}^{\spadesuit\heartsuit}$ $(c = 0.7)$	$\mathbf{0.289}^{\heartsuit\heartsuit}$ $(c = 0.7)$	—
	f+f⊕tf+tf (tf+tf(AL): 0.212)	$\mathbf{0.286}^{\spadesuit\heartsuit}$ $(c = 0.9)$	$\mathbf{0.288}^{\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)$	$\mathbf{0.294}^{\spadesuit\spadesuit\heartsuit\heartsuit}$ $(c = 0.9)$	$\mathbf{0.285}^{\spadesuit\spadesuit}$ $(c = 0.9)^{\diamondsuit}$	—
(c) **PDoc10/ALL rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.360
manual abstracts	f+f⊕a+a	—	—	$\mathbf{0.400}^{\spadesuit}$ $(c = 0.7)$	—	—
automatic extracts	f+f⊕ld+ld	$\mathbf{0.396}$ $(c = 0.6)$	$\mathbf{0.400}^{\heartsuit\heartsuit}$ $(c = 0.4)$	$\mathbf{0.400}^{\heartsuit}$ $(c = 0.5)$	$\mathbf{0.388}^{\heartsuit}$ $(c = 0.6)$	—
	f+f⊕tf+tf	$\mathbf{0.372}^{\heartsuit}$ $(c = 0.4)$	$\mathbf{0.372}$ $(c = 0.5)$	$\mathbf{0.388}^{\heartsuit}$ $(c = 0.4)$	$\mathbf{0.384}$ $(c = 0.5)$	—
(d) **PDoc10/HIGHLY rel.**	strategy	5%	10%	AL	30%	100%
fulltext baseline	f+f	—	—	—	—	0.260
manual abstracts	f+f⊕a+a	—	—	$\mathbf{0.308}^{\spadesuit\spadesuit}$ $(c = 0.7)$	—	—
automatic extracts	f+f⊕ld+ld	$\mathbf{0.296}^{\spadesuit}$ $(c = 0.8)$	$\mathbf{0.324}^{\spadesuit}$ $(c = 0.5)$	$\mathbf{0.340}^{\spadesuit\heartsuit}$ $(c = 0.5)$	$\mathbf{0.292}$ $(c = 0.5)$	—
	f+f⊕tf+tf	$\mathbf{0.292}$ $(c = 0.3)$	$\mathbf{0.308}$ $(c = 0.3)$	$\mathbf{0.296}^{\spadesuit}$ $(c = 0.7)$	$\mathbf{0.284}^{\spadesuit}$ $(c = 0.8)$	—

ld+ld(AL). Similar trends have been observed for lead extracts at other CRs and for *tf-idf* extracts. The performance gains from combinations are thus typically very substantial.

Sections 4.2-4.6 contain further discussions based on Tables 1-4. Moreover, Section 4.7 discusses the relationship between the effectiveness of combination and the *overlap* of document sets retrieved by the component runs.

4.2 Combinations vs Components

As explained in Section 4.1, wherever both "♠" and "♡" are attached to a combined performance value in Tables 1-3, that combination is significantly better than either of its components. These symbols draw attention to some general trends across the three tables, namely, that the effect of fulltext/summary combination is (1) more significant for *highly* relevant documents than for *all* relevant documents; and (2) more significant for AveP than for PDoc10. However, all the results seem promising (except perhaps for Part (c) of each table) in that there are significant gains over the fulltext baselines. Moreover, as has been illustrated in Section 4.1, there are substantial average performance gains over the fulltext and summary component runs, regardless of the summary index being used. In short, *the combination-of-evidence approach works well with the fulltext and summary indexes.* Moreover, Tables 1-3 show that these combination strategies are most effective when $c > 0.5$, that is, *when the fulltext evidence is emphasized.* This finding is also in agreement with previous ones on combination of evidence, namely, that when one source of evidence is weaker than others, this should be reflected by giving it less weight in the combination [3, 15].

4.3 Abstracts vs Extracts

As explained in Section 4.1, "◇"'s indicate that *abstract-based* combined runs are significantly better than *extract-based* combined runs, that is, that there are significant gaps between the upperbounds and the lowerbounds. These differences are apparent with AveP (Parts (a) and (b) of Tables 1-3). However, note that the differences between the abstract performances and the *Abstract-Length* extract performances are *not* statistically significant, except for two occasions in Table 1(a) and (b). Moreover, looking down the "AL" columns of Tables 1-3 shows that the abstract performances and the Abstract-Length extract performances are in general comparable. This suggests that, *while the summary quality can make a difference, even simple extracts may be sufficient as components of combination, provided* *that an appropriate CR is chosen.*

4.4 Extracts vs Extracts

Table 4(1) shows that there are only a small number of significant differences between the lead-based combined runs and the *tf-idf*-based combined runs. Moreover, the lead-based runs outperform the *tf-idf*-based ones in some cases, but not in others. Thus we cannot draw a conclusion as to which extract type is the superior.

4.5 Effect of CR

Table 4(2) shows that there are some significant effects of varying the CR for the lead and *tf-idf* extracts, but only with *highly* relevant documents. The somewhat artificial *Abstract-Length* extracts generally outperform extracts with fixed CRs, which again seems to underscore the importance of selecting the right CR.

Another interesting fact is that, with AveP for highly relevant documents, f⊕ld(5), f+f⊕ld(5), and f+f⊕ld+ld(5) significantly outperform their 10% counterparts, suggesting that *a lead extract index with a CR of 5% or below may be particularly suitable as a component of combination.* The fact that there are no significant differences between 5% lead extracts and *abstracts* in Tables 1-3 (implied by the lack of "◇"'s) supports this view. Thus, with AveP for highly relevant documents, Table 1(b) shows that f⊕ld(5) is (0.282/0.249=)13% better than f; Table 2(b) shows that f+f⊕ld(5) is 4% better than f+f; Table 3(b) shows that f+f⊕ld+ld(5) is 7% better than f+f. With PDoc10 for highly relevant documents, Table 1(d) shows that f⊕ld(5) is (0.340/0.252=)35% better than f; Table 2(d) shows that f+f⊕ld(5) is 15% better than f+f; Table 3(d) shows that f+f⊕ld+ld(5) is 14% better than f+f.

4.6 Comparison across Tables

Table 4(3) compares the corresponding cells from Tables 1-3 in terms of statistical significance. Some runs in Table 1 (i.e. **Combination 1**) were significantly outperformed by those in Table 2 (i.e. **Combination 2**), which in turn were significantly outperformed by those in Table 3 (i.e. **Combination 3**). One exception is the last line in Table 4(3): with AveP for highly relevant documents, f+f⊕tf(AL) (a **Combination 2** run) significantly outperformed f+f⊕tf+tf(AL) (a **Combination 3** run), even though this difference is not apparent in terms of average performance in Tables 2(b) and 3(b). However, in general, it seems that the relative effectiveness of the component runs is reflected to some extent in the relative effectiveness of the combined runs.

4.7 Overlap of Retrieved Document Sets

This section attempts to explain the results of our combination experiments by analyzing the actual document sets retrieved by the fulltext and summary component runs. Table 5 provides pertinent facts about the overlaps between each pair of document sets, where Parts (1)-(3) correspond to the component runs of **Combinations 1-3**, respectively. Each cell contains three numbers, namely, the *Retrieved Document Overlap* (RDO), the fulltext-oriented Retrieved *Relevant* Document Overlap ($RRDO_{fulltext}$), and the summary-oriented Retrieved Relevant Document Overlap ($RRDO_{summary}$), all averaged over the 25 requests based

Table 4: Statistically significant differences.

(1) lead *vs* *tf-idf* for fixed CRs	all relevant	highly relevant
f ⊕ ld(5) *vs* f ⊕ tf(5)	-	* †
f ⊕ ld(AL) *vs* f ⊕ tf(AL)	*	** ††
f+f ⊕ ld(30) *vs* f+f ⊕ tf(30)	-	*
f+f ⊕ ld+ld(5) *vs* f+f ⊕ tf+tf(5)	-	*
f+f ⊕ ld+ld(AL) *vs* f+f ⊕ tf+tf(AL)	-	*
(2) Effect of CR for fixed extract types	**all relevant**	**highly relevant**
f ⊕ ld(5) *vs* f ⊕ ld(10)	-	**
f ⊕ ld(5) *vs* f ⊕ ld(AL)	-	*
f ⊕ ld(AL) *vs* f ⊕ ld(30)	-	** †
f ⊕ tf(10) *vs* f ⊕ tf(AL)	-	**
f+f ⊕ ld(5) *vs* f+f ⊕ ld(10)	-	* †
f+f ⊕ ld(5) *vs* f+f ⊕ ld(30)	-	*
f+f ⊕ ld(10) *vs* f+f ⊕ ld(AL)	-	*
f+f ⊕ ld(10) *vs* f+f ⊕ ld(30)	-	†
f+f ⊕ ld(AL) *vs* f+f ⊕ ld(30)	-	* †
f+f ⊕ ld+ld(5) *vs* f+f ⊕ ld+ld(10)	-	*
(3) Comparison of cells across Tables 1-3	**all relevant**	**highly relevant**
f ⊕ ld(5) *vs* f+f ⊕ ld(5)	*	-
f ⊕ ld(5) *vs* f+f ⊕ ld+ld(5)	**	-
f+f ⊕ ld(5) *vs* f+f ⊕ ld+ld(5)	-	*
f ⊕ ld(10) *vs* f+f ⊕ ld(10)	-	†
f+f ⊕ ld(10) *vs* f+f ⊕ ld+ld(10)	-	**
f ⊕ ld(AL) *vs* f+f ⊕ ld+ld(AL)	*	-
f ⊕ ld(30) *vs* f+f ⊕ ld+ld(30)	*	-
f ⊕ tf(5) *vs* f+f ⊕ tf(5)	*	*
f ⊕ tf(5) *vs* f+f ⊕ tf+tf(5)	*	-
f ⊕ tf(10) *vs* f+f ⊕ tf(10)	*	*
f+f ⊕ tf(10) *vs* f+f ⊕ tf+tf(10)	**	-
f+f ⊕ tf(AL) *vs* f+f ⊕ tf+tf(AL)	-	*

Significant differences with AveP are represented by $*$ ($\alpha = 0.05$) or $**$ ($\alpha = 0.01$). Those with PDoc10 are represented by † ($\alpha = 0.05$) or †† ($\alpha = 0.01$).

on *all* relevant documents:

$$RDO = Ret_{fulltext} \cap Ret_{summary}$$

$$RRDO_{fulltext} = RelRet_{fulltext} \cap RelRet_{summary} / RelRet_{fulltext}$$

$$RRDO_{summary} = RelRet_{fulltext} \cap RelRet_{summary} / RelRet_{summary}$$

where

$Ret_{fulltext}$:	set of documents Retrieved by the fulltext component run
$Ret_{summary}$:	set of documents Retrieved by the summary component run
$RelRet_{fulltext}$:	set of *Relevant* documents Retrieved by the fulltext component run
$RelRet_{summary}$:	set of *Relevant* documents Retrieved by the summary component run

For example, at the very top of Table 5, it is shown that:

1. RDO between f and a is approximately 33% in percentage (since each component run usually retrieves 1000 documents [10, 11]).

2. 72% of the relevant documents retrieved by f were also retrieved by a ($RRDO_{fulltext} = .72$);

3. 94% of the relevant documents retrieved by a were also retrieved by f ($RRDO_{summary} = .94$);

Table 5(1)-(3) reveal that the fulltext and summary indexes retrieve many different documents: thus, RDO is well below 50% in percentage, even at 30% CR. Compared to this, the overlap percentages for the *relevant* documents are much higher: $RRDO_{fulltext}$ increases with CR, from around 50% to 75%, while $RRDO_{summary}$ remains around 90%-95%. The high values of $RRDO_{summary}$ imply that *the primary role of the summary index in the combination is to promote some of the relevant documents already retrieved by the fulltext index, rather than to discover new relevant documents.* These trends are in agreement with previous findings on the combination of evidence, namely, that components that retrieve similar relevant documents and dissimilar nonrelevant documents produce the best results in combination [6, 15].

Finally, Table 5(4) provides the document overlap information for an alternative combination mentioned in Section 3.2, namely, that of the fulltext-fulltext run with a *summary-fulltext* run. As explained in Section 3.2, this combination was less successful than **Combination 3** (i.e. that with a *summary-summary* run) despite the fact that the summary-fulltext component runs themselves were generally more effective than the summary-summary component runs [10, 11]. Table 5(4) seems to provide an explanation for this: thus, RDO (and therefore $RRDO_{fulltext}$) between f+f and the summary-fulltext runs are considerably greater than those in Table 5(1)-(3), because the summary-fulltext runs use the fulltext index for final search just like f+f does. Again, these results are in agreement with previous findings on the combination of evidence, namely, that components should have a *low* Retrieved Document Overlap in order to produce good results in combination [3, 5].

Table 5: Overlap of retrieved document sets.

(1)	5%	10%	AL	30%
f *vs* a	—	—	334.9 **.72/.94**	—
f *vs* ld	236.5 **.48/.92**	297.6 **.56/.94**	316.7 **.60/.95**	463.6 **.75/.95**
f *vs* tf	280.2 **.53/.90**	325.4 **.60/.94**	415.1 **.71/.95**	487.8 **.78/.95**
(2)	5%	10%	AL	30%
f+f *vs* a	—	—	289.7 **.68/.90**	—
f+f *vs* ld	217.1 **.47/.93**	269.0 **.54/.93**	270.5 **.58/.93**	394.9 **.71/.92**
f+f *vs* tf	244.6 **.51/.87**	284.2 **.58/.91**	345.2 **.69/.92**	409.5 **.75/.91**
(3)	5%	10%	AL	30%
f+f *vs* a+a	—	—	300.2 **.71/.91**	—
ld+ld f+f *vs*	195.7 **.50/.92**	267.1 **.58/.93**	260.7 **.60/.92**	408.7 **.71/.92**
f+f *vs* tf+tf	219.4 **.52/.88**	283.7 **.59/.92**	338.4 **.67/.92**	438.8 **.76/.90**
(4)	5%	10%	AL	30%
f+f *vs* a+f	—	—	665.0 **.95/.94**	—
f+f *vs* ld+f	559.7 **.90/.94**	611.4 **.92/.94**	624.0 **.93/.94**	699.2 **.90/.92**
f+f *vs* tf+f	565.0 **.91/.94**	616.7 **.92/.94**	654.8 **.89/.94**	712.4 **.94/.94**

5. CONCLUSIONS

This paper has demonstrated the advantage of combining a ranked output from a fulltext index and another from an index generated using generic summaries. Our main findings are as follows:

- The combination of a fulltext-based run and a summary-based run can significantly outperform either alone, especially when the fulltext evidence is emphasized.
- The primary role of the summary index in the above combination is to promote some of the relevant documents already retrieved by the fulltext index, rather than to discover new relevant documents.
- Moreover, though the summary quality can make a difference for this combination approach, even simple extracts such as 5% "lead" may give significant improvements.

One weakness of our study is the limited size of the test collection, which prevented us from conducting a training/testing-type evaluation. Although a large test collection with a manual abstract for each document is hard to come by, our results require re-examination with such data when they become available.

Acknowledgments

This work was done while visiting the University of Cambridge Computer Laboratory, and the author would like to thank Karen Sparck Jones for her many useful comments on the earlier draft of this paper.

6. REFERENCES

[1] Brandow, R., Mitze, K. and Rau, L. F: Automatic Condensation of Electronic Publications by Sentence Selection, *Information Processing and Management* 31(5), pp. 675–685 (1995).

[2] Callan, J. P.: Passage-Level Evidence in Document Retrieval, *ACM SIGIR '94 Proceegings*, pp. 302–310 (1994).

[3] Croft, W. B.: Combining Approaches to Information Retrieval, *Advances in Information Retrieval - Recent Research from the Center for Intelligent Information Retrieval* (Croft, W. B. (ed.)), Kluwer Academic Publishers, pp. 1-36 (2000).

[4] Fox, E. A. and Shaw, J. A: Combination of Multiple Searches, *TREC-2 Proceedings* (1994).

[5] Lee, J. H.: Combining Multiple Evidence from Different Properties of Weighting Schemes, *ACM SIGIR '95 Proceedings*, pp. 180–188 (1995).

[6] Lee, J. H.: Analyses of Multiple Evidence Combination, *ACM SIGIR '97 Proceedings* , pp. 267–276 (1997).

[7] McCarley, J. S.: Should we Translate the Documents or the Queries in Cross-language Information Retrieval?, *ACL '99 Proceedings*, pp. 208–214 (1999).

[8] Sakai, T., Kajiura, M. and Sumita, K.: Generation and Evaluation of Search Queries Using Boolean Expressions and Document Structure for Information Filtering (in Japanese), *IPSJ Journal*, Vol.39, No.11, pp. 3076-3083 (1998).

[9] Sakai, T., Robertson, S. E. and Walker, S.: Flexible Pseudo-Relevance Feedback via Direct Mapping and Categorization of Search Requests, *BCS-IRSG ECIR 2001 Proceedings*, pp. 3–14 (2001).

[10] Sakai, T. and Sparck Jones, K.: Generic Summaries for Indexing in Information Retrieval - Detailed Test Results, TR513, Computer Laboratory, University of Cambridge (2001). `http://www.ftp.cl.cam.ac.uk/ftp/papers/reports/#TR513`

[11] Sakai, T. and Sparck Jones, K.: Generic Summaries for Indexing in Information Retrieval, *ACM SIGIR 2001 Proceedings, to appear* (2001).

[12] Sakai, T. and Robertson, S. E.: Flexible Pseudo-Relevance Feedback Using Optimization Tables, *ACM SIGIR 2001 Proceedings, to appear* (2001).

[13] Strzalkowski, T., Wang, J. and Wise, B.: Summarization-based Query Expansion in Information Retrieval, *COLING-ACL '98 Proceedings*, pp. 1258–1264 (1998).

[14] Tombros, A. and Sanderson, M.: Advantages of Query Biased Summaries in Information Retrieval, *ACM SIGIR '98 Proceedings*, pp. 2–10 (1998).

[15] Vogt, C. C. and Cottrell, G. W.: Predicting the Performance of Linearly Combined IR Systems, *ACM SIGIR '98 Proceedings*, pp. 190–196 (1998).