Appears in ***ASIST 2007****, November 3-8, 2007, Milwaukee, WI, USA.*

The Benefits of Skimming in Data Fusion

Anselm Spoerri

Department of Library and Information Science
School of Communication, Information and Library Studies
Rutgers University
4 Huntington Street, New Brunswick, NJ 08901, USA
Email: aspoerri@scils.rutgers.edu.

Data fusion methods commonly use and compare all the documents returned by multiple retrieval systems to create a new result list. On the one hand, as documents further down in the result lists are considered, a document's probability of being relevant decreases significantly. On the other hand, retrieval systems tend to find similar relevant documents when searching the same database, but they do not find them in the same rank positions. Thus, data fusion methods need to consider all of the documents returned by the retrieval systems. Using TREC 3, 6, 7, 8, 12 and 13 data, this paper examines how "skimming", where the number of documents examined in the result lists is gradually increased, can help to identify relevant documents. It is shown that "gradual skimming" and what can be learned as the list depth is increased can help to improve the retrieval effectiveness of data fusion methods.

Introduction

The goal of data fusion is to combine the result lists of multiple retrieval systems to produce a new ordering of the documents that moves potentially relevant documents further toward the top of the fused result list and non relevant ones toward its bottom (Fox & Shaw 1994, Callan, 2000). According to www.dictionary.com, to "skim" can refer to a) "removing cream" from the top of milk or b) to "take the best items" from a collection of things. Both these meanings apply in the context of data fusion. The latter meaning refers to the fact that data fusion aims to identify the relevant documents contained in multiple result lists. The former meaning corresponds to the fact that the relevant documents tend to be located at the top of a result list. A key problem data fusion methods have to solve is how to identify relationships between the different result lists that make it possible to detect all or many of the relevant documents (high recall) and move relevant documents closer toward the top of the fused result list (high precision). These relationships can be easily computed: a) the number of lists that contain the same document and b) the average of a document's different positions in the lists that contain it. Many of the most effective data fusion methods use these relationships to decide how best to merge the result lists (Fox & Shaw, 1994; Lee 1997). Further, it is possible to compute these relationships between the result lists as the number of documents examined in the lists, called the *list depth*, is gradually increased. A contribution of this paper is that it shows the potential of "gradual skimming" to improve data fusion.

The work described in this paper is part of a research program that investigates what can be learned by only comparing and analyzing the result lists of different retrieval systems searching the same database without the need for any other sources of information or analytical tools (Spoerri 2005, 2006a, 2006b). Specifically, Spoerri (2005, 2006b) has shown that a document's probability of being relevant increases exponentially as the number of systems retrieving it increases – called the *Authority Effect*. Further, he has shown that a document placed higher up in multiple result lists and found by more systems is more likely to be relevant – called the *Ranking Effect*. Thus, easily observable relationships between the different lists can be used to identify potentially relevant documents. Spoerri (2006a) has also shown that the Authority and Ranking Effects are present as the list depth is increased, but the total number of relevant documents found by the systems affects when the Ranking Effect fully emerges. This paper examines how gradually increasing the list depth can improve the ability to identify relevant documents.

Data fusion methods commonly use all the documents returned by different retrieval systems. It can be argued that "more is better" and the fact that a document is found by many systems, although placed by some toward the bottom of their result lists, provides very useful information to identify relevant documents. Specifically, retrieval systems tend to find similar relevant documents when searching the same database, but they do not find them in the same rank positions (Spoerri, 2006a). This suggests that data fusion methods are well advised to consider all of the documents returned by the retrieval systems. However, retrieval methods aim to place the potentially relevant documents toward the top of their result lists and as documents further down in the result lists are considered, a document's probability of being relevant decreases significantly. Using Text REtrieval Conference (TREC) 3, 6, 7, 8, 12 and 13 data, this paper addresses the problem of how the information gained for documents that are contained in the top of the lists and found by many systems is not "diluted and lost" as the list depth is increased and an increasing number of non-relevant documents are found by many systems, thus introducing a major source of noise. In the context of the "skimming cream" analogy, this paper investigates how to ensure that more relevant documents stay and/or raise to the top as gradually more documents are examined.

This paper is organized as follows: first, related work is discussed. Second, the methodology employed is described. Third, data is presented that estimates a document's probability of being relevant as a function of list depth, the number of systems that have found it and its average rank position. Fourth, it is shown how "gradual skimming" is able to improve the retrieval effectiveness of data fusion. Fifth, it is discussed how "gradual skimming" could be used to develop more effective data fusion methods.

Related Work

Prior research relevant or related to the work presented in this paper falls into two categories: 1) research providing direct support for the Authority and Ranking Effects; 2) methods proposed for fusing multiple results lists.

Support for Authority and Ranking Effects

Spoerri (2005) summarized previous research that provides indirect support for the Authority and Ranking Effects (Saracevic & Kantor, 1988; Turtle & Croft, 1991; Foltz & Dumais, 1992; Belkin et al., 1993). He conducted a systematic analysis of the overlap between the search results of the retrieval systems that participated in the *short* tracks in TREC 3, 6, 7 and 8. He provides direct support for the Authority and Ranking Effects by demonstrating that a document's probability of being relevant is correlated to the number of retrieval systems that find it and the document's positions in the lists that contain it. Spoerri (2006a) extends this study by also analyzing data from the *manual* and *long* tracks in TREC 6, 7 and 8. Both analyses showed that a document's probability of being relevant increases exponentially as the number of systems retrieving it increases, thereby providing direct support for the Authority Effect. It was also demonstrated that the placement of the relevant documents in ranked lists is not a random process. Instead, as the number of systems retrieving the same relevant document increases, a relevant document is increasingly located toward the top of the systems' lists. Finally, it was shown that a document's probability of being relevant increases greatly as more systems find it and the higher up it is placed in the multiple ranked lists that contain it, thereby providing direct support for Ranking Effect.

Spoerri (2006a) also studied how varying the number of documents examined in the results (*list depth*) impacts the Authority and Ranking Effects. He analyzed the overlap between the search results of retrieval systems that participated in the ad hoc track in TREC 3, 7 and 8, the robust track in TREC 12 and the web track (distillation task) in TREC 13. First, as noted above, it was demonstrated that the retrieval systems find similar relevant documents, but they do not find them in the same rank positions or at the same list depth levels. Specifically, it was shown that the relevant documents are gradually found by more systems and the number found only by a single system decreases rapidly as the list depth is increased. Second, it was shown that the Authority Effect is present at all list depth levels. Third, it was shown that the Ranking Effect is present at all list depth levels, but if the systems in the same TREC year retrieve a large number of relevant documents, then the Ranking Effect only begins to emerge as more systems have found the same document and/or the list depth increases (as will be discussed in the Results section). Fourth, it was demonstrated that the Authority and Ranking Effects are not an artifact of how the TREC test collections have been constructed, where only top 100 documents are pooled and examined to identify relevant documents. Specifically, it was shown that the majority and increasing percentage of new documents, which are found at a specific list depth, are only found by a single system and very few new documents are found by more than 3 or 4 systems for list depths greater than 100 and approaching 1000 documents. This implies that the unjudged relevant documents will be found by few systems at most, only to be "overshadowed" by the many non-relevant documents found by few systems, and thus will not impact the Authority and Ranking Effects for documents found by many systems.

Data Fusion Methods

Many and some of the most effective data fusion methods use merging and voting methods, where documents that are found by multiple methods and placed high up in the respective lists are promoted. Specifically, Fox and Shaw (1994) introduced a set of major methods for combining multiple results sets, such as CombMNZ and CombSUM. When a document is found by a system, it receives a retrieval score and has a specific position in the ranked list returned by the system. Further, a document can be found by multiple systems. If a document's retrieval scores or rank positions are normalized to a score between 0 and 1 (the higher up in the result list, the greater the score), then the sum of a document's scores will be less or equal to the number of systems retrieving it. CombSUM only sums a document's scores. CombMNZ sums a document's scores by the different systems that find it and then multiplies this sum by the number of systems that retrieve the document. CombMNZ and CombSUM exploit to varying degrees the Authority and Ranking Effects. Both of them make use of the Ranking Effect, because they sum the normalized rank positions – the higher up a document in multiple lists, the greater the sum. This summing operation also incorporates the Authority Effect, because the more systems that find a document, the more scores are added. The Authority Effect is more dominant for CombMNZ than for CombSUM, since CombMNZ multiples CombSUM by the number of systems that find a document. Lee (1997) demonstrated that CombMNZ performs best, followed by CombSUM, in terms of retrieval effectiveness. CombMNZ has proven to be an effective fusion method that it is used by many researchers as the baseline method to compare with their newly developed fusion methods.

Aslam and Montague (2001, 2002) developed two fusion methods that make use of democratic voting methods that can deal with few voters (retrieval systems) and many candidates (documents). The Borda-fuse method (Aslam & Montague, 2001) assigns a score to each document that is related to the sum of its positions in each result list that contains it, making Borda-fuse very similar to CombSUM. The Condorcet-fuse method (Montague & Aslam, 2002) ranks documents based on a pair-wise comparison of their rank positions. A document is ranked above another if it appears above it in more result sets.

Vogt and Cottrell (1999) refer to the phenomenon described by the Authority Effect as the "Chorus Effect" and they also suggest that the "Skimming Effect" plays a role in data fusion. The latter effect refers to the fact that multiple result sets are more likely to result in a larger number of relevant documents being included in the fused list than when only a single list is considered. Thus, a data fusion method can "skim" the top documents from each result list, since many the relevant documents are located toward the top of the lists. Vogt & Cottrell's analysis of the pair-wise comparison of result lists showed that the Authority Effect tends to be the dominant effect. Further, Lillis et al. (2006) have developed probFuse, which uses a probabilistic approach to data fusion and exploits the Skimming Effect with some success. A contribution of this paper is that it provides insight into how "skimming" can help to improve the effectiveness of data fusion methods because it "sharpens" the Ranking Effect.

Methodology

The TREC workshops provide IR researchers with a controlled setting, a set of search topics and, most importantly, human relevance judgments to make it possible to compare and analyze the effectiveness of different retrieval methods that search the same large document collections (Voorhees & Harman, 1994, 1997, 1998, 1999; Voorhees, 2003, 2004; Craswell & Hawking, 20040. This paper uses the ranked lists returned by the retrieval systems that took part in the *ad hoc track* in TREC 3, 6, 7 and 8, the *robust track* in TREC 12 and the *web track (distillation task)* in TREC 13 to examine what can be learned as the list depth is varied to identify relevant documents and improve data fusion effectiveness. First, these tracks were chosen, because they represent a diverse subset of all the TREC years, where the retrieval systems participating in the selected years search different document collections for the different 50, 75 and 100 provided topics in the ad hoc, web (distillation task) and robust tracks, respectively. Second, the systems in the chosen tracks submit a ranked list (also called a run) of usually 1000 documents per topic for evaluation, which provides data fusion methods with a large number of documents to fuse as well as makes it possible to study what can be learned as the list depth is gradually increased. In order to identify the relevant documents, the top 100 retrieved documents (or the top 125 documents for topics 51 – 100 in the TREC 12 Robust track) are pooled from each result list per topic and then a TREC evaluator determines the relevance of each document in the pool. The systems are evaluated based upon different measures of recall and precision. *Recall* assesses the fraction of relevant documents that were found by a system, while *precision* assesses the fraction of a system's retrieved documents that are relevant. The average precision for a specific topic is the mean of the precision after each relevant document is found. The *mean average precision* for all topics is the mean of the average precision scores. The retrieval effectiveness of retrieval systems is measured in terms of the mean average precision achieved.

Each system participating in TREC can submit multiple runs for evaluation. A run can either be *automatic* or *manual*. For the former, the query is created without human intervention based on the complete topic statement (called a *long* run) or only the title and description fields (called a *short* run). In this paper, the short runs in TREC 3, 6, 7 and 8 are used, because a greater number of systems submitted short runs (Voorhees & Harman, 1994; 1997, 1998; 1999). The TREC 12 Robust track is chosen, since the 100 topics used in this track consist of 50 topics selected from the ad hoc tracks in TREC 6-8 that proved especially difficult for most retrieval systems and 50 new topics that were selected with the expectation to be difficult as well (Voorhees, 2003). This makes it possible to examine the potential benefits of "gradual skimming" for poorly performing topics. For the TREC 13 Web track, the 75 distillation topics are chosen, since for these topics the systems needed to find more than one relevant web page and tend to return 1000 documents per topics (Voorhees, 2004; Craswell & Hawking, 2004). The selected ad hoc, robust and web TREC tracks make it possible to investigate what can be learned to help identify relevant documents in diverse and difficult settings.

In this paper, only one run of the runs submitted by the same systems is considered. Specifically, the "best" run with the highest mean average precision is used in this study (but any run submitted by a system could be used). This greatly reduces the noise introduced if multiple runs by the same system are included in the analysis (Wu & Crestani, 2003). There are 18 (19), 24, 25 (28), 35, 16 (17) and 11 (17) best runs for TREC 3, 6, 7, 8, 12 and 13 respectively, that are analyzed in this paper. The numbers in brackets indicate the total number of different systems, and some systems were not included in this study because they submitted significantly less than 1,000 documents on average per topic. Once the best run for each system has been identified, the overlap between the result sets of the different systems in the same year is computed for each topic. Next, averaging across all topics, the number of documents found by a specific number of systems is computed for all documents (relevant and non-relevant) and all relevant documents, respectively. The average number of unique relevant documents found per topic is 188, 78, 85, 87, 72, 33 and 21 for TREC 3, 6, 7, 8, 12a (topics 1 – 50), 12b (topics 51 – 100) and 13, respectively. TREC 3 has more than double the average number of relevant documents found per topic than TREC 6, 7, 8 or 12a (topics 1 – 50); six and nine times the average number of relevant documents than TREC 12b (topics 51 – 100) and TREC 13, respectively. As noted above, Spoerri (2006a) showed that the number of relevant documents in a TREC year affects when the "regular" Ranking Effect occurs as the list depth is increased (see Fig.1 and the graph for the Top 50 documents).

Estimating A Document's Probability of Being Relevant

As mentioned, Spoerri (2006a) studied how varying the number of documents examined in the ranked results (list depth) impacts the Authority and Ranking Effects. He presented empirical data of a document's probability of being relevant as a function of the list depth level used (top 50 to 1000 documents with a document step size of 50), the number of systems that have found it and its average rank position in the lists containing it. This paper additionally examines the smaller list depths of the top 10, 20, 30 and 40 documents, which makes it possible to better identify the many relevant documents that tend to be located at the very top of a result list.

In order to estimate a document's relevance probability as a function of list depth, the number of systems that have found it and its average rank position, the following steps are taken. First, the result sets of all the systems in the same TREC year are compared and the average number of documents found by 1, 2, 3, …or all systems is computed for each topic. If a document is found by multiple systems, then it will have multiple rank positions, which need to be averaged. The rank position is normalized so that the top document has a value of 1 and the very bottom document has a value of 1 divided by the *ListDepth*, which is equal to the maximal number of documents currently being compared. Specifically, a document *i* with the rank position in the result list of system S(j), called *doc(i)_RankPos_S(j)*, will have a normalized rank position that is equal to 1 – ((doc(i)_RankPos_S(j) – 1) / ListDepth). For example, if ListDepth is equal to the top 50 documents, then the document in the 11th rank position will have a normalized rank value equal to 0.80. If a document is found by multiple systems, then its multiple normalized rank positions are averaged. If a document has an average normalized rank position equal to 1, then this implies that it is the top document in all the ranked lists that contain it. Conversely, if a document has an average normalized rank position close to 0, then this implies that it is located close to the bottom of the list depth currently being used.

Second, the documents found by a specific number of systems are placed into different buckets based their average normalized rank values. The range of consecutive rank positions that are collected in the same bucket is equal to the list depth divided by 10 for list depths equal to 10, 20, 30, 40 or 50 top documents. For list depths equal or greater than 100, a range equal to the list depth divided by 20 is used (i.e., if ListDepth is equal to 500, then 25 consecutive rank values are aggregated). For example, a document with an average rank position of 21 will be placed in the bucket that aggregates the documents with average rank positions of 1 to 25 (or in terms of normalized rank positions, a value of 0.96 will be aggregated with the values ranging from 0.952 to 1.0). Thus, a fixed number of buckets are used to estimate the probability distribution instead of using a fixed number of consecutive rank positions.

Third, for each bucket, the probability that a document is relevant is estimated by dividing the number of relevant documents by the total number of documents in a bucket. As noted above, the data from all the topics is then averaged. It is required that at least three topics have documents in the same bucket so that a few data points cannot introduce spurious effects when the percentage of documents that are relevant is calculated. The rank positions are normalized and documents placed in consecutive buckets to make it possible to "stitch together" the relevance plots for documents found by a specific number of systems and their average rank positions. Specifically, a graph can be created, whose x-axis has continuous values ranging from 1 to the maximum number of systems being compared in a TREC year (see Fig. 1). Each segment along the x-axis represents the documents found by a specific number of systems. Further, the average normalized rank value increases from left to right in each segment. The number of systems that found a document and its average normalized rank position determine a document's position along the x-axis. For example, using a ListDepth equal to 50 documents, a document, which is found by 20 systems and has an average rank position of 11, will have a normalized rank value of 0.80 and will be located toward to the right end of the segment with label ''20'' in Fig. 1, since it will have a value of 20.80 on the x-axis. The steps just described make it possible to visualize in a compact way the percentage of documents that are relevant as a function of the number of systems that find them, their average rank positions and list depth.

This paper addresses the question of what can be learned to improve the ability to identify relevant documents as the list depth is gradually increased. To achieve this goal, it is important to track and not to lose the relevance estimates inferred for documents already encountered at a shallow list depth as gradually more documents located further down in the result lists are included in the comparison computation. At the same time, more documents will be found by multiple systems as the list depth is increased, which in turn may make it possible to infer a higher relevance probability for such documents. Thus, as the list depth is gradually increased, a document's probability of being relevant at a specific list depth will be set equal to the maximum of the probability estimates inferred at the list levels examined so far. This will make it possible to construct a "best case" scenario of what can be learned as the result lists are gradually skimmed.

Results

If the 35 systems in TREC 8 are compared, then Figure 1 displays the average percentage of documents that are relevant as a function of the number of systems retrieving them and the average of their rank positions for the list depths of the *top 50* (grey line) and *top 1000* (black line) documents, respectively. The ascending "saw tooth" patterns in Figure 1 illustrate that as both the number of systems retrieving the same document and the average of its normalized rank positions increases, the probability that the document is relevant tends to increase. Within a segment, which contains documents found by the same number of systems, the relevance probability tends to increase from left to right as the average normalized rank value increases from left to right in each segment. However, if a list depth of the top 50 documents is used, then a document, which is found by few systems and has a low average rank position, has a greater probability of being relevant than a document with a high average rank position.

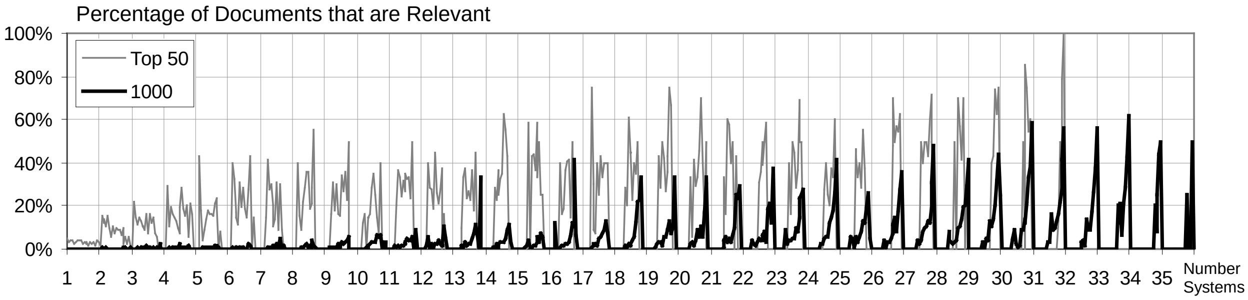


Fig 1: Plots of the percentage of TREC 8 documents that are relevant based on the number of systems retrieving them and the average of their rank positions for the list depths of the top 50 (grey line) and top 1000 (black line) documents, respectively.

This is the opposite result than predicted by the Ranking Effect (see gray graph in Fig.1). Once a list depth of 250 or more documents is used, then the "regular" Ranking Effect occurs for all segments, as shown by the black graph in Figure 1, which plots the probability distribution for a list depth of 1000 documents. Spoerri (2006a) provides a more detailed presentation of how the probability distribution changes as the list depth is increased for the TREC 3, 6, 7, 8, 12 and 13 (and how the number of relevant documents found in a TREC year affects the Ranking Effect). For the purpose of this paper, it is worth observing in Figure 1 that, as the list depth increases, the "saw tooth" pattern shifts to the right as documents found by few systems become increasingly less likely to be relevant and more documents are found by (almost) all systems. Figure 1 shows how a large percentage of relevant documents is located toward the top of the result lists and that the ability to identify relevant documents decreases as the list depth is increased. Specifically, if a list depth of the top 50 documents is used, then a document in the very top of the result lists and found by many systems has a very high probability of being relevant, whereas the ability to identify relevant documents decreases if all 1000 documents are compared.

What Can be Learned As List Depth is Increased

The goal of this paper is to investigate how to have more relevant documents "stay and/or float" to the top of the fused list as gradually more documents are compared and "skimmed". As the list depth is increased, it is important not to lose track of the relevance estimates inferred for documents already encountered at a shallow list depth. At the same time, more documents will be found by multiple and an increasing number of systems as the list depth is increased, which in turn may make it possible to infer a higher relevance probability for such documents. Thus, a document's relevance probability at a specific list depth level is set equal to the *maximum* of the relevance estimates assigned at the list levels examined so far. This way the information gained for documents that are contained in the top of the lists and found by many systems is not "diluted and lost" as the list depth is increased and a growing number of non-relevant documents are returned by many systems. To measure what can be learned as the list depth is gradually increased, it is useful to compute the difference between the *maximum relevance estimate*, which is inferred if the list depths of the top 10, 20, 30, 40, 50, 100, 200, … ,1000 documents are gradually compared, and the *ALL 1000 relevance estimate*, which is equal to the percentage of the top 1000 documents that are relevant. Figure 2 displays the difference between the "maximum" and "ALL 1000" relevance estimates as a function of the number of systems that have found them and their average rank positions for TREC 3, 6, 7, 8, 12a (topics 1 – 50), 12b (topics 51 – 100) and 13, respectively. It is worth noting that the two distributions being compared are computed in slightly different ways. The "maximum" relevance estimate for a bucket is equal to the average of the maximum relevance estimates assigned to the documents in the same bucket. The "ALL 1000" relevance estimate for a bucket is computed by dividing the number of relevant documents by the number of all documents contained in a bucket.

Figure 2 shows that the information that can be learned as the list depth is gradually increased helps to "sharpen" the Ranking Effect. The information gained helps to significantly increase a document's estimate of being relevant if it is placed very high up the lists that contain it, especially if it has been found by many, but not necessarily all systems. The gains are greatest for TREC 3, which contains more than double the number of relevant documents than TREC 6, 7, 8 or 12a; six and nine times the average

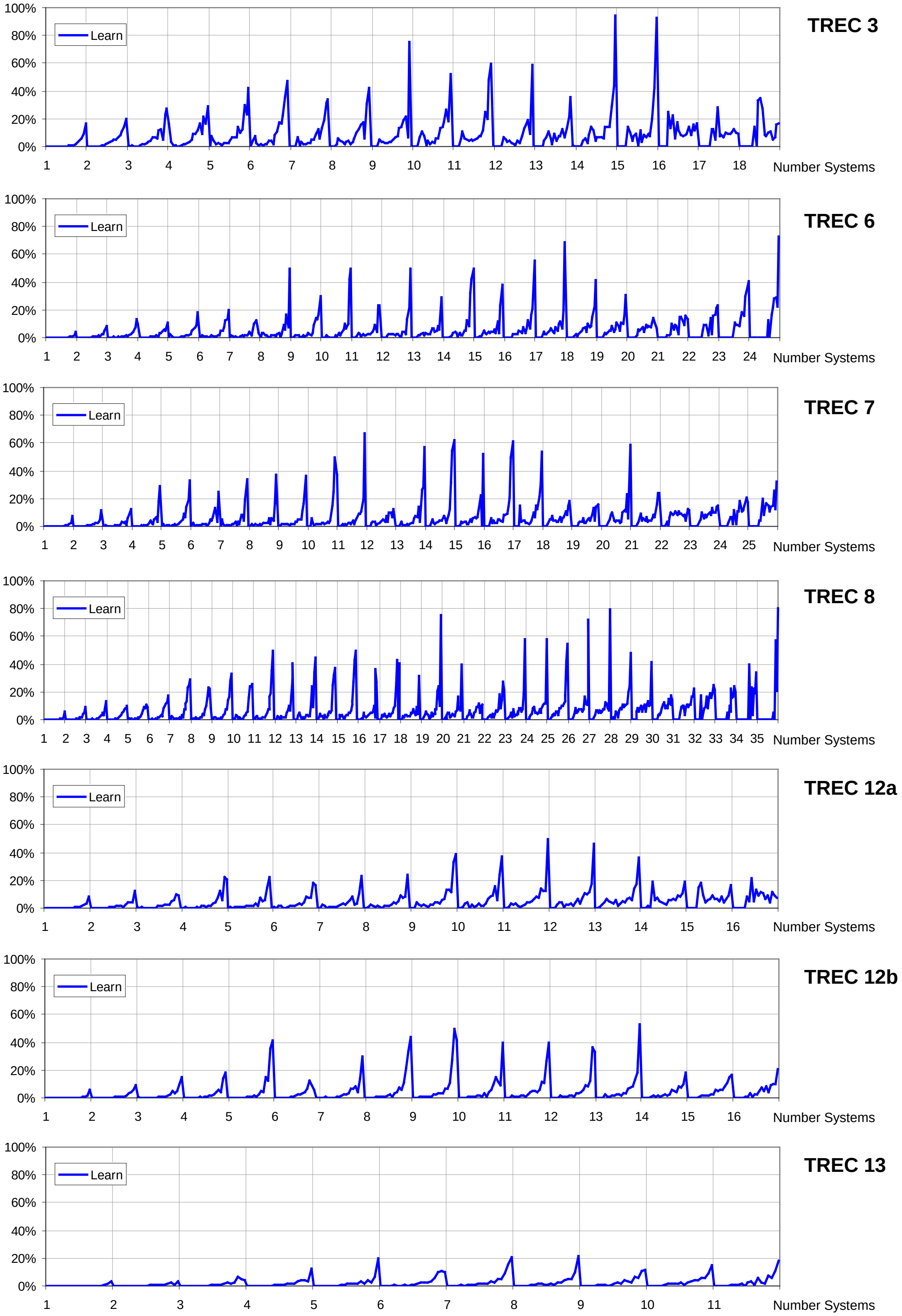
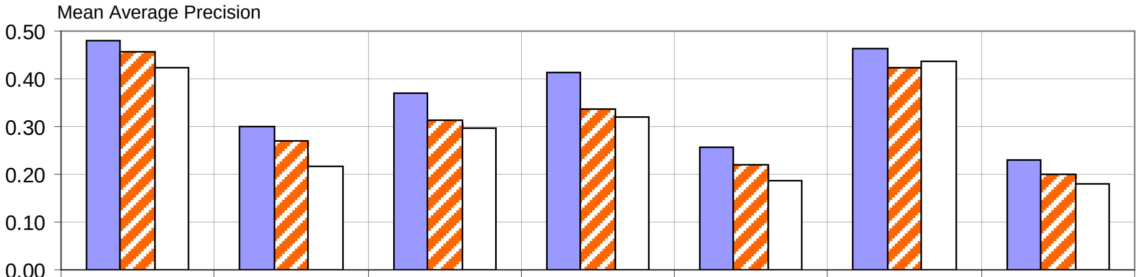


Fig 2: displays what can be learned if the list depth is gradually increased by plotting the difference between the “maximum” and “ALL 1000” relevance estimates as a function of the number of systems that have found them and their average rank positions for TREC 3, 6, 7, 8, 12a (topics 1 – 50), 12b (topics 51 – 100) and 13, respectively.



	TREC 3	TREC 6	TREC 7	TREC 8	TREC 12a	TREC 12b	TREC 13
■ BestCase	0.4803	0.2997	0.3716	0.4138	0.2555	0.4636	0.2295
▨ ALL1000	0.4577	0.2707	0.3126	0.3360	0.2184	0.4229	0.1987
□ BestSys	0.4226	0.2164	0.2961	0.3207	0.1854	0.4369	0.1790

Fig 3: displays the mean average precision (MAP) scores for BestCase, ALL1000 and BestSys.

number of relevant documents than TREC 12b and TREC 13, respectively. The gains are smallest for TREC 13, which contains the smallest number of relevant documents. The more relevant documents found in a TREC year, the greater the gains to be expected by "gradual skimming."

The reason for studying what can be learned as the result lists are gradually "skimmed" is to gain insights into how data fusion can be performed more effectively. Thus, the unique documents for each topic are sorted based on the "maximum" relevance estimates assigned to them, which will be referred to as the *BestCase* fusion method. This method is not equivalent to knowing which specific documents are actually relevant, but instead it uses the maximum of the empirical relevance estimates and it represents what can at best be expected in terms of fusion effectiveness when using voting and merging methods. In this paper, the retrieval effectiveness of the "BestCase" method is compared to the effectiveness of the *ALL1000* method, which uses the "ALL 1000" relevance estimates (shown as the black line in Figure 1 for TREC 8). Once the sorted lists are generated for the "BestCase" and "ALL1000" methods, respectively, the first 1000 documents are used to produce a fused list for each method. The mean average precision (MAP) score is calculated for each fused list. Figure 3 displays the MAP scores for "BestCase", "ALL1000" and the best system (*BestSys*) in each TREC year. As is to be expected, the MAP score for "BestCase" is greater than all the other scores for all years and the average MAP improvement of "BestCase" with respect "BestSys" is more than 25% for the TREC years studied in this paper. The bar chart in Figure 3 visualizes the orderly relationship between the MAP scores for "BestCase", "ALL1000" and "BestSys", where the bar heights are decreasing from left to right for each TREC year (except for TREC12b, where "BestSys" has a greater MAP score than "ALL1000"). Figure 3 suggests that "gradually skimming" the different result lists holds the potential to produce more effective fusion methods. It also provides an insight into the effectiveness improvement that can be expected if the result lists are gradually fused.

Discussion and Future Work

To be able to take advantage of the benefits of "gradual skimming" in practice, a model needs to be developed that can be used to estimate a document's relevance probability based on the number of systems that found it, its average normalized rank positions and the list depth used. Preliminary research shows that a combination of piecewise linear and exponential functions can be used to approximate the Ranking Effect for a diverse set of TREC data. The Authority Effect can be modeled by using an exponential function. This makes it possible to develop a (relatively) simple model of a document's probability of being relevant, where the key variables depend in a predominately linear fashion on only one a priori variable, namely the average number of relevant documents found per topic. Future research will a) investigate how to generalize this preliminary model so that it can accommodate any number of systems being compared and b) experiment with different ways to estimate the average number of relevant documents found per topic. Ongoing research suggests that the more accurately a data fusion method is able to capture the exact nature of the Authority and Ranking Effects as well as leverage what can be learned as the list depth is gradually increased, the greater its retrieval effectiveness.

Figure 2 shows that for documents only found by one system, which is the great majority of all the documents returned by the systems in the same TREC year, a weak Ranking Effect can be observed (it is strongest for TREC3, since it contains the greatest number of relevant documents of the TREC years studied in this paper). When fusing multiple result lists, this will have the effect of moving highly ranked found by a single system further up in the fused list, which in turn helps to improve the fused list's precision. The use of the "maximum" relevance estimates will cause very highly ranked documents only found by a single system to move further up in the "BestCase" fused list. This helps to improve the precision of the fused list, yet at the same time it also promotes more non-relevant documents, since the probability of such "single system" documents being relevant is usually less than 10%. This can have the effect of pushing out of the fused list of 1000 documents some of the relevant documents that would otherwise be located toward the bottom of the list.

Conclusion

A key challenge in data fusion is how to identify relationships between the different result lists that make it possible to detect all or many of the relevant documents (high recall) and to move relevant documents closer toward the top of the fused result list (high precision). Since retrieval systems searching the same database tend to find similar relevant documents, but not in the same rank positions, data fusion methods benefit from comparing all the documents found by the different retrieval systems. However, a major source of noise is introduced in the fusion process as documents further down in the result lists are considered, since a document's probability of being relevant decreases significantly. Using TREC 3, 6, 7, 8, 12 and 13 data, this paper examined how "gradual skimming", where the number of documents examined in the result lists is gradually increased, can help to ensure that more relevant documents stay and/or raise to the top in the fused list as steadily more documents are examined. The selected ad hoc, robust and web TREC tracks made it possible to investigate the benefits of "skimming" in diverse and difficult settings.

First, empirical data was presented about a document's probability of being relevant as a function of the list depth level used, the number of systems that have found it and its average rank position in the lists containing it. Second, this data was used compute a document's "maximum" relevance probability, which is equal to the maximum of its probability estimates as the list depth is gradually increased. This way the information gained for documents that are contained in the top of the lists and found by many systems is not "diluted and lost" as the list depth is increased and an increasing number of non-relevant documents are found by many systems. Third, the difference between the "maximum" relevance estimates (inferred if the list depths of the top 10, 20, 30, 40, 50, 100, 200, … , 1000 documents are gradually compared) and the "ALL 1000" relevance estimates (equal to the percentage of the top 1000 documents that are relevant) was computed to measure what can be learned as the list depth is gradually increased. It was shown that "gradual skimming" helps to "sharpen" the Ranking Effect, since a document's estimate of being relevant is greatly increased if the document is placed very high up the lists that contain it and it is found by many systems. Fourth, the "maximum" and "ALL 1000" relevance estimates were sorted to produce new fused result lists, called "BestCase" and "ALL1000", respectively. The mean average precision (MAP) scores were calculated for the result lists of "BestCase", "ALL1000" and the best system ("BestSys") in a TREC year, where the average MAP improvement of "BestCase" with respect to "BestSys" was more than 25% for the TREC years studied in this paper. The presented results suggest that "gradual skimming" and using what can be learned as the list depth is gradually increased holds the potential to produce more effective data fusion methods.

ACKNOWLEDGMENTS

The author would like to thank Nick Belkin and Paul Kantor for their help and feedback. This research has been supported by a Rutgers Research Council Grant. The TREC data used in the research reported in this paper has been provided by NIST and can be downloaded at http://trec.nist.gov/.

REFERENCES

Aslam J. A. & Montague M. (2001). Models for metasearch. In SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 276–284, New York, NY, USA, 2001. ACM Press.

Banks D., Over P. & Zhang N. (1999). Blind Men and Elephants: Six Approaches to TREC data. Information Retrieval 1, 7–34.

Belkin, N. J., Kantor, P., Fox, E. A., & Shaw, J. A. (1995). Combining Evidence of Multiple Query Representations for Information Retrieval. Information Processing & Management, 31(3), 431-448.

Callan. J. (2000). Distributed information retrieval. In Croft W.B. (Ed.), Advances in Information Retrieval. (pp. 127-150). Kluwer Academic Publishers.

Foltz, P. & Dumais, S. (1992). Personalized information delivery: An analysis of information-filtering methods. Communications of the ACM, 35, 12:51-60.

Fox, E. & Shaw, J. (1994). Combination of Multiple Searches. 2nd Annual Text Retrieval Conference (TREC-2), NIST, Gaithersburg, MD.

Lee, J. H. (1997). Analyses of Multiple Evidence Combination. Proc. of the 20th Intl. Conf. on Research and Development in Information Retrieval (SIGIR '97), pages 267–276, 1997.

Lillis D., Toolan F., Collier R . & Dunnion J. (2006). ProbFuse: a probabilistic approach to data fusion. In SIGIR ’06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 276–284, New York, NY, USA, 2001. ACM Press.

Montague M. & Aslam J. A. (2002). Condorcet fusion for improved retrieval. In CIKM '02: Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM '02), pages 538–548, New York, NY, USA, 2002. ACM Press.

Saracevic, T. & Kantor, P. (1988). A study of information seeking and retrieving. III. Searchers, searches and overlap. Journal of the American Society for Information Science. 39, 3, 197-216.

Spoerri, A. (2005). How the Overlap Between Search Results of Different Retrieval Systems Correlates with Document Relevance. Proceedings of the 68th Annual Meeting of the American Society for Information Science and Technology (ASIST '05).

Spoerri, A. (2006a). Examining the Authority and Ranking Effects as the Result List Depth Used in Data Fusion is Varied. Information Processing & Management, 43 (4), 1044-1058.

Spoerri, A. (2006b) Authority and Ranking Effects in Data Fusion. Journal of the American Society for Information Science (in Press).

Turtle, H., Croft, B. (1991). Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems, 1991.

Vogt C. C. & Cottrell G. W. (1999). Fusion via a linear combination of scores. Information Retrieval, 1(3):151–173, 1999.

Voorhees, E. & Harman, D. (1994) Overview of the Third Text REtrieval Conference (TREC-3). The Third Text REtrieval Conference (TREC-3), Gaithersburg, MD, USA, 1994. U.S. Government Printing Office, Washington.

Voorhees, E. & Harman, D. (1997) Overview of the Sixth Text REtrieval Conference (TREC-6). The Seventh Text REtrieval Conference (TREC-7), Gaithersburg, MD, USA, 1997. U.S. Government Printing Office, Washington.

Voorhees, E. & Harman, D. (1998) Overview of the Seventh Text REtrieval Conference (TREC-7). The Seventh Text REtrieval Conference (TREC-7), Gaithersburg, MD, USA, 1998. U.S. Government Printing Office, Washington.

Voorhees, E. & Harman, D. (1999) Overview of the Eighth Text REtrieval Conference (TREC-8). The Eighth Text REtrieval Conference (TREC-8), Gaithersburg, MD, USA, 1999. U.S. Government Printing Office, Washington.

Voorhees, E. (2003) Overview of the TREC 2003 Robust Retrieval Track. The Twelfth Text REtrieval Conference (TREC-12), Gaithersburg, MD, USA, 2003. U.S. Government Printing Office, Washington.

Voorhees, E. (2004) Overview of TREC 2004 (TREC-13). The Thirteenth Text REtrieval Conference (TREC-13), Gaithersburg, MD, USA, 2004. U.S. Government Printing Office, Washington.

Wu, S. & Crestani, F. (2003). Methods for Ranking Information Retrieval Systems Without Relevance Judgements. Proceedings of the 2003 ACM Symposium on Applied Computing (SAC '03).