Measuring the Effectiveness of Selective Search Index Partitions without Supervision

Yubin Kim
Carnegie Mellon University
Pittsburgh, PA
yubink@cmu.edu

Jamie Callan
Carnegie Mellon University
Pittsburgh, PA
callan@cs.cmu.edu

ABSTRACT

Selective search architectures partition a document collection into topic-oriented index *shards*, usually using algorithms that have random components. Different mappings of documents into index shards (*shard maps*) produce different search accuracy and consistency, however identifying which shard maps will deliver the highest average effectiveness is an open problem.

This paper presents a new metric, Area Under Recall Curve (AUReC), to evaluate and compare shard maps. AUReC is the first such metric that is independent of resource selection and shard cut-off estimation. It does not require an end-to-end evaluation or manual gold-standard judgements. Experiments show that its predictions are highly-correlated with evaluating end-to-end systems of various configurations, while being easier to implement and computationally inexpensive.

KEYWORDS

selective search, clustering, evaluation, cluster-based retrieval

ACM Reference Format:
Yubin Kim and Jamie Callan. 2018. Measuring the Effectiveness of Selective Search Index Partitions without Supervision. In *2018 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '18), September 14–17, 2018, Tianjin, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3234944.3234952

1 INTRODUCTION

Selective search is a cluster-based distributed search architecture that increases the efficiency of the early stages of a large-scale, pipelined retrieval system [6, 12]. A large collection of D documents is divided into n topically focused index *shards*. At query time, k shards selected by a resource selection algorithm are searched, typically $k << n << D$.

The mapping of documents into shards, called a *shard map*, can be generated using many different *partitioning schemes* (e.g. text similarity, source domain, random) [3, 4, 9] and many resource selection algorithms are available for selective search [1, 8, 14]. The shard map can greatly affect the accuracy of the end-to-end selective search system and is one of multiple such components [7]. Thus, it is desirable to be able to measure the quality of shard maps independent of the other components in the selective search system. In prior work, the effectiveness of shard maps were evaluated with end-to-end selective search systems using queries with relevance judgements [9]. This method is affected by the choices made in other system components. Building an end-to-end system and gathering relevant judgements also can be costly. A measurement of shard map quality independent of other system components would provide better diagnostic information and allow a selective search system to be tuned more easily and quickly.

This paper introduces a method for estimating shard map effectiveness called Area Under Recall Curve (AUReC). AUReC is calculated independent of other selective search components. It requires only a set of queries and the results of the full collection retrieval for those queries; from this data, AUReC can be used to quickly identify good shard maps for an efficient selective search system. To our knowledge, AUReC is the first to fully decouple shard map evaluation from other selective search components.

We compare AUReC against end-to-end system evaluations using three different resource selection algorithms with relevance judgements to answer the following research questions:

RQ 1. *Can AUReC generate a relative ordering of multiple shard maps that is consistent with orderings generated by end-to-end system evaluations?*

RQ 2. *Is AUReC consistent with end-to-end evaluations on whether differences between two shard maps are statistically significant?*

RQ 3. *Is AUReC robust when compared to end-to-end systems using different resource selection algorithms; compared to a variety of IR metrics at different retrieval depths; when the search engine generating the top-k retrieval is weaker; and when it is calculated with a different set of queries than the end-to-end evaluation queries?*

2 RELATED WORK AND MOTIVATION

A *shard* is a search index that contains a subset of the total document collection. A *shard map* is the mapping that describes to which shard each document in the collection belongs, which is determined by the *partitioning scheme*, a method or technique for generating shard maps, e.g. text-similarity clustering, grouping documents by geography, or splitting a collection randomly.

Three areas of prior research created and used shard maps. *Cluster-based retrieval* systems sought to improve accuracy by dividing the corpus into topical clusters and searching just within cluster(s) most relevant to each query [11, 19]. A common method of assessing partition quality was to measure search results produced by a complete ("end-to-end") system that selects the optimal

ICTIR '18, September 14–17, 2018, Tianjin, China

ACM ISBN 978-1-4503-5656-5/18/09…$15.00
https://doi.org/10.1145/3234944.3234952

cluster [11, 18]. *Distributed retrieval* systems partition a large corpus into shards so that processing effort can be divided among multiple processors to improve response time (*latency*). The efficiency enabled by different partitioning schemes is well-studied [3, 4]. However, the effectiveness of shard maps has received little attention. *Selective search* improves the efficiency of large scale search while maintaining the same quality as searching the entire collection (*exhaustive search*) [13]. A *resource selection* algorithm selects the shards to be searched for each query [1, 8, 14, 17].

Prior work on selective search evaluated shard maps using the end-to-end retrieval accuracy (e.g., MAP, NDCG@k, or P@k) of a complete system using that shard map. This methodology is realistic, but requires manual relevance assessments and the results are more representative of the effectiveness of the particular choice of components and how well they are tuned. A thorough system designer would generate many shard maps and test each one against a variety of system configurations [9]. This is cumbersome, so it is not done often. There is a need for shard map evaluation that is decoupled from other system components and relevance judgements to enable rapid tuning of new selective search systems.

One method of decoupling evaluation from relevance judgments is to compare the results of a less thorough (less expensive) search to the results of a more thorough (more expensive) search that exhaustively searches the entire index. The more thorough, exhaustive search system is treated as the 'gold standard' that the less thorough system is intended to mimic. Exhaustive search has been used to measure a new method's accuracy [5] and efficiency [12]. It has also been used as a target to train a supervised resource selection algorithm [8]. Clarke et al. [6] used overlap with exhaustive retrieval to evaluate the effectiveness of early-stage filters in a multi-stage retrieval system. In a selective search context, this is equivalent to evaluating the end-to-end selective search system by overlap with exhaustive results. We are the first to apply this concept to scoring shard maps, independent of the other stages in selective search.

To score shard maps, we borrow ideas from French and Powell [10], who describe a recall-like measure, $\hat{R}_k(E,B)$, used to compare the performance of resource selection algorithms in federated search. Let B represent an ideal ordering for query q of a shard map containing n shards; and E the estimated ordering produced by a resource selection algorithm. Let B_i represent the number of relevant documents in the i'th ranked shard in B, and similarly for E_i and E. It should be the case that $B_i \geq B_{i+1}$ and this ordering of shards is called relevance-based ranking (RBR). It is not generally the case that $E_i \geq E_{i+1}$.

$$\hat{R}_k(E,B) = \frac{\sum_{i=1}^{k} E_i}{\sum_{i=1}^{n} B_i}$$

French and Powell plot $\hat{R}_k(E,B)$ from $k = \{1 \ldots n\}$ to visualize and compare the effectiveness of various resource selection algorithms. Our work develops a variation of the $\hat{R}_k$ curve to evaluate shard maps quantitatively, thereby transforming a visualization aid into a powerful diagnostic tool.

3 AUREC

Our goal is a method of evaluating shard maps that that does not require relevance judgements and is independent of the resource selection algorithm, other system components, and parameter settings. While different resource selection algorithms have some differences, their shared goal is to find the shards with the most relevant documents and they tend to return similar shards. Thus, our hypothesis is that the quality of a shard map can be treated as an inherent property that can be measured on its own.

Selective search reduces costs by searching as few shards as possible while trying to produce results that are as accurate as searching all shards (*exhaustive search*). For a given query q, an ideal shard map i) groups the important documents D_q in as few shards as possible, and ii) places them in shards that contain many similar documents. The first condition enables most shards to be ignored. The second condition makes the right shards easy to recognize.

There are different ways to define D_q. French and Powell [10] defined it as documents that have been judged relevant by human assessor. However, we would like to avoid reliance on human relevance judgements. In addition, considering only the relevant documents when evaluating shard maps creates data sparsity issues. Often only a small number of queries with relevance judgements are available and some queries have very few relevant documents. The sparsity of data can produce high variance results [7].

Alternatively, note the goal of selective search to meet, rather than to exceed, the accuracy of exhaustive search, just at a much lower cost. Thus, D_q can be defined as the documents that an exhaustive search system would return for the query. For the exhaustive search system, a well-tuned, strong ranker is preferred (e.g. a trained learning to rank system) for more accurate signals on quality of the documents. This method avoids the sparsity issues of relevance judgements; our experiments showed that the additional data generated by this method increases the robustness of AUReC.

Given a shard map p containing n_p shards, a query q, and D_q, the documents that should be retrieved for query q, we can calculate the maximum possible recall of D_q when searching up to a limit of k shards as follows. For query q, let $count(D_q, s_i^p), i = \{1 \ldots n_p\}$ represent the number of documents in D_q present in shard s_i^p, where the shards were ordered such that $count(D_q, s_i^p) > count(D_q, s_{i+1}^p)$. $R_q(p,k)$ is the percentage of documents in D_q that appear in the first k shards of shard map p, that is:

$$R_q(p,k) = \begin{cases} 1 & |D_q| = 0 \\ 0 & |D_q| > 0, k = 0 \\ \frac{\sum_{i=1}^{k} count(D_q, s_i^p)}{|D_q|} & |D_q| > 0, k = \{1 \ldots n_p\} \end{cases}$$

R_q loosely represents recall (what portion of relevant documents can be found in the given shards) and k represents efficiency (searching fewer shards is more efficient). The relationship between the recall and efficiency in selective search can be described in graphical form by plotting k vs. $R_q(p,k)$ for $k = \{0 \ldots n_p\}$. It is a convex, monotonically increasing curve, and the x-axis is normalized to be between $[0,1]$ by dividing by n_p. When comparing multiple shard maps for a given query, the curves of effective shard maps will be higher and to the left of less effective shard maps.

Our method calculates the area under this recall versus fraction of shards searched curve and is named *Area Under Recall Curve* (AUReC). AUReC of query q for shard map p can be calculated as follows, using the formula for calculating the area of trapezoids:

$$AUReC_q(p) = \int R_q(p,k)\,dk = \frac{1}{n_p} \sum_{k=0}^{n_p-1} \frac{R_q(p,k) + R_q(p,k+1)}{2}$$

AUReC scores have the range $[0.5, 1.0]$. 0.5 indicates that the documents from D_q are completely evenly distributed across the shards and is the worst possible score. Scores closer to 1.0 indicate that the documents from D_q are tightly clustered in very few shards. 1.0 occurs if D_q is empty.

$R_q(p,k)$ is comparable to $\hat{R}_k(B,B)$ [10], where B is the ideal ordering of shards in shard map p (refer to Section 2). It differs by using exhaustive search runs rather than human-judged relevant documents. It also has defined values when $|D_q| = 0$ and $k = 0$, whereas the equivalent in $\hat{R}_k$ are undefined. The $k = 0$ case for R_q is critical when comparing shard maps that contain a different number of shards, and it enables us to determine that a shard map p_{100} consisting of 100 shards with $R_q(p_{100}, 1) = 1$ is better than a shard map p_2 consisting of 2 shards with $R_q(p_2, 1) = 1$, since the important documents are more densely clustered. Without this modification, AUReC plotted from $k = \{1 \dots n_p\}$ (as opposed to $k = \{0 \dots n_p\}$) is biased towards shard maps with fewer shards.

AUReC is calculated on a per query basis similar to traditional information retrieval metrics and can be averaged across queries to generate a summary value. Paired significance testing is possible. In this paper, unless otherwise specified, the AUReC score refers to the $AUReC_q$ averaged across all queries in the query set.

Note that no relevance judgements are used to calculate AUReC and no other component of selective search needs to be implemented. In particular, there is no need to tune a shard cut-off value or to pick an efficiency level for resource selection. Instead, the shard-recall curve describes how well a given shard map performs for a query over all cut-offs. Using only the results of an exhaustive search engine, AUReC can be used to quickly tune a new, effective selective search system to replace the less efficient exhaustive search system. An open source implementation is available[1].

4 EXPERIMENTAL SETUP

4.1 Data and query sets

The experiments were conducted on two large web data sets. Gov2 is a TREC collection containing 25 million `.gov` domain websites[2]. ClueWeb09 Category B (ClueWeb09 B) is the first 50 million documents of the ClueWeb09 web crawl[3]. The shard maps used in evaluation were generated by Dai et al. [9] using three different methods: `KLD-Rand`, `QKLD-Rand` and `QKLD-Qinit`, where each successive method produces shard maps of higher quality than the last. These partitioning schemes have random components and different random seed initializations produce different shard maps. Using 10 different random seeds, the authors generated 10 shard maps for each method for a total of 30 shard maps per data set, 60 shard maps overall[4]. This data set contains both shard maps that of similar and different qualities, and is ideal for testing a new evaluation measure which should be able to distinguish the two cases.

To generate an end-to-end system evaluation, we use queries and relevance judgements published by TREC. Gov2 was evaluated with the TREC Terabyte Track queries spanning 2004–2006 and ClueWeb09 B with the TREC Web Track queries spanning 2009–2012. The relevance judgements for the TREC Web Track queries were filtered such that only documents in Category B were present. Furthermore, queries with zero judged relevant documents were removed from consideration (topics 20, 112, 143, 152 in the Web Track queries), since an end-to-end system evaluation of these queries cannot be used to rank shard maps. AUReC is calculated using the same queries but the relevance judgements are not used.

4.2 Document retrieval

Calculating AUReC requires knowing a set of documents that should be retrieved for a given query q, D_q. We create D_q using the top 1000 results of a strong retrieval engine to create the best possible substitute for relevance judgements. To create competitive near state-of-the-art results, SlideFuse-MAP [2], a data fusion technique was used to fuse the top 10 runs submitted to TREC in each year as ordered by MAP. In Gov2, the fused result had a MAP of 0.41, up from the 0.38 of the best submitted run. In ClueWeb09 B, the fused run had a MAP of 0.29, up from 0.25 of the best submitted run. The scores of SlideFuse-MAP were also used to rank the documents in the end-to-end selective search system. Thus, the evaluation of an end-to-end system uses retrieval results of quality on par with the information used by AUReC, enabling apple-to-apple comparisons.

4.3 Resource selection and baselines

To compare AUReC to the current state-of-the-art, we must create several different configurations of end-to-end selective search systems, requiring a few different resource selection algorithms. Three resource selection algorithms were used in the experiments: Taily [1], a term-based algorithm; Rank-S [14], a sample-based algorithm; and oracle method, relevance-based ranking (RBR) at various static shard cut-offs. As discussed in Section 2, RBR orders shards based on the relevance judgements of the query. Shards with the more relevant documents are ranked higher.

The parameters for Rank-S and Taily were set using the values from Kim et al. [12] and are summarized in Table 1. When searching the CSI for Rank-S, SlideFuse-MAP is used to rank the documents. While Rank-S and Taily produce dynamic shard cut-offs (i.e. number of shards to search per query is built-in to the method and varies by query), RBR does not. Therefore, we evaluate RBR selective search systems at three different static shard cut-offs, $k = \{1, 3, 5\}$, which is a common range for shard cut-offs used in prior work [12] and expresses a range of efficiency levels.

The three algorithms used for end-to-end evaluation present two different families of resource selection, two different strategies for shard cut-offs, and the upper bound of resource selection performance. By experimenting with three dissimilar resource selection algorithms, we demonstrate the robustness of AUReC.

[1] http://anonymized.com
[2] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html
[3] https://lemurproject.org/clueweb09/
[4] Downloaded from http://boston.lti.cs.cmu.edu/appendices/CIKM2016-Dai/

Data set	Algorithm	Parameters
Gov2	Taily	$n = 400$, $v = 50$
	Rank-S	1% CSI sample, $base = 3$
ClueWeb09 B	Taily	$n = 400$, $v = 50$
	Rank-S	1% CSI sample, $base = 5$

Table 1: Parameter settings for Rank-S and Taily used in the end-to-end selective search system evaluations.

As AUReC is the first method to completely decouple shard map evaluation from other selective search components, there are no apple-to-apple baselines to which it can be directly compared. However, a common heuristic used in cluster-based retrieval systems is optimal cluster effectiveness, the effectiveness of performing retrieval on the one shard that contains the most relevant documents [18]. This is exactly equivalent to an end-to-end system evaluation using RBR resource selection with $k = 1$. Although this method has dependencies on relevance judgements and shard cut-offs, we include this common option as a baseline in our experiments to provide context and a point of comparison. Note that variations of this baseline that do not use relevance judgements can be easily created as noted in Section 2. However, these baselines are weaker because they use less information. Thus, we chose to present the strictly stronger baseline of optimal cluster effectiveness.

4.4 Metrics

To establish AUReC as a robust method, we compare it to end-to-end selective search evaluations of multiple different settings. An end-to-end evaluation score $EtE(p)$ of shard map p is generated by evaluating queries on a selective search system based on shards dictated by shard map p. Multiple versions of $EtE(p)$ are possible dependent on the type of resource selection algorithm used and the evaluation metric used to score the results of selective search.

The primary evaluation metric used in this work is Precision at 1000 (P@1000); as a first-stage retrieval system, selective search must return deeper result lists to support later stage re-rankers [16]. Other metrics are explored in Section 5.2.

AUReC and the end-to-end system evaluation are compared in two ways: list-wise and pair-wise. In the list-wise comparison, the correlation of the scores of $EtE(p_i)$ and $AUReC(p)i)$ are calculated using Pearson's r. Pearson's r was chosen over non-parametric, ordinal methods such as Kendall's τ so that the correlation in the difference in the magnitude of the scores were measured, not just the relative rankings. This is important as the shard maps used in the experiments were generated with three different methods, with each method generating ten shard maps each. Thus, shard maps generated by the same method are expected to have similar scores, whereas shard maps generated by different methods should have larger score differences.

In the pair-wise comparison, for each pair of shard maps (p_i, p_j) where $i, j = \{1 \ldots 30\}, i \neq j$, we determine the ordering of $AUReC(p_i)$ and $AUReC(p_j)$ and whether the difference is statistically significant under a paired two-tailed t-test using significance level $\alpha = 0.05$. This is repeated for $EtE(p_i)$ and $EtE(p_j)$. Given that there are 30 shard maps for a data set, the total number of pair-wise comparisons is $\binom{30}{2} = 435$. Based the decisions of AUReC and end-to-end evaluation on the direction and significance of the differences between p_i and p_j, a contingency table can be built as shown in Figure 1. Note that cells d and f are split in half. This is to indicate two scenarios; in $d1$ and $f2$, end-to-end evaluation and AUReC agreed in the direction of the difference, but only AUReC found the difference to be significant. In $d2$ and $f1$, the two methods disagreed on the direction of differences.

EtE \ AUReC	$p_i > p_j$	No sig. diff.	$p_i < p_j$
$p_i > p_j$	a	b	c
No sig. diff.	$d1$	e	$f1$
	$d2$		$f2$
$p_i < p_j$	g	h	i

Figure 1: Contingency table for pair-wise comparison of shard maps using AUReC and end-to-end system evaluation (EtE).

Three summary metrics are reported from this table. First is *Pairs recall* $= \frac{a+i}{a+b+c+g+h+i}$. This indicates the fraction of statistically significant differences found by the end-to-end evaluation that was also recovered by AUReC. Second is *Overlap* $= \frac{a+e+i}{435}$ the number of shard map pairs where the end-to-end system and AUReC agreed exactly on the direction and significance of the differences. Lastly, *Additional pairs* $= \frac{d1+f2}{435}$ is reported (abbreviated as *Addit. pairs*. As described above, this indicates situations where AUReC was able to detect significant differences where the end-to-end evaluation could not, due to the greater amount of information used by AUReC and the sparsity of relevance judgements used by the end-to-end evaluation. Additionally, *Overlap+Addit. pairs* is reported together, loosely representing the total agreement between AUReC and end-to-end evaluation. Other cells indicate varying levels of disagreement between the two methods. Cells c, g indicates strong disagreements where the two methods disagreed on the direction of significance; b, h indicate statistically significant differences that were missed by AUReC; and $d2, f1$ are situations of slight disagreement where AUReC specifies statistically significant differences, but in a different direction from the end-to-end evaluation.

5 EXPERIMENTAL RESULTS

5.1 Comparison with end-to-end systems

AUReC and optimal cluster effectiveness (equivalent to end-to-end evaluation with RBR, $k = 1$) are compared against end-to-end selective search systems utilizing various resource selection techniques. The results are presented in Table 2.

The experiments demonstrate that AUReC is highly correlated with all end-to-end system evaluations in both list-wise and pair-wise comparisons. The high Pearson's r indicates that the AUReC and end-to-end evaluation scores order shard maps similarly throughout the entire score range. Furthermore, the best shard map selected by AUReC and the end-to-end systems were either identical or were shard maps where there were no statistically significant differences, indicating agreement in the highest end of the scores.

EtE	r	Pairs recall	Overlap+Addit. pairs
Optimal cluster effectiveness			
Rank-S	0.94	176/236 = 0.75	0.84 + 0.03 = 0.86
Taily	0.90	161/184 = 0.88	0.89 + 0.06 = 0.94
RBR ($k = 1$)	-	-	-
RBR ($k = 3$)	0.95	181/228 = 0.79	0.88 + 0.01 = 0.89
RBR ($k = 5$)	0.91	173/215 = 0.80	0.87 + 0.03 = 0.90
AUReC			
Rank-S	0.96	221/236 = 0.94	0.71 + 0.21 = 0.93
Taily	0.92	181/184 = 0.98	0.65 + 0.23 = 0.88
RBR ($k = 1$)	0.94	185/187 = 0.99	0.66 + 0.27 = 0.93
RBR ($k = 3$)	0.98	219/228 = 0.96	0.72 + 0.23 = 0.96
RBR ($k = 5$)	0.96	206/215 = 0.96	0.69 + 0.25 = 0.94

(a) ClueWeb09 B

EtE	r	Pairs recall	Overlap+Addit. pairs
Optimal cluster effectiveness			
Rank-S	0.95	169/226 = 0.75	0.84 + 0.03 = 0.87
Taily	0.86	115/149 = 0.77	0.77 + 0.15 = 0.92
RBR ($k = 1$)	-	-	-
RBR ($k = 3$)	0.93	169/224 = 0.75	0.85 + 0.02 = 0.87
RBR ($k = 5$)	0.89	166/229 = 0.72	0.82 + 0.03 = 0.85
AUReC			
Rank-S	0.95	197/226 = 0.87	0.89 + 0.05 = 0.93
Taily	0.84	121/149 = 0.81	0.71 + 0.20 = 0.92
RBR ($k = 1$)	0.93	166/181 = 0.92	0.85 + 0.12 = 0.96
RBR ($k = 3$)	0.93	193/224 = 0.86	0.87 + 0.05 = 0.92
RBR ($k = 5$)	0.92	195/229 = 0.85	0.87 + 0.03 = 0.90

(b) Gov2

Table 2: Comparison of AUReC against the evaluation of end-to-end (EtE) systems with various resource selection algorithms. The end-to-end systems used P@1000 to measure the effectiveness of the shard maps. r is Pearson's correlation. Optimal cluster effectiveness is a commonly used heuristic in prior work that is identical to an end-to-end system evaluation using RBR $k = 1$.

That is, AUReC and end-to-end system evaluations will make very similar decisions when picking the best shard map.

AUReC also had a high *Pairs recall*, indicating that it recovered most of the statistically significant differences between shard map pairs found by the end-to-end systems. Finally, the high *Overlap+Addit. pairs* sum shows that AUReC and end-to-end evaluations mostly agree on the direction of differences between pairs of shard maps. In addition, while not shown in the table, there were no pairs in which AUReC and end-to-end evaluations disagreed in the direction of statistical significance in any of the settings; that is, there were no pairs of shard maps (p_i, p_j) where AUReC believed that p_i was statistically significantly better than p_j but an end-to-end evaluation believed vice versa.

Furthermore, AUReC correlates better with end-to-end systems than optimal cluster effectiveness, across both datasets and nearly all settings and comparisons, including systems that use RBR with different cut-offs. In particular, AUReC is superior in replicating statistically significant differences between pairs of shard maps (*Pairs recall*). This is because optimal cluster effectiveness produces high query variance. Other resource selection algorithms select 3–5 shards on average. Selecting only one shard means optimal cluster effectiveness sees fewer relevant documents and has less information to make judgements. Consequently, compared to AUReC, optimal cluster effectiveness produces higher variance and is less able to distinguish statistically significant differences and is less correlated with other systems overall. While often used in prior research [18], it is a less reliable metric.

There are two places where optimal cluster effectiveness correlates better. First is the pair-wise comparisons against a Taily system in ClueWeb09 B, where optimal cluster effectiveness has a higher overall agreement (*Overlap+Addit. pairs*) than AUReC. This was because Taily also produces high variance results [7] and most of the *Overlap* was from pairs where both methods did not find significant differences (58% of *Overlap* pairs). In contrast, most of the *Overlap* between AUReC and Taily system was from concordance on statistically significant differences (64% of *Overlap* pairs), which is more informative.

Optimal cluster effectiveness also produced a slightly higher Pearson's r than AUReC when compared to the Taily system in Gov2. However, this difference is less meaningful; due to the high variance of in the Taily systems, most of the pair-wise differences between shard maps in the end-to-end evaluation are not significant. This means that in an ordered list of shard maps, there may be sections where multiple shard maps are interchangeable in order because the confidence intervals of the shard maps' scores overlap each other. This introduces noise when calculating Pearson's r against a shard map ordering created by a Taily system and thus small differences become less meaningful.

The TREC Web Track queries have shallow judgement pools and produce high variance due to the dearth of signals [15]. In the work that described the shard maps used in our experiments, Dai et al. [9] discovered that while the differences in the means for the end-to-end evaluation scores of the three categories of partitioning schemes (`KLD-Rand`, `QKLD-Rand`, and `QKLD-QInit`) were greater in ClueWeb09 B than Gov2, the differences were sometimes not statistically significant in ClueWeb09 B due to the higher variance. It was unclear which data set benefits more from the improved partitioning scheme. This can now be answered using AUReC.

AUReC is calculated based on 1000 retrieved documents and uses more information than evaluating an end-to-end retrieval with sparse relevance judgements. Thus, AUReC finds significant differences where an end-to-end system evaluation cannot. Note the amount of *Addit. pairs* found by AUReC on the ClueWeb09 B dataset. These are shard map pairs where the end-to-end evaluation saw the same direction of difference as AUReC but could not conclude their statistical significance. The end-to-end evaluation found as few as 42% of shard map pairs to be significantly different (the denominator of the *Pairs recall* column), but AUReC found significant differences in 76% of shard map pairs in ClueWeb09 B (cells $a + d + g + c + f + i$ in Figure 1). In contrast, Terabyte Track queries for Gov2 have deeper pooling for the relevance judgements, producing stable results. AUReC finds fewer *Addit. pairs* in Gov2 and found significant differences in 50% of shard map pairs in Gov2,

similar to end-to-end evaluations. Thus, thanks to evidence supplied by AUReC, we can now conclude that the choice of partitioning scheme affects ClueWeb09 B more strongly than Gov2.

AUReC typically generates fewer *Addit. pairs* in Gov2. However, when the resource selection algorithm used by the end-to-end system has high variance (i.e. Taily and RBR $k = 1$) AUReC finds significant differences that the end-to-end systems miss.

With this experiment, RQ 1, RQ 2 and a portion of RQ 3 were answered. The ordering of shard maps generated by AUReC is highly correlated with the ordering determined by end-to-end system evaluations (RQ 1). In pair-wise comparisons, AUReC found significant differences in pairs of shard maps where the end-to-end system found differences. In addition, AUReC was able to discover additional pairs of significantly different shard maps, where end-to-end systems were unable to ascertain the significance (RQ 2). Finally, AUReC performed well across two data sets and three different resource selection algorithms, and substantially better than the common baseline heuristic used by prior cluster-based retrieval research (RQ 3). Further sections elaborate on RQ 3 and examine AUReC under less ideal scenarios to test its robustness.

5.2 Different metrics and evaluation depths

So far P@1000 was used for the end-to-end system evaluation, which is a metric directly comparable to AUReC. AUReC is calculated with a top 1000 retrieval ordering of shards that maximizes the recall. Because there are 1000 documents in D_q, in most cases this is equivalent to maximizing P@1000 (where a document is considered 'relevant' if it is in D_q). However, not all metrics are so directly related to AUReC. If a metric takes rank position into account, a recall-oriented ordering of shards is not always optimal. In fact, for metrics such as NDCG and MAP, the optimal solution set of k shards is not always a subset of the optimal set for $k + 1$ shards. To explore how well AUReC correlates with metrics that are less directly related, AUReC is compared against end-to-end systems evaluations using two rank sensitive metrics, MAP and NDCG at two different evaluation depths, 1000 and 100. The results are presented in Table 3. The table shows results for Rank-S end-to-end systems for brevity. Other resource selection choices behaved similarly, to the exception of RBR $k = 1$.

Optimal cluster effectiveness (i.e. RBR $k = 1$) was much less robust than AUReC and experienced a sharp drop in correlation in every setting and method of comparison. Both the change of metric and the change of evaluation depth adversely affected the correlation and a combination of both produced dismal results, especially in ClueWeb09 B (*Pairs recall* was 0.04 for MAP@100).

AUReC remains highly correlated with MAP and NDCG, and metrics evaluated at a shallower depth. In particular, differences in the type of evaluation metric did not impact the correlation in a major way. In Gov2, AUReC was slightly *better* correlated with NDCG, a rank sensitive metric, than Precision.

Evaluation depth had a stronger effect on the correlation of AUReC, yet still not necessarily negative. The results show that at shallower evaluation depths, the variance of the end-to-end systems increase, as seen by the lower number of statistically significant shard map pairs found by the end-to-end system, e.g. a P@1000 Gov2 system found 226 pairs with significant differences whereas

Metric	r	Pairs recall	Overlap+Addit. pairs
Optimal cluster effectiveness			
P@1000	0.94	176/236 = 0.75	0.84 + 0.03 = 0.86
NDCG@1000	0.89	97/248 = 0.39	0.64 + 0.01 = 0.65
MAP@1000	0.78	17/217 = 0.08	0.53 + 0.01 = 0.54
P@100	0.86	116/204 = 0.57	0.76 + 0.03 = 0.79
NDCG@100	0.75	10/214 = 0.05	0.53 + 0.00 = 0.53
MAP@100	0.62	8/190 = 0.04	0.57 + 0.01 = 0.58
AUReC			
P@1000	0.96	221/236 = 0.94	0.71 + 0.21 = 0.93
NDCG@1000	0.94	229/248 = 0.92	0.72 + 0.19 = 0.91
MAP@1000	0.91	204/217 = 0.94	0.68 + 0.22 = 0.90
P@100	0.90	194/204 = 0.95	0.66 + 0.21 = 0.88
NDCG@100	0.90	196/214 = 0.92	0.65 + 0.23 = 0.88
MAP@100	0.88	175/190 = 0.92	0.61 + 0.27 = 0.88

(a) ClueWeb09 B

Metric	r	Pairs recall	Overlap+Addit. pairs
Optimal cluster effectiveness			
P@1000	0.95	169/226 = 0.75	0.84 + 0.03 = 0.87
NDCG@1000	0.94	153/221 = 0.69	0.78 + 0.06 = 0.84
MAP@1000	0.94	126/228 = 0.55	0.73 + 0.03 = 0.77
P@100	0.84	68/173 = 0.39	0.73 + 0.03 = 0.76
NDCG@100	0.84	50/131 = 0.38	0.77 + 0.05 = 0.81
MAP@100	0.80	36/115 = 0.31	0.79 + 0.03 = 0.82
AUReC			
P@1000	0.95	197/226 = 0.87	0.89 + 0.05 = 0.93
NDCG@1000	0.96	196/221 = 0.89	0.89 + 0.05 = 0.94
MAP@1000	0.96	197/228 = 0.86	0.88 + 0.05 = 0.93
P@100	0.91	145/173 = 0.84	0.77 + 0.16 = 0.93
NDCG@100	0.88	112/131 = 0.85	0.71 + 0.23 = 0.95
MAP@100	0.86	95/115 = 0.83	0.67 + 0.27 = 0.95

(b) Gov2

Table 3: Comparison of AUReC against a Rank-S end-to-end system at different rank depths.

the P@100 system found only 173. When compared against the shallower, more variable system evaluations, AUReC found more *Addit. pairs* and Pearson's r was reduced. This is in line with the observations from Table 2 where similar effects were present when AUReC was compared to a Taily-based end-to-end system which has higher variance.

The increased variance was more pronounced in Gov2 than ClueWeb09 B. The queries for Gov2 have more relevant documents per query than those of ClueWeb09 B (182 vs. 50) and the pooling during the assessing phase was deeper. Gov2 queries have more judged documents further down in the ranked list than ClueWeb09, and thus were impacted more by the loss of this information with the use of a shallower evaluation depth.

Results for shallower depths were not reported because the increase in variance makes the results less interesting. For example, in ClueWeb09 B with P@10, more than 75% of shard map pairs have

Type	MAP	r	Pairs recall	Overlap+Addit. pairs
SDM	0.21	0.95	213/236 = 0.90	0.82 + 0.12 = 0.93
BOW	0.20	0.95	214/236 = 0.91	0.83 + 0.11 = 0.94

(a) ClueWeb09 B

Type	MAP	r	Pairs recall	Overlap+Addit. pairs
SDM	0.32	0.95	204/226 = 0.90	0.89 + 0.06 = 0.94
BOW	0.29	0.92	185/226 = 0.82	0.86 + 0.04 = 0.90

(b) Gov2

Table 4: Comparsion of AUReC scores against Rank-S end-to-end system evaluation using P@1000, when D_q was generated from weaker rankers. MAP column is the accuracy of the weak rankers (compare with Section 4.2).

no significant differences in the Rank-S system. With such noise, list-wise comparison using Pearson's r becomes less informative and any cluster effectiveness metric that is high in variance would have a good pair-wise overlap with the end-to-end evaluation.

This experiment demonstrates that AUReC is well-correlated with different metrics and different evaluation depths. AUReC is a robust, low-variance method that can identify significant differences better than end-to-end system evaluations that have high variance configurations, e.g. using a shallow evaluation depth.

5.3 Weaker retrieval engine

In previous sections, AUReC was calculated based on a very high-quality ranker. However, results from a strong retrieval engine may not be easy to obtain. In this experiment, we investigate the results of using a good, but less accurate retrieval engine to generate D_q.

We use the Indri[5] search engine to generate D_q using default search parameters. The indexes for both Gov2 and ClueWeb09 B were stopped and stemmed using the Indri stopword list[6] and the Krovetz stemmer. Runs of two different types were generated: bag-of-word queries and sequential dependency model (SDM) queries with 0.8 weight given to the original query and 0.1 to the bigrams and 0.1 to the unordered windows. In a post-retrieval step, ClueWeb09 B results were filtered for spam, removing any documents that had a Waterloo Fusion spam score[7] of below 50. AUReC scores were generated these weaker runs and were compared to a full end-to-end selective search evaluation, unchanged from Table 2. The accuracy of the Indri-based runs and correlation of the resulting AUReC scores are presented in Table 4. For brevity, only Rank-S end-to-end system results are shown. Other resource selection methods displayed similar trends.

Despite the fact that the Indri-based runs were significantly worse (approximately −30% change in MAP scores) than the data fusion run, the resulting AUReC scores were well-correlated with the end-to-end evaluation. However, there is some degradation of results. The bag-of-words Gov2 run especially saw a reduction in correlation scores across all methods of comparison.

[5]https://www.lemurproject.org/indri/
[6]http://www.lemurproject.org/stopwords/stoplist.dft
[7]http://plg.uwaterloo.ca/ gvcormac/clueweb09spam/

The ClueWeb09 B results were also degraded, but in a more subtle way. While the final correlation scores were mostly similar, the composition of the overall agreement scores in the Indri runs shifted; the weaker Indri based runs were less able to uncover additional significant differences and instead agreed more with the end-to-end system that the differences were not significant. For example, AUReC scores based on the data fusion run and Indri SDM run both agreed with the end-to-end system evaluation on 93% of all shard map pairs. However, 21% of pairs were *Addit. pairs* in the data fusion run whereas only 12% of pairs were new significance discoveries in the Indri SDM run.

Although AUReC using the Indri-based runs produced slightly worse correlation with the end-to-end evaluations, the reduction of effectiveness was much less than the difference in accuracy of the runs. This implies that while strong retrieval results can improve AUReC, the robustness of metric indicates that a good out-of-the-box retrieval engine is sufficient to calculate reliable scores.

5.4 Different queries

The final experiment further tests the robustness of AUReC by experimenting with queries that lack relevance judgements and high accuracy runs, using an out-of-the-box search engine. The experiments use 10,000 queries from the 2009 TREC Million Query Track (MQT). Queries which were duplicates of the Terabyte Track query set or Web Track query set were removed. The queries were then expanded into SDM queries and 1000 results were retrieved per query from the Indri index using settings and post-retrieval spam filtering for ClueWeb09 B as described in Section 5.3.

In order to explore the effect of the number of queries on the reliability of AUReC, AUReC scores were generated using various sized subsets of the MQT queries. These runs were then compared against end-to-end systems that are, as usual, evaluated with queries that have relevance judgements. The results are presented in Table 5. For brevity, only the Rank-S end-to-end system results are shown. Other resource selection methods displayed similar trends.

AUReC generalizes well to different queries and the scores calculated are well-correlated with a full system evaluation. When the best result of Table 5 is compared to the Rank-S entry of Table 2, in ClueWeb09 B, AUReC computed with MQT queries converges to a slightly lower Pearson's r than what could be achieved if the TREC queries are used. In Gov2, the convergence value is slightly higher than in Table 2. Both values remain high and suggest that AUReC shard map rankings remain robust when queries change.

As the number of queries increase, there is a general increase in correlation between AUReC and the end-to-end system evaluation. However, there is a clear diminishing returns effect on list-wise comparisons as more queries are used to calculate AUReC scores. Pearson's r hits a saturation point quickly and moves slowly or not at all with increasing queries. The pair-wise correlation comparisons continue to grow somewhat and *Addit. pairs* in particular continues to grow more with additional queries. This is expected. As additional queries are utilized, the mean of the AUReC scores for a given shard map converge quickly. Thus, the shard map's position in an ordered list shifts rarely. However, with more evidence, the confidence interval around the mean continues to shrink and AUReC uncovers more significant differences. Note that *Overlap*

decreases because e in Figure 1 decreases (i.e. cases where both AUReC and end-to-end believe the differences between the two shard maps are not significant).

We have calculated AUReC with queries without relevance judgements or high accuracy runs, using an out-of-the-box search engine. The results were highly correlated with full end-to-end system evaluations. When this less optimal configuration is compared to the optimal cluster effectiveness baseline from Section 5.1 which used the same queries and relevance judgements as the end-to-end system evaluations, AUReC had better pair-wise correlation and better or near-equal list-wise correlation while using a different set of queries and no relevance judgements, demonstrating that it is robust and reliable.

Num of qrys	r	Pairs recall	Overlap+Addit. pairs
100	0.89	186/236 = 0.79	0.79 + 0.08 = 0.86
400	0.92	202/236 = 0.86	0.77 + 0.12 = 0.89
800	0.93	217/236 = 0.92	0.70 + 0.21 = 0.91
1000	0.93	219/236 = 0.93	0.66 + 0.24 = 0.90
10000	0.93	224/236 = 0.95	0.58 + 0.28 = 0.86

(a) ClueWeb09 B

Num of qrys	r	Pairs recall	Overlap+Addit. pairs
100	0.89	178/226 = 0.79	0.72 + 0.12 = 0.84
400	0.93	211/226 = 0.93	0.74 + 0.16 = 0.90
800	0.95	218/226 = 0.96	0.71 + 0.20 = 0.91
1000	0.94	218/226 = 0.96	0.69 + 0.21 = 0.90
10000	0.96	225/226 = 1.00	0.58 + 0.28 = 0.86

(b) Gov2

Table 5: Comparsion of AUReC and Rank-S end-to-end system evaluation using P@1000, when using varying number of MQT queries. The end-to-end system was evaluated with TREC queries, as usual.

6 CONCLUSION AND RECOMMENDATIONS

Prior work evaluated shard maps by measuring the accuracy of end-to-end selective search systems. This is a cumbersome method that relies on relevance judgements and is sensitive to the specific system configuration. This paper introduces AUReC, a new way to measure the effectiveness of shard maps that does not require gathering relevance judgments and is the first to completely decouple shard map evaluation from other components and parameters of a selective search system. By freeing shard map quality from other system components, AUReC provides robust diagnostic information that can be used to quickly sort through a large number of shard maps to tune a new selective search system, a process which was previously time-consuming and difficult.

AUReC evaluates shard maps by the area under a recall curve using the retrieval results of an exhaustive search system. It is highly-correlated to end-to-end selective search system evaluations while being simple to implement and not requiring: the implementation of other selective search components; picking a fixed efficiency level; or human-assessed relevance judgements. An examination of the effectiveness and robustness of AUReC found it produces scores that are highly-correlated with the evaluation of end-to-end systems under a variety of configurations.

Given a set of shard maps, the ordering of the shard maps determined by AUReC scores closely resembled the ordering by different end-to-end evaluations, usually with Pearson's $r > 0.9$. When pairs of shard maps were compared, most shard maps that had significant differences under an end-to-end evaluation also were significantly different when compared with AUReC scores. AUReC scores are calculated from easy-to-generate, plentiful data points and therefore produces stable results. Thus, AUReC was able to ascertain significant differences in pairs of shard maps where end-to-end system evaluations could not due to the scarcity of relevance data.

AUReC allows system designers to quickly test a large number of shard maps to tune the accuracy of a new selective search system, a task which used to be prohibitively expensive. We end the paper with practical guidelines on using AUReC to tune a system. First, to generate D_q, the set of documents that should be retrieved for query q, the strongest search engine available is preferred. However, an out-of-the-box retrieval still produces reliable results. More queries generate more consistent results with less variance. However, there are diminishing gains after about 800 queries.

REFERENCES

[1] Robin Aly, Djoerd Hiemstra, and Thomas Demeester. 2013. Taily: Shard Selection Using the Tail of Score Distributions. In *Proceedings of SIGIR*. 673–682.

[2] Yael Anava, Anna Shtok, Oren Kurland, and Ella Rabinovich. [n. d.]. A Probabilistic Fusion Framework. In *Proceedings of CIKM*.

[3] Ulf Brefeld, B. Barla Cambazoglu, and Flavio P. Junqueira. 2011. Document Assignment in Multi-site Search Engines. In *Proceedings of WSDM*. 575–584.

[4] B. Barla Cambazoglu, Emre Varol, Enver Kayaaslan, Cevdet Aykanat, and Ricardo Baeza-Yates. 2010. Query Forwarding in Geographically Distributed Search Engines. In *Proceedings of SIGIR*. 90–97.

[5] David Carmel, Doron Cohen, Ronald Fagin, Eitan Farchi, Michael Herscovici, Yoelle S. Maarek, and Aya Soffer. 2001. Static Index Pruning for Information Retrieval Systems. In *Proceedings of SIGIR*. 43–50.

[6] Charles L. A. Clarke, J. Shane Culpepper, and Alistair Moffat. 2016. Assessing efficiency–effectiveness tradeoffs in multi-stage retrieval systems without using relevance judgments. *Inf. Ret.* 19, 4 (2016), 351–377.

[7] Zhuyun Dai, Yubin Kim, and Jamie Callan. 2015. How Random Decisions Affect Selective Distributed Search. In *Proceedings of SIGIR*. 771–774.

[8] Zhuyun Dai, Yubin Kim, and Jamie Callan. 2017. Learning To Rank Resources. In *Proceedings of SIGIR*. 837–840.

[9] Zhuyun Dai, Chenyan Xiong, and Jamie Callan. [n. d.]. Query-Biased Partitioning for Selective Search. In *Proceedings of CIKM*.

[10] James C. French and Allison L. Powell. 2000. Metrics for evaluating database selection techniques. *World Wide Web* 3, 3 (2000), 153–163.

[11] Alan Griffiths, H.Claire Luckhurst, and Peter Willett. 1986. Using Inter-document Similarity Information in Document Retrieval Systems. *J. Am. Soc. Inf. Sci.* 37 (1986), 3–11.

[12] Yubin Kim, Jamie Callan, J. Shane Culpepper, and Alistair Moffat. 2017. Efficient distributed selective search. *Inf. Ret.* 20, 3 (2017), 221–252.

[13] Anagha Kulkarni and Jamie Callan. 2010. Document Allocation Policies for Selective Searching of Distributed Indexes. In *Proceedings of CIKM*. 449–458.

[14] Anagha Kulkarni, Almer Tigelaar, Djoerd Hiemstra, and Jamie Callan. 2012. Shard Ranking and Cutoff Estimation for Topically Partitioned Collections. In *Proceedings of CIKM*. 555–564.

[15] Xiaolu Lu, Alistair Moffat, and J. Shane Culpepper. 2016. The Effect of Pooling and Evaluation Depth on IR Metrics. *Inf. Ret.* 19, 4 (2016), 416–445.

[16] Craig Macdonald, Rodrygo L. T. Santos, and Iadh Ounis. 2013. The whens and hows of learning to rank for web search. *Inf. Ret.* 16, 5 (2013), 584–628.

[17] Luo Si and Jamie Callan. 2003. Relevant Document Distribution Estimation Method for Resource Selection. In *Proceedings of SIGIR*. 298–305.

[18] Anastasios Tombros, Robert Villa, and C.J Van Rijsbergen. 2002. The effectiveness of query-specific hierarchic clustering in information retrieval. *Inf. Process. Manage.* 38, 4 (2002), 559–582.

[19] Ellen M. Voorhees. 1985. *The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval.* Technical Report. Cornell University.