Visual Pool: A Tool to Visualize and Interact with the Pooling Method

Aldo Lipani
TU Wien
Institute of Software Technology &
Interactive Systems, Vienna, Austria
aldo.lipani@tuwien.ac.at

Mihai Lupu
TU Wien
Institute of Software Technology &
Interactive Systems, Vienna, Austria
mihai.lupu@tuwien.ac.at

Allan Hanbury
TU Wien
Institute of Software Technology &
Interactive Systems, Vienna, Austria
allan.hanbury@tuwien.ac.at

ABSTRACT

Every year more than 25 test collections are built among the main Information Retrieval (IR) evaluation campaigns. They are extremely important in IR because they become the evaluation praxis for the forthcoming years. Test collections are built mostly using the pooling method. The main advantage of this method is that it drastically reduces the number of documents to be judged. It does so at the cost of introducing biases, which are sometimes aggravated by non optimal configuration. In this paper we develop a novel visualization technique for the pooling method, and integrate it in a demo application named *Visual Pool*. This demo application enables the user to interact with the pooling method with ease, and develops visual hints in order to analyze existing test collections, and build better ones.

CCS CONCEPTS

• **Information systems → Test collections**; *Relevance assessment*; • **Human-centered computing → Visualization techniques**;

KEYWORDS

Visualization; Test Collections; Pooling Method; Pooling Strategies

1 INTRODUCTION

Test collection based evaluation in IR is a cornerstone of the IR experimentation. Most often, test collections are built using the pooling method. This method refers to a sampling procedure, according to a given strategy, of documents to be judged. This demo aims to visualize this procedure, allowing the user deeper insights.

A test collection is composed of a collection of documents, a set of topics, and a set of relevance judgements. A relevance judgment (or qrel) expresses the relevance of a document for a given topic. Due to the size of the modern collection of documents, to produce a *complete set* of relevance judgements is impossible. For example, if we examine what today would be considered a small test collection, with 500,000 documents and 50 topics (approximately the size of the TREC Ad Hoc 8 test collection [16]), the total relevance judgments to be made would be 25×10^6. At an optimistic rate of 120 seconds per judgment, this represents the equivalent of around 400 years of work for one person [4]. To solve this problem, early in the modern IR history, a sampling method was developed, the *pooling method* [14].

The pooling method consists in building a test collection by using the results provided by a set of search engines. These are usually systems designed by participants of challenges organized by IR evaluation campaigns such as: TREC, CLEF, NTCIR, or FIRE. In these challenges, every participant is provided a collection of documents and a set of topics. Their task is to develop a search engine to produce a result that maximizes the goal defined by the challenge. This result is then sent to the organizers, who now have everything they need to implement a pooling strategy.

The most common pooling strategy is the Depth@K strategy. This consists of creating a pool by selecting the top K documents from the results submitted by each system of each participant. The pool is given to the relevance assessors, who will produce a set of relevance assessments, which are then used in combination with an IR evaluation measure to rank the performance of the systems of the participants. These test collections are then used later by researchers to evaluate their systems. However, when comparing a new system with the search engines that participated in the challenge, the pooled systems have an advantage given by the guarantee that at least their top K documents have been judged, while for the new system this guarantee does not exist. This effect goes under the name of *pool bias*, which manifests itself when the evaluated system retrieves documents that will never be considered relevant [5] because they had never been seen by the human assessors.

This bias can be mitigated by increasing: 1) the depth of the pool, which decreases the probability of retrieving a non-judged document; 2) the number of topics, which reduces the variability of the bias making it easier to correct; and, 3) the number (assumed to be proportional to the variety) of the submitted results by the participants, which leads to a better exploration of the information space. However, all of these solutions result in a mere increase of the number of documents to be judged and therefore in an increase of the cost of the test collection. The research in the IR community to reduce the pool bias has branched out into two directions: (a) identifying a pooling strategy and a set of parameters that manifests a lower bias, and (b) estimating the bias to correct the score obtained by the search engine. The former direction has lead to the development of new pooling strategies [7, 10, 11], the latter instead to the development of new pool bias estimators [6, 8, 9]. Moreover, a hybrid approach has been also explored developing IR evaluation measures in combination with pooling strategies in order to minimize the pool bias [15, 17].

In this paper we present a demo that enables the user to visualize and interact with the pooling method. This demo addresses the needs of four classes of users: test collection builders, researchers, lecturers, and students. This solution aims to, by exploiting the users' sight, develop visual cues to guide the development of more sophisticated analyses. This solution is open source (MIT licensed) and is available on the website of the first author.

The reminder of the paper goes as follows. We first present our solution in Section 2. Then we present the three use cases in Section 3. In Section 4 we present the technology used. Finally, we discuss and conclude in Section 5.

2 VISUAL POOL

Visual Pool gives its users a new perspective over the pooling method, integrating a novel information visualization technique. This section is divided into three parts: we first present our pooling visualization technique, then we explain how this is integrated into the demo application, and we conclude listing the features of the demo. The authors have not found any solution that addresses a similar issue, which makes this solution unique in its kind.

In Figure 1 we see an example of the pool visualization technique. In this case we have applied a Depth@K pooling strategy. On the left, the run view highlights how the documents are distributed among the runs. On the center, the unique documents runs view where all the duplicated documents retrieved at a lower rank are removed. On the right, the pool view shows the distribution of unique documents at varying of the rank.

In Figure 2 we show a screen-shot of the Visual Pool application. In this user interface we identify the following sections (the numbers correspond to those in the figure):

(1) *Pooling Strategy Selection and Configuration.* We can select the pooling strategy among the 22 implemented. Every pooling strategy is configurable, if needed.

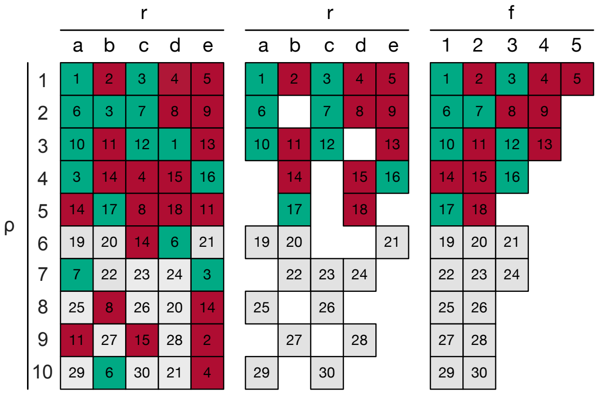


Figure 1: Example of three types of visualization of the same pool after the application of the pooling strategy Depth@5. Every square represents a document identified by its unique id. The documents' color means, if gray that is not judged, if green that is relevant, and if red that is not relevant. The y-axis is always the rank ρ at which a document has been retrieved. On the left the run view, where on the x-axis we find the list of runs. On the center the unique documents runs view, where, w.r.t. to the previous view, all the duplicate documents have been removed starting from the top left corner. On the right the pool view, where, w.r.t. to the previous view, the documents have been pushed to the left, and the x-axis is now instead the frequency of unique documents at rank ρ.

(2) *Visualization Control.* We can select which topic to visualize, and we can control which pool visualization view to display: run, unique documents runs, or pool.

(3) *Pool Strategy Control.* We can control the progress of the pooling strategy. We can here decide if to step the pooling strategy forward by one document or till the end, for the current topic, or for all the topics.

(4) *Visualization.* We visualize the pool using the previously described visualization technique.

(5) *Analytics.* We have a set of analytics that show statistics about the pool and display the current status of the pooling strategy.

(6) *Log.* The log of the pooling strategy is displayed, where we show the status of the processed documents.

(7) *Run/QRels upload.* We can upload the set of runs to be analyzed. It is possible also to indicate at which size to cut the runs. When an existing test collection is to be analyzed, we can also upload the set of relevance assessments, which will be used to visualize the process of assessment.

(8) *QRels download.* We can download the current qrels file, e.g. the current set of relevance assessments as generated by the pooling strategy.

In summary, here we list all the features implemented in the version of the demo presented at SIGIR:

- Load runs files in TREC format with a given size;
- Load a qrels file in TREC format;
- Select a pooling strategy and configure its parameters;
- Select which topic to visualize;
- Control the progress of the pooling strategy;
- Visualize the pool in three views: runs, unique documents runs, or pool;
- Visualize the log of the progress of the pooling strategy;
- Visualize the statistics about the pool and the status of the pooling strategy.
- Save the progress of the pooling strategy as a qrels file in TREC format;
- If required by the pooling strategy ask the user to judge a document;
- Offer API for controlling the pooling strategy in order to perform the judgment with an external application.

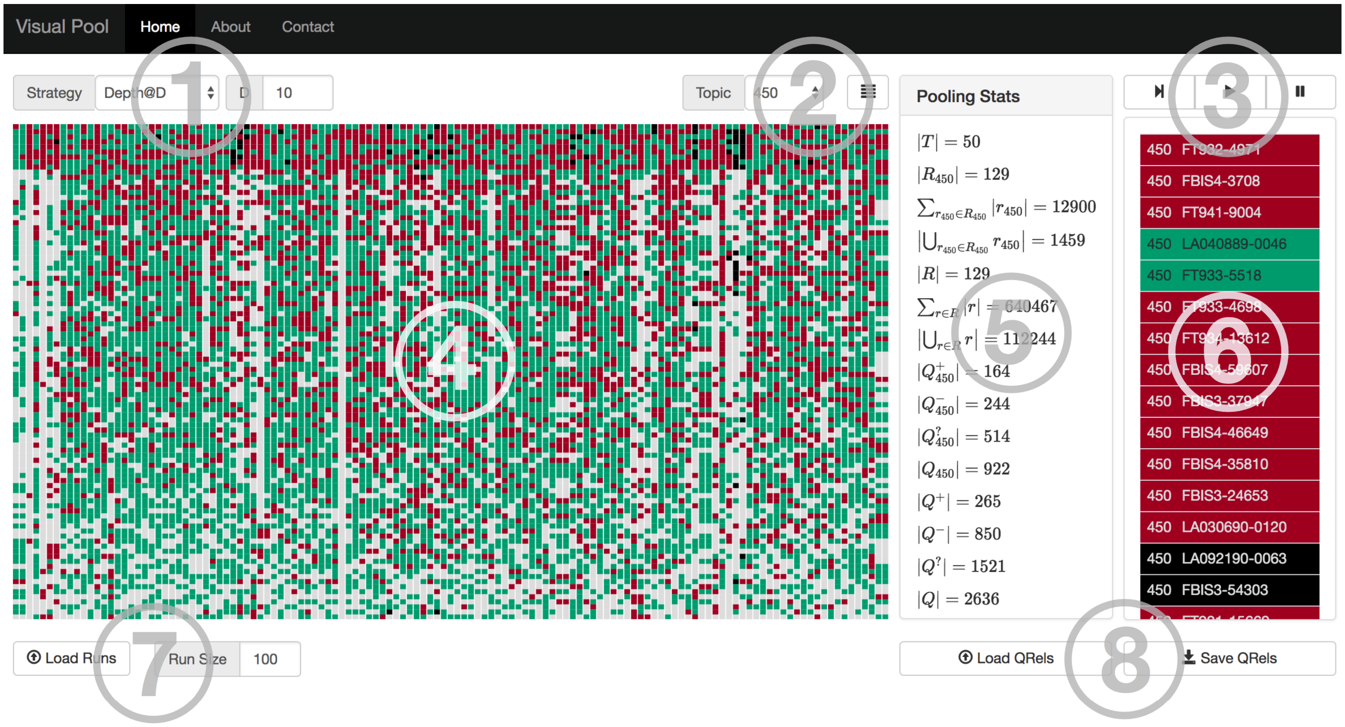
In Table 1 are listed all the pooling strategies already implemented in the demo.

3 USE CASES

In this section we present three use cases that cover the main user needs expressed by the four classes of users we aim to address. The first use case is about the visualization of an existing test collection. The second use case is about the analysis of a pooling strategy. Finally, the third use case is about building a test collection.

3.1 Visualizing a Test Collection

This use case addresses the needs of researchers when (a) interested in checking the properties of a test collection, e.g. visualize the pooled runs, assess the behavior of each topic, bias of the non-pooled or new systems, or (b) interested in juxtaposing two or more test collections to compare their properties.



Strategy Depth@D D 10

Topic 450

Run Size 100

Pooling Stats

$|T| = 50$

$|R_{450}| = 129$

$\sum_{r_{450} \in R_{450}} |r_{450}| = 12900$

$|\bigcup_{r_{450} \in R_{450}} r_{450}| = 1459$

$|R| = 129$

$\sum_{r \in R} |r| = 640467$

$|\bigcup_{r \in R} r| = 112244$

$|Q^{+}_{450}| = 164$

$|Q^{-}_{450}| = 244$

$|Q^{?}_{450}| = 514$

$|Q_{450}| = 922$

$|Q^{+}| = 265$

$|Q^{-}| = 850$

$|Q^{?}| = 1521$

$|Q| = 2636$

450 FT932-4971
450 FBIS4-3708
450 FT941-9004
450 LA040889-0046
450 FT933-5518
450 FT933-4698
450 FT934-13612
450 FBIS4-59607
450 FBIS3-37947
450 FBIS4-46649
450 FBIS4-35810
450 FBIS3-24653
450 LA030690-0120
450 LA092190-0063
450 FBIS3-54303

Figure 2: Screen-shot of the Visual Pool application taken after having: uploaded the runs with run size 100, uploaded the qrels, and executed the evaluation procedure as dictated by the selected pooling strategy Depth@10. In addition to the colors' meanings presented in Fig. 1, the color black indicates a document that has been pooled but it is not contained in the provided qrels.

Pooling Strategy			
Depth@K	[14]	RRFTake@N	[2]
Take@N	[7]	RBPTake@N	[13]
BordaTake@N	[1]	RBPAdaptiveTake@N	[13]
CondorcetTake@N	[9]	RBPAdaptive*Take@N	[13]
CombMAXTake@N	[12]	MTFTake@N	[3]
CombMINTake@N	[12]	HedgeTake@N	[11]
CombMEDTake@N	[12]	MABRandomTake@N	[11]
CombSUMTake@N	[12]	MABGreedyTake@N	[11]
CombANZTake@N	[12]	MABUCBTake@N	[11]
CombMNZTake@N	[12]	MABBetaTake@N	[11]
DCGTake@N	[10]	MABMaxMeanTake@N	[11]

Table 1: List of the implemented pooling strategies and their respective references.

For this use case, it is required from the user to provide as input both the runs files and the qrels file. Then, select the pooling strategy used to build the test collection, select the appropriate parameters, and execute the pooling strategy. Now, the application will display a visualization similar to Figure 1, where the user can select dynamically which view, and topic to visualize. When multiple test collections are to be compared, the user can repeat the process with a new instance of the application for each test collection.

3.2 Analyzing of a Pooling Strategy

This use case addresses the needs of lecturers to help them explain the pooling method to students, and to address the needs of students to better understand the algorithm. However, also researchers benefit from this use case, e.g. when interested in juxtaposing the results obtained with different pooling strategies.

For this use case, it is required from the user to provide as input both the runs files and the qrels file. Then, the user can select a pooling strategy to be analyzed, and configure it. Now, the application of the pooling strategy can be controlled by the controllers in the pooling controller section that allows the user to follow the pooling strategy at her/his own pace. To compare different pooling strategies, the user can repeat the process with a new instance of the application for each pooling strategy.

3.3 Building a Test Collection

This use case addresses the needs of a test collection builder to help them control the assessments of the selected documents using the application as a dashboard. This is achieved by making use of the API offered by the application, which allows a third party application to query the application about which document should be judged, and send a response back with the label.

Name	Type	Ignored?
topic_id	String+	No
iteration_id	String+	Yes
document_id	String+	No
rank	Integer	Yes
score	Float32	No
run_name	String	No

Table 2: Space separated fields of a runs file and their type. The column 'Ignored?' indicates if the field is ignored or not by the application.

Name	Type	Ignored?
topic_id	String+	No
iteration_id	String+	Yes
document_id	String+	No
score	Integer	No

Table 3: Space separated fields of a qrels file and their type. The column 'Ignored?' indicates if the field is ignored or not by the application.

For this use case, it is required from the user to provide as input the runs files, select a pooling strategy to be used, and configure it. Then, generate a unique key that will be used by the third party application to communicate with the application. At this point the user is able to follow the judgment process on-line. The application allows the user to change strategy if required, by downloading the current qrels and giving them as input to a new instance of the application.

4 TECHNOLOGY

This demo has been developed as a modern web application in JavaScript for the front-end and Scala for the back-end. The front-end is based on the web framework Ember.js[1], and on the visualization library p5.js[2], which is based on the Processing[3] language. The back-end is based on the Play Framework[4] and for in-memory storage on Redis[5], which is required only to support the API module.

The input files to be provided to the application are based on the de facto standard format of trec_eval[6]. The format is a non-breakable space separated file. In Table 2 we show the fields in the correct order as they should be contained by a runs file, and in Table 3 we show the same but for a qrels file. As indicated in the tables, some of the fields are ignored because they are redundant. The type String+ is a String type that does not contain spaces.

5 DISCUSSION & CONCLUSION

In this demo paper we have presented Visual Pool, an application to help test collection builders, researchers, lecturers, and students to visualize the pooling method. We believe that this technology will have a commercial impact because it allows the building of more efficient test collections but at the same cost, through the application of more efficient pooling strategies. We also believe it will have a research impact because it enables the analysis of new pooling strategies. Finally, it will have an educational impact because it supports lecturers in explaining and students in understanding the pooling method.

6 ACKNOWLEDGMENTS

This research was partly supported by the Austrian Science Fund (FWF) project number P25905-N23 (ADmIRE). This work has been supported by the Self-Optimizer project (FFG 852624) in the EUROSTARS programme, funded by EUREKA, the BMWFW and the European Union.

REFERENCES

[1] Javed A. Aslam and Mark Montague. 2001. Models for Metasearch. In *Proc. of SIGIR.*

[2] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proc. of SIGIR.*

[3] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. 1998. Efficient Construction of Large Test Collections. In *Proc. of SIGIR.*

[4] Bevan Koopman and Guido Zuccon. 2014. Why assessing relevance in medical IR is demanding. In *Medical Information Retrieval (MedIR) Workshop.*

[5] Aldo Lipani. 2016. Fairness in Information Retrieval. In *Proc. of SIGIR.*

[6] Aldo Lipani, Mihai Lupu, and Allan Hanbury. 2015. Splitting Water: Precision and Anti-Precision to Reduce Pool Bias. In *Proc. of SIGIR.*

[7] Aldo Lipani, Mihai Lupu, and Allan Hanbury. 2016. The Curious Incidence of Bias Corrections in the Pool. In *Proc. of ECIR.*

[8] Aldo Lipani, Mihai Lupu, Evangelos Kanoulas, and Allan Hanbury. 2016. The Solitude of Relevant Documents in the Pool. In *Proc. of CIKM.*

[9] Aldo Lipani, Mihai Lupu, Joao Palotti, Guido Zuccon, and Allan Hanbury. 2017. Fixed Budget Pooling Strategies Based on Fusion Methods. In *Proc. of SAC.*

[10] Aldo Lipani, Joao Palotti, Mihai Lupu, Florina Piroi, Guido Zuccon, and Allan Hanbury. 2017. *Fixed-Cost Pooling Strategies Based on IR Evaluation Measures.*

[11] David E. Losada, Javier Parapar, and Alvaro Barreiro. 2017. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing & Management* 53, 5 (2017).

[12] Craig Macdonald and Iadh Ounis. 2006. Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In *Proc. of CIKM.*

[13] Alistair Moffat, William Webber, and Justin Zobel. 2007. Strategic System Comparisons via Targeted Relevance Judgments. In *Proc. of SIGIR.*

[14] K. Spärck Jones and C. J. van Rijsbergen. 1975. Report on the need for and provision of an 'ideal' information retrieval test collection. *British Library Research and Development Report No. 5266* (1975).

[15] Ellen M. Voorhees. 2014. The Effect of Sampling Strategy on Inferred Measures. In *Proc. of SIGIR.*

[16] E Voorhes and Donna Harman. 1999. Overview of the eighth text retrieval conference. In *Proc. of TREC.*

[17] Emine Yilmaz and Javed A. Aslam. 2006. Estimating Average Precision with Incomplete and Imperfect Judgments. In *Proc. of CIKM.*

[1] https://emberjs.com

[2] https://p5js.org

[3] https://processing.org

[4] https://www.playframework.com

[5] https://redis.io

[6] https://github.com/usnistgov/trec_eval