Towards Stochastic Simulations of Relevance Profiles

Kevin Roitero[1], Andrea Brunello[1], Julián Urbano[2], Stefano Mizzaro[1]
[1] University of Udine, Italy
[2] Delft University of Technology, The Netherlands
mizzaro@uniud.it

ABSTRACT

Recently proposed methods allow the generation of simulated scores representing the values of an effectiveness metric, but they do not investigate the generation of the actual lists of retrieved documents. In this paper we address this limitation: we present an approach that exploits an evolutionary algorithm and, given a metric score, creates a simulated relevance profile (i.e., a ranked list of relevance values) that produces that score. We show how the simulated relevance profiles are realistic under various analyses.

KEYWORDS

Test collections, genetic algorithms, stochastic simulations.

ACM Reference Format:
Kevin Roitero[1], Andrea Brunello[1], Julián Urbano[2], Stefano Mizzaro[1]. 2019. Towards Stochastic Simulations of Relevance Profiles. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358123

1 BACKGROUND AND MOTIVATION

In the context of Information Retrieval (IR) evaluation, there are several works employing stochastic approaches to model and assess the various components that comprise an evaluation experiment. Research on the simulation of user behavior is an example that has received considerable attention in recent years [2, 3, 10, 21, 22]. Others have studied simulation in the traditional setting of evaluation with test collections. For instance, Cooper [5] developed a model to study the effect that query parameters had on retrieval, and Tague et al. [17] developed models to generate queries and their corresponding judgments. Azzopardi et al. [1] presented a method to simulate queries using documents and known items, and test their representativeness and validity against real queries produced from the same documents and known items. Robertson and Kanoulas [14] investigated how to model the document corpus of a test collection as a sample from a larger population.

More recently, Urbano [18] introduced a simple method for the simulation of effectiveness scores, and used it to compare different measures of the reliability of evaluation experiments, to guide in the design of test collections. Urbano and Nagler [19] refined this work and showed how to build generative stochastic models of the joint distribution of effectiveness for a set of systems, so that one can simulate results on new random topics. Their method simulates realistic evaluation scores for various metrics such as AP or P@n, but it does not generate the associated ranked lists of documents and associated relevance values that produce those scores. Therefore, its application is restricted to problems pertaining to distributions of scores, such as reliability of statistical tests or topic set size design, but it cannot be used to study lower level problems pertaining to system runs, such as properties of evaluation metrics or pooling.

The capability to simulate system runs has multiple applications in IR research. The first application is for effectiveness evaluation without relevance judgments, that originally aimed at ranking systems without using human relevance judgments [15] and more recently is also used in query performance prediction [11] and in the reduction of a topic set to a few good topics [15, 16]. One of the main problems in this respect is the availability of large training datasets of system runs, which can be addressed by our approach. The second sample application is ranking fusion [4, 6, 13]. The generation of simulated but realistic system runs would allow to have more data to work on, and thus improve the existing fusion techniques for IR. The third sample application relates to judgment allocation, where methods like estimation of evaluation scores [23], pooling or total recall [7] would benefit from the simulation of system runs without judgment incompleteness.

In this paper we present a first attempt at addressing the problem of generating ranked lists. More precisely, we do not aim to provide actual ranked lists of retrieved documents. Instead, our approach produces what we call *relevance profiles*, that is, ranked lists of relevance values. To the best of our knowledge, this is a novel problem that has never been studied before. Our approach is able to adapt to different researcher needs, producing relevance profiles having different properties such as a given number of relevant documents, or a given error on the effectiveness value, etc. More in general, our proposal can work as a data augmentation technique, to solve specific IR problems including but not limited to the three above, for example by producing more data to be exploited by machine learning techniques. Additionally, it can work with arbitrary effectiveness measures and multi-level relevance scales. However, because of space constrains we focus in this paper on binary relevance and Average Precision alone.

We focus on the following three research questions:

RQ1. Given a target AP value, are we always able to generate a simulated relevance profile having that AP value?
RQ2. Are the simulated relevance profiles different to the real ones (i.e., they are not simple replicas)?
RQ3. Are the simulated relevance profiles realistic (i.e., they show similar features to real relevance profiles)?

We develop a reasonably efficient solution, based on the NSGA-II evolutionary algorithm, to a computationally challenging problem,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358123

Table 1: The four variants of the algorithm

	V1	V2	V3	V4
(i) (p)	1	1	0.1	0.1
(ii) (Initial population)	Uniform	Geometric	Geometric	Uniform
(iii) (Mutation)	Uniform	Geometric	Uniform	Uniform

i.e., that of generating simulated but realistic relevance profiles having a target AP score. We demonstrate the effectiveness of our approach through a series of experiments with real TREC data. We also provide the code for our algorithm (available upon publication to preserve anonymity) and briefly discuss efficiency. Our results show that we are able to generate relevance profiles that produce a given AP score, that they are different from the original real relevance profiles that produced the AP score, and that the simulated relevance profiles are realistic.

2 GENERATION OF RELEVANCE PROFILES

We model the stochastic generation of relevance profiles as an optimization problem, adopting an evolutionary based approach. Specifically, our implementation is based on the well-known evolutionary algorithm NSGA-II [8], as implemented in the jMetal framework [9]. Each relevance profile is represented by an integer array of length 1000 (the number of retrievable documents). Each array element is bound to belong to the interval $[0, L]$, where L is the maximum relevance value (in this paper, since we consider binary relevance, $L = 1$ and array elements are either 0 or 1). The initial population consists in a set of profiles, each of which contains somewhere between 0 and $R \cdot p$ relevants, where R is the number of relevant documents for the considered topic and p is a random real value in $[0, 1]$; such a requirement is also enforced throughout the evolutionary process, by means of suitable constraints; crossover is performed by element-wise sum or product (modulus $L + 1$); mutation happens by applying random modifications to the arrays, involving swaps between two elements, or the sum of a random quantity to an element (modulus $L + 1$); finally, the fitness function seeks to minimize the absolute error between the target and actual AP values of the relevance profile corresponding to a given solution.

Let us remark that the computational complexity of the problem is daunting. When considering 1000 retrieved documents and binary relevance (as is usual in TREC), the number of different relevance profiles is $2^{1000} \approx 10^{300}$. On the other hand, the number of AP values with 4 significant digits (again the de-facto standard in TREC) is $10000 = 10^4$. Such a large number of combinations carries both bad and good news. A brute-force approach is simply unfeasible, but we can be confident that multiple relevance profiles exist for the target AP value. Conversely, it can be difficult to generate a simulated profile which is similar to the real ones because these represent a very small sample.

There are several variables and parameters that can be configured in the algorithm. For example, the initial population might be set according to a given amount of relevant documents (as a function of p above, and taking into account either R or 1000, the retrieval depth, as an upper bound), and/or to a given distribution of relevant documents (eg., uniformly distributed or with a bias towards the first rank positions, such as exponential or geometric).

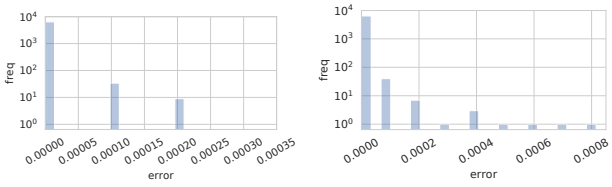


Figure 1: Two distributions of absolute error for V1 and V3.

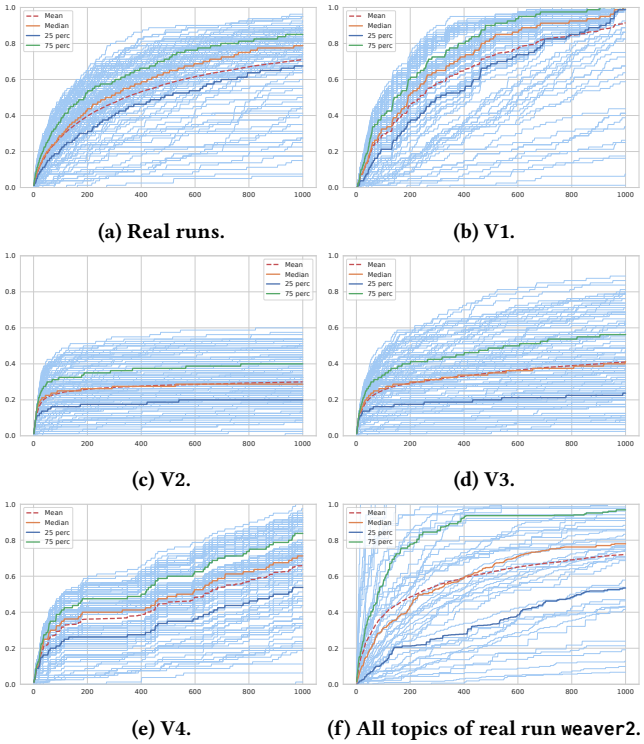
Mutations could also happen uniformly over the list or with a bias towards early ranks. A complete and systematic analysis of all possible configurations would be impossible in a short paper, and it is left for future work. Here we focus on exploring only a sensible selection of configurations that vary in terms of: (i) the number of relevant documents in the initial population, determined by p, (ii) the probability of the documents in the initial population to be relevant as a function of their rank position (uniform and geometric); and (iii) the probability of mutation (uniform and geometric). By setting these parameters we select four variants V1–V4 of the approach, as shown in Table 1.

3 EXPERIMENTS AND RESULTS

In this section we describe our experiments and results, addressing each research question. All experiments, when not otherwise specified, are performed on TREC8 data.

RQ1: Are Simulated Profiles Correct? We first report on the differences between the actual APs of the simulated relevance profiles and the original APs by measuring the absolute error. The average absolute error, for all four variants V1–V4, is less than 10^{-4}. The maximum absolute error, on all variants, is 0.0017, obtained with variant V3. Figure 1 shows two representative distributions of absolute error, for V1 and V3. This result is confirmed from Pearson's ρ and Kendall's τ correlation values between the real and the simulated AP scores: for all four versions they are equal to 0.999 and significant to the $p < 0.01$ level. We also ran some preliminary experiments on other TREC collections (TREC7, TREC2001, Robust2004, TeraByte2006) and obtained similar values. These results clearly show that our method is capable of generating simulated relevance profiles having the target AP score.

RQ2: Are Simulated Profiles New? We now verify that our method does not generate simulated relevance profiles that are simple replicas of the real ones (a result that could be obtained in a much simpler way, and that would not be much interesting). We start by analyzing the distribution of relevant documents in the real relevance profiles by means of graphical representations. Figure 2(a) shows the cumulative relevance profiles for all TREC8 runs on topic 402; the X axis represents each of the 1000 rank positions, and the Y axis represents the recall at those points. Every line represents a TREC8 run, with additional lines representing the mean, median, and quantiles over runs. We use *cumulative* profiles since they are perhaps more intuitive than density distributions, which in turn would be almost unreadable and not comparable (they would be histograms with 1000 bars). Figure 2 also shows four simulated cumulative relevance profiles for topic 402 ((b)–(e)). When comparing with Figure 2(a) it is clear that the simulated relevance profiles are

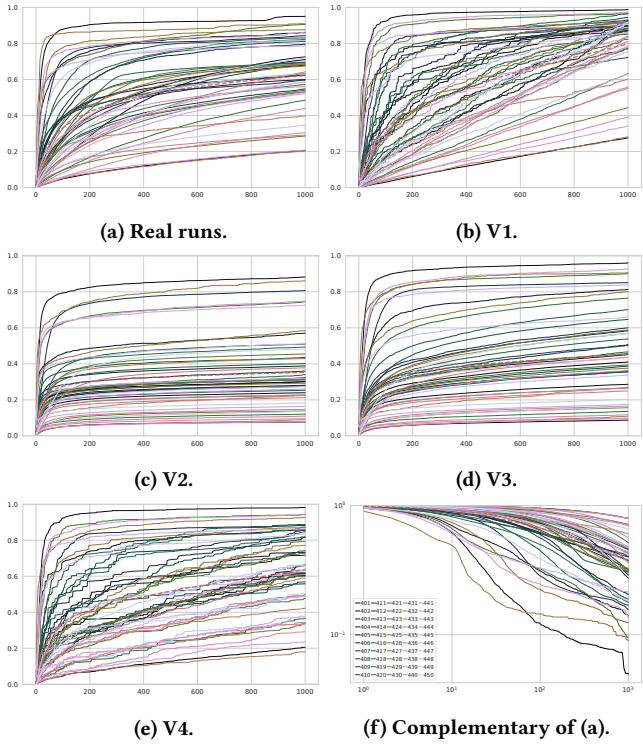


(a) Real runs. (b) V1.

(c) V2. (d) V3.

(e) V4. (f) All topics of real run `weaver2`.

Figure 2: Cumulative relevance profiles for topic 402.



(a) Real runs. (b) V1.

(c) V2. (d) V3.

(e) V4. (f) Complementary of (a).

Figure 3: Aggregated cumulative relevance profiles. Red dashed line is the average.

different from the original/real ones. The last chart in Figure 2(f) shows, in a similar way, a single run over all topics, i.e., the real cumulative relevance profiles for all TREC8 topics of run `weaver2`. It is clear that there is a high variation in the real relevance profiles, across both runs (a) and topics (f).

So far, this analysis has been based on a single topic that, although useful for presentation purposes, does not represent all topics in the collection. We therefore turn to a more general analysis next. Table 2 shows the number (and fraction) of simulated relevance profiles generated by the four versions that are identical, item by item and at various cutoffs, to the corresponding real profile. Of course the smaller the cutoff, the shorter the relevance profile, and the higher the number of simulated profiles that are identical to the real one. The values in the table clearly show that the simulated relevance profiles are usually different from the original ones. The cutoff 5 shows that differences occur also in the early rank positions. By looking at the first 10 rank positions, the simulated relevance profiles that are identical to the real ones are in the 4%–9% range. At a cutoff of 50 or higher, the percentage drops to less than 1%. When considering at least 10 retrieved documents, about 90%–99% of the simulated relevance profiles are new.

RQ3: Are Simulated Profiles Realistic? Having shown that the simulated relevance profiles generated by our method are both correct and different from the real ones, we now turn to the more interesting and difficult RQ3, namely whether the simulated relevance profiles are realistic. This question is more interesting since, given the high number of possibilities (see Section 2), it is perhaps not surprising that additional simulated profiles can be generated for a given AP value; it is more difficult also because it is unclear how to determine the degree of realism of a relevance profile.

As an initial observation, by comparing again Figure 2(a) with (b)–(e), one can notice that, overall, the set of simulated relevance profiles is different from the real one; this is clear, for example, by comparing the means. But again this argument lacks generality since it focuses on a single topic. We attempt a more general analysis in Figure 3. In this figure, each cumulative relevance profile is not obtained using a single run as in Figure 2(a)–(e), but using all 129 TREC8 runs and taking the average number of relevant documents retrieved at a given rank position. Each line in this chart still corresponds to a topic (and topic colors are the same across the charts; see the legend in Figure 3(f)). At a first glance, the V3 chart in (d) might seem the most similar to that in (a). However, the V1 and V4 charts in (b) and (e) feature more variability in steepness and coverage of values when flattening. V4 is also similar, especially when looking at the mean (red dashed line). On the other hand, V1 lower lines are too straight when compared to (a) and the steps that can be seen in V1 and especially V4 mean that the relevant documents tend to cluster at specific rank positions. Although these vary across topics, this is anyway an effect of the optimization process that is probably undesired as it introduces a subtle bias. We can state that it is difficult to single out the variant, among those that we have tried so far, that best approximates the real relevance profile; overall, the simulated relevance profiles generated by V1, V3, and V4 are realistic. Also, we have not yet tried to merge relevance profiles generated by different versions, but this seems a promising

Table 2: Number (and fraction in parentheses) of simulated relevance profiles identical to the real one with the same AP value up to a cutoff rank position. We consider only the 6283 AP values that are different from zero, out of the total 6450.

Cutoff	5	10	20	50	100	1000
V1	565 (.091)	244 (.039)	61 (.010)	20 (.003)	18 (.003)	15 (.002)
V2	1306 (.209)	546 (.088)	171 (.027)	53 (.008)	47 (.008)	32 (.005)
V3	1197 (.192)	462 (.074)	101 (.016)	28 (.004)	26 (.004)	17 (.003)
V4	879 (.141)	230 (.037)	61 (.010)	24 (.004)	21 (.003)	17 (.003)

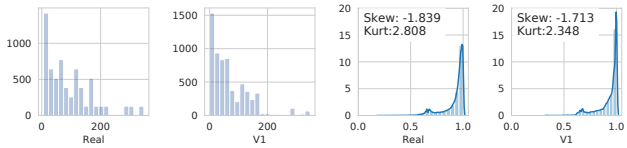


Figure 4: Comparison of real and simulated (V1) relevance profiles: distributions of retrieved relevant documents for each AP score (first two plots), and distributions of RBO values between pairs of relevance profiles (last two plots).

direction as, for example, the relevance profiles in the charts for V1 (b) and V2 (c) seem to be, respectively, too high and too low when compared to the real ones (a). It can be also noted that the topics are sorted in similar ways in the five charts: also in this respect, the simulated relevance profiles are similar to the real ones, and can be considered realistic.

As a last remark, Figure 3(f) shows the complementary cumulative distribution of the real relevance profile on a log-log scale, to understand if the relevance profiles are long-tailed (actually, power law): if so, the lines should be straight lines [12, §8.3, 4]. This could be a useful information to generate the simulated relevance profiles. The situation is unclear and requires further analyses.

Another confirmation that our method produces realistic relevance profiles can be derived from the two charts on the left in Figure 4, that show the distributions of the number of retrieved relevant documents for all the AP scores, for the real and simulated (V1) relevance profiles. As it can be seen, the number of retrieved relevant documents in the simulated profile tends to be similar to that real profiles. Figure 4 also shows, in the two charts on the right, the result of another analysis. We consider for each topic all possible pairs of runs, and we compute the RBO [20] of the two relevance profiles. The chart on the left shows the RBO distribution for the real relevance profiles; the one on the right for the simulated profiles (V1). Although these distributions show that our method tends to generate relevance profiles that are slightly more similar to each other than the real ones, we can state that the two distributions are very similar. As a final analysis, we consider again for each topic all possible pairs of runs. For each pair we compute the RBO of the two relevance profiles, for both the real and simulated relevance profiles. We then compute the Pearson's correlation between the two RBO vectors, obtaining $\rho = 0.68$ ($p < 0.01$). This is yet another confirmation that the simulated relevance profiles are in some sense similar to the real ones, and thus realistic.

4 CONCLUSIONS AND FUTURE WORK

We have shown that generating simulated relevance profiles is possible. Although we have not yet performed a systematic analysis of all possible variants and parameter configurations, we have been able to obtain relevance profiles that are correct, different from the real ones, and realistic according to our analyses, therefore providing a positive answer to the three RQs. Even though the problem is computationally challenging, we are able to obtain, on a common laptop, a simulated relevance profile of length 1000 in about a minute in the worst case (less than a minute on average); the 6450 TREC8 profiles were generated in about a day. In the future, we intend to perform a systematic analysis of several variants of the evolutionary algorithm, that may be obtained for example by taking into account other external factors such as topic properties (e.g., topic difficulty) not simply as a parameter, but to vary the algorithm to be able to mimic topics and runs with different characteristics. We will include other datasets, metrics, as well as non binary relevance.

ACKNOWLEDGMENTS

JU was funded by the EU H2020 programme (770376-2 TROMPA).

REFERENCES

[1] L. Azzopardi, M. de Rijke, and K. Balog. 2007. Building Simulated Queries for Known-item Topics: An Analysis Using Six European Languages. In *ACM SIGIR*.
[2] L. Azzopardi, K. Järvelin, J. Kamps, and M.D. Smucker. 2011. Report on the SIGIR 2010 Workshop on the Simulation of Interaction. *SIGIR Forum* (2011).
[3] F. Baskaya, H. Keskustalo, and K. Järvelin. 2013. Modeling Behavioral Factors Ininteractive Information Retrieval. In *ACM CIKM*.
[4] R. Benham and J.S. Culpepper. 2017. Risk-Reward Trade-offs in Rank Fusion. In *Australasian Document Computing Symposium*.
[5] M.D. Cooper. 1973. A simulation model of an information retrieval system. *Information Storage and Retrieval* (1973).
[6] G.V. Cormack, C.L.A. Clarke, and S. Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *ACM SIGIR*.
[7] G.V. Cormack and M.R. Grossman. 2018. Beyond Pooling. In *ACM SIGIR*.
[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Comp.* (2002).
[9] J.J. Durillo, A.J. Nebro, and E. Alba. 2010. In *IEEE CEC 2010*.
[10] D. Maxwell and L. Azzopardi. 2016. Agents, Simulated Users and Humans: An Analysis of Performance and Behaviour. In *ACM CIKM*.
[11] S. Mizzaro, J. Mothe, K. Roitero, and M.Z. Ullah. 2018. Query Performance Prediction and Effectiveness Evaluation Without Relevance Judgments: Two Sides of the Same Coin. In *ACM SIGIR*.
[12] M. Newman. 2010. *Networks: An Introduction*. Oxford University Press, Inc.
[13] R. Nuray and F. Can. 2006. Automatic ranking of information retrieval systems using data fusion. *Inf. Proc. & Mgmt.* (2006).
[14] S.E. Robertson and E. Kanoulas. 2012. On Per-topic Variance in IR Evaluation. In *ACM SIGIR*.
[15] K. Roitero, M. Passon, G. Serra, and S. Mizzaro. 2018. Reproduce. Generalize. Extend. On Information Retrieval Evaluation Without Relevance Judgments. *J. Data and Information Quality* (2018).
[16] K. Roitero, M. Soprano, A. Brunello, and S. Mizzaro. 2018. Reproduce and Improve: An Evolutionary Approach to Select a Few Good Topics for Information Retrieval Evaluation. *J. Data and Information Quality* (2018).
[17] J. Tague, M. Nelson, and H. Wu. 1981. Problems in the Simulation of Bibliographic Retrieval Systems. In *ACM SIGIR*.
[18] J. Urbano. 2016. Test collection reliability: a study of bias and robustness to statistical assumptions via stochastic simulation. *Inf. Retrieval Journal* (2016).
[19] J. Urbano and T. Nagler. 2018. Stochastic Simulation of Test Collections: Evaluation Scores. In *ACM SIGIR*.
[20] W. Webber, A. Moffat, and J. Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM TOIS* (2010).
[21] R.W. White. 2006. Using searcher simulations to redesign a polyrepresentative implicit feedback interface. *Inf. Proc. & Mgmt.* 42, 5 (2006), 1185 – 1202.
[22] R.W. White, I. Ruthven, J. Joemon, and C.J. van Rijsbergen. 2005. Evaluating Implicit Feedback Models Using Searcher Simulations. *ACM TOIS* (2005).
[23] E. Yilmaz, . Kanoulas, and J.A. Aslam. 2008. A Simple and Efficient Sampling Method for Estimating AP and NDCG. In *ACM SIGIR*.