Reproduce and Improve: An Evolutionary Approach to Select a Few Good Topics for Information Retrieval Evaluation

KEVIN ROITERO, MICHAEL SOPRANO, ANDREA BRUNELLO, and
STEFANO MIZZARO, University of Udine

Effectiveness evaluation of information retrieval systems by means of a test collection is a widely used methodology. However, it is rather expensive in terms of resources, time, and money; therefore, many researchers have proposed methods for a cheaper evaluation. One particular approach, on which we focus in this article, is to use fewer topics: in TREC-like initiatives, usually system effectiveness is evaluated as the average effectiveness on a set of n topics (usually, $n = 50$, but more than 1,000 have been also adopted); instead of using the full set, it has been proposed to find the best subsets of a few good topics that evaluate the systems in the most similar way to the full set. The computational complexity of the task has so far limited the analysis that has been performed. We develop a novel and efficient approach based on a multi-objective evolutionary algorithm. The higher efficiency of our new implementation allows us to reproduce some notable results on topic set reduction, as well as perform new experiments to generalize and improve such results. We show that our approach is able to both reproduce the main state-of-the-art results and to allow us to analyze the effect of the collection, metric, and pool depth used for the evaluation. Finally, differently from previous studies, which have been mainly theoretical, we are also able to discuss some practical topic selection strategies, integrating results of automatic evaluation approaches.

CCS Concepts: • **Information systems** → **Test collections**; • **Computing methodologies** → *Multi-objective evolutionary computation*;

Additional Key Words and Phrases: Test collection, few topics, reproducibility, topic sets, topic selection strategy, evolutionary algorithms

ACM Reference format:
Kevin Roitero, Michael Soprano, Andrea Brunello, and Stefano Mizzaro. 2018. Reproduce and Improve: An Evolutionary Approach to Select a Few Good Topics for Information Retrieval Evaluation. *J. Data and Information Quality* 10, 3, Article 12 (September 2018), 21 pages.
https://doi.org/10.1145/3239573

1 INTRODUCTION

Test-collection-based evaluation of Information Retrieval (IR) systems is a widely adopted approach. Test collections were born with the Cranfield Collection in the '60s, and are maintained in the current form of competition by many initiatives such as TREC, CLEF, NTCIR, FIRE, and so on. A test collection is composed of a set of documents, a set of queries (called topics), and a

Authors' addresses: K. Roitero, M. Soprano, A. Brunello, and S. Mizzaro, University of Udine, Via delle Scienze 206, Udine, 33100, ITA; emails: {Mizzaroroitero.kevin, soprano.michael, andrea.brunello, mizzaro}@uniud.it.

1936-1955/2018/09-ART12 $15.00
https://doi.org/10.1145/3239573

set of relevance judgments for each ⟨topic, document⟩ pair. The relevance judgments are made by human experts, and this is perhaps the most expensive step of the whole evaluation process. To reduce the cost of the whole process, and in particular of gathering the judgments, several solutions have been proposed. One notable example is represented by the pooling strategy adopted in most oTREC-like initiatives: only the documents retrieved by at least one system in the first N ranking positions (with N classically equal to 100) are judged [22].

In this article, we focus on one particular approach to reduce the cost of the evaluation process, which consists of limiting the number of topics used in the evaluation. This approach has been studied by Guiver et al. [9], Robertson [13], and Berto et al. [3]. Their results have been obtained by using the BestSub software, which presents several limitations, discussed in the following.

Our contribution in this article is fourfold:

(1) We re-implement the BestSub software using a novel approach based on a multi-objective Evolutionary Algorithm (EA). We also release the software, making it freely available to the research community.
(2) We then reproduce the main results by Guiver et al. [9], Berto et al. [3], and Reference [13], using the novel implementation, as well as discuss its advantages w.r.t. the original approach.
(3) The novel and more efficient implementation allows us to run a battery of new experiments. We therefore generalize the previous results to other datasets and collections.
(4) Finally, we extend such results by performing some experiments on an effective and *a priori* (i.e., before the human relevance assessment takes place) topic selection strategy. The previous studies failed in finding such a practical result. We rely on methods by Soboroff et al. [17] and Robertson [13] to estimate some topic features. We then compare the evaluation of systems on the basis of the subset of topics selected in this way with the evaluation on the basis of a random selection of topics, showing that the former evaluation has a higher correlation with the original one, based on the full set of topics.

We remark that reproduction seems particularly important in this case, since the previous results have been obtained by a single research group that used a specific, *ad hoc*, software that has never been released widely and officially, and they have been published in potentially high impact venues: Reference [9] has been published in an important journal (ACM TOIS, 1.070 impact factor); Berto et al. [3] in ICTIR, and Reference [13] in ECIR, two important conferences for the IR community.

This article is structured as follows: Section 2 details the background, Section 3 discusses the reimplementation of BestSub software, detailing the EA approach we used, Section 4 presents the experiments, and Section 5 concludes and provides directions for future work.

2 BACKGROUND—RELATED WORK

In the following, we describe the state-of-the art in the evaluation of IR systems using fewer topics (Section 2.1), focusing on the original software BestSub (Section 2.2), and discussing its limitations (Section 2.3).

2.1 Fewer Topics

The attempts to reduce the number of topics used in the evaluation of system effectiveness can be classified into two main approaches; namely, either select a subset of topics with a lower cardinality by random sampling or investigate the best possible choice of an optimal topic subset.

Many researchers have studied the first approach, although arriving at different conclusions in the various studies. Sparck Jones and van Rijsbergen [18, page 63] analyze the number of topics

Table 1. AP and MAP for n Topics and m Systems (Adapted from Reference [9])

	t_1	$\cdots$	t_n	MAP
s_1	$\mathrm{AP}(s_1, t_1)$	$\cdots$	$\mathrm{AP}(s_1, t_n)$	$\mathrm{MAP}(s_1)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s_m	$\mathrm{AP}(s_m, t_1)$	$\cdots$	$\mathrm{AP}(s_m, t_n)$	$\mathrm{MAP}(s_m)$

used in evaluation by means of test collections, concluding that "250 (topics) are usually acceptable, and 1,000 are sometimes needed." Many years later, Zobel [25], focusing on pool depth, concludes that 25 topics are reasonably good in predicting the effectiveness evaluation of systems on a different set of 25 topics, and Buckley and Voorhees [4, page 39] concludes that "25 topics is just barely enough for an experiment but that 50 topics is stable." More recently, Webber et al. [23] and Sakai [15, 16] use statistical power, obtaining that "an experiment should contain at least 150 topics" [23, page 5], and "as different evaluation measures can have vastly different within-system variances, they require substantially different topic set sizes under the same set of statistical requirements" [16, page 256].

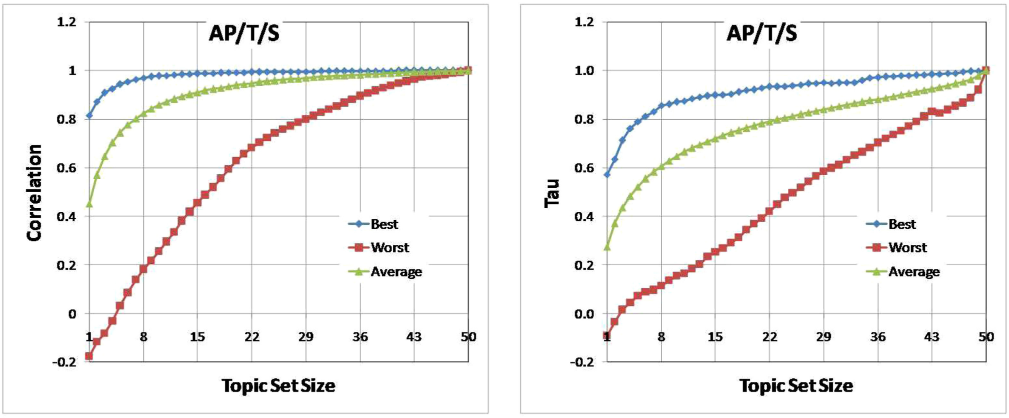
Other researchers show strong evidence of interactions between systems and topics; these works include the ANOVA analysis by Banks et al. [2], and results by Mizzaro and Robertson [12] and Roitero et al. [14], which show the interactions between systems and topics using network analysis, and results by Kleinberg [10].

Turning to the second approach of topic set reduction based on the theoretical best possible choice, the three main contributions are by Guiver et al. [9], Robertson [13], and Berto et al. [3]. Since in this article, we focus on reproducing those results, we describe such articles more in detail.

Guiver et al. [9] propose a theoretical analysis on topic set reduction. Their analysis starts from TREC evaluation results as represented in Table 1. The process is described as follows [9, page 21:4]:

> The basic method is as follows. We start from a set of n topics ($n = 50$ or 25 in the experiments that follow). We now consider, for any $c \in \{1, \ldots, n\}$ and for any subset of topics of cardinality c, the corresponding values of MAP for each system calculated on just this subset of topics: that is, we average only a selected set of c of the n columns in Table I [our Table 1]. For each such subset, we calculate the correlation of these MAP values with the MAP values for the whole set of topics. This correlation measures how well the subset predicts the performance of different systems on the whole set. Now for each cardinality c, we select the best subset of topics, that is the one with the highest correlation. We also select the worst, and finally we calculate an average correlation over all subsets of size c.

The outcome of the process is shown in Figure 1, which shows for each cardinality (x-axis) the correlation values (y-axis): the best possible correlations are the curves in blue, the average correlation (i.e., that obtained by a random topic selection) is in green, and the worst is in red. Results show that the best topic subset is much better than the average one in predicting the system performance on the full set of topics. Furthermore, the worst topic subset is much worse than the average one, and the gap between the two correlations is high. To make an example for the Pearson Correlation, with just 8 best topics we obtain correlations higher than 0.95; to achieve the same result we would need 23 topics for the average series, and more than 40 for the worst one. Furthermore, results appear stable across measures: Guiver et al. include in the study R-prec, P10, and logAP (or GMAP). Whereas the usefulness of the Best series is intuitive, the interestingness of the Worst series is perhaps less straightforward and deserves a brief justification, besides simply



(a) (b)

Fig. 1. Correlation values from Pearson's ρ (a) and Kendall's τ (b), obtained with BestSub on TREC-8 (from Reference [9, Figures 2 and 3]).

stating that it was studied by the previous authors and we are reproducing it. Indeed, knowing how a topic subset can rank the systems in a so different way from the official ranking (i.e., the one provided by TREC) is useful to understand how "wrong" one can be when using a topic set of a given cardinality. The Best series is an optimum to aim at; the Worst series is something that needs to be avoided.

It has to be noted that the computational complexity is high, and finding an exact solution becomes intractable even for rather small n, since the number of subsets to be analyzed increases exponentially. For this reason, Guiver et al. rely on a heuristic search in their analysis, that works as follows [9, page 21:10]:

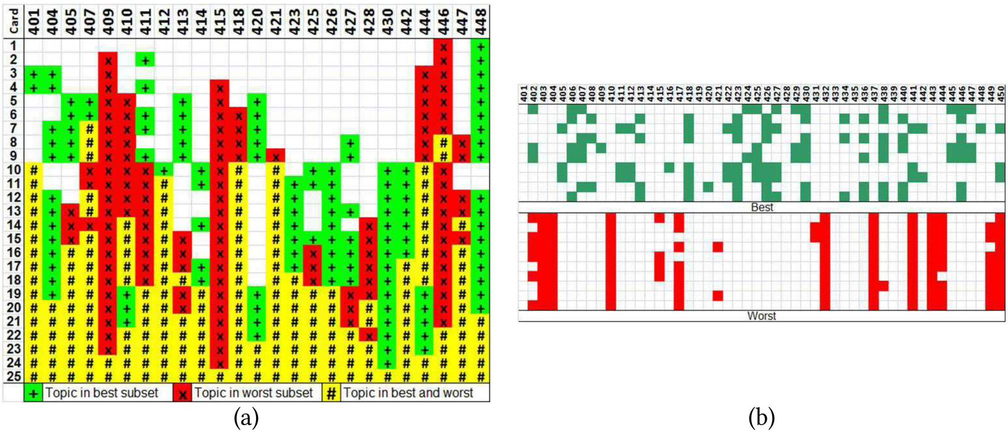
> a heuristic is to search recursively: having identified the best set for cardinality c, to seek the best for cardinality $c + 1$ among sets which differ from the best c set by not more than 3 topics (the number 3 was chosen primarily because 4 is intractable).

The effects of the heuristic on the results are discussed in Section 2.3.

Guiver et al. also addressed two questions that are strongly related to the heuristic algorithm used:

(i) How much difference is there considering the topics of the Best/Worst topic subset at a given cardinality (c), and the topics of the subset at the next one ($c + 1$)?
(ii) What happens when performing a neighborhood analysis, i.e., when selecting not only the single Best/Worst subset but also the second Best/Worst subset and the subsequent Best/Worst ones? In particular, they analyzed the 10 Best/Worst topic subsets for each cardinality.

Results are shown in Figure 2. Figure 2(a) is a topic by cardinality pixel-map, for ρ correlation; in each cell the value is "+" if for that cardinality the topic is part of the Best subset, "x" if the topic is in the Worst subset, and "#" if the topic is present in both sets. Figure 2(a) shows that, in general, single topics appear to be either good or bad; this statement is more true for the bad topic subset: once a topic enters in the worst set at a given cardinality it tends to be in the worst set also for the next cardinality, while for the best topic subset some variation arises in this pattern.



(a) (b)

Fig. 2. Stability pixel-maps for Pearson's ρ for the whole dataset (a) and for Pearsons's ρ for the top 10 sets at cardinality 12 (b) (from Reference [9, Figures 5 and 6]).

Figure 2(b) is a topic by "goodness" (i.e., first Best/worst set, second, etc.) pixel-map, for cardinality 12 for ρ correlation; in the upper part (i.e., first 10 rows) of the map the Best 10 subsets are shown, while in the lower part (i.e., last 10 rows) of the map the Worst 10 subsets are shown; in each cell the value is colored if for that set the topic is part of the Best subset (upper part of the map), or in the Worst subset (lower part of the map). This figure confirms that while the quality of being a bad topic is true for individual topics, a good topic set is not formed by individual good topics only, but by a set of topics that somehow contribute in predicting systems effectiveness.

The second contribution is by Robertson [13], who extended the work by Guiver et al. in four ways:

- He used two more collections: TREC87 and another one, named Terrier, in which he used a set of different configurations of the Terrier system[1] to obtain another system population.
- He used a different evaluation measure (i.e., logitAP).
- He studied a particular method to build good topic subsets. He built a matrix, as proposed by Mizzaro and Robertson [12], to represent interactions between systems and topics, and in particular correlations between topic ease and the ability to predict system effectiveness; on that matrix, using the HITS algorithm [10], he computed topic hubness, and he analyzed if such a feature can be used to find a few good topics.
- He performed three generalization experiments [13, page 138]:
  (1) a HITS analysis: he compared hubness vectors of topics from the three collections for a given metric;
  (2) a Best/Worst subset analysis: he used the Best/Worst subsets computed on the TREC data on the Terrier collection;
  (3) a topic selection strategy: he used the HITS analysis of TREC data to predict good topic sets on Terrier.

Conclusions of the analysis confirm that choosing a good topic subset is not just a matter of selecting good individual topics.

Berto et al. [3] generalize the work by Guiver et al. [9] and Robertson [13] by (i) investigating how many good topic subsets exist, (ii) extending the generalization analysis of Robertson,

[1] http://terrier.org.

considering various evaluation metrics, and (iii) extending the results of Guiver et al. on the best 10 topic subsets by including a generalization experiment (i.e., what happens when considering the rank of the top topic subsets and a different topic subset). Results show that (i) many good topic subsets exist, so there is hope that some of them are general; (ii) even if the single best topic subset is not able to generalize to a new system population, some of the subsequent ones might be adequate under certain conditions; and (iii) the metric has a major role when dealing with best topic subsets.

Note that results by Guiver et al. [9], Robertson [13], and Berto et al. [3] are *a posteriori*: their analysis is conducted after the whole evaluation process is finished; thus, their results are not immediately applicable to obtain a practical topic selection strategy, which should be able to identify the few good topics *a priori* (or, at least, during the relevance assessment process). We agree with the authors of the previous studies that, even if *a posteriori*, such a theoretical approach is interesting and useful since the theoretical maximum, minimum, and average correlation values for the different topic subsets constitute useful baselines to compare with when testing and proposing a novel topic selection strategy. Moreover, practical *a priori* approaches based on this theoretical method are indeed possible: we discuss two of them in Section 4.6.

2.2 BestSub

The above results have been obtained by using the BestSub software, specifically implemented in 2006 (and revised some years later) to study the fewer topics approach. BestSub is written in C#, and receives in input the topic-system table (see Table 1) and a parameter k, used for the heuristic search. BestSub searches the Best/Worst topic subsets (i.e., the one which correlates more/less with the ground truth) at each cardinality c between 1 and n (i.e., the number of topics). BestSub computes the Best/Worst sets for each cardinality using the Pearson's ρ, and Kendall's τ correlations, plus the Error-Rate measure [9]. The output of BestSub can be represented as in Figure 1.

2.3 Issues with BestSub

BestSub has been a valuable and useful tool that allowed us to obtain the previous results, but it is not free from limitations, as we now discuss.

The heuristic is quite rough. In the BestSub implementation, the side effect of the heuristic is that the topic subset at cardinality c and the one at cardinality $c + 1$ differ for at most k elements. The more k is close to 1, the more the search process becomes a greedy algorithm. Guiver et al. set this parameters to a maximum number of 3. Robertson investigated the outcome of the experiments with a parameter k close to 1. Concerning exhaustive search, Guiver et al. [9, page 21:10] write:

> when using the Kendall's τ, searching exhaustively takes around 7 days for $c = 11$, and around 20 days for $c = 12$, even with efficient $O(n \log n)$ calculation of τ using Knight's algorithm [Boldi et al. 2005]. Exhaustive search for correlation runs is much faster due to the simpler calculation, and wider scope for optimization of the algorithm; however, even there, computation becomes a real issue beyond $c = 15$.

It is particularly worrying that the heuristic can distort the stability results. Although the correlation values obtained are probably not heavily affected by using the heuristics, some effect on stability (see Figure 2) cannot be excluded.

Finally, BestSub efficiency is not ideal. The search algorithm of BestSub, even if optimized, results in an extremely slow search even for a small topic set. With $k = 3$ on a top class PC (2013 Mac Pro) for 50 topics, the algorithm takes approximately 10.5 hours to finish, and for 250 topics with $k = 2$ (i.e., almost a greedy search) the algorithm takes more than 1 month to finish the computation.

3 NEWBESTSUB

We now turn to our first aim, see item (1) in Section 1. In this section, we first briefly present some background on EAs; we then detail our overall approach and the NewBestSub software, the reimplementation of BestSub by means of EAs.

3.1 Evolutionary Algorithms

In the current work, we make use of an approach based on EAs, i.e., population-based metaheuristics that rely on mechanisms inspired by the process of biological evolution and genetics to solve optimization problems [6]. Unlike blind random search algorithms, EAs are capable of exploiting historical information to direct the search into the most promising regions of the search space, relying on methods designed to imitate the processes that in natural systems lead to adaptive evolution.

In nature, a population of individuals tends to evolve to adapt to the environment in which they live; in the same way, EAs are characterized by a population, where each individual represents a possible solution to the optimization problem. Every solution is evaluated with regard to its degree of "adaptation" to the problem through a single- or multi-objective *fitness* function.

During the computation of the algorithm, the population iteratively goes through a series of *generations*. At each generation step, some of the individuals are picked by a *selection strategy*, and go through a process of reproduction, by the application of suitable *crossover and mutation operators*. The selection strategy is one of the main distinguishing factors between meta-heuristics, although typically individuals with high degree of adaptation are more likely to be chosen.

NSGA-II [5], on which our method is based, uses a Pareto-based multi-objective strategy with a *binary tournament selection* and a *rank crowding better* function. To the selected individuals, operations such as crossover and mutation are applied with a certain degree of probability, with the goal of generating new offspring, creating a new generation of solutions. Crossover is the EA equivalent of natural reproduction, by which the characteristics of two individuals are combined. Mutation is used to maintain the genetic diversity in the elements of the population, through applying random changes in the encoding of the selected solution. Typically, a high crossover probability tends to pull the population towards a local minimum or maximum, while a high degree of mutation allows to explore the search space more broadly.

The algorithm terminates when a predefined criteria is satisfied, which can be a bound on the number of generations, or a minimum fitness increment that must be achieved between subsequent evolution steps of the population.

Multi-objective EAs are designed to solve a set of minimization/maximization problems for a tuple of n functions $f_1(\overrightarrow{x}), \ldots, f_n(\overrightarrow{x})$, where $\overrightarrow{x}$ is a vector of parameters belonging to a given domain. A set $\mathcal{S}$ of solutions for a multi-objective problem is said to be *non-dominated* (or *Pareto optimal*) if and only if for each $\vec{x} \in \mathcal{S}$, there exists no $\vec{y} \in \mathcal{S}$ such that: (i) $f_i\ (\vec{y})$ improves $f_i\ (\vec{x})$ for some i, with $1 \leq i \leq n$, and (ii) for all j, with $1 \leq j \leq n$ and $j \neq i$, $f_j\ (\vec{x})$ does not improve $f_j\ (\vec{y})$. The set of non-dominated solutions from $\mathcal{S}$ is called *Pareto front*.

Note that, although we are indeed interested in finding a set of Pareto optimal individuals (the best or worst topic set for each cardinality), it would also be possible, in principle, to choose a single one of them as the final solution. This presupposes the existence of a suitable *a posteriori* selection strategy, such as, for example, "keep the subset characterized by the highest correlation value, among the ones having less than 10 topics."

3.2 EAs and Fewer Topics Subsets

Multi-objective approaches are particularly suitable for solving multi-objective optimization problems, as the one treated in this work, because they are capable of searching for multiple optimal solutions in parallel. Indeed, given the wide search space of topic sets, and the two antithetical

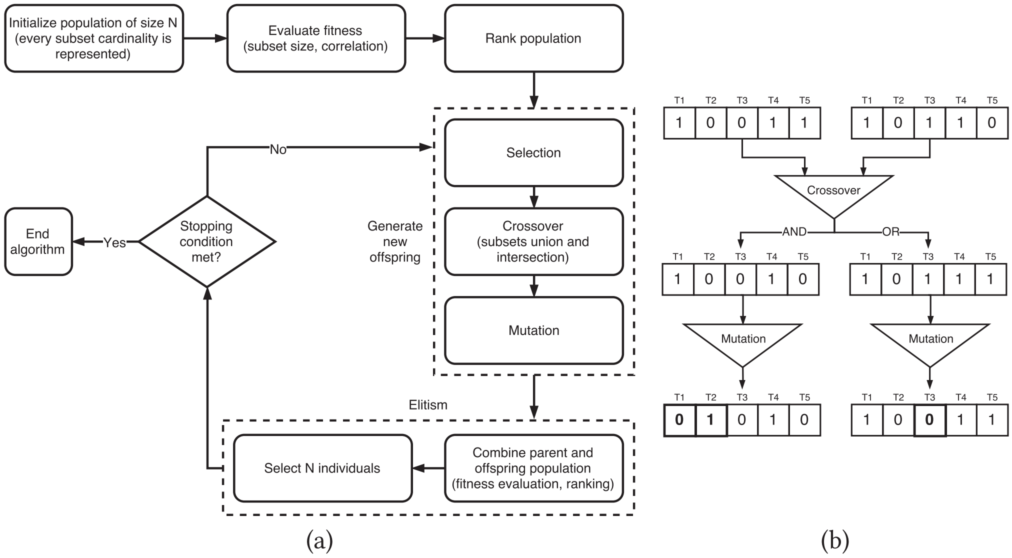


Fig. 3. Overview of the algorithm NSGA-II applied to the few topic selection problem (left), and detail of the crossover and mutation process (right); the bold squares represent the mutated topics.

objectives which characterize the topic reduction process (i.e., reducing the topic subset size while keeping a high correlation value), EAs seem to be a natural solution to solve such a problem. We propose a method capable of optimizing, together, the size of a subset of topics and its accuracy in evaluating information systems, with respect to the whole set of topics. For each subset cardinality, we look for the best and the worst topics to be included for such an extent.

An overview of the operation of the NSGA-II EA, applied to the topic selection problem, is presented in Figure 3; further details are provided in the following.

3.3 Initial Population

Each individual of the population is represented by a binary array, having length equal to the number of topics in the full topic set (n). The ith cell of the array tracks whether the corresponding topic is included in the encoded solution or not. Note that in the population, topic subsets of the different cardinalities $1 \leq c \leq n$ co-exist.

As for the size of the initial population, the minimum allowed value is equal to the number of topics to ensure that every subset cardinality is initially represented; also, to guarantee population heterogeneity, we simply randomly generate a set of individuals for each subset cardinality.

3.4 Operators

The crossover operator merges information from two (or more) randomly selected parents, generating one or more offspring. The underlying idea is that of combining the features of two different but desirable individuals. In our implementation, given two parent solutions, two children are generated by simply performing a pairwise logic AND and a pairwise logic OR of the two corresponding binary arrays. Although many other options for the operators are possible [6], we chose to rely on those two since they have an intuitive, clear meaning for the problem at hand: one child will contain only the topics which are in common between the two parents, and the other one will contain the union of the topics of the two parents. A graphical representation of the crossover operator is shown in Figure 3(b).

A proper mutation operator should not try to improve a solution on purpose, since this would bias the evolution process of the population. Rather, it should cause random, unbiased changes in the solution components. In our implementation, mutation is performed by scanning the array left to right and deciding, for each cell, whether to flip its value or not according to a given probability. That is, for every topic in the vector representing the solution, we randomly generate a number between zero and 1: if the draw number is less than the given probability we perform the mutation (in doing so, if the topic is included in the solution then we remove it, and vice versa). This is a classic strategy, which has been described, for example, in [6]. A graphical representation of the mutation operator is shown in Figure 3(b).

For the selection of the parent candidates, i.e., the sets of topics to be used for the application of crossover and mutation operators, we rely on the classic strategy implemented in NSGA-II, based on the concepts of ranking and crowding distance: the entire population is sorted into fronts, according to non-domination. A first front is made by the individuals which are non-dominated. A second one is composed of the individuals which are dominated by the elements in the first front only, and so on for the remaining fronts. Then, pairs of individuals are randomly selected from the population; finally, a *Binary Tournament* selection is carried out for each pair, considering a better-function based on the concepts of *rank* (which considers the front which the instance belongs to) and *crowding distance* (intuitively, it measures how close an individual is to its neighbors). For further details see Reference [5].

Observe that, after the offspring generation phase, the population has doubled in size. To select the individuals to pass to the next generation, the entire population is sorted again based on non-domination, and the best ones, according to the same function as before, are selected (elitist criterion).

3.5 Fitness Function

We use two fitness functions to optimize two antithetical objectives: the cardinality of the topic subset, and its correlation value with respect to the full topic set. In particular, we investigate two instances of the problem, forcing the constraint to find a best/worst topic set for each cardinality:

- Finding the *Best* topic set: the cardinality has to be minimized, while the correlation has to be maximized.
- Finding the *Worst* topic set: the cardinality has to be maximized, while the correlation has to be minimized.

3.6 Choice of the Final Solution

Given the aim of the work, we are not interested in finding a single solution, but rather we collect the best (or worst, depending on the instance of the problem to be solved) solution for each possible topic subset cardinality. To do that, we store the Pareto-front of the initial population. Then, at the end of each iteration of the algorithm, we merge the current population with the stored ones, keeping only the non-dominated solutions (if two solutions for the same cardinality have also the same correlation value, then one of the two is randomly taken). We choose to consider the best 10 subsets for each cardinality, but a deeper analysis is possible.

3.7 Implementation

We implemented our algorithm named NewBestSub extending the jMetal framework[2] (version 5.0); jMetal is an Object Oriented framework based on Java, used for multi-objective optimization problems through meta-heuristics. More in detail, we extended the NSGA-II algorithm.

[2] https://github.com/jMetal/jMetal.

Concerning the parameters used for the collections (which mostly depend on the number of topics of the collection), we use: 2,000 as *population number*, i.e., the number of individuals in the population which, in our case, correspond to individual topic sets; 10 million as *max evals*, i.e., an upper bound on the number of evaluations carried out during the computation, and used as a stopping condition for the algorithm; 0.3 as *mutation probability* and 0.7 as *crossover probability*, which represent respectively the probability of applying the mutation and crossover operators to the selected individuals (i.e., the individual topic sets); finally, we choose 5,000 as *average repetitions*.

We implemented the software using the Kotlin[3] programming language, a multi-platform programming language developed by JetBrains, fully interoperable with Java.[4] The software is about 2,000 lines of code (plus comments); the full project code is available at https://github.com/Miccighel/newbestsub.

3.8 Discussion

As we will discuss later in more detail, compared to BestSub, our implementation is capable of achieving higher/lower correlation values; to do so, several runs of the EA are carried out starting from different initial populations. The final result is obtained by merging the several intermediate outcomes; this is in fact a commonly used (and trivial) practice to improve the results of EAs algorithms, and simultaneously to avoid overfitting. We run some experiments with 10 executions on various datasets: when comparing the correlation values obtained by running NewBestSub 10 times with the ones obtained by a single run of BestSub we found a small but not significant improvement in the correlation values of the Best/Worst topic in datasets with 50 topics, while major improvements (i.e., higher/lower correlation values for the Best/Worst subset) are observed for datasets with more than 50 topics.

The software presents also some limitations: in the current version it is not granted that, for each cardinality c, we can obtain the Best/Worst x (let us say 10) sets. However, experimentally we found that we obtain at least 10 solutions for each cardinality when using a ground truth of 50 topics, and at least 10 solutions for most of the cardinalities using a larger ground truth (i.e., 1,000 topics); we leave for future work to analyze the relation between the number of Best/Worst solutions we can obtain for a given cardinality and the selected algorithm parameters, as well as to study an alternative approach able to avoid this problem completely.

We remark that we decided to use a state-of-the-art setting, without any fine tuning of the parameters. By doing so, we aim to (i) avoid overfitting, (ii) keep the implementation simple, and (iii) obtain a fast algorithm, rather than focusing on finding the absolute Best and Worst topic subsets, which are extreme results per-se. The latter remark is also motivated by the result by Berto et al. [3], that shows that a high number of good topic sets exist; in detail, Reference [3, Figure 3] shows that for the TREC96 collection (our AH99_top96 dataset, see below), at a cardinality of half of the full topic set (i.e., 25 topics out of a ground truth of 50) more than 50% of the topics are "good" (i.e., subsets with $\tau > 0.85$ and $\rho > 0.96$), and 99% of the topics are good after cardinality 35.

4 EXPERIMENTS

4.1 Aims, Motivations

With the novel implementation of NewBestSub, we now turn to: (i) reproduce previous work, to see if the past results hold; (ii) compare the efficiency of NewBestSub with BestSub; and (iii) generalize

[3] https://kotlinlang.org/.

[4] See https://kotlinlang.org/docs/reference/ and https://kotlinlang.org/docs/reference/comparison-to-java.html.

Table 2. The Test Collections Used in the Experiments

	Acronym	TREC Official Name	Year	Topics	Runs
1.	AH99	Ad Hoc	1999	50	129
2.	AH99_top96	Ad Hoc	1999	50	96
3.	AH99_logAP	Ad Hoc	1999	50	129
4.	AH99_logAP_top96	Ad Hoc	1999	50	96
5.	AH99@20	Ad Hoc	1999	50	129
6.	AH99@20_top96	Ad Hoc	1999	50	96
7.	WEB14	Web Track (ad Hoc Task)	2014	50	30
8.	WEB14_top25	Web Track (ad Hoc Task)	2014	50	25
9.	WEB14B	Web Track (ad Hoc Task)	2014	50	30

and extend some results, performing some novel experiments that would not be feasible with the old BestSub.

4.2 Data

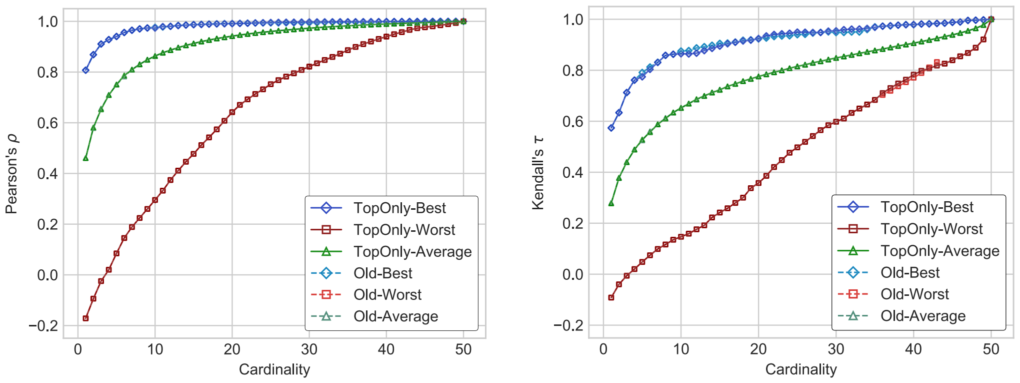
In our experiments we use the following nine datasets derived from TREC and summarized in Table 2:

1. AH99: the dataset obtained using the full TREC-8 Ad Hoc collection, with all the runs.
2. AH99_top96: the dataset obtained selecting from AH99 only the top96 runs, i.e., circa the top 75% of the most effective systems (this is the choice done by Guiver et al. [9]).
3. AH99_logAP: the dataset obtained using the logAP metric (this is equivalent to using the logarithm of AP values in Table 1).
4. AH99_logAP_top96: the dataset obtained using the logAP metric on the top runs, to further study the effect of considering the top 75% of the most effective systems.
5. AH99@20: the dataset obtained using a shallow pool (AP@20 is used in place of AP: values are computed considering the first 20 retrieved documents only).
6. AH99@20_top96: the dataset obtained combining shallow pool and the top 75% of the most effective systems.
7. WEB14: the dataset obtained from the TREC Web Track of 2014. This allows us to include in our analysis a more recent collection, and to compare it with TREC-8. This seems important to us since previous results were obtained on rather old collections.
8. WEB14_top25: the dataset obtained when selecting the circa top 75% of the most effective systems for WEB14.
9. WEB14B: WEB14 also allows us to use the official NDCG metric as well as two binarized AP metrics: in the former we consider as not relevant the qrels values −2 and 0, and as relevant the qrels values 1, 2, and 3; in the latter we consider as not relevant the qrels values −2, 0, and 1, and as relevant the qrels values 2 and 3; in the following, we report results for the former binarization since it leads to better results.

We decided to focus our analysis on the effects of different evaluation metrics, pool depth, number of systems, and collections; to do so, we choose all our datasets to have a fixed number of 50 topics. We leave as future work the study of the effects of varying the number of topics.

4.3 Reproduce Previous Work

The first experiment is to reproduce the same results as Guiver et al. [9]. Figure 4 shows the comparison between BestSub and NewBestSub for the Best, Worst, and Average series.



(a) (b)

Fig. 4. Correlation curves, Pearson's ρ (a) and Kendall's τ (b) for BestSub and NewBestSub on AH99_top96. For most of the cardinalities, the series are indistinguishable.

NewBestSub obtains almost the same results (i.e., correlation values) as BestSub. For some cardinalities, NewBestSub provides slightly different correlation values from the original BestSub: for example see, in the τ chart, cardinalities around 12 and 31 for the Best series and cardinalities around 40 for the Worst series. The Average series appears to be stable with 10,000 repetitions, and overlaps perfectly with the BestSub Average series.

4.4 Efficiency

Having shown that we are able to reproduce previous results, we now focus on the efficiency of NewBestSub. To test efficiency, we run two kinds of experiments: (i) we keep constant the number of topics (we use 50 topics, as in AH99_top96) and we vary the number of systems/runs, from 5 to 96 (the number of runs in AH99_top96) and (ii) we keep constant the number of systems (we use 96 runs, as in AH99) and we vary the number of topics, from 50 to 1,100 (the latter being an approximation of the number of topics in the largest collection available, i.e., the Million Query 2007 collection [1]).

The effect of the number of runs can be seen by analyzing the results of experiment (i), in Table 3(a). Of course, the more the number of runs m increases, the longer will be the vectors used to compute the correlation values; in NewBestSub, as in BestSub, the complexity is $O(m)$ for ρ and $O(m \log m)$ for τ. However, in practice the effect is rather small. The time complexity appears to be more than linear, but with a quite small growth.

The number of topics affects computation time to a much greater extent, as it can be seen in Table 3(b). The efficiency of NewBestSub is mainly influenced by the number of topics: the time complexity appears to be slightly more than linear, but with a much higher growth.

The rightmost three columns in Table 3(b) compare the efficiency of NewBestSub with the old BestSub. Speedup is defined as

$$\text{Speedup} = \frac{\text{Time(BestSub)}}{\text{Time(NewBestSub)}}.$$

The speedup obtained by NewBestSub, even when the old BestSub is used with a smaller k for a faster heuristics, is clear and very large. Even though the computation time is drastically reduced, the results are not affected: the correlation values obtained are similar when the comparison between the two softwares is possible, i.e., up to 250 topics.

Table 3. Time Comparison between BestSub and NewBestSub, on Varying the Number of Runs (a) and the Number of Topics (b)

Topics	Runs	Time NewBestSub
50	5	2 min, 00 sec
50	10	2 min, 04 sec
50	25	2 min, 08 sec
50	40	2 min, 15 sec
50	50	2 min, 18 sec
50	75	2 min, 30 sec
50	90	2 min, 38 sec
50	96	2 min, 41 sec

(a)

Topics	Runs	Time NewBestSub	k	Time BestSub	Speedup
50	96	3 min	2	30 min	10x
50	96	3 min	3	12 hours	240x
250	96	10 min	2	1 month	4380x
250	96	10 min	3	>4 months*	> 17,520x
500	96	20 min	2	>1 year*	> 26,280x
750	96	35 min	2	>1 year*	> 15,017x
1000	96	60 min	2	≫1 year*	≫ 8760x
1100	96	80 min	2	≫1 year*	≫ 6570x

*The execution was stopped before terminating.

(b)

Even better, NewBestSub is also much more effective when focusing on extreme value results (which might not be necessary, as detailed in Section 3.8): we run 10 executions of NewBestSub to maximize/minimize the correlation of Best/Worst subset as much as possible; we use 96 runs and 250 topics (the third row in Table 3(b)). The total time for NewBestSub is 10 times 10 minutes = 100 minutes. Correlation appears to be much lower for the Worst set (i.e., with a τ correlation difference up to 0.2) when compared with results of BestSub, with $k = 2$ as heuristic parameter; we also run BestSub with $k = 3$ only for correlation up to cardinality 50 using a ground truth of 250 (results obtained in a 1.5 months circa); results of NewBestSub with 10 executions are still significantly more optimal, especially for the Worst series.

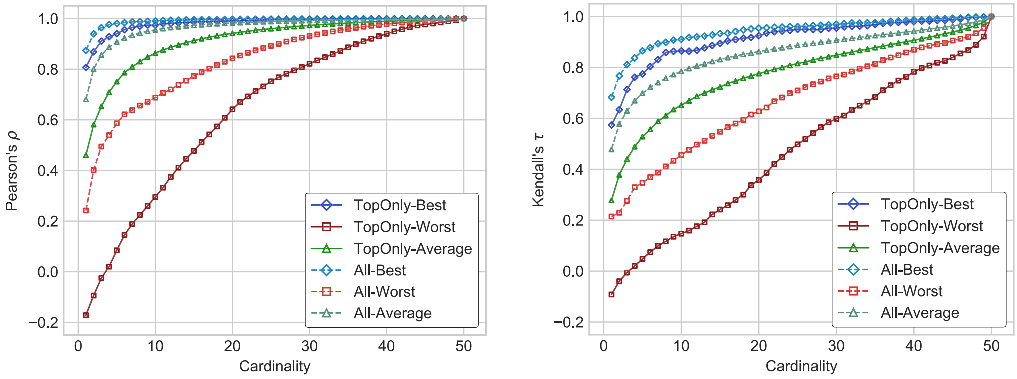
The much higher efficiency allows us to conveniently perform many novel experiments. In the remaining part of this article we discuss some of them.

4.5 Generalize the Results

In this section, we generalize previous work, considering the inclusion of the all runs or just the most effective one (Section 4.5.1), the stability of the Best/Worst sets (Section 4.5.2), and of the top 10 Best/Worst sets (Section 4.5.3), as well as a more recent collection, new metrics, and shallow pool effects (Section 4.5.4).

4.5.1 Top-only vs. All. One issue that, in general, hinders reproducibility is not using a whole dataset [7, 8]. In our case, this corresponds to not using the whole set of runs of a TREC track. This is a common practice in TREC data analysis [21], and it is usually justified by the need of removing buggy and not informative systems from the analysis. Figure 5 compares Correlation curves (both ρ and τ) obtained when using the whole AH99 dataset, to those obtained when including only the top 75% best runs, as it is done in the original AH99_top96 dataset. Considering only the top 75% of all systems leads to rather different results: the Best (and, especially, the Worst) series appear to achieve much higher (respectively, lower) correlation values. For example, when considering τ correlation and the Worst series, a correlation of 0.4 is achieved with 8 topics in AH99_top96, while 21 topics are needed in the case of considering the full AH99 dataset.

4.5.2 Stability of the Best and Worst Sets. As already suggested in Section 2.3, the heuristic adopted in BestSub might seriously affect the stability results. Conversely, also NewBestSub is



(a) (b)

Fig. 5. Comparison of correlation curves when including top 75% or all runs, for Pearson's ρ (a) and Kendall's τ (b) correlation measures.

based on a, yet more sophisticate, heuristic, and its effect on the stability of the Best/Worst sets needs to be studied.

Figure 6 shows two pixel-maps created on NewBestSub output. When comparing them to the old BestSub pixel-maps in Figure 2, it might appear that the stability of the Best topic subset, and of the 10 Best topics subset, is higher with NewBestSub (Worst topic subsets seems instead similar). To quantify this sensation, we use the measure of stability of the Best/Worst set defined as follows by Guiver et al. [9, page 21:14]:

> For each topic in a set of a given size, a counter is incremented if the topic is in the set the next size up. For the exhaustive search, this counter has a minimum and maximum possible value, where the minimum is greater than 0 due to the fact that as topic sets get larger there is some inevitable overlap from one topic set to the next. The stability value (expressed as a percentage) is given by the ratio
>
> $$\frac{\text{counter actual} - \text{counter min}}{\text{counter max} - \text{counter min}}.$$
>
> For Worst subsets, the average stability for Average precision across the three goodness measures is 93%. Best subsets are somewhat less consistent, with an average of 86%.

Note that Guiver et al. [9] do not specify how to compute the counter max/min values. We derive that they can be computed as follows (where n is the number of topics, as usual):

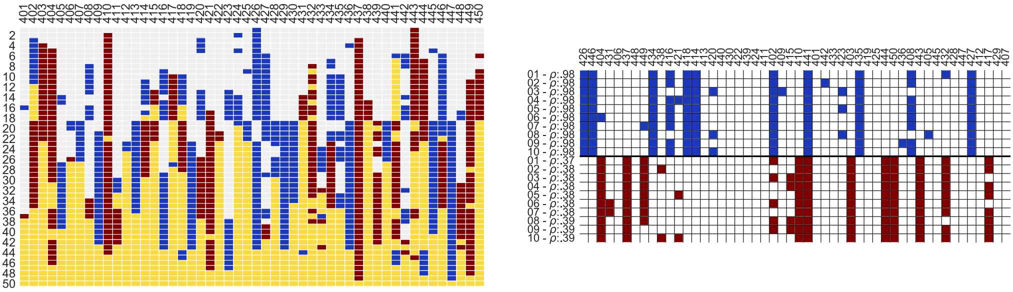
$$\text{counter min} = \begin{cases} 0, & \text{if } c \leq \lfloor \frac{n}{2} \rfloor - 1 \\ 2c + 1 - n, & \text{otherwise} \end{cases}$$

$$\text{counter max} = c.$$

Table 4 shows the stability values for the Best/Worst set for all the dataset used, plus the values for BestSub on AH99_top96. When comparing BestSub and NewBestSub (i.e., rows 0 and 2 of the table), different from the first sensation from Figure 6, the stability values are almost identical. This is also clear when comparing the averages of stability for NewBestSub across all nine datasets (last row in the table) with the old BestSub. Also, in general, correlation values are similar across all datasets, with the stability values for the Worst set always higher than the value for the Best set;

Table 4. Stability Values for the Best and Worst Sets Using Guiver et al. [9] Measure, for ρ and τ Correlation

		Pearson's ρ		**Kendall's τ**	
		Best set	Worst set	Best set	Worst set
0.	BestSub on AH99_top96	.88	.98	.84	.95
1.	AH99	.85	.97	.86	.97
2.	AH99_top96	.88	.98	.88	.95
3.	AH99_logAP	.83	.96	.85	.92
4.	AH99_logAP_top96	.89	.95	.82	.93
5.	AH99@20	.83	.96	.85	.92
6.	AH99@20_top96	.89	.94	.89	.90
7.	Web14	.88	.92	.84	.89
8.	Web14_top25	.87	.95	.82	.88
9.	Web14B	.85	.97	.61	.91
	Average	.86	.95	.82	.92



(a) (b)

Fig. 6. Stability pixel-maps for Pearson's ρ for the whole AH99_top96 dataset (a) and for Pearsons's ρ for the top 10 sets at cardinality 12 (b). The color blue is used to represent topics in the Best set (different from Figure 2), red for the topics in the Worst set, and yellow for the topics in the Best and Worst set.

the only peculiar dataset is WEB14B, but the lower value for the τ Best set stability can be caused by the binarization process.[5] Therefore, these results confirm previous findings:

- once a topic set enters in the Worst set at a certain cardinality, it tends to remain in the Worst set also for the consequent cardinalities;
- the previous finding is less true for Best topics;
- a Worst topic set is formed by individual Worst topics, while Best topic sets are not necessarily formed by individual Best topics. In other words, a set formed by Worst individual topics in general is Worst, while this is not true for Best topics.

Thus, we can conclude that the stability results of Guiver et al. [9] still hold with the completely different heuristic that we used, and therefore it seems unlikely that they depend on the (quite rough) heuristic they used.

[5] As well as by the particular nature of that dataset that is known to be rather incomplete due to shallow pools and low number of participants [11].

Table 5. Stability Values for the Best/Worst 10 Sets for τ Correlation

		Best 10 sets						**Worst 10 sets**					
	Collection	cardinality						cardinality					
		5	10	20	30	40	45	5	10	20	30	40	45
1.	AH99	.60	.74	.89	.92	.80	.65	.74	.84	.94	.93	.84	.71
2.	AH99_top96	.49	.82	.92	.90	.84	.64	.71	.84	.93	.91	.72	.67
3.	AH99_logAP	.58	.78	.88	.88	.85	.73	.73	.83	.91	.92	.88	.73
4.	AH99_logAP_top96	.60	.81	.80	.86	.62	.62	.60	.86	.91	.88	.86	.71
5.	AH99@20	.54	.74	.90	.83	.62	.62	.67	.82	.90	.92	.78	.56
6.	AH99@20_top96	.56	.77	.94	.89	.70	.56	.67	.82	.92	.89	.71	.51
7.	WEB14	.67	.72	.91	.61	.53	.22	.73	.83	.92	.93	.86	.58
8.	WEB14_top25	.47	.64	.89	.46	.21	.20	.78	.82	.90	.89	.84	.62
9.	WEB14B	.29	.20	.47	.43	.25	.18	.69	.85	.75	.89	.82	.60
	Average	.53	.69	.84	.75	.60	.49	.70	.83	.90	.91	.81	.63

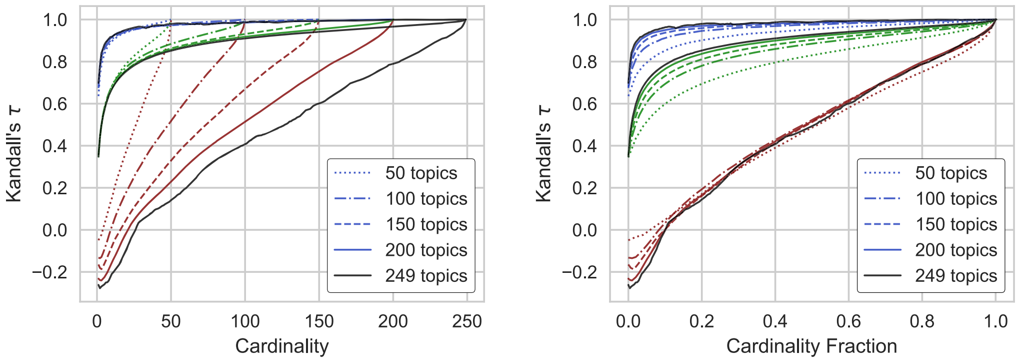
4.5.3 Stability of the Top 10 Best and Worst Sets. We turn now to study the stability of the top 10 Best/Worst sets, as done by Guiver et al. [9] and Berto et al. [3]. We can generalize the stability measure used in Section 4.5.2 to include the top p sets (in this case $p = 10$) at a given cardinality c. Thus, as in Section 4.5.2, we can compute a stability value counter actual as done by Guiver et al. [9] where, in this case,

$$\text{counter min} = \begin{cases} (2c - n)(p - 1), & \text{if } c > \lfloor \frac{n}{2} \rfloor \\ 0, & \text{otherwise} \end{cases}$$
$$\text{counter max} = c(p - 1).$$

Table 5 shows the stability values for the Best/Worst 10 sets at some selected cardinality. We report the results for τ correlation values only; the outcome for ρ correlation is similar. As we can see reading Table 5 column-wise (i.e., for each cardinality), results are quite similar across collections, with the exception of (again) WEB14 and, to a lesser extent, WEB14_top25 (see again Footnote 5). As we can see, reading the table row-wise (i.e., for each collection), the stability values are different at different cardinality values: the maximum stability value is reached around cardinality 20 and 30 for all the collections, whereas the value is minimal at cardinalities of 5 and 45. But the main remark is that, generalizing the result of Section 4.5.2, it is clear that the 10 Worst sets are much more stable than the 10 Best ones. Including the top 75% of the runs or the whole datasets changes the stability values: for example, considering cardinality 5, the stability of the Best 10 sets for AH99 is 0.60, while for AH99_top96 is 0.49.

4.5.4 A Recent Collection, other Evaluation Metrics, and a Shallow Pool. We use NewBestSub on a more recent collection with a new evaluation metric: the NDCG metric of the WEB14 collection. We do not show the corresponding charts as the overall results look very similar to those of the other collections. Some of the more specific results are quite similar as well. Including or not the top 75% of the runs still makes a difference with WEB14 and NDCG, as results are comparable with Figure 5; and stability values for the Best/Worst single sets are similar to the other collections. On the contrary, stability values for the Best/Worst 10 sets are lower for Web14 than for the other collections, especially at cardinalities 5 and 45.

[5] As well as by the particular nature of that dataset that is known to be rather incomplete due to shallow pools and low number of participants [11].



(a)

(b)

Fig. 7. τ correlation series when starting from a topic population of various sizes taken from Robust 2004 dataset, as a function of the absolute cardinalities (a) and of the cardinality fractions (b).

Concerning using a shallow pool (i.e., AP@20), we see that when comparing AH99 with AH99@20, results are similar both for the stability of single and Best/Worst 10 sets.

As a final remark, we can state that the reproduced results (Section 4.3) alsoseem quite stable on the more recent collection when using NDCG and/or with a shallower pool, with very few exceptions.

4.5.5 Larger Ground Truth. We now briefly discuss the effect of the size of the initial topic set on the Best and Worst series. To this aim, we design an experiment as follows: we make use of the Robust track from 2004, which includes 249 topic and 110 runs [20]. We simulate five larger ground truths of different sizes by randomly sampling $t \in \{50, 100, 150, 200, 249\}$ topics. We then run NewBestSub on the selected topic set. To limit noise and give stability to the results, we repeat the process 20 times for each sampling. Finally, we compute for each cardinality $c \in \{1, \ldots, t\}$, the average τ correlation over the 20 repetitions.

Figure 7 shows the five Best, five Worst, and five Average series for the different five initial topic set sizes. Besides the usual correlation chart with the cardinality on the horizontal axis (Figure 7(a)), we also show the chart with the correlation as a function of the cardinality fraction of the full topic set (Figure 7(b)), which is perhaps more informative in this case. From this second chart, we see that the initial size of the ground truth has an impact on both the Best and Worst series; the larger the initial topic set, the higher (respectively, lower) the correlation for the Best (respectively, Worst) series. The behavior of the Average series is similar to that of the Best one.

So far, the effects of using fewer topics has been studied on collections having a ground truth of 50 topics. These results hint that these effects still hold on larger collections, and they even become more extreme. However, it has to be noted that there are many variables that need to be taken into consideration, such as the topic set (i.e., different collections have different topics), the collection task and track, the participating runs, the effectiveness metric, and so on. Therefore, we leave for future work a more systematic study of the effect of the initial ground truth size, which is now made possible by NewBestSub.

4.6 Expand the Results

In this section, we now turn to our last aim (item (4) in Section 1): to exploit the novel NewBestSub to perform some experiments that go beyond those of the previous studies. More in detail, we compute an approximated version of the original matrix, using two methods detailed in the following: the method by Soboroff et al. [17], and the hubness analysis, proposed by Mizzaro and Robertson [12].

The methodology developed by Soboroff et al. [17] works as follows. To compute an effectiveness evaluation metric, human annotators are required to produce a relevance judgment for each document retrieved by the IR systems and present in the pool of documents, forming the so-called *qrels*. Instead of using human annotators, Soboroff et al. method builds a fully automatic so-called *pseudo-qrels* file. This is done by simply randomly selecting (pseudo) relevant documents from the pool. Such a methodology has limited effectiveness, as it produces a τ correlation with the ground truth of circa 0.5 in the best cases. However, its outcome is an approximated topic-system table, as the one in Table 1: we denote it by AH99*.

The hubness analysis was proposed by Mizzaro and Robertson [12] and used as a strategy for topic selection by Robertson [13], as detailed in Section 2. Such a methodology allows us to compute a hubness score for each topic, representing its ability to identify effective systems.

With the approximated matrix AH99*, and with the topic hubness computed for each topic of AH99*, it can be the case that if we use NewBestSub to obtain the Best/Worst few topic subsets for the approximated matrix AH99*, these subsets are general enough to be Best/Worst topic subsets also on the original matrix. It can also be the case that, instead, the topic with highest/lowest hubness are general enough to be Best/Worst topic subsets, also the original matrix.

Note that these experiments aim to provide effective practical ways to select a subset of a few Best/Worst topics: the methodology by Soboroff et al. [17] is completely *a priori* (i.e., before the evaluation process) and the hubness analysis is performed on such a matrix.

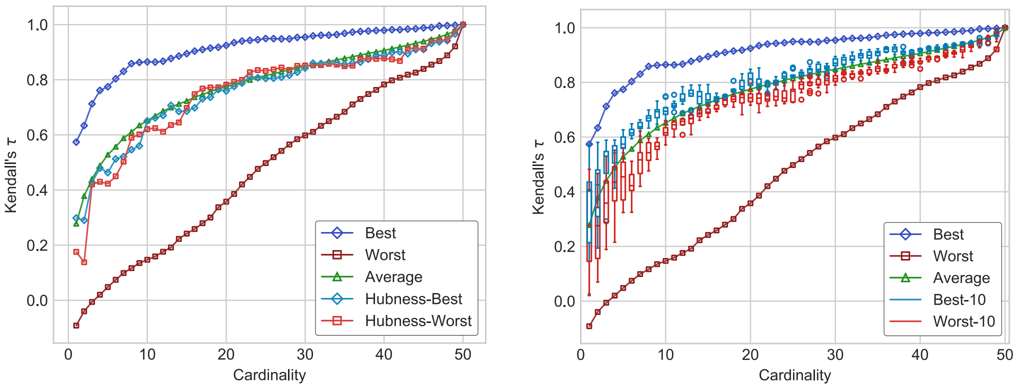
Thus we perform the following experiments. Given a true AP matrix (e.g., AH99), we produce, using Soboroff et al. [17], the corresponding approximated AP matrix, in this case AH99*. Now we compare two topic selection strategies:

(1) We run NewBestSub on AH99* and we select for each cardinality the 10 Best topic subsets (as well as the 10 Worst ones).
(2) We select the topic sets from AH99* according to their hubness value; to form a Best/Worst topic set at cardinality c we select the Best/Worst c topics sorted by descending/ascending hubness value.

Finally, we use for each cardinality c the Best/Worst topic sets selected according to (1) and (2) (i.e., from the AH99* matrix), to check their correlation with the actual ground truth (i.e., in the original matrix).

Figure 8 shows the series Best, Worst, and Average on the AH99_top96 dataset, and the series computed on AH99_top96* and used in the AH99_top96 dataset. Figure 8(a) shows that hubness (computed on the approximate matrix) does not help in finding a subset of a few good topics; in fact, both the hubness series are below the Average series. This can be caused by (i) generality of the measure: it can be the case that the hubness measure is not general enough to be computed on the approximated matrix and used in the original one; (ii) the approximated matrix is not enough representative of the real matrix: the approximated matrix computed using Soboroff et al. [17] has τ values around 0.5, thus in general is not representative of the real matrix.

Figure 8(b) shows the Best/Worst 10 topic subsets series (computed on the approximate matrix) and used in the real matrix; we represent, for each cardinality, the result of the evaluation of the Best/Worst 10 sets with a box-plot. On the contrary to Figure 8(a), Figure 8(b) shows that the 10 Best/Worst topic set computed on the approximate matrix are general; that is, almost all Best (Worst) sets are above (below) the average series. This outcome provides an effective and practical topic selection strategy: if a researcher wants to perform experiments on a few good topic sets, he



(a) (b)

Fig. 8. τ correlation series when selecting few topics using Hubness (a), and Soboroff et al. Best/Worst 10 topics (b) on the AH99_top96 dataset.

can simply run NewBestSub on the approximated matrix built using Soboroff et al. [17] and use one of the Best 10 topic subsets at the desired cardinality.

5 CONCLUSIONS AND FUTURE WORK

Our contribution is fourfold: (i) we re-implement BestSub using a novel approach based on EAs; (ii) we successfully reproduce the results by Guiver et al. [9], Robertson [13], and Berto et al. [3]; (iii) we generalize such results to other metrics and collections; and (iv) we extend such results by proposing an effective *a priori* topic selection strategy.

NewBestSub, the novel implementation of BestSub, besides allowing us to obtain the previous results, also leaves plenty of space for future work. The first improvement that we can study is a fine tuning of the algorithm parameters, such as experimenting with other operators to perform crossover, like XAND and XOR; study the relations between and give an initial accurate estimate of *population number* and *number of iterations*; and, in the case of more than one execution and the merge of the results, find the optimal number of executions. We also aim at modifying NewBestSub such that it finds the most (or a more) general topic set, i.e., a topic set which maximizes both the correlation with the ground truth and the ability to be a Best/Worst set in other collections.

The consistent speedup allows for many new interesting research possibilities. For example, we aim at reproducing and extending the generalization experiments of Guiver et al. [9], not by using a single split of the original AP matrix (both topic-wise and system-wise), but by performing many iterations of the process. With the old BestSub this experiment would be unfeasible. We aim also at developing the effective topic selection strategy, integrating other state-of-the-art methods to compute approximated matrix, such as the work by Spoerri [19] and Wu and Crestani [24]. We believe that NewBestSub will be a useful tool to perform these experiments, as well as several other ones.

ACKNOWLEDGMENTS

We would like to thank Luca Di Gaspero for providing CPU time for our experiments, as well as for interesting and stimulating discussions on the original idea of implementing an efficient version of BestSub.

REFERENCES

[1] James Allan, Ben Carterette, Javed A. Aslam, Virgil Pavlu, Blagovest Dachev, and Evangelos Kanoulas. 2007. *Million Query Track 2007 Overview.* Technical Report. NIST. http://trec.nist.gov/pubs/trec18/papers/MQ09OVERVIEW.pdf.

[2] David Banks, Paul Over, and Nien-Fan Zhang. 1999. Blind men and elephants: Six approaches to TREC data. *Information Retrieval* 1, 1 (1999), 7–34.

[3] Andrea Berto, Stefano Mizzaro, and Stephen Robertson. 2013. On using fewer topics in information retrieval evaluations. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval (ICTIR'13).* ACM, New York, NY, Article 9, 8 pages. DOI: https://doi.org/10.1145/2499178.2499184

[4] Chris Buckley and Ellen M. Voorhees. 2000. Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00).* ACM, New York, NY, 33–40. DOI: https://doi.org/10.1145/345508.345543

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197. DOI: https://doi.org/10.1109/4235.996017

[6] Agoston E. Eiben and J. E. Smith. 2003. *Introduction to Evolutionary Computing.* Springer-Verlag.

[7] Nicola Ferro. 2017. Reproducibility challenges in information retrieval evaluation. *Journal of Data and Information Quality* 8, 2, Article 8 (Jan. 2017), 4 pages. DOI: https://doi.org/10.1145/3020206

[8] Nicola Ferro, Norbert Fuhr, Kalervo Järvelin, Noriko Kando, Matthias Lippold, and Justin Zobel. 2016. Increasing reproducibility in IR: Findings from the Dagstuhl seminar on reproducibility of data-oriented experiments in E-science. In *ACM SIGIR Forum*, Vol. 50. ACM, 68–82.

[9] John Guiver, Stefano Mizzaro, and Stephen Robertson. 2009. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Transactions on Information Systems* 27, 4, Article 21 (Nov. 2009), 26 pages. DOI: https://doi.org/10.1145/1629096.1629099

[10] Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46, 5 (Sept. 1999), 604–632. DOI: https://doi.org/10.1145/324133.324140

[11] Xiaolu Lu, Alistair Moffat, and J. Shane Culpepper. 2016. The effect of pooling and evaluation depth on IR metrics. *Information Retrieval* 19, 4 (Aug. 2016), 416–445. DOI: https://doi.org/10.1007/s10791-016-9282-6

[12] Stefano Mizzaro and Stephen Robertson. 2007. Hits hits TREC: Exploring IR evaluation results with network analysis. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07).* ACM, New York, NY, 479–486. DOI: https://doi.org/10.1145/1277741.1277824

[13] Stephen Robertson. 2011. On the contributions of topics to system evaluation. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval, Volume 6611 (ECIR'11).* Springer-Verlag, New York, 129–140. DOI: https://doi.org/10.1007/978-3-642-20161-5_14

[14] Kevin Roitero, Eddy Maddalena, and Stefano Mizzaro. 2017. *Do Easy Topics Predict Effectiveness Better Than Difficult Topics?* Springer International Publishing, Cham, 605–611. DOI: https://doi.org/10.1007/978-3-319-56608-5_55

[15] Tetsuya Sakai. 2014. Designing test collections for comparing many systems. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14).* ACM, New York, NY, USA, 61–70. DOI: https://doi.org/10.1145/2661829.2661893

[16] Tetsuya Sakai. 2016. Topic set size design. *Information Retrieval Journal* 19, 3 (1 Jun 2016), 256–283.

[17] Ian Soboroff, Charles Nicholas, and Patrick Cahan. 2001. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01).* ACM, New York, NY, 66–73. DOI: https://doi.org/10.1145/383952.383961

[18] Karen Sparck Jones and Cornelis Joost van Rijsbergen. n.d. Information retrieval test collections. *Journal of Documentation* 32, 1. DOI: https://doi.org/10.1108/eb026616

[19] Anselm Spoerri. 2005. How the overlap between the search results of different retrieval systems correlates with document relevance. *Proceedings of the American Society for Information Science and Technology* 42, 1 (2005). DOI: https://doi.org/10.1002/meet.14504201175

[20] Ellen M. Voorhees. 2004. Overview of the TREC 2004 robust track. In *Proceedings of The Thirteenth Text Retrieval Conference (TREC'04).* Vol. 4. http://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf.

[21] Ellen M. Voorhees and Chris Buckley. 2002. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02).* ACM, New York, NY, 316–323. DOI: https://doi.org/10.1145/564376.564432

[22] Ellen M. Voorhees and Donna Harman. 2000. Overview of the *Proceedings of the 8th Text REtrieval Conference (TREC-8).* 1–24.

[23] William Webber, Alistair Moffat, and Justin Zobel. 2008. Statistical power in retrieval experimentation. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08).* ACM, New York, NY, 571–580. DOI: https://doi.org/10.1145/1458082.1458158

[24] Shengli Wu and Fabio Crestani. 2003. Methods for ranking information retrieval systems without relevance judgments. In *Proceedings of the 2003 ACM Symposium on Applied Computing (SAC'03)*. ACM, New York, NY, 811–816. DOI: https://doi.org/10.1145/952532.952693

[25] Justin Zobel. 1998. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*. ACM, New York, NY, 307–314. DOI: https://doi.org/10.1145/290941.291014

Received October 2017; revised April 2018; accepted July 2018