Deciding on an Adjustment for Multiplicity in IR Experiments

Leonid Boytsov
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
leo@boytsov.info

Anna Belova
Abt Associates Inc.
Bethesda, MD, USA
anna@belova.org

Peter Westfall
Texas Tech University
Lubbock, TX, USA
peter.westfall@ttu.edu

ABSTRACT

We evaluate statistical inference procedures for small-scale IR experiments that involve multiple comparisons against the baseline. These procedures adjust for multiple comparisons by ensuring that the probability of observing at least one false positive in the experiment is below a given threshold. We use only publicly available test collections and make our software available for download. In particular, we employ the TREC runs and runs constructed from the Microsoft learning-to-rank (MSLR) data set. Our focus is on non-parametric statistical procedures that include the Holm-Bonferroni adjustment of the permutation test p-values, the MaxT permutation test, and the permutation-based closed testing. In TREC-based simulations, these procedures retain from 66% to 92% of individually significant results (i.e., those obtained without taking other comparisons into account). Similar retention rates are observed in the MSLR simulations. For the largest evaluated query set size (i.e., 6400), procedures that adjust for multiplicity find at most 5% fewer true differences compared to unadjusted tests. At the same time, unadjusted tests produce many more false positives.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance Evaluation*

General Terms

Experimentation

Keywords

Statistical significance, multiple comparisons, t-test, MaxT, permutation test, randomization test, Holm-Bonferroni.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

1. INTRODUCTION

1.1 Description of the Problem

Multiple comparisons/testing is a fundamental experimental issue that arises when a certain hypothesis is being repeatedly tested in different settings. For example, a researcher proposes a new retrieval algorithm and verifies its effectiveness against a baseline. In reality, this method is equivalent to the baseline, but, after exhaustive testing with different collections and parameter settings, he observes a statistically significant improvement, which has happened by chance. Most false positives arising from multiple testing can be eliminated by considering a family of tests as a whole and requiring stronger evidence, i.e., smaller p-values, in each test. This approach is commonly referred to as an adjustment for multiple comparisons (testing).

The multiple comparisons issue received a lot of attention in a bio-medical research. In clinical trials, the cost of making a wrong conclusion is high. Thus, the US Food and Drug Administration strongly recommends to employ adjustments for multiple comparisons and requires a justification if multiplicity adjustments are not performed [1]. In contrast, in IR experiments, multiplicity issues are rarely taken into account. Yet, there is a non-negligible cost related to (1) human effort in reproducing experimental results, (2) computational effort related to aggregating results from several retrieval methods. These efforts are wasted on methods whose improvement over the baseline was observed due to spurious, i.e., random effects. This is why we believe that the IR community should also adopt the practice of reporting corrected p-values.

How do we define a family of tests where p-values should be adjusted for multiple comparisons? It turns out that the choice of the family is very subjective [6, 33]. Westfall and Young state that

> ...there can be no universal agreement: statisticians have argued back and forth (sometimes vehemently) over this issue, often arriving at dramatically different conclusions [33].

They note, however, that there is more agreement on adjusting p-values in a single experiment. This is especially pertinent when results are summarized in a single conclusion [2]. For example, the researcher may compare 10 methods against a baseline, adjust p-values, and state that only 3 differences are jointly significant.

In our work we adopt this point of view and focus on adjustments that provide a strong control of a family-wise

error rate (FWER) at a significance level α. In other words, the probability of observing a false positive among all tests is at most α. We also limit our attention to the case when a small number of methods are compared against a single baseline. This is a common scenario in the TREC setting, where a group submits 2-3 official runs that are evaluated by TREC organizers. Additionally, the group may evaluate several unofficial runs on their own (using relevance judgements produced by TREC assessors). There are several other approaches to deal with multiple testing that provide a weaker control: e.g., limiting the probability to observe at most $k > 1$ false positives [15, 35] or controlling a false discovery rate (FDR) [3]. We believe that these methods are less useful for the purpose of discovering and publishing significant results, but they may be appealing to practitioners, e.g., those who seek to reduce dimensionality of machine learning models [37, 21].

1.2 Related Work

There are several papers covering a wide range of reliability issues in IR experiments [22, 5, 31, 25, 38]. We encourage the reader to follow these articles and references therein.

Wilbur [34] carried out the first comprehensive assessment of methods for testing statistical significance in IR. He used several pre-TREC collections and evaluated the Wilcoxon test, the sign test, the permutation test (also known as the randomization test), and several modifications of bootstrapping. According to Wilbur, the permutation test and the bootstrapping test had comparable statistical power, superior to that of the Wilcoxon and the sign test. These findings were confirmed by Smucker et al. [27] who conducted similar experiments using several much larger TREC collections. The experiments of Cormack and Lynam [10], though, showed that both the Wilcoxon and the sign test were sufficiently accurate and powerful, but somewhat inferior to the t-test. In addition, they discovered that there was a strong agreement among the t-test, the bootstrapping test, and the permutation test. Savoy [23] recommended to use bootstrapping to estimate the sample median instead of the sample mean.

There are also several papers focusing on multiple testing adjustments in IR experiments. Tague-Sutcliffe and Blustein carried out a statistical analysis of TREC-3 results [29] and adjusted them using the Scheffé's method [24]. They found that only large differences in performance metrics could be considered significant. Blanco and Zaragoza [4] presented an experimental analysis of spurious effects in IR and advocated for adoption of multiple comparisons adjustments. Carterette [7] modeled randomness with a linear regression and adjusted p-values for multiplicity using a single-step method that relied on multivariate Student distribution. He found that in TREC-8 relative pairwise differences in the mean average precision smaller than about 50% were insignificant, which is in line with earlier findings of Tague-Sutcliffe and Blustein [29].

The focus of our paper is on permutation tests. These procedures were independently proposed by Pitman [18] and Fisher [12] in the 1930s, long before advances in computer hardware made this approach practical. A straightforward generalization of the permutation test that accounts for multiple testing is based on the closure principle proposed by Marcus et al. [16]. It entails verification of up to $2^m - 1$ null hypotheses (m is the number of tests). Westfall and Young proposed a computational shortcut, which allows one to consider only m hypotheses [33, 32]. One method of Westfall and Young, called the MaxT permutation test, was shown to have high statistical power among methods that provided a strong control of the FWER in microarray experiments [11].

2. HYPOTHESIS TESTING

We consider a standard experimental setting in IR. There is a set of queries, which represent user's information needs, ground truth relevance judgements for these queries, and several retrieval systems. Selection of queries can be thought of as a random sampling from an infinite (or very large) population. The relevance judgements are compared against ranked sets of documents (called runs) retrieved by these systems in response to q queries. Effectiveness of retrieval is characterized by scores computed separately for each query using a performance metric, such as the Expected Reciprocal Rank at depth 20 (ERR@20). The mean of query-specific scores is then used to evaluate the overall performance of retrieval systems.

Let scores of systems X and Y be represented by vectors $x = (x_1, x_2, \ldots, x_q)$ and $y = (y_1, y_2, \ldots, y_q)$ with mean values equal to $\bar{x}$ and $\bar{y}$, respectively. Even if $\bar{x}$ is substantially larger than $\bar{y}$, we cannot safely infer that Y is inferior to X in the long run. The scores are highly variable across queries [29] and it is not uncommon for an inferior system to outperform a superior system on some subpopulation of queries. Performance of X and Y in this subpopulation is not a good indication of the relative performance in the entire population of queries. There is always a chance that our test sample has a lot of queries for which the inferior system outstrips the superior one. Thus, the measured difference between X and Y could be attributed to random sampling effects.

Significance testing is a standard approach to deal with this problem. Testing involves the following steps:

1. An IR researcher formulates a null hypothesis H (or simply a null), e.g., by assuming that there is no difference in ERR@20 (or some other performance metric) between X and Y. That is, the population means are equal. In addition, he sets a significance level α that controls the rate of false rejections (i.e., false positives).

2. He chooses a test statistic $T(x, y)$ (a function of measured systems' scores) that provides evidence against the null hypothesis H. One example is the sample mean difference: $T(x, y) = \bar{x} - \bar{y}$. Small absolute values of this statistic present evidence in favor of equality of population means, while large ones may signify that H is not true. Another well-known statistic is the paired t-statistic:

$$\frac{(\bar{x} - \bar{y})\sqrt{q(q-1)}}{\sqrt{\sum_{i=1}^{q}(x_i - y_i - \bar{x} + \bar{y})^2}}. \quad (1)$$

3. The researcher quantifies the evidence provided by the test statistic. Formally, he computes a statistic value $t = T(x, y)$ from the sample data. Then, he estimates the probability of obtaining a test statistic value at least as extreme as t *under the null hypothesis* (i.e., when H is true). This probability is known as a p-value. If the p-value is less than the chosen significance level

α, the observed value of the statistic is unlikely to happen by chance (i.e., due to *randomness in selection of queries*). Thus, the researcher can reject the null hypothesis with confidence $1 - \alpha$. We discuss this approach in Section 2.2 in more detail.

To compute the p-value, we need to know the distribution of the test statistic under the null. In a parametric approach, we assume that data follows a theoretical distribution, which allows us to derive the distribution of the test statistic analytically. A widely used parametric test is the Student's t-test. In a non-parametric approach, the distribution of the test statistic is estimated through resampling of observed data (see Sections 2.2-2.3).

When, we observe an unusually small p-value this may be due to the following:

1. The null hypothesis is not true;
2. The null hypothesis is true and extreme statistic value is observed by chance;
3. Some underlying assumptions are violated.

The null hypothesis can be true even when the statistic value is extreme. Yet, if we reject the null only when the corresponding p-value is less than α, we ensure that in a series of repeated experiments the probability to incorrectly reject the true null is α. Thus, in the frequentist approach, one should avoid a temptation to interpret the p-value as the probability of the null hypothesis being true or as another measure that quantifies the veracity of the null.

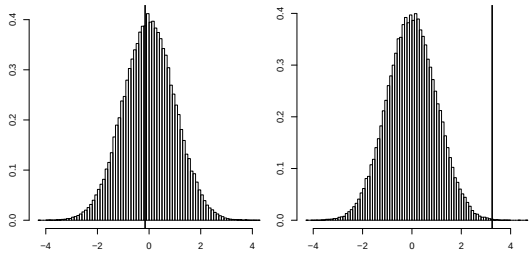
Note especially the third case. If the statistical procedure relies on the distributional assumptions (such as the normality assumption for the t-test) and these assumptions are violated, this may also lead to a rejection of the null. Unfortunately, there is no good way to control a rate of false rejections due to assumption violations. Thus, it is very desirable to use tests requiring minimal assumptions such as the non-parametric randomization procedures assessed in our work.

Effectiveness of a testing procedure can be characterized by a proportion of *true positives* (correctly rejected false null hypotheses) and by a proportion of *false positives* (incorrectly rejected true null hypotheses).

2.1 Multiple Testing

The significance level α controls the probability of a false positive under the true null hypothesis only in a single test. Consider an example, where the researcher slightly modifies a baseline method 100 times and measures changes in performance. The significance level in each test is $\alpha = 0.05$. Suppose that these modifications of the baseline method did not result in any real improvements. Therefore, he may expect to obtain at least one false positive with the probability of $1 - (1 - \alpha)^{100} \approx 0.99$, and five false positives on average. If the researcher is sufficiently naïve, he may decide that merely obtaining relevance judgements for a larger set of queries will help to overcome this problem. However he would still obtain about five false positives on average, irrespective of the number of queries used. One can easily verify this statement using the simulation approach presented in Section 3.3.

This problem can be addressed by using an adjustment for multiplicity in testing. The classic adjustment method is the Bonferroni procedure. Let $p_1, p_2, \ldots, p_m$ be a set of unadjusted p-values. The Bonferroni method consists in multiplying each p_i by the number of tests m (values larger than 1 are set to 1). Then, we reject hypotheses with p-values smaller than α. This procedure is equivalent to enforcing a significance level of α/m in each of the m tests. In other words, the probability to observe a false positive in a single experiment should be α/m, assuming that the null is true. Using the union bound, we obtain that in a series of m experiments, the probability to encounter at least one false positive is controlled at the level α.



(a) Identical systems (b) Different systems

Figure 1: Distribution of statistic values obtained through random 100,000 permutations. Thick vertical lines denote statistic values computed from non-permuted system scores.

The Bonferroni adjustment provides a strong control of a family-wise error rate (FWER) at the significance level α, but it is conservative. The Holm-Bonferroni adjustment [13] is a slightly more powerful method. Let $p_1 \leq p_2 \leq \ldots \leq p_m$ be an ordered set of unadjusted p-values. The Holm-Bonferroni adjustment entails multiplying p_i by $m - i + 1$ and enforcing monotonicity of obtained values. Formally, the i-th adjusted p-value is equal to:

$$\min(1, \max_{j \leq i} p_j \cdot (m - j + 1)). \quad (2)$$

In the subsequent sections we present several non-parametric adjustment methods based on randomization. The discussion starts with a description of the permutation test for two systems.

2.2 Permutation Test (Two Systems)

We compare two systems represented by performance scores $x = (x_1, x_2, \ldots, x_q)$ and $y = (y_1, y_2, \ldots, y_q)$. The hypothesis of interest is whether systems' mean population values of the performance metric (e.g., ERR@20) are equal. We additionally assume that under the null hypothesis H the values of x and y are outcomes of *exchangeable* multivariate random variables X and Y. This can be viewed as a combination of two random processes. The first random process generates a pair of scores in response to a query. The second process randomly labels one of the scores as belonging to X and another as belonging to Y (with equal probabilities).

From the practical perspective, this means that the distribution of the test statistic under the null hypothesis can be computed by the following randomization procedure. First, the vectors of performance scores x and y are stored in the

form of the matrix with each vector representing a row:

$$\left| \begin{array}{cccc} x_1 & x_2 & \ldots & x_q \\ y_1 & y_2 & \ldots & y_q \end{array} \right|.$$

Then we repeatedly obtain new pseudo-observation vectors $\widetilde{x}$ and $\widetilde{y}$ by randomly exchanging (i.e., permuting) values in the columns of this matrix. If the hypothesis H is true, all such observations are equally likely outcomes of exchangeable variables X and Y. If, in addition, we compute the value of the statistic $T(\widetilde{x}, \widetilde{y})$ for all possible 2^q permutations, we obtain an exact distribution of the test statistic (under the null). Computing all 2^q statistic values is intractable for all but very small q. Instead, the distribution could be approximated by carrying out sufficiently many random permutations B.

In Figure 1, there are two approximate distributions of the t-statistic for $B = 100,000$. The thick vertical lines indicate the values of the statistic $t = T(x, y)$ computed using non-permuted vectors x and y. The rightmost distribution in Figure 1 was computed for different systems. The value is $t \approx 3.5$ and only about one in 2,000 of computed statistic values exceeds t. The p-value is 0.0005, which means that we can reject the hypothesis that the two systems are identical at $\alpha = 0.05$. The leftmost distribution in Figure 1, was computed using very similar systems. The statistic value $t \approx 0$ and the p-value ≈ 0.5 Hence, H cannot be rejected.

The described procedure is a one-sided (one-tailed) test, because we reject the null, when the statistic value falls into the right tail of the statistic distribution. If a statistic distribution is symmetric (as distributions in Figure 1), we may choose to reject the null, when the statistic value falls into the left tail, i.e., to compute the p-value as the probability to observe a statistic value at least as low as $-T(x, y)$. If we use the paired t-statistic, the one-tailed test allows us to make statistical inference about directionality of the difference (i.e., which system has significantly better scores). For instance, if we observe a high positive value of the $T(x, y)$ we can reject the hypothesis that Y is better (has a higher average score) than X.

If we do not know a priori which method is better, we may choose only to test whether methods are different or not. To this end, one can employ a two-sided (two-tailed) test, where a p-value is computed as the probability of observing statistic values that are at least as high as $T(x, y)$ or at least as low as $-T(x, y)$. In this paper we focus on two-sided tests and leave evaluation of one-sided tests for future work. One approach to directional inference involves carrying out a two-sided test and comparing mean performance scores if the difference is significant. This approach is widespread, but not fully rigorous, because it offers no protection against choosing the wrong direction [26].

From a computational perspective, there is no need to evaluate the distribution of $T(x, y)$ explicitly. One can emulate this process using a counter C, initially set to zero. In each permutation step, we compute $T(\widetilde{x}, \widetilde{y})$ and verify if $|T(\widetilde{x}, \widetilde{y})| \geq |T(x, y)|$. When this condition is true, the counter C is incremented. For a one-sided test, one increments the counter when $T(\widetilde{x}, \widetilde{y}) \geq T(x, y)$. Finally, the p-value is computed as C/B, where B is the number permutations.

How many permutation steps are sufficient? The coefficient of variation (the standard error divided by the mean) of an estimated p-value is equal to $\sqrt{(1-p)/(pB)}$, where p is the actual p-value [12]. For $B = 20,000$ (the minimum number of permutations used in our tests) the coefficient of variation for $p = 0.05$ is approximately equal to 0.03. Using the Chebyshev's inequality, we get that the estimate is accurate within 15% for 96% of computed p-values (within 5 standard deviations).

Various test statistics can be employed with the permutation test. We use the paired t-statistic given by Equation (1), because it is asymptotically standard normal when differences in query-specific scores are independent or weekly dependent [28]. The sample mean difference does not have this property, because the variance of the statistic converges to zero when q grows. Our preliminary experiments showed that tests based on the sample mean difference sometimes suffer from subsantial loss of power.

2.3 Permutation Test (Multiple Systems)

2.3.1 Generalizing Permutation Algorithm

There are m retrieval systems and a baseline. We formulate m null hypotheses H_i by assuming that there is no difference between the system i and the baseline. Our focus is on testing all H_i jointly and controlling the FWER.

One may assume that the permutation algorithm of Section 2.2 can be generalized to deal with joint testing in the following fashion. Let $m + 1$ vectors x_i represent performance scores of $m + 1$ systems, where x_0 is the baseline. These vectors are combined in the matrix X (each vector is a row):

$$X = \left| \begin{array}{cccc} x_{01} & x_{02} & \ldots & x_{0q} \\ x_{11} & x_{12} & \ldots & x_{1q} \\ & \ldots & & \\ x_{m1} & x_{m2} & \ldots & x_{mq} \end{array} \right|.$$

Hypothesis H_i states that there is no difference between system i and the baseline. Each H_i is associated with a test statistic $T_i(X) = T(x_i, x_0)$, where $T(x, y)$ is a paired t-statistic (Equation 1), and the counter C_i, initially set to zero. The method involves sufficiently many permutation steps, each of which includes:

- Randomly permuting values inside columns of X and obtaining a permuted matrix $\widetilde{X}$;
- Computing m statistic values $T_i(\widetilde{X}) = T(\widetilde{x}_i, \widetilde{x}_0)$ and comparing them with statistic values obtained for the original matrix X. Formally, we increment all counters C_i such that $|T_i(\widetilde{X})| \geq |T_i(X)|$.

After B iterations, we compute the i-th p-value as C_i/B. This simple procedure runs in time $O(mB)$, but it fails to produce p-values *adjusted* for multiplicity.

One can modify this method to verify the complete null hypothesis, i.e., that *all* methods are not distinguishable from the baseline. It is used as a part of the permutation-based closed testing presented in Section 2.3.3. When the complete null does not hold, there is at least one system different from the baseline. When, we reject the complete null, we are confident that not all systems are identical, but we cannot infer which systems are actually different.

To implement this modification, we need an aggregate statistic that incorporates all m individual statistics $T_i(X)$. We use the maximum of all statistics:

$$T_{\text{aggr}}(X) = \max_{1 \leq i \leq m} |T_i(X)|.$$

Similarly to the permutation algorithm for two systems (see Section 2.2), we have one (complete) null hypothesis, one (aggregate) statistic, and a single counter C initialized with zero. We carry out B permutations steps, each of which includes random reshuffling of values inside columns of X to obtain the permuted matrix $\widetilde{X}$. If $T_{\text{aggr}}(\widetilde{X}) \geq T_{\text{aggr}}(X)$, we increment C. Finally, we reject the complete null if $C/B < \alpha$. To verify a partial hypothesis that methods i_1, i_2, ..., i_k are the same, one should apply this algorithm to a sub-matrix containing only rows i_1, i_2, ..., and i_k.

2.3.2 The MaxT Permutation Test

Westfall and Young proposed the MaxT permutation test, which is an extension of the generic procedure presented in Section 2.3.1. It uses the following m functions:

$$\text{MaxT}_i(X) = \max_{i \leq j \leq m} |T_j(X)|.$$

Let $t_i = |T_i(X)|$ be the statistic values computed for the *unmodified* matrix of observations X. Without a loss of generality we assume that t_i are *sorted in the non-increasing order*: $t_1 \geq t_2 \geq \ldots \geq t_m$. There are also m counters C_i (one for each hypothesis) initially set to zero.

We carry out B iterations. In each iteration, we obtain the permuted matrix $\widetilde{X}$ and compute m values $\text{MaxT}_i(\widetilde{X})$. If $\text{MaxT}_i(\widetilde{X}) \geq \text{MaxT}_i(X)$ we increment the counter C_i. In the end, we obtain p-values using the formula:

$$p\text{-value}_i = \max_{1 \leq j \leq i} C_j/B.$$

The MaxT permutation test runs in $O(mB)$ time and controls the FWER under the assumption of subset pivotality. Subset pivotality means that distributions of observed p-values under any partial null hypothesis should be the same as under the complete null hypothesis. Subset pivotality does not always hold in practice. Departure from this assumption may result in a low test power or inflated rate of false positives [35].

2.3.3 Joint Hypotheses and Closed Testing

Consider hypotheses H_1, H_2, H_3 and assume that we decided to reject at least one of them. In doing so, we express our belief that the respective joint hypothesis $H_1H_2H_3 = H_1 \cap H_2 \cap H_3$ is not true. This observation suggests that, before applying individual tests to H_1, H_2, and H_3, we could test the joint hypothesis (also known as an *intersection* hypothesis). If the intersection hypothesis is rejected, we may make additional tests to decide which individual hypotheses can be rejected. However, if $H_1H_2H_3$ is not rejected, we retain all the implied hypotheses and no further testing is needed.

This observation is the foundation of the *closed testing* procedure proposed by Marcus et al. [16]. In closed testing, *all joint* hypotheses are tested directly. If we fail to reject some joint hypothesis, we do not reject all implied hypotheses either. To test joint hypotheses, we use the permutation method from Section 2.3 and call this approach the permutation-based closed testing.

Assume now that H_1, H_2, and H_3 are true null hypotheses and, therefore, $H_1H_2H_3$ is also a true null. It is tested using an α-level test. Thus, the probability of not rejecting this *true* joint hypothesis is at least $1-\alpha$. According to the closed testing principle, if we do not reject the joint hypothesis $H_1H_2H_3$, we do not reject any of the implied hypotheses H_1, H_2, and H_3 either. Consequently, the probability of not rejecting any of them is at least $1-\alpha$, and the probability of rejecting at least one of them is at most α. In other words, this test controls the family-wise error rate (FWER) in the family of three hypotheses H_1, H_2, and H_3.

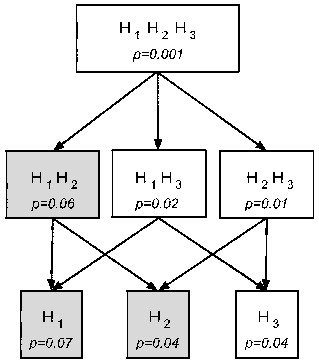


Figure 2: An example of closed testing with three hypotheses and significance level $\alpha = 0.05$. We give a p-value for each intersection hypothesis. Gray denotes hypotheses that cannot be rejected.

An example of closed testing with three hypotheses is given in Figure 2. We test 7 intersection hypotheses (including elementary hypotheses H_i as a special case) at the significance level $\alpha = 0.05$. In that, H_3 is rejected, because H_3 itself as well as all three hypotheses that include H_3, i.e., H_1H_3, H_2H_3, and $H_1H_2H_3$, have p-values smaller than α. H_1 and H_2 cannot be rejected, because we could not reject H_1H_2.

Closed testing is computationally intensive. Given that we have m individual hypotheses, there are $2^m - 1$ intersection hypotheses each of which can be tested in $O(mB)$ time. Thus, the overall runtime of the permutation-based closed testing is $O(m2^mB)$.

To reduce runtime of the complete closed testing, one could start with the narrowest intersection hypothesis (that includes all m individual hypotheses) and proceed to more generic ones. Once a hypothesis H cannot be rejected, all hypotheses implied by H need not be considered. In the example of Figure 2, we could have started with $H_1H_2H_3$ and proceeded to testing H_1H_2. Because H_1H_2 could not be rejected, H_1 and H_2 could not be rejected as well. Therefore, there is no need to test them explicitly. According to our experiments, this modification of closed-testing procedure is 10-20% faster than complete closed testing, but the overall runtime is still exponential in m.

2.4 TEST COLLECTIONS AND SYSTEMS

2.4.1 TREC Data

Similar to previous studies [27, 10, 7], we analyze data from several TREC ad hoc tasks.[1] These tasks can be divided into two groups: TRECs 3-5, 7,8, and the latest Web tasks in TRECs 19, 20. The relevance judgements in the first group are binary. They were collected through pooling to depth at least 100 [5]. The second group has 5-grade relevance judgments obtained through pooling at depth 20

[1] `http://trec.nist.gov/pubs.html`

[9]. The average number of judgements per query varies among tasks: It is roughly 2,000 in the early TRECs and is about 400 in TRECs 19, 20. The proportion of documents considered (somewhat) relevant is typically 10-20%.

The retrieval systems were represented by official runs produced by TREC participants. We downloaded raw run data and relevance judgements from the TREC website and computed the mean average precision (MAP), ERR@20 [8], and NDCG@20 using utilities `trec_eval`, and `gdeval`, which are provided by TREC organizers.

2.4.2 Learning-to-Rank Set MSLR-WEB30K.

This data set is provided by Microsoft[2]. It contains machine learning data with 5-grade relevance judgements, which were obtained from an obsolete training set used internally by Microsoft Bing.

MSLR-WEB30K has relevance judgments for 30,000 queries, which makes it possible to derive reliable conclusions about relative standings of retrieval systems. The judgements were collected in a way similar to a standard pooling. (Personal communication with Tao Qin, Microsoft Research Asia.)

The Microsoft data set is different from TREC collections in several important ways. First of all, it contains machine learning data, where a response of a retrieval system to a specific query is represented by a set of feature vectors such as BM25 scores [20] or document quality scores. Each document returned for a given query is represented by exactly one feature vector and a single label that expresses the degree of relevance between the document and the query.

It is not possible to access the original collection as well as to implement a system that uses data other than a set of precomputed features (stored in the MSLR-WEB30K file). In contrast to TREC runs (potentially representing thousands of documents per query), the average number of judged documents per query in MSLR-WEB30K is only 126.

Microsoft provides a rich set of 136 features, which are not scaled and have clear descriptions (i.e., they are not anonymized). This allows us to generate runs closely resembling runs obtained from a real retrieval system without actually implementing such a system. We use the following three methods or a combination thereof:

Method 1 computes a weighted sum of BM25 scores [20] as well as scores based on the language models [19] with two types of smoothing: Jelinek-Mercer and Dirichlet [36]. The value of this sum may be additionally multiplied by a linearly transformed quality score (feature 133).

Method 2 employs randomization to produce Oracle runs that improve over BM25 in a predictable way. We randomly select queries to be improved (with the probability p). A weight of each document returned in response to the selected queries is multiplied by $1+r(2^l-1)$, where r is a magnitude of improvement and l is the document relevance label.

Method 3 employs randomization to produce different runs, which nevertheless have almost identical performance scores. To make a randomized version of a run, we modify scores by adding a number drawn from a uniform distribution (with support 0–0.2) as proposed by Blanco and Zaragoza [4].

We evaluate methods using only ERR@10, which ignores documents at depths higher than 10.

[2] `http://research.microsoft.com/en-us/projects/mslr/`

Table 1: Fractions of Individually Significant Results Deemed *Insignificant* due to Adjustments for Multiplicity (smaller is better), $\alpha = 0.05$, ERR@20.

TREC	Closed test	MaxT	Holm Bonf
3	16.1%	16.4%	19.1%
4	12.7%	12.7%	15.4%
5	7.5%	8.7%	10%
7	15%	15.4%	17.3%
8	8.2%	8.2%	9.5%
19	31.1%	32.1%	32.1%
20	33.5%	33.5%	38.1%
All	16.4%	16.8%	18.8%

3. EXPERIMENTS

3.1 Statistical Tests Employed

Our experiments involve several statistical tests including permutation-based closed testing, the MaxT permutation test, and the Holm-Bonferroni adjustment (Equation 2) of the unadjusted permutation test p-values (see Section 2.2).

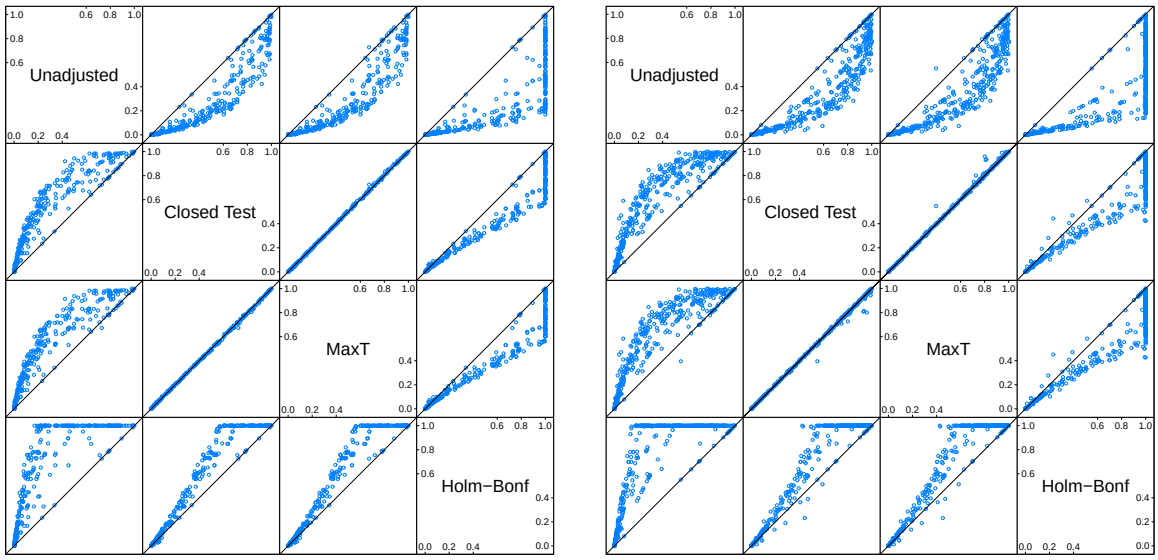
The permutation tests were implemented in C++. We use the Mersenne Twister generator of random numbers [17], which has a period of $2^{19937}-1$. In the analysis of the TREC data, the number of permutations $B = 50,000$; in the simulation studies with MSLR-WEB30K data, $B = 20,000$. Our code is available for download at `https://github.com/searchivarius/PermTest`.

3.2 TREC data

Our main goal is to assess (1) an agreement among different tests and (2) a degree of conservativeness of multiple comparisons adjustment procedures. To this end, we used TREC data to randomly choose retrieval systems and compare them against a randomly chosen baseline. Because closed testing run time is exponential in the number of compared systems, the number of systems is limited to 10. We carried out 50 iterations for several TREC data sets (see Table 1).

The agreement among these four statistical tests for TREC data is shown in Figure 3a. One can see that all tests that adjust for multiplicity produce larger p-values than the unadjusted permutation test. When we compare only among tests that adjust p-values for multiple comparisons, we can see that there is very little difference in p-values smaller than 0.1. The application of the Holm-Bonferroni adjustment does result in much larger p-values, but only for p-values that are already large (> 0.1). These two observations are consistent with findings by Dudoit et al. [11]. Also note that the outcomes from the permutation-based closed testing and the permutation MaxT tests are almost identical.

There is no ground truth information about relative performance of systems in TREC. This is why we can compare the power of tests only approximately, by examining the number of significant results. According to Table 1, multiple comparisons adjustments "kill" from 8 to 38 percent of results that were deemed significant by the unadjusted permutation test. In that, there is very little difference among tests. Closed testing is being slightly better than MaxT, and MaxT is slightly better than the Holm-Bonferroni adjustment. These is observed in all TRECs, but the difference



(a) TREC data, 10 runs in a comparison

(b) MSLR "Language models" data, 8 runs in a comparison

Figure 3: Agreement of p-values among adjustment methods. A performance metric is ERR@10.

is too small to be of practical importance. Note that the fraction of results that became insignificant due to multiple comparisons adjustments vary greatly among TRECs. Only about 10% of all results became insignificant in TREC-5, but in TREC-20 we lose almost half of the results, if multiple comparisons adjustments are used.

3.3 MSLR-WEB30K data

For MSLR-WEB30K we carried out a simulation study, in which we generated runs with 30,000 queries (one run for each retrieval system). These runs are "populations" that represent *long-term* performance of retrieval systems. Systems' responses to a smaller set of q queries were generated through *repeated sampling* from 30,000 queries. As a result, we obtained simulated runs representing performance of each system for selected q queries (as measured by ERR@10). In addition, mean value for each metric was computed. We evaluated several scenarios where q varied from 50 to 6,400. The sample size of 50 is intended to represent a typical TREC experimental setting, while much larger sets of queries mimic experimental environment accessible by a commercial search engine developer/researcher.

Even though the sample mean of a performance metric may vary considerably in each simulation step, the average values of ERR@10 *converge* to the mean population values (of 30,000 queries) as the number of simulation steps increases. Thus, we can use population values of performance metrics to establish ground truth relative standings among systems. To this end, we select a cutoff value $\gamma = 0.5\%$ and consider all pairs of systems with percent differences in ERR@10 (computed for the whole population) smaller than γ as identical. By repeating our analysis for $\gamma \in \{0.05, 0.1, 0.5, 1\}$, we confirmed that conclusions did not depend on the choice of this cutoff value.

The cutoff-based approach reflects a point of view that small differences in system performance may be due to sampling uncertainty of our 30,000 queries from a much larger super population. In fact, some statisticians question

> ... whether it makes sense to even consider the possibility that the means under two different experimental conditions are equal. Some writers contend that a priori no difference is ever zero (for a recent defense of this position, see Tukey 1991, 1993). Others, including this author, believe that it is not necessary to assume that every variation in conditions must have an effect [26].

The latter point is supported by studies showing that small differences may not affect user experience [30].

We generated three sets of populations: "Language Models", "Oracle 0.25", and "Oracle 0.5" (see Section 2.4). Language models were generated using Method 1. Oracle runs were generated by Method 2 with probabilities of improvement 0.25 and 0.5. The best runs in the "Language Models", "Oracle 0.25", and "Oracle 0.5" improved over the BM25 by 17%, 14%, and 27%, respectively. In each of the three population sets, we took a system with performance close to the median of the set, and replicated it 4 times using randomization (Method 3). This system is considered to be a *baseline*. Overall, each population set had 8 runs, half of which were almost identical (in terms of the mean value of ERR@10).

Given a specific sample size and a set of runs, we carried out 500 iterations of the resampling process and tested if the differences between simulated runs were significant (using selected statistical tests). Because we knew the ground truth relative standings of retrieval systems, it was possible to determine the number of false positives and negatives.

The agreement among tests is plotted in Figure 3b. The plot includes the "Language Model" runs only (all query set sizes from 50 to 6,400), because results obtained for Oracle runs are almost identical. One can see that agreement graphs are also similar to those for the TREC data:

(1) multiple comparisons adjustment procedures produce larger p-values than the unadjusted permutation test, (2) permutation-based closed testing and the MaxT permutation test agree almost ideally for the whole range of p-values, (3) the Holm-Bonferroni adjustment applied to p-values of the unadjusted permutation test is more conservative than MaxT and closed testing. Similar to TREC experiments, there is virtually no difference among all multiple comparisons methods for small p-values (< 0.1). Observation (2) is important, because we can use the MaxT permutation test instead of considerably less efficient closed testing (whose run time is exponential in the number of systems m).

Table 2: The Percent of False Negatives/Positives for different query set sizes ($\alpha = 0.05$)

	Query Set Size				
	50	100	400	1600	6400
"Lang. Models": 4 out of 8 runs same as the baseline					
Unadjusted	85.7/14.4	80.8/11.6	53.9/10.0	25.9/15.4	2.5/17.0
Closed Test	92.9/0.0	88.8/0.2	69.5/1.7	36.6/3.1	5.2/6.8
MaxT	93.9/0.0	91.8/0.2	68.0/1.2	35.7/3.0	6.3/6.6
Holm-Bonf.	94.9/2.0	92.5/1.8	69.6/2.6	37.0/3.2	6.5/6.2
"Oracle 0.25": 4 out of 8 runs same as the baseline					
Unadjusted	91.6/12.9	86.0/14.1	56.9/13.9	22.9/14.5	0.3/9.3
Closed Test	98.9/1.8	97.8/1.1	73.8/2.1	35.3/2.8	1.1/3.2
MaxT	97.3/2.0	96.4/3.0	74.4/3.0	36.1/5.5	1.0/4.6
Holm-Bonf.	98.2/2.4	97.0/3.4	74.9/2.6	37.4/4.8	1.9/4.2
"Oracle 0.5": 3 out of 8 runs same as the baseline					
Unadjusted	87.2/8.1	76.0/8.5	49.0/9.5	22.0/8.9	18.6/6.9
Closed Test	98.2/1.1	93.8/0.4	62.5/2.5	26.0/2.1	19.6/2.1
MaxT	96.9/1.2	93.3/1.6	61.4/2.6	26.5/3.2	19.4/2.8
Holm-Bonf.	97.7/1.0	91.5/2.2	62.9/2.0	27.3/2.8	19.5/2.0

Format: false negative rate (blue)/false positive rate (red).

Using ground truth relative standings for system performance, we computed the rates of false positives and false negatives for different query set sizes. In Table 2, we present results of this evaluation. Surprisingly, there is very little difference in the rate of false negatives (approximately within 10%) between the unadjusted permutation test and any test that takes multiplicity into account. However, when the number of queries is small (*as in TREC*) and the number of false negatives is close to 100%, the number of detected differences in system performance may vary greatly. For instance, in the case of 50 queries and "Language Models" runs, the unadjusted permutation test detects 14.3% of all true differences (85.7% false negative rate), while the MaxT permutation test detects only 6.1% (93.9% false negative rate). Detection of these additional 8.2% true differences comes at a price of at least one false finding in 14.4% of all experimental series. In contrast, the number of false positives for the MaxT test is zero in this case.

If the researcher does not know the true number of different systems, he may conclude that the MaxT test performs much worse than the unadjusted permutation test from the perspective of detection of true differences. Yet, in our opinion, both tests perform rather poorly in this situation. When there is a sufficient number of queries, all the tests detect more than 80-90% of true differences. In that, only the tests that adjust for multiplicity have the false positive rate close to the nominal level of $\alpha = 0.05$, i.e., they perform better than the unadjusted test, without being overly conservative.

Consider a somwehat extreme example where out of 100 systems 90 are equivalent to the baseline. For $\alpha = 0.05$, unadjusted tests may find 4-5 statistically significant differences, which represent false positives. It is possible that for small sets of queries no true difference will be detected, if false negatives rates are as high as those listed in the first column of Table 2.

3.4 Discussion

Our results indicate that multiple comparisons adjustments can be conservative when the number of queries is small. Yet, as the number of queries increases, the FWER approaches the nominal level α. When the number of queries is large, both types of tests (with and without multiplicity adjustment) detect similar number of true differences, but only adjustments for multiple comparisons allow us to control the number of false positives.

This conclusion may be affected by a small scale of our experiments (a joint test involves at most 10 systems). Yet, a small-scale experiment is not unusual for studies with both an exploratory and a confirmatory step. In the exploratory step, the researcher may "play" with a large number of systems and choose various heuristics to assess systems' performance. Multiple comparisons adjustments are typically not used in this step. The outcome is a small number of systems to be tested rigorously. During the confirmatory step, the researcher formulates the null hypotheses and carries out a statistical test using *previously unseen* data. We argue that in this step multiple comparisons adjustments are essential.

We found that the Holm-Bonferroni adjustment was only slightly more conservative than the MaxT permutation test and/or the permutation-based closed testing, which was true for both the TREC and the MSLR experiments. This is surprising, because performance scores across systems are correlated. In the presence of correlations, the MaxT permutation test and the permutation-based closed testing are expected to be more powerful than the Holm-Bonferroni adjustment.

However, permuting the data, subtracting the baseline row, and computing the t-statistic is equivalent to first subtracting the baseline row, then permuting the differences, and computing the t-statistic. Thus, it is the correlations among the deviations from the baseline that matter. We found that these correlations are small. For instance, for the TREC-8 data and ERR@20, the correlation is almost zero on average. This explains similar relative performance of the Holm-Bonferroni adjustment and the other two procedures. Yet, this may not generally hold.

We carry out an artificial experiment in which we took two vectors of performance scores such that there was a significant statistical difference between them with a p-value equal to β. Then, we replicated one of the vector several times, which is equivalent to having a number of identical systems evaluated against the baseline. The p-value computed using either the MaxT permutation test or the permutation-based closed testing procedure was approximately β in all experiments. The Holm-Bonferroni correction produced a p-value of $m\beta$, where m is the number of times the system was replicated. Thus, using the MaxT permutation test or the permutation-based closed testing can be advantageous.

While the run-time of the permutation-based closed testing procedure is exponential in the number of systems being evaluated, the run-time of the MaxT permutation test is reasonably short. For example, it takes 6 minutes to carry out 100K iterations of the MaxT permutation test to assess the joint statistical significance of 8 system runs represented by performance scores for as many as 30K queries.[3]

One may find our use of machine learning data objectionable, because it requires assumptions regarding what can be considered a retrieval system. Note, however, that the learning-to-rank community already made these assumptions and models the behavior of retrieval systems in the same fashion as we constructed "Language Model" runs. The only difference is that we designed a (semi)-linear ranking function with coefficients tuned by hand. They, instead, replace this step with a machine learning algorithm. They also evaluate performance of constructed runs using ERR@10 and employ statistical tests. Thus, it is important to show that the statistical tests work well in the learning-to-rank setting. Also note that all our tests exhibit similar behavior for both the TREC and MSLR data, which supports the hypothesis that MSLR runs are similar to those produced by real retrieval systems.

Even though permutation tests do not make strong distributional assumptions such as the normality or i.i.d, they are not assumption free. Exchangeability means that we test the equality of distributions instead of sample means. This may appear problematic, because sometimes the test may reject the null due to, e.g., a difference in variances. In particular, the simulation studies of Huang et al. [14] showed that inequality of distributions sometimes results in inflated rates of false positives. Yet, as noted by Efron and Tibshirani [12], permutation tests typically perform well in practice, even if the equality of distributions is not a reasonable assumption. They also suggest that the permutation test should be applied in all circumstances when there is "something to permute", even if other methods such, as the bootstrap test, are applicable as well. In addition, the equality of distributions is an underlying assumption for a number of statistical tests, such as the Student's t-test, already used by the IR community.

4. CONCLUSIONS

We carried out a comparative assessment of non-parametric testing procedures appropriate in the presence of multiplicity. To the best of our knowledge, such comparisons have not been done previously in the IR setting. We use only publicly available test collections and make our software available for download.

The experiments employ the realistic TREC runs and runs constructed from the Microsoft learning-to-rank dataset. The latter is a novel approach, which allows us to (1) obtain ground truth relative standings among systems, (2) experiment with much larger sets of queries and relevance assessments compared to the TREC setting.

Our recommendation is to employ adjustments for multiple comparisons in confirmatory experiments. When the number of queries is small, these procedures may, indeed, detect many fewer significant results than standard procedures such as the Student's t-test. However, the advantage of the tests without adjustments may be illusory. In this case, both the unadjusted tests and tests that adjust for multiplicity detect only a small fraction of all true differences. In that, results obtained using unadjusted tests may contain a lot of false positives, possibly, more than significant results. When there is a large query set, both types of tests may have enough power to detect true differences among systems. Yet, only the procedures adjusting for multiplicity control the rate of false positives.

The permutation-based closed testing relies on fewer assumptions than the MaxT permutation test, yet, it is impractical for all but very small sets of runs. Our recommendation is to use the MaxT permutation test, which seems to produce very similar results while being reasonably fast. In our experiments, the Holm-Bonferroni adjustments performed as well as the other adjustment methods. Yet, this may be due to specifics of our simulations, where there are small correlations among deviations from the baseline. As the example in Section 3.4 shows, permutation methods can be much more powerful when strong correlations are present.

[3]The CPU is Intel Core i7 (3.4 GHz).

5. ACKNOWLEDGMENTS

We thank Tao Qin (Microsoft Research Asia) for information about the MSLR collection. Leonid Boytsov was partially supported by a SIGIR Student Travel Grant. Dr. Westfall was partially supported by the following grants: NIH RO1 DK089167. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the funding agencies.

6. REFERENCES

[1] Anonymous. Guidance for Industry - E9 Statistical Principles for Clinical Trials. Technical report, U.S. Department of Health and Human Services - Food and Drug Administration, Center for Drug Evaluation and Research, Center for Biologics Evaluation and Research, ICH, 1998.

[2] R. Bender and S. Lange. Adjusting for multiple testing—when and how? *Journal of Clinical Epidemiology*, 54(4):343 – 349, 2001.

[3] Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[4] R. Blanco and H. Zaragoza. Beware of relatively large but meaningless improvements. Technical report YL-2011-001, Yahoo! Research, 2011.

[5] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling for large collections. *Information Retrieval*, 10:491–508, 2007.

[6] R. J. Cabin and R. J. Mitchell. To Bonferroni or not to Bonferroni: when and how are the questions. *Bulletin of the Ecological Society of America*, 81(3):246–248, 2000.

[7] B. A. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans. Inf. Syst.*, 30(1):4:1–4:34, Mar. 2012.

[8] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceeding of the 18th ACM conference on*

Information and knowledge management, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM.

[9] C. L. A. Clarke, N. Craswel, I. Soboroff, and G. V. Cormack. Overview of the TREC 2010 Web track. In *TREC-19: Proceedings of the Nineteenth Text REtrieval Conference*, 2010.

[10] G. V. Cormack and T. R. Lynam. Validity and power of t-test for comparing map and gmap. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 753–754, New York, NY, USA, 2007. ACM.

[11] S. Dudoit, J. Schaffer, and J. Boldrick. Multiple hypothesis testing in microarray experiments. *Statistical Science*, 18(1):71–103, 2003.

[12] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1993.

[13] S. Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[14] Y. Huang, H. Xu, V. Calian, and J. C. Hsu. To permute or not to permute. *Bioinformatics*, 22(18):2244–2248, 2006.

[15] E. L. Lehmann and J. P. Romano. Generalizations of the familywise error rate. *Annals of Statistics*, 33(3):1138–1154, 2005.

[16] R. Marcus, P. Eric, and K. R. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660, 1976.

[17] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, Jan. 1998.

[18] E. Pitman. Significance tests which may be applied to samples from any population. *Royal Statistical Society, Supplement*, 4:119–130, 1937.

[19] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[20] S. Robertson. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60:503–520, 2004.

[21] Y. Saeys, I. n. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, Oct 2007.

[22] M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 162–169, New York, NY, USA, 2005. ACM.

[23] J. Savoy. Statistical inference in retrieval effectiveness evaluation. *Information Processing & Management*, 33(4):495 – 512, 1997.

[24] H. Scheffé. A method for judging all contrasts in the analysis of variance. *Biometrika*, 40(1-2):87–110, 1953.

[25] F. Scholer, A. Turpin, and M. Sanderson. Quantifying test collection quality based on the consistency of relevance judgements. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1063–1072, New York, NY, USA, 2011. ACM.

[26] J. P. Shaffer. Multiple hypothesis testing. *Annual Review of Psychology*, 46(1):561–584, 1995.

[27] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 623–632, New York, NY, USA, 2007. ACM.

[28] J. Sunklodas. Approximation of distributions of sums of weakly dependent random variables by the normal distribution. In Y. Prokhorov and V. Statulevičius, editors, *Limit Theorems of Probability Theory*, pages 113–165. Springer Berlin Heidelberg, 2000.

[29] J. Tague-Sutcliffe and J. Blustein. A statistical analysis of TREC-3 data. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 385–398, 1994.

[30] J. Urbano, J. S. Downie, B. Mcfee, and M. Schedl. How significant is statistically significant? the case of audio music similarity and retrieval. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, pages 181–186, Porto, Portugal, October 8-12 2012.

[31] W. Webber, A. Moffat, and J. Zobel. Statistical power in retrieval experimentation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 571–580, New York, NY, USA, 2008. ACM.

[32] P. H. Westfall and J. F. Troendle. Multiple testing with minimal assumptions. *Biometrical Journal*, 50(5):745–755, 2008.

[33] P. H. Westfall and S. S. Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley-Interscience, 1 edition, Jan. 1993.

[34] W. J. Wilbur. Non-parametric significance tests of retrieval performance comparisons. *J. Inf. Sci.*, 20:270–284, April 1994.

[35] H. Xu and J. C. Hsu. Applying the generalized partitioning principle to control the generalized familywise error rate. *Biometrical Journal*, 49(1):52–67, 2007.

[36] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

[37] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar. Streamwise feature selection. *Journal of Machine Learning Research*, 7:1861–1885, 2006.

[38] J. Zobel, W. Webber, M. Sanderson, and A. Moffat. Principles for robust evaluation infrastructure. In *Proceedings of the 2011 workshop on Data infrastructures for supporting information retrieval evaluation*, DESIRE '11, pages 3–6, New York, NY, USA, 2011. ACM.