IMS @ TREC 2017 Core Track

Nicola Ferro

Department of Information Engineering, University of Padua
Via Gradenigo 6/B, 35131, Padova, Italy
ferro@dei.unipd.it

Abstract

We report our participation to the TREC 2017 Core Track. Our objective is to systematically investigate the use of weak baselines, namely off-the-shelf open source IR systems, and understand how they compare with respect to TREC state-of-the-art ones. We are also interested in understanding how IR components – namely stop lists, stemmers/n-grams, and IR models – contribute to the overall performances and, specifically, to the pools.

1. Introduction

The goal of the TREC 2017 Common Core Track[1] is to explore new ways to test collection construction with a specific focus on relevance assessment creation and how to go beyond standard depth-k pooling, both in a traditional NIST assessors-based setting and in a crowd-sourcing context.

The TREC 2017 Common Core Track is a standard ad-hoc task, using a set of 249 topics taken from the TREC 2004 Robust Track [40] against a newswire corpus by the New York Times[2], consisting of 1.8 million articles from 1987 to 2007; 50 of these topics have been assessed by NIST and are used in the present report, while the remaining 199 have been assessed by using crowd-sourcing. The TREC 2017 Common Core Track solicited the participation of as diverse as possible *Information Retrieval (IR)* systems, focused on retrieving as much unique relevant documents as possible, in order to support the exploration of new pooling techniques. For more details on the TREC 2017 Common Core Track, please see the overview of the track [1].

The objective of our participation is to investigate how weak baselines, namely off-the-shelf open-source IR systems, perform in this task and how they compare to state-of-the-art and optimized TREC systems participating in the track. Indeed, it has been repeatedly shown that best historical TREC systems still outperforms off-the-shelf open source systems [4, 5, 21, 26] but these comparisons happened somehow post-hoc in the sense that the analysed open source systems typically did not participate in the examined TREC tracks and their configurations have not been systematically explored and exploited to see whether they can get closer to best TREC systems.

Therefore, we used Apache Lucene[3] 6.6.0 to create a *Grid of Points (GoP)* [13, 14] made up of hundreds of runs resulting from all the possible combinations of a set of selected stop lists, stemmers, and IR models. We then exploited data fusion techniques [10, 41] to merge these runs and get out the maximum from them.

In our previous work, we conducted a study on how to break-down the overall IR system into the contributions of the individual components [14–16], using GoPs created with Terrier[4] on several past TREC and CLEF ad-hoc collections. An additional objective of our participation is to create a GoP using Lucene instead of Terrier, based on a test collection where the GoP systems actually participated, and see whether there are differences with respect to our previous findings.

[1] `https://trec-core.github.io/2017/`
[2] `https://catalog.ldc.upenn.edu/ldc2008t19/`
[3] `http://lucene.apache.org/`
[4] `http://terrier.org/`

Finally, we take the opportunity to start investigating how IR system components contribute to pool and how to break-down their contributions.

The paper is organized as follows: Section 2 provides some background information on the statistical tools used for conducting the analyses; Section 3 introduces the adopted approach; Section 4 describes the implemented software library, which is available for further re-use; Section 5 reports the analyses on the experimental outcomes; finally, Section 6 draws conclusions and wraps up the discussion on the main findings of the study.

2. Background on GLMM and ANOVA

A *General Linear Mixed Model (GLMM)* [36] explains the variation of a dependent variable ("Data") in terms of a controlled variation of independent variables ("Model") in addition to a residual uncontrolled variation ("Error"): Data = Model + Error.

ANalysis Of VAriance (ANOVA) expresses a GLMM, for example, as $Y_{ij} = \mu + \alpha_j + \varepsilon_{ij}$, where Y_{ij} is the i-th subject's dependent variable score in the j-th experimental condition, the parameter μ is the grand mean of the experimental condition population means that underlies all subjects' dependent variable scores, the parameter α_j is the effect of the j-th experimental condition and the random variable ε_{ij} is the error term, which reflects variation due to any uncontrolled source.

For a given model, the ANOVA table summarizes the outcomes of the ANOVA test indicating, for each factor, the *Sum of Squares (SS)*, the *Degrees of Freedom (DF)*, the *Mean Squares (MS)*, the F statistics, and the p-value of that factor, which allows us to determine the significance of that factor. In the following, we consider a confidence level $\alpha = 0.05$ to determine if a factor is statistically significant.

We are not only interested in determining whether a factor effect is significant, but also which proportion of the variance is due to it, that is we need to estimate its *effect-size measure* or *Strength of Association (SOA)*. The SOA is a "standardized index and estimates a parameter that is independent of sample size and quantifies the magnitude of the difference between populations or the relationship between explanatory and response variables" [32, 37].

$$\hat{\omega}^2_{\langle fact \rangle} = \frac{df_{fact}(F_{fact} - 1)}{df_{fact}(F_{fact} - 1) + N} \tag{1}$$

is an unbiased estimator of the variance components associated with the sources of variation in the design, where F_{fact} is the F-statistics and df_{fact} are the degrees of freedom for the factor while N is the total number of samples.

The common rule of thumb [36] when classifying $\hat{\omega}^2_{\langle fact \rangle}$ effect size is: 0.14 and above is a *large size effect*, 0.06–0.14 is a *medium size effect*, and 0.01–0.06 is a *small size effect*. $\hat{\omega}^2_{\langle fact \rangle}$ values could happen to be negative and in such cases they are considered as zero.

A *Type I* error occurs when a true null hypothesis is rejected and the significance level α is the probability of committing a Type I error. When performing multiple comparisons, the probability of committing a Type I error increases with the number of comparisons and we keep it controlled by applying the Tukey *Honestly Significant Difference (HSD)* test [19] with a significance level $\alpha = 0.05$. Tukey's method is used in ANOVA to create confidence intervals for all pairwise differences between factor levels, while controlling the family error rate; it is an effective method generally more powerful than other popular statistical methods like the Bonferroni one [28]. Two levels u and v of a factor are considered significantly different when

$$|t| = \frac{|\hat{\mu}_u - \hat{\mu}_v|}{\sqrt{MS_{error}\left(\frac{1}{n_u} + \frac{1}{n_v}\right)}} > \frac{1}{\sqrt{2}} q_{\alpha,k,N-k} \tag{2}$$

where $\hat{\mu}_u$ and $\hat{\mu}_v$ are the marginal means, i.e. the main effects, of the two factors; n_u and n_v are the sizes of levels u and v; $q_{\alpha,k,N-k}$ is the upper $100 * (1 - \alpha)$th percentile of the studentized range distribution with parameter k and $N - k$ degrees of freedom; k is the number of levels in the factor and N is the total number of observations.

3. Approach

3.1. Grid of Points

We consider three types of components of an IR system: stop list, stemmer and n-grams, and IR model. We select a set of alternative implementations of each component and, by using the Lucene open source system, we create a run for each system defined by combining the available components in all the possible ways. The components we experiment are:

- *Stop list* (6 components): `nostop`, `indri`, `lucene`, `smart`, `snowball`, `terrier`;
- *Stemmer* (5 components): `nolug`, `krovetz`, `lovins`, `porter`, `5grams`;
- *Model* (11 components): `bm25`, `dfichi`, `dfiis`, `dfrinb2`, `dfrinexpb2`, `dfrinl2`, `iblgd`, `ibspl`, `lmd`, `lmjm`, `lucene`.

Stop lists differ in the number of composing terms: `lucene` has 33 terms, `snowball` has 174 terms, `indri` has 418 terms, `smart` has 571 terms, and `terrier` 733 terms.

Stemmers can be classified into aggressive and weak stemmers. `lovins` [27] is the most aggressive stemmer; `porter` [33] si weaker than `lovins`; `krovetz` [23] is as aggressive as `porter` and weaker than `lovins`. When it comes to n-grams, we use $n = 5$, which is one of the best performing lengths according to previous findings [16, 29].

The models we employ are classified into the three main approaches currently adopted by search engines [35]: the vector space model [38] (`lucene`), the probabilistic model – comprehending the `bm25` model [34], *Divergence From Randomness (DFR)* models [2] (`dfrinexpb2` and `dfrinl2`), *Divergence From Independence (DFI)* models [22] (`dfichi` and `dfiis`), and information-based models [9] (`iblgd` and `ibspl`) – and *Language Models (LMs)* [42] (`lmd` and `lmjm`). For all the models, we considered their off-the-shelf implementation with default parameters.

More in detail, `lucene` is the classic similarity of Lucene[5]; `dfichi` is DFI model with normalized chi-squared measure of distance from independence while `dfiis` is a DFI model with saturated measure of distance from independence; `dfrinb2` is a DFR model with Inverse Document Frequency model with Bernoulli after-effect and normalisation 2, `dfrinexpb2` is a DFR model with Inverse Expected Document Frequency model with Bernoulli after-effect and normalisation 2, and `dfrinl2` is a DFR model with Inverse Document Frequency model with Laplace's law of succession after-effect and normalisation 2; `iblgd` is an information-based model with log-logistic distribution, lambda as average number of documents where w occurs and normalization 2 while `ibspl` is an information-based model with smoothed power-law (SPL) distribution, lambda as average number of documents where w occurs, and normalization 2; finally, `lmd` and `lmjm` are LMs with, respectively, Dirichlet and Jelinek-Mercer smoothing.

Overall, we create GoPs consisting of $6 \times 5 \times 11 = 330$ system runs. They represent nearly all the possible off-the-shelf configurations of Lucene made up with core components and thus provide quite an accurate view on what is actually possible to achieve.

3.2. Data Fusion

We adopted basic data fusion techniques for two main reasons. Firstly, they typically improve over the performance of the merged runs and thus this represents a further chance for off-the-shelf systems to get closer to best TREC systems. Secondly, the Core Track guidelines limited submissions to 3 official runs for each participant plus possibly 7 additional runs; clearly, the GoP above consisting of 330 systems does not fit into these limits. Therefore, data fusion represents a way to "summarize" the runs in the GoP in order to give their documents a chance for being pooled anyway.

We took two approaches to data fusion, one unsupervised and the other supervised.

As unsupervised approach, we merged the above GoP runs with the CombSum algorithm [18] using the Min-Max normalization [24] (run ID `ims_cmbsum`).

[5] `https://lucene.apache.org/core/6_6_0/core/org/apache/lucene/search/similarities/ClassicSimilarity.html`

As supervised approach, we merged the above GoP runs with the Weighted CombSum algorithm [3] using the Min-Max normalization. We trained the weighting for each pair of topics and runs by creating another GoP with the same set of systems on the TREC 2004 Robust track collection [40], which uses the same topics as the Core Track but a different set of documents.

Besides using *Average Precision (AP)* as weighting measure (run ID `ims_wcmbsum_ap`), as done in [3], we also experimented with Precision at 10 (run ID `ims_wcs_p10`), Recall (run ID `ims_wcs_recall`), R-prec (run ID `ims_wcs_rprec`),*Rank-Biased Precision (RBP)* [30] (run ID `ims_wcs_rbp`), *Normalized Discounted Cumulated Gain (nDCG)* [20] (run ID `ims_wcs_ndcg`), *Expected Reciprocal Rank (ERR)* [8] (run ID `ims_wcs_err`), and Twist [17] (run ID `ims_wcs_twist`).

Finally, since one of the goals of the Core Track was to retrieve as much unique relevant documents as possible, we also tried a very simplistic approach merging different runs but pushing on top the unique documents in this set of runs (run ID `ims_wcs_ap_uf`). In particular, we adopted a two layers merging approach to try to mitigate the noise at least a bit. Instead of directly merging the GoP runs, in the lower layer we used the native Lucene merger[6] which implements the CombSum algorithm without any score normalization. We used it to merge the 11 IR models above for each combination of stop list and stemmer but without 5-grams to further reduce noise. This originated a set of $6 \times 4 = 24$ runs which then we merged using AP Weighted CombSum with Min-Max normalization but putting on top the unique documents in this set of 24 runs.

4. Implementation

We developed a library which acts a generic tool kit for creating GoPs using Lucene and then applying different data fusion techniques. The library is open source and available at `https://bitbucket.org/frrncl/trec-core-2017` to ease the reproducibility of the experiments [11, 12].

The main challenges in developing this library were to have a simple and declarative way to define which components have to be combined to create a GoP and, especially, scalability and efficiency issues to allow the creation of hundreds of indexes and runs as well as merging hundreds of runs.

The simple declarative way to specify GoP components is achieved with a parametric Java properties file where it is possible to specify lists of components, the Java classes which implement them, and input parameters to them, such as the files containing the stop words.

When it comes to efficiency, the library allows us to create the indexes and then the runs in a multi-threaded way by using thread pools with separate sets of workers for indexing and searching. The number of workers and the memory allocated to each worker can be configured in the Java properties file mentioned above.

Another scalability issue is related to the limits of the Java virtual machine and the garbage collector which are put under pressure during the merging phase. Indeed, we needed to merge 330 runs of 250 topics each, retrieving 10,000 documents for each topic, amounting to 825,000,000 document identifiers to be managed as Java String objects. In order to avoid the creation of so many Java objects as well as to reduce the requested memory footprint, we implemented a caching mechanism for Java String objects in order to return the same Java object when the same document appears in more runs and topics, instead of creating a new String object for each these replicates.

The library is organized into the following packages:

- `it.unipd.dei.ims.treccore.analysis`: extends Lucene basic text processing components by adding a generic analyzer for creating GoPs, n-grams, and the Lovins stemmer, wrapping the implementation[7] of Eibe Frank at University of Waikato, New Zeland.
- `it.unipd.dei.ims.treccore.index`: provides a generic mechanism for developing parsers for experimental collections and gives two concrete instances, a very basic one for TIPSTER, derived from sample code[8] by Ian Soboroff at NIST, USA, and one for the New York Times. These parsers re-

[6] `https://lucene.apache.org/core/6_6_0/core/org/apache/lucene/search/similarities/MultiSimilarity.html`
[7] `http://www.cs.waikato.ac.nz/~eibe/stemmers/`
[8] `https://github.com/isoboroff/trec-demo/blob/master/src/TrecDocIterator.java`

cursively index a directory tree containing the collection, processing a document at time. However, due to need of producing hundreds of indexes and to avoid opening and closing millions of files, we also developed pre-processing parsers which scan the whole directory tree, parse the documents, and produce a single text file where each line is the pre-processed content of a document; in this way, the subsequent indexers have to open just one single file for the whole collection and, apart from operating system resources, this also saves a bit of time in terms of pre-processing.

- `it.unipd.dei.ims.treccore.similarities`: extends the Lucene similarities, i.e. the IR models, in order to provide concrete instances for the DFR, DFI, LM and information-based models described in Section 3 as well as instantiating the native Lucene merger with those IR models.
- `it.unipd.dei.ims.treccore.search`: extends Lucene searchers to allow its easy configuration in order to create a GoP and working in a pool of threads.
- `it.unipd.dei.ims.treccore.gop`: manages the multi-threading approach and the pools of workers needed for indexing and searching a GoP and it is the entry point for managing the GoP creation process.
- `it.unipd.dei.ims.treccore.merge`: provides a generic infrastructure for developing data fusion algorithms, both supervised and unsupervised, in an efficient way and contains several instantiations of them; besides those used in the official runs described in Section 3, it also implements CombSum with ZMUV normalization [31] and MapFuse [25].
- `it.unipd.dei.ims.treccore.uniquerel`: provides helper classes for counting unique relevant documents and conducting the analyses reported in this paper.

Finally, the above repository contains also a `matlab` folder containing all the Matlab[9] code required for conducting the the analyses reported in this paper. The Matlab code is based on the *MATlab Toolkit for Evaluation of information Retrieval Systems (MATTERS)* library available at `http://matters.dei.unipd.it/`.

5. Analysis

First, in section 5.1, we conduct a preliminary analysis of the 330 GoP runs, which constitute the basis for the official runs submitted to the Common Core track. Then, section 5.2 analyses the official runs.

Section 5.3 uses the GoP runs to break-down the overall system performance into the contribution of stop lists, stemmers, and IR models.

Finally, Section 5.4 uses the GoP runs to explore how the IR system components contributed to pools.

We adopt the official measures of the Common Core track, namely *Average Precision (AP)*, P10, and *Normalized Discounted Cumulated Gain (nDCG)*; note that AP and nDCG are computed on the full ranked result list, i.e. on the 10,000 returned documents. We also consider Recall, as additional measure, again computed on the full ranked result list.

We use $\alpha = 0.05$ as significance level.

5.1. GoP Runs

Table 1 reports the average performance of the GoP runs over the 50 topics assessed by NIST, showing the minimum, maximum, mean and median performance while Figure 1 shows the boxplot of the performances scores across the topics for each run and measures to give an idea of the distribution of the scores.

We can see how the GoP runs have reasonable performance in terms of AP, P10, and nDCG, as you can expect from weak baselines, and quite good performance in terms of recall; this is also due to the very long

[9] `https://www.mathworks.com/`

	Average across topics			
	AP	**P10**	**nDCG**	**Recall**
Minimum	0.0998	0.2600	0.2422	0.5399
Mean	0.2114	0.4911	0.4383	0.7384
Median	0.2261	0.5280	0.4586	0.7496
Max	0.2634	0.6020	0.5059	0.8042

Table 1: Performance of the GoP runs over the 50 topics assessed by NIST.

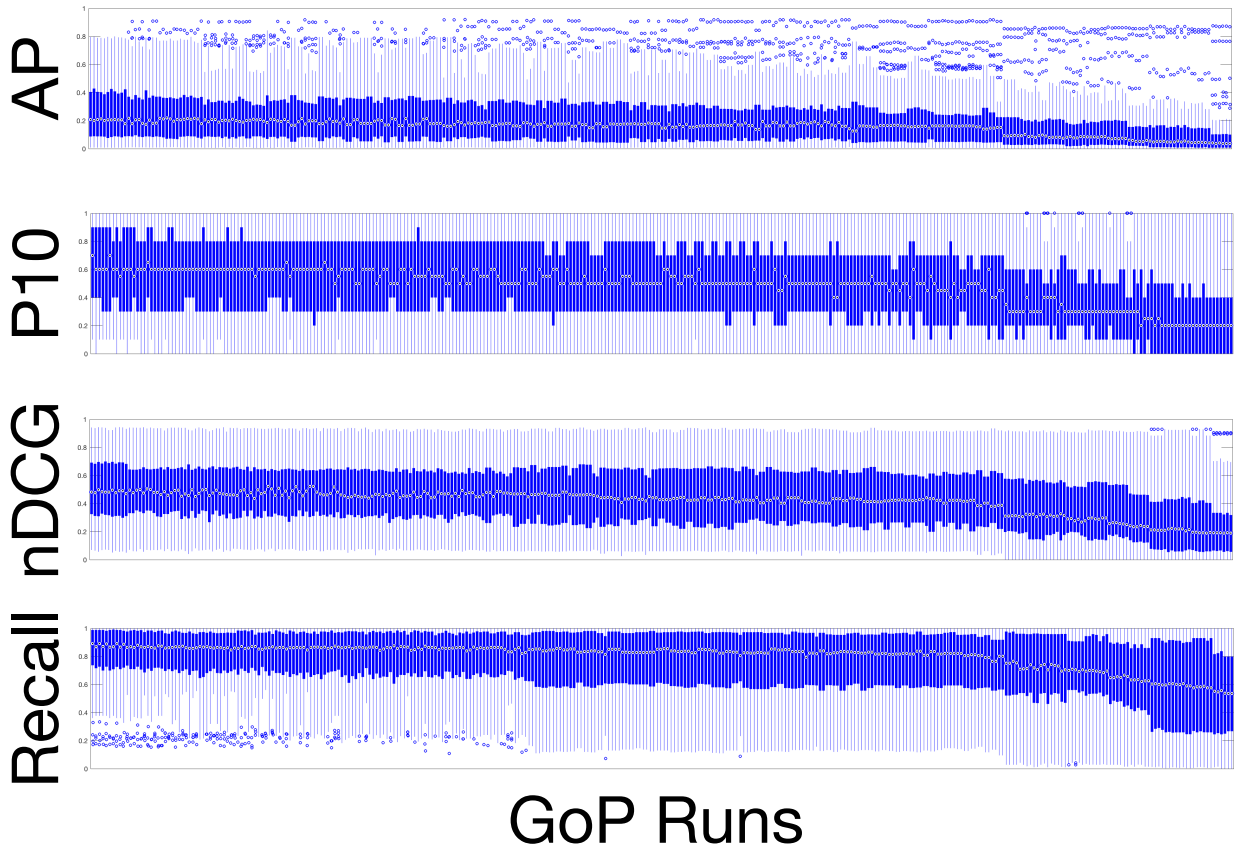

Figure 1: Box plot of the performance distributions of the GoP runs for the different evaluation measures.

runs (10,000 retrieved documents) that give the opportunity to retrieve relevant documents also at very low rank positions.

Overall, we can note how GoP runs are weak baselines mainly because how they rank the relevant documents rather than not retrieving enough of them. We can also observe that we have the highest variability with AP which has many more outliers than the other evaluation measures.

In order to assess the significant differences among the GoP runs, we use the GLMM shown in equation (3) below, which breaks down the performance into a topic (τ_i) effect and a system effect (σ_j), as previously done in the literature [6, 39] to conduct this kind of analysis:

$$Y_{ij} = \mu_{..} + \tau_i + \sigma_j + \quad \varepsilon_{ij} \tag{3}$$

Tables from 1(a) to 1(d) report the results of the GLMM of equation (3) applied to all the GoP runs; besides the ANOVA table, they also report the $\hat{\omega}^2_{\langle fact \rangle}$ effect size for the topic and system effects as well as the number of significantly different system pairs according to the Tukey HSD test.

We can observe as, consistently with previous findings in the literature, both the topic and system effects are statistically significant, for all the measures; moreover, they are large size effects and, as expected, the topic effect is the most prominent one. We can also observe that, depending on the measure, between $\frac{1}{4}$ and $\frac{1}{3}$ of the system pairs are significantly different.

Since as it emerges in Section 5.3 and it is known from the literature [7], n-grams deteriorate retrieval performances, we repeated the above analysis removing the runs containing the `5grams` component, i.e. using a smaller GoP consisting of 264 runs, in order to check that the spread in performance observed in the above analyses is not mainly due to the `5grams` component. Tables from 2(a) to 2(d) report the outcomes of this analysis. We can observe both the topic and the system effects are statistically significant but while the topic effect is still a large-size effect, the system effect is now a medium-size effect, close to a small-size one in the case of AP and P10. Also, the number of significantly different system pairs has drastically reduced, being between 2% and 3% of the total possible pairs, an order of magnitude less than in the previous case. This clearly indicates that much of the variability was due to low performing n-grams while all the other systems are somehow more homogeneous.

5.2. Official Runs

Table 4 reports the average performance of the official runs over the 50 topics assessed by NIST, where the maximum score for each measure is highlighted in bold. It clearly emerges that `ims_wcs_ap_uf`, i.e. merging by putting unique documents first, does not work at all and it is a far too simplistic idea. The training experiments on the TREC 2004 Robust track were not so negative but, in the training phase, runs retrieved 1,000 documents while the Core Track runs retrieve 10,000 documents and this gives room for much more noise, especially in the unique documents. `ims_wcmbsum_ap`, i.e. CombSum with Min-Max normalization weighted by AP, is the best method in terms of AP and nDCG while `ims_wcs_rbp` is the best one in terms of P10. We can also observe a kind of trend in the case of P10 where merging with weights based on a more top-heavy measure (RBP, ERR, P10) works slightly better.

We applied the GLMM of eq. (3) to the official runs, excluding `ims_wcs_ap_uf` which clearly performs significantly worse than all the other runs. The ANOVA test reports significant differences among the official runs (without `ims_wcs_ap_uf`) and Tables from 4(a) to 4(d) show how runs are grouped according to the Tukey HSD test. It emerges, consistently across all the evaluation measures, that weighted combsum approaches are better than plain combsum, confirming previous findings in the literature, while there are no significant differences among the measures used for weighting.

5.3. Grid of Points

We broke-down the contribution of the different components in the GoP – namely, stop lists, stemmers and n-grams, IR models – according to the methodology we proposed in [14, 16].

We define a three factors design where we manipulate factors A, B and C corresponding to the stop lists, the stemmers/n-grams and the IR models respectively; with this design we can also study the interaction

(a) AP.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	626.2727	49	12.7810	2652.1999	0.0000	0.8873
System	28.5738	329	0.0869	18.0224	0.0000	0.2543
Error	77.6879	16121	0.0048			
Total	732.5344	16499				
Significantly Different Pairs	15,693 out of 54,285 possible pairs (28.91%)					

(b) P10.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	972.6175	49	19.8493	766.1859	0.0000	0.6944
System	133.2276	329	0.4049	15.6310	0.0000	0.2258
Error	417.6416	16121	0.0259			
Total	1523.4868	16499				
Significantly Different Pairs	13,364 out of 54,285 possible pairs (24.62%)					

(c) nDCG.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	788.9116	49	16.1002	2078.6696	0.0000	0.8605
System	71.2373	329	0.2165	27.9553	0.0000	0.3496
Error	124.8644	16121	0.0077			
Total	985.01343	16499				
Significantly Different Pairs	17,407 out of 54,285 possible pairs (32.07%)					

(d) Recall.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	788.9116	49	16.1002	2078.6696	0.0000	0.8605
System	71.2373	329	0.2165	27.9553	0.0000	0.3496
Error	124.8644	16121	0.0077			
Total	985.01343	16499				
Significantly Different Pairs	12,937 out of 54,285 possible pairs (23.83%)					

Table 2: ANOVA tables for the GLMM of eq. (3) on the GoP runs using different evaluation measures.

(a) AP.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	559.48381	49	11.4180	4980.1578	0.0000	0.9487
System	2.6065	263	0.0099	4.3226	$<$ e-5	0.0621
Error	29.5461	12887	0,0023			
Total	591.6364	13199				
Significantly Different Pairs	649 out of 34,716 possible pairs (1.87%)					

(b) P10.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	920.3072	49	18.7817	1181.6869	0.0000	0.8142
System	19.1455	263	0.0727	4,5801	$<$ e-5	0.0666
Error	204.8265	12887	0.0159			
Total	1144.2793	13199				
Significantly Different Pairs	855 out of 34,716 possible pairs (2.46%)					

(c) nDCG.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	679.2036	49	13.8612	4621.3181	0.0000	0.9449
System	4.4164	263	0.0168	5.5985	$<$ e-5	0.0839
Error	38.6536	12887	0,0030			
Total	722.2736	13199				
Significantly Different Pairs	1,184 out of 34,716 possible pairs (3.41%)					

(d) Recall.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	674.8188	49	13.7718	3652.3937	0.0000	0.9313
System	8.5998	263	0.0327	8.6721	$<$ e-5	0.1326
Error	48.5920	12887	0.0038			
Total	732.0108	13199				
Significantly Different Pairs	2,539 out of 34,716 possible pairs (7.31%)					

Table 3: ANOVA tables for the GLMM of eq. (3) on the GoP runs without the `5grams` component for different evaluation measures.

Run	AP	P10	nDCG	Recall
ims_cmbsum	0.2625	0.5720	0.5104	0.8092
ims_wcmbsum_ap	**0.2795**	0.5900	**0.5223**	0.8158
ims_wcs_ap_uf	0.0400	0.0160	0.1361	0.8046
ims_wcs_err	0.2776	0.6000	0.5202	0.8151
ims_wcs_ndcg	0.2719	0.5840	0.5183	0.8158
ims_wcs_p10	0.2781	0.5980	0.5195	0.8146
ims_wcs_rbp	0.2794	**0.6040**	0.5197	0.8122
ims_wcs_recall	0.2704	0.5860	0.5171	**0.8166**
ims_wcs_rprec	0.2759	0.5880	0.5199	0.8147
ims_wcs_twist	0.2765	0.5920	0.5197	0.8159

Table 4: Average performance of the official runs over the 50 topics assessed by NIST. The maximum score of each column is highlighted in bold.

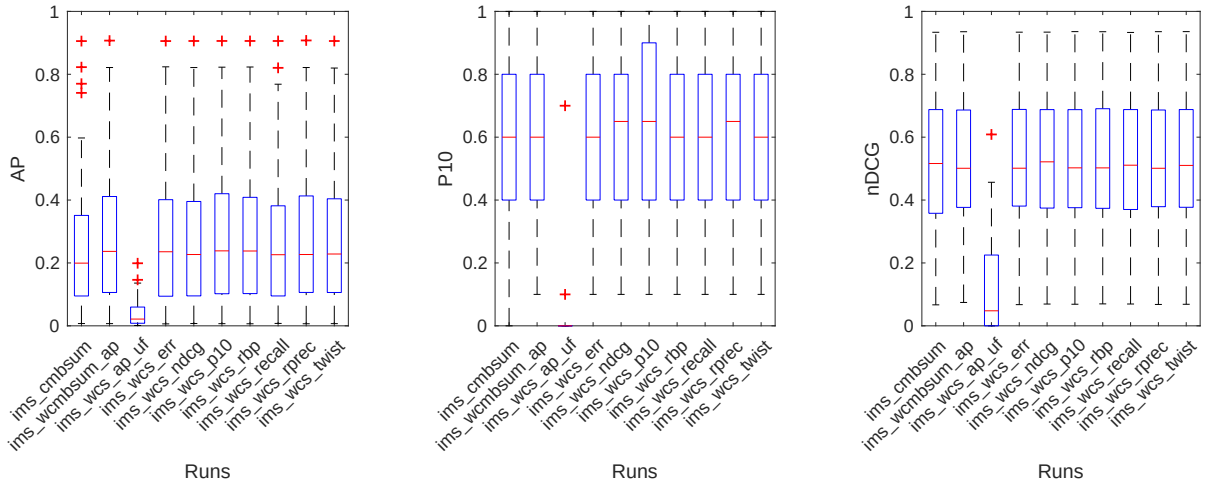

Figure 2: Box plot of the performance distributions of the official runs for the different evaluation measures.

between component pairs. The full GLMM model for the described factorial ANOVA for repeated measures is:

$$Y_{ijkl} = \underbrace{\mu.... + \tau_i + \alpha_j + \beta_k + \gamma_l}_{\text{Main Effects}} + \underbrace{\alpha\beta_{jk} + \alpha\gamma_{jl} + \beta\gamma_{kl} + \alpha\beta\gamma_{jkl}}_{\text{Interaction Effects}} + \underbrace{\varepsilon_{ijkl}}_{\text{Error}} \tag{4}$$

where: Y_{ijkl} is the score of the i-th subject in the j-th, k-th, and l-th factors; $\mu....$ is the grand mean; τ_i is the effect of the i-th subject, i.e. topics, where $\tau_i = \mu_{i...} - \mu....$ and $\mu_{i...}$ is the mean of the i-th subject; $\alpha_j = \mu_{.j..} - \mu....$ is the effect of the j-th factor, i.e. stop lists, where $\mu_{.j..}$ is the mean of the j-th factor; $\beta_k = \mu_{..k.} - \mu....$ is the effect of the k-th factor, i.e. stemmers or n-grams, where $\mu_{..k.}$ is the mean of the k-th factor; and, $\gamma_l = \mu_{...l} - \mu....$ is the effect of the l-th factor, i.e. IR models, where $\mu_{...l}$ is the mean of the l-th factor; ε_{ijkl} is the error committed by the model in predicting the score of the i-th subject in the three factors j, k, l. It consists of all the interaction terms between the subjects and the fixed factors, such as $(\tau\alpha)_{ij}$, $(\tau\beta)_{ik}$ and so on, plus the error ϵ_{ijkl} which is an additional error due to uncontrolled sources of variance.

Tables from 5(a) to 5(c) report the summary ANOVA tables for the conducted analyses on the different evaluation measures. It emerges that stop lists, stemmers/n-grams, and IR models are always a significant effect while only the interaction between stemmers/n-grams and IR models is significant. These results are

(a) AP.

AP	Systems	Groups	
0.2795	ims_wcmbsum_ap	X	
0.2794	ims_wcs_rbp	X	
0.2781	ims_wcs_p10	X	
0.2776	ims_wcs_err	X	
0.2765	ims_wcs_twist	X	
0.2759	ims_wcs_rprec	X	
0.2719	ims_wcs_ndcg	X	X
0.2704	ims_wcs_recall	X	X
0.2625	ims_cmbsum		X

(b) P10.

P@10	Systems	Groups	
0.6040	ims_wcs_rbp	X	
0.6000	ims_wcs_err	X	
0.5980	ims_wcs_p10	X	
0.5920	ims_wcs_twist	X	X
0.5900	ims_wcmbsum_ap	X	X
0.5880	ims_wcs_rprec	X	X
0.5860	ims_wcs_recall	X	X
0.5840	ims_wcs_ndcg	X	X
0.5720	ims_cmbsum		X

(c) nDCG.

nDCG	Systems	Groups	
0.5223	ims_wcmbsum_ap	X	
0.5202	ims_wcs_err	X	
0.5199	ims_wcs_rprec	X	
0.5197	ims_wcs_twist	X	
0.5197	ims_wcs_rbp	X	
0.5195	ims_wcs_p10	X	
0.5183	ims_wcs_ndcg	X	
0.5171	ims_wcs_recall	X	X
0.5104	ims_cmbsum		X

(d) Recall.

recall	Systems	Groups	
0.8166	ims_wcs_recall	X	
0.8159	ims_wcs_twist	X	X
0.8158	ims_wcs_ndcg	X	X
0.8158	ims_wcmbsum_ap	X	X
0.8151	ims_wcs_err	X	X
0.8147	ims_wcs_rprec	X	X
0.8146	ims_wcs_p10	X	X
0.8122	ims_wcs_rbp	X	X
0.8092	ims_cmbsum		X

Table 5: Tukey HSD Test for the GLMM of eq. (3) on the official runs for different evaluation measures..

partially discordant with previous findings [14, 16] where the only significant interaction effect was the Stop List*IR Model one. Another difference with respect to our previous work is that now the stop list effect is an extremely (almost negligible) small-size effect while before was a small to medium-size effect; also stemmers/$n-$grams were small to medium size effects while now they are large-size effects.

Figure 3 shows the main effects plots for the stop lists, stemmers/n-grams, and IR models for the different evaluation measures. Even in this case there are some discrepancies with respect to previous findings [14, 16]. Whilst the main effects of stop lists and stemmers present minor discrepancies, even if we used exactly the same stop lists and stemmers as in the previous work, the relative ranking of IR models turns out to be different, also among those models which are the same as in previous work.

Figure 4 shows the interaction effects plots for the stop lists, stemmers/n-grams, and IR models for the different evaluation measures. As expected in the interaction between stemmers/n-grams and IR models, we observe a noticeable difference between the group of stemmers and the n-grams, which greatly lower the performance of all the models. However, also in this case, we observe different with respect to our previous work. For example, the interaction between `bm25` and `nostop` was negative and the absence of a stop list lowered the performance while now it does not change much for `bm25` whether a stop list is applied or not.

All these differences with respect to our previous work can be certainly due to the change in the experimental collections but they could be also due to intrinsic differences between Lucene, used here, and Terrier, used in previous work, even if many of the investigated components are the same.

5.4. Contribution of the Components to the Pool

In this section we try to analyze of the different components contribute to the pool or, more precisely, how they retrieve relevant documents.

For each relevant document in the pool, NIST has provided the number of teams which have retrieved that relevant document. Therefore, unique relevant documents are those retrieved by just one team (labelled

(a) AP.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	626.2727	49	12.7811	2652.1999	0.0000	0.8873
Stop list	0.0922	5	0.0184	3.8251	0.0018	0.0008
Stemmer	25.7429	4	6.4357	1335.4788	0.0000	0.2444
IR Model	0.6560	10	0.0656	13.6134	< e-5	0.0076
Stop list*Stemmer	0.0244	20	0.0012	0.2535	0.9997	–
Stop list*IR Model	0.1235	50	0.0025	0.5127	0.9983	–
Stemmer*IR Model	1.9027	40	0.0476	9.8706	< e-5	0.0211
Stop list* Stemmer*IR Model	0.0321	200	0.0001	0.0333	1.0000	–
Error	77.6879	16121	0.0048			
Total	732.5345	16499				

(b) P10.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	972.6175	49	19.8493	766.1859	0.0000	0.6944
Stop list	0.2937	5	0.0587	2.2674	0.0451	0.0003
Stemmer	108.8487	4	27.2122	1050.3922	0.0000	0.2028
IR Model	12.1523	10	1.2152	46.9078	< e-5	0.0271
Stop list*Stemmer	0.2789	20	0.0139	0.5384	0.9520	–
Stop list*IR Model	0.7741	50	0.0155	0.5976	0.9893	–
Stemmer*IR Model	10.2303	40	0.2558	9.8723	< e-5	0.0211
Stop list* Stemmer*IR Model	0.6495	200	0.0032	0.1254	1.0000	–
Error	417.6417	16321	0.0259			
Total	1523.4868	16499				

(c) nDCG.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	788.9117	49	16.1002	2078.6696	0.0000	0.8605
Stop list	0.1710	5	0.0342	4.4152	0.0005	0.0010
Stemmer	64.2259	4	16.0565	2073.01818	0.0000	0.3344
IR Model	0.6950	10	0.0695	8.9728	< e-5	0.0048
Stop list*Stemmer	0.0331	20	0.0017	0.2136	0.9999	–
Stop list*IR Model	0.2140	50	0.0043	0.5526	0.9957	–
Stemmer*IR Model	5.8485	40	0.1462	18.8772	< e-5	0.0415
Stop list* Stemmer*IR Model	0.0498	200	0.0002	0.0322	1.0000	–
Error	124.8644	16321	0.0077			
Total	985.0134	16499				

Table 6: ANOVA tables for the GLMM of eq. (4) on the GoP runs using different evaluation measures.

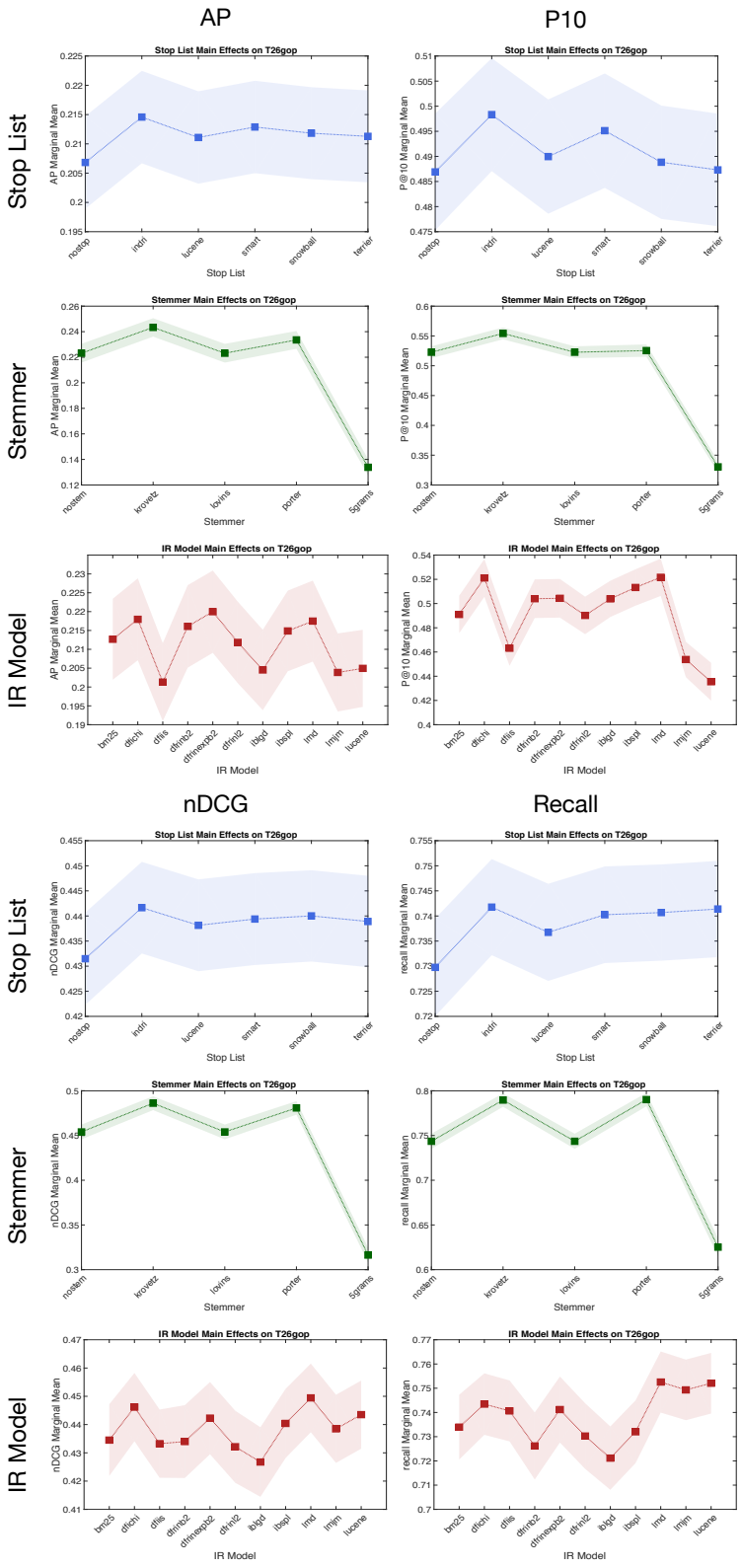

Figure 3: Main effects of stop lists, stemmers/n-grams and IR models for different evaluation measures.

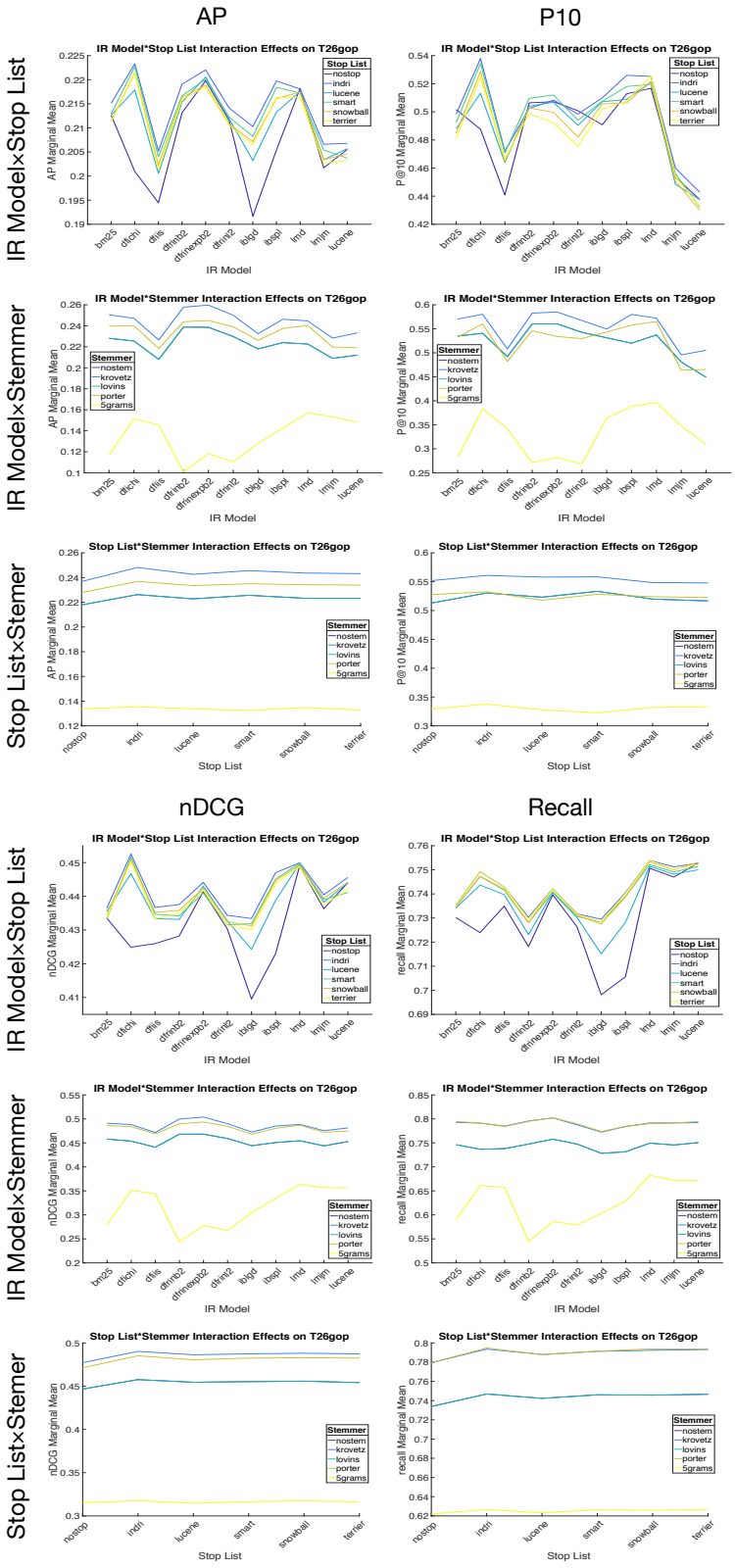

Figure 4: Interaction effects of stop lists, stemmers/n-grams and IR models for different evaluation measures.

Source	SS	DF	MS	F	p	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	9093454.0338	49	185580.6945	17803.7398	0.0000	0.7790
Stop list	1069.6607	5	213.9321	20.5236	< e-5	0.0004
Stemmer	153938.0836	4	38484.5209	3692.0241	0.0000	0.0563
IR Model	5838.2465	10	583.8247	56.0094	< e-5	0.0022
Team	8302968.0506	14	593069.1464	56896.2673	0.0000	0.7630
Topic*Team	20667655.8943	686	30127.7782	2890.3174	0.0000	0.8890
Stop list*Team	2520.2141	70	36.0031	3.4539	< e-5	0.0006
Stemmer*Team	201517.5295	56	3598.5273	345.2258	0.0000	0.0723
IR Model*Team	10204.1723	140	72.8869	6.9924	< e-5	0.0033
Error	2568449.7401	246405	10.4237			
Total	41009345.9873	247439				

Table 7: Four-way ANOVA of eq. (5).

`1T`); according to NIST data, the maximum is 15 teams retrieving a given relevant document (labelled `15T`), with all the other possible bins in-between.

We processed all the 330 GoP runs and counted, for each topic: how many `1T` documents they retrieved – i.e. documents retrieved by just one team; how many `2T` documents they retrieved – i.e. documents retrieved by just two teams; and so on up to how many `15T` documents they retrieved.

We then analysed these data according to the following GLMM

$$Y_{ijklm} = \underbrace{\mu..... + \tau_i + \alpha_j + \beta_k + \gamma_l + \delta_m}_{\text{Main Effects}} + \underbrace{\tau\delta_{im} + \alpha\delta_{jm} + \beta\delta_{km} + \gamma\delta_{lm}}_{\text{Interaction Effects}} + \underbrace{\varepsilon_{ijklm}}_{\text{Error}} \quad (5)$$

where, as above, τ is the topics effect, α is the stop list effect, β is the stemmer/n-gram effect, and γ is the IR model effect while δ is the effect of the number of teams retrieving a given document, followed by the interactions among these effects.

Table 7 reports the ANOVA table for the conducted analysis. We can see that all the main effects are statistically significant, suggesting that all the components play a role in determining how relevant documents are retrieved. When it comes to interactions, only the Stemmer*Team interaction is significant.

Figure 5 shows the main effects plots for stop lists, stemmers/n-grams and IR models. We can see how all the stop lists give somehow similar contributions in terms of retrieved relevant documents, while n-grams have the most negative impact, being `krovetz` and `porter` the best options among stemmers. When it comes to IR models, the vector space model and the LM perform better than others.

6. Conclusions

We participated in the Core Track with the goal of getting a better understanding on how off-the-shelf open source systems perform with respect to state-of-the-art TREC ones and how IR components contribute to the pools.

To this end, we used Lucene to create a GoP of 330 runs, with different kinds of stop lists, stemmers/n-grams, and IR models, basically representing almost everything you can get out-of-the-box from Lucene. We also applied data fusion techniques on these GoP runs in order further improve the final performance. We found out that the merged runs perform reasonably well, that there is not much difference among the various fusion strategies adopted, and that putting unique retrieved documents first, without any further care, was not effective at all.

We then analysed the GoP runs themselves to break-down their performance into those of their constituting components. We found some discrepancies with our previous work, which might be due to various factors besides experimenting on a new collection. For example, previous work was build on Terrier while here we experimented with Lucene. Therefore, we could have many differences, starting from tokenization

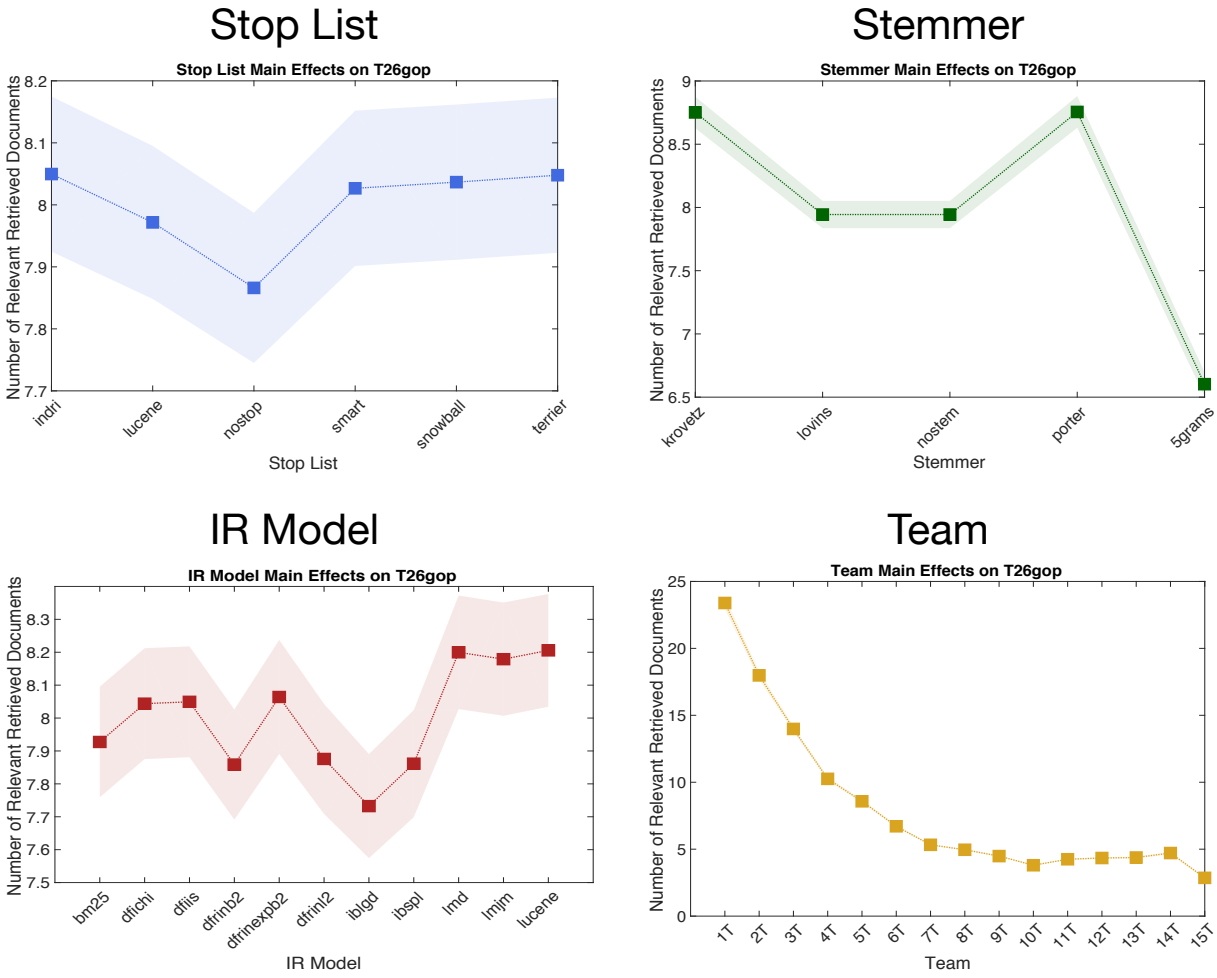

Figure 5: Main effects of stop lists, stemmers/n-grams and IR models in terms or relevant documents retrieved.

and going onwards up to the implementation of the IR models. So, further investigation is needed in this respect.

Finally, we started to envision a way to understand how component contribute to the pools and found some preliminary results in terms of the effects of stop lists, stemmers, and IR models. Also in this case, this is just an initial step and further modelling and analysis are needed.

References

[1] Allan, J., Harman, D. K., Kanoulas, E., Li, D., Van Gysel, C., Voorhees, E. M., 2018. TREC 2017 Common Core Track Overview. In: Voorhees, E. M., Ellis, A. (Eds.), The Twenty-Sixth Text REtrieval Conference Proceedings (TREC 2017). National Institute of Standards and Technology (NIST), Special Publication, Washington, USA.

[2] Amati, G., van Rijsbergen, C. J., 2002. Probabilistic Models of Information Retrieval based on measuring the Divergence From Randomness. ACM Transactions on Information Systems (TOIS) 20 (4), 357–389.

[3] Anava, Y., Shtok, A., Kurland, O., Rabinovich, E., 2016. A Probabilistic Fusion Framework. In: Zhai, C., Mukhopadhyay, S., Bertino, E., Crestani, F., Mostafa, J., Tsinghua, J. T., Si, L., Zhou, X. (Eds.), Proc. 25th International Conference on Information and Knowledge Management (CIKM 2016). ACM Press, New York, USA, pp. 1463–1472.

[4] Arguello, J., Crane, M., Diaz, F., Lin, J., Trotman, A., December 2015. Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). SIGIR Forum 49 (2), 107–116.

[5] Armstrong, T. G., Moffat, A., Webber, W., Zobel, J., 2009. Improvements That Don't Add Up: Ad-Hoc Retrieval Results Since 1998. In: Cheung, D. W.-L., Song, I.-Y., Chu, W. W., Hu, X., Lin, J. J. (Eds.), Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009). ACM Press, New York, USA, pp. 601–610.
[6] Banks, D., Over, P., Zhang, N.-F., May 1999. Blind Men and Elephants: Six Approaches to TREC data. Information Retrieval 1 (1-2), 7–34.
[7] Büttcher, S., Clarke, C. L. A., Cormack, G. V., 2010. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press, Cambridge (MA), USA.
[8] Chapelle, O., Metzler, D., Zhang, Y., Grinspan, P., 2009. Expected Reciprocal Rank for Graded Relevance. In: Cheung, D. W.-L., Song, I.-Y., Chu, W. W., Hu, X., Lin, J. J. (Eds.), Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009). ACM Press, New York, USA, pp. 621–630.
[9] Clinchant, S., Gaussier, E., 2010. Information-Based Models for Ad Hoc IR. In: Crestani, F., Marchand-Maillet, S., Efthimiadis, E. N., Savoy, J. (Eds.), Proc. 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010). ACM Press, New York, USA, pp. 234–241.
[10] Croft, W. B., 2000. Combining Approaches to Information Retrieval. In: Croft, W. B. (Ed.), Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval. Kluwer Academic Publishers, Norwell (MA), USA, pp. 1–36.
[11] Ferro, N., February 2017. Reproducibility Challenges in Information Retrieval Evaluation. ACM Journal of Data and Information Quality (JDIQ) 8 (2), 8:1–8:4.
[12] Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., Zobel, J., June 2016. Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". SIGIR Forum 50 (1), 68–82.
[13] Ferro, N., Harman, D., 2010. CLEF 2009: Grid@CLEF Pilot Track Overview. In: Peters, C., Di Nunzio, G. M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (Eds.), Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments – Tenth Workshop of the Cross–Language Evaluation Forum (CLEF 2009). Revised Selected Papers. Lecture Notes in Computer Science (LNCS) 6241, Springer, Heidelberg, Germany, pp. 552–565.
[14] Ferro, N., Silvello, G., 2016. A General Linear Mixed Models Approach to Study System Component Effects. In: Perego, R., Sebastiani, F., Aslam, J., Ruthven, I., Zobel, J. (Eds.), Proc. 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016). ACM Press, New York, USA, pp. 25–34.
[15] Ferro, N., Silvello, G., 2016. The CLEF Monolingual Grid of Points. In: Fuhr, N., Quaresma, P., Gonçalves, T., Larsen, B., Balog, K., Macdonald, C., Cappellato, L., Ferro, N. (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventh International Conference of the CLEF Association (CLEF 2016). Lecture Notes in Computer Science (LNCS) 9822, Springer, Heidelberg, Germany, pp. 16–27.
[16] Ferro, N., Silvello, G., February 2018. Toward an Anatomy of IR System Component Performances. Journal of the American Society for Information Science and Technology (JASIST) 69 (2), 187–200.
[17] Ferro, N., Silvello, G., Keskustalo, H., Pirkola, A., Järvelin, K., 2016. The Twist Measure for IR Evaluation: Taking User's Effort Into Account. Journal of the American Society for Information Science and Technology (JASIST) 67 (3), 620–648.
[18] Fox, E. A., Shaw, J., 1993. Combination of Multiple Searches. In: Harman, D. K. (Ed.), The Second Text REtrieval Conference (TREC-2). National Institute of Standards and Technology (NIST), Special Publication 500-215, Washington, USA, pp. 243–252.
[19] Hochberg, Y., Tamhane, A. C., 1987. Multiple Comparison Procedures. John Wiley & Sons, USA.
[20] Järvelin, K., Kekäläinen, J., October 2002. Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems (TOIS) 20 (4), 422–446.
[21] Kharazmi, S., Scholer, F., Vallet, D., Sanderson, M., June 2016. Examining Additivity and Weak Baselines. ACM Transactions on Information Systems (TOIS) 34 (4), 23:1–23:18.
[22] Kocabaş, İ., Dinçer, B. T., Karaoğlan, B., April 2014. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. Information Retrieval 17 (2), 153–176.
[23] Krovetz, R., April 2000. Viewing morphology as an inference process. Artificial Intelligence 118 (1–2), 277–294.
[24] Lee, J. H., 1995. Combining Multiple Evidence from Different Properties of Weighting Schemes. In: Fox, E. A., Ingwersen, P., Fidel, R. (Eds.), Proc. 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995). ACM Press, New York, USA, pp. 180–188.
[25] Lillis, D., Zhang, L., Toolan, F., Collier, R. W., Leonard, D., Dunnion, J., 2010. Estimating Probabilities for Effective Data Fusion. In: Crestani, F., Marchand-Maillet, S., Efthimiadis, E. N., Savoy, J. (Eds.), Proc. 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010). ACM Press, New York, USA, pp. 347–354.
[26] Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., Macdonald, C., Vigna, S., 2016. Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge. In: Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G. M., Hauff, C., Silvello, G. (Eds.), Advances in Information Retrieval. Proc. 38th European Conference on IR Research (ECIR 2016). Lecture Notes in Computer Science (LNCS) 9626, Springer, Heidelberg, Germany, pp. 357–368.
[27] Lovins, J. B., January/February 1971. Error Evaluation for Stemming Algorithms as Clustering Algorithms. Journal of the American Society for Information Science (JASIS) 22 (1), 28–40.
[28] Maxwell, S., Delaney, H. D., 2004. Designing Experiments and Analyzing Data. A Model Comparison Perspective, 2nd Edition. Lawrence Erlbaum Associates, Mahwah (NJ), USA.
[29] McNamee, P., Mayfield, J., January 2004. Character N-Gram Tokenization for European Language Text Retrieval. Information Retrieval 7 (1-2), 73–97.
[30] Moffat, A., Zobel, J., 2008. Rank-biased Precision for Measurement of Retrieval Effectiveness. ACM Transactions on

Information Systems (TOIS) 27 (1), 2:1–2:27.
[31] Montague, M., Aslam, J. A., 2001. Relevance Score Normalization for Metasearch. In: Pu, C., Paques, H., Liu, L., Grossman, D. (Eds.), Proc. 10th International Conference on Information and Knowledge Management (CIKM 2001). ACM Press, New York, USA, pp. 427–433.
[32] Olejnik, S., Algina, J., December 2003. Generalized Eta and Omega Squared Statistics: Measures of Effect Size for Some Common Research Designs. Psychological Methods 8 (4), 434–447.
[33] Porter, M. F., July 1980. An algorithm for suffix stripping. Program 14 (3), 130–137.
[34] Robertson, S. E., Zaragoza, U., 2009. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval (FnTIR) 3 (4), 333–389.
[35] Roelleke, T., 2013. Information Retrieval Models. Foundations and Relationships. Morgan & Claypool Publishers, USA.
[36] Rutherford, A., 2011. ANOVA and ANCOVA. A GLM Approach, 2nd Edition. John Wiley & Sons, New York, USA.
[37] Sakai, T., June 2014. Statistical Reform in Information Retrieval? SIGIR Forum 48 (1), 3–12.
[38] Salton, G., McGill, M. J., 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York, USA.
[39] Tague-Sutcliffe, J. M., Blustein, J., 1994. A Statistical Analysis of the TREC-3 Data. In: Harman, D. K. (Ed.), The Third Text REtrieval Conference (TREC-3). National Institute of Standards and Technology (NIST), Special Publication 500-225, Washington, USA, pp. 385–398.
[40] Voorhees, E. M., 2004. Overview of the TREC 2004 Robust Track. In: Voorhees, E. M., Buckland, L. P. (Eds.), The Thirteenth Text REtrieval Conference Proceedings (TREC 2004). National Institute of Standards and Technology (NIST), Special Publication 500-261, Washington, USA.
[41] Wu, S., 2012. Data Fusion in Information Retrieval. Springer-Verlag, Heidelberg, Germany.
[42] Zhai, C., 2008. Statistical Language Models for Information Retrieval. A Critical Review. Foundations and Trends in Information Retrieval (FnTIR) 2 (3), 137–213.