Unfolding Off-the-shelf IR Systems for Reproducibility

Emanuele Di Buccio
Dept. Information Engineering
University of Padua, Italy
dibuccio@dei.unipd.it

Giorgio Maria Di Nunzio
Dept. Information Engineering
University of Padua, Italy
dinunzio@dei.unipd.it

Nicola Ferro
Dept. Information Engineering
University of Padua, Italy
ferro@dei.unipd.it

Donna Harman
National Institute of Standards
and Technology (NIST), USA
donna.harman@nist.gov

Maria Maistro
Dept. Information Engineering
University of Padua, Italy
maistro@dei.unipd.it

Gianmaria Silvello
Dept. Information Engineering
University of Padua, Italy
silvello@dei.unipd.it

ABSTRACT

In this position paper, we discuss the issue of how to ensure reproducibility of the results when off-the-shelf open source Information Retrieval (IR) systems are used. These systems provided a great advancement to the field but they rely on many configurations parameters which are often implicit or hidden in the documentation and/or source code. If not fully understood and made explicit, these parameters may make it difficult to reproduce results or even to understand why a system is not behaving as expected.

The paper provides examples of the effects of hidden parameters in off-the-shelf IR systems, describes the enabling technologies needed to embody the approach, and show how these issues can be addressed in the broader context of component based IR evaluation.

We propose a solution for systematically unfolding the configuration details of off-the-shelf IR systems and understanding whether a particular instance of a system using is behaving as expected. The proposal requires to: 1) build a taxonomy of components used by off-the-shelf systems, 2) uniquely identify them and their combination in a given configuration, 3) run each configuration on standard test collections, 4) compute the expected performance measures for each run, 4) and publish on a Web portal all the gathered information in order to make accessible and comparable for everybody how an off-the-shelf system with a given configuration is expected to behave.

1. INTRODUCTION

Nowadays much of the research in the *Information Retrieval (IR)* field does not start from scratch but builds on off-the-shelf open source IR systems widely available [31], such as Lucene[1], Terrier[2], or Indri as part of the Lemur project[3]. These systems are typically used as either starting point by new researchers or students who are moving their first steps in IR or as building blocks in more complex systems by experienced researchers. Both these uses have a great impact on follow-up results, repeatability, reproducibility, and generalizability.

Indeed, in the first case, an off-the-shelf system may be used without a deep comprehension of its internals and this can influence both the obtained performances and the possibility for others to reproduce them, since these settings are often left implicit. For instance, it can happen that the implementation of the BM25 formula [28], by e.g. Terrier or Lucene, is slightly different from the actual formula, and young reserachers may not understand why the results produced by these systems are not equal to the theoretical ones. In the second case, the focus is on the newly developed components, e.g. a new stemmer or a new feedback module, and the off-the-shelf systems are on the background. Therefore, even if some careful tuning of their internals has been carried out, this is often not perceived as important as the description of the new component and details about it are simply left out. Moreover, when reporting about off-the-shelf systems, it is common to write something like "we used Lucene" which is not enough to guarantee future reproducibility of results due to the many differences among its various versions. Saying "we used Lucene 4.0.x", to indicate a whole branch of Lucene versions, may raise problems as well, because its different sub-versions can have significant differences. Even saying "we used Lucene 4.0.3" may be not enough; indeed each version has many settings, some of which are hidden into the documentation and/or implementation, and, if not made explicit, may cause you to report something (slightly) different from what you have actually done, as described in detail in the case study in the next section.

Therefore, we think that a better comprehension of how off-the-shelf IR systems work and they are tuned are a fundamental pre-requisite for ensuring the reproducibility of the experimental results and to guaranteeing that the researchers can choose the tool that best fits their tasks (see for example [9]). For this reason, we believe that a detailed documentation of the retrieval models available in open source libraries is needed. This documentation will supply researchers with tables and configuration files that are needed to obtain the same figures on the same experimental collections.

[1] `http://lucene.apache.org/`
[2] `http://terrier.org/`
[3] `http://www.lemurproject.org/`

The paper is organized as follows: Section 2 provides a more detailed example of how implicit internal settings of an off-the-shelf IR system can influence the outcomes of the retrieval; Section 3 describes the approach we propose to address these issues and to ensure better reproducibility when off-the-shelf IR systems are exploited; Section 4 proposes a possible strategy for implementing the proposed approach, by both combining existing building blocks and developing new ones specifically needed for the purpose; Section 5 draws some conclusions and provides an outlook for future work.

2. CASE STUDY

In this section, we present some real case situations that show how some of the details about the implementation of these softwares may be easily overlooked. These small details may lead to important differences in the behaviour of the ranking list produced by these systems. We discuss two common problems: the computation of the document length and the management of ties in the ranked list of retrieved documents.

2.1 BM25 document length and Lucene

BM25 is one of the most successful ranking functions used by search engines to rank matching documents according to their relevance to a given search query. The building block of the BM25 is the weight w_i^{BM25} that is assigned to each token t_i (word, term, n-gram, etc.) that represents the document to be retrieved:

$$w_i^{BM25} = \frac{tf_i}{tf_i + K} \cdot w_i^{BIM}$$

where tf_i is the frequency of the term t_i in the document, w_i^{BIM} is the weight of the token t_i computed with the Binary Independence Model (usually approximated with the inverse document frequency), and K is a function of some parameters about the global statistics of the collection of documents:

$$K = k_1 * ((1 - b) + b * dl/\Delta)$$

where k_1 and b are usually set to 1.2 and 0.75, respectively, dl is the length of document d, and Δ is the average document length.

For example, the BM25 implementation of Lucene is an approximation of the above formulas. Indeed, the available implementation[4] uses an approximation of the length of the document, not the actual document length. This has been documented in the javadoc[5]. This problem has also been raised in some blogs[6]

> Also note that this [document length] is heavily quantized: we use only a single byte to encode the length by default. If the quantization or mixing of boost is a problem you can also make your own Similarity impl and store docLen yourself (eg as an int)?

Therefore, if a researcher wants to replicate the exact scores of the theoretical BM25 formula in Lucene, s/he needs to implement a custom function for the relevance scores of the token of a document.

The approximation of the document length may have a strong effect on the computation of the BM25 score, hence an impact on the ranked list produced by the search engine. For instance, using Lucene 4.7.2 with its own standard implementation of BM25 and with a custom one exactly implementing the BM25 formula, the MAP obtained with the TREC 2004 Robust Collection [33] is 0.2302 when approximated document length is used and 0.2352 when the length is not approximated.

It would also be very useful to know when this computation is done during the pipeline of the index construction (for example, whether document length is computed before or after stop words removal, word stemming, word de-compounding, and so on). This is important because the choice of the point in the pipeline changes the output of the statistics of document length and, consequently, the value of the BM25 score [32].

2.2 Score Ties and trec_eval

'Ties' happen when two (or more) documents have the same retrieval score computed by a retrieval function. During an experimental laboratory evaluation, it is necessary to define a clear procedure to manage this situation to avoid that a particular random permutation of the documents produces better results than another random permutation of the same documents. For this reason, the `trec_eval`[7] software performs a re-ordering of the documents of a run submitted for a task before computing the actual performance measures.

When importing runs, `trec_eval` may modify the actual ordering of the items in the file since it sorts items in descending order of `score` and descending lexicographical order of `document-id`, when `score`s are tied and this may have an impact on the computed performance measures and modify the ranking of the systems [20]. For example in TREC 5 [34], when ties and reordering happen, they cause average changes in *Average Precision (AP)* around 1% but with a maximum of 139%. Similar effects have been observed also by [13].

In theory, this issue should be mitigated in the case of sophisticated retrieval functions, since they rarely assign the same score to two different documents, but, in practice, this may happen also for them because of numerical approximations introduced by the implementations of the retrieval functions, as discussed before. Moreover, for some simple models (for example the Binary Independence Model) ties happen very frequently. Since some of these models are used as a baseline to compare with, we want to be certain that two experiments that use the same model produce the exact same ranked list given the same pre-processing pipeline.

For example, let N be the number of documents in the collection that is used in a TREC task. Let us assume that the guidelines of this task require participants to submit runs with at most the top H documents per query (usually, H = 1,000). If the program that implements the retrieval func-

[4] `https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/BM25Similarity.html`

[5] `https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/BM25Similarity.html#computeNorm(org.apache.lucene.index.FieldInvertState,org.apache.lucene.index.Norm)`

[6] `http://blog.mikemccandless.com/2012/03/new-index-statistics-in-lucene-40.html?showComment=1345804703310#c7345673044266042473`

[7] `http://trec.nist.gov/trec_eval/`

tion uses an approach like Document At A Time (DAAT) and a data structure like a priority queue to keep in memory the top 1,000 documents, the ordering criterion would be the following:

- if the score of document d_1 is greater than the score of document d_2, $score(d_1) > score(d_2)$, then rank d_1 before d_2.
- if the two scores are equal, $score(d_1) = score(d_2)$, then order documents according to their ID by lexicographic order.

However, the resulting list depends on when (and how) the order is performed. For simplicity, suppose that H = 3 and the collection has only four documents d_1, d_2, d_3, d_4, and the only relevant document given a query q is d_3. Now, we have two search engines s_1 and s_2 that implement the exact same retrieval function, produce the same scores for each document, in case of ties re-order documents in reverse lexicographic order, and follow the DAAT approach. The only difference between s_1 and s_2 is that, in case of ties, s_1 re-orders documents as soon as the document is ranked, while s_2 re-orders documents after the last document has been ranked. Let us assume that $score(d_1) = score(d_2) = score(d_3) < score(d_4)$. Both search engines observe documents in the following order: d_1, d_2, d_3, d_4. Since $H = 3$ the ranked list for s_1 is (step by step):

1. d_1
2. d_2, d_1
3. d_3, d_2, d_1
4. d_4, d_3, d_2

Search engine s_2 produces the following ranked list:

1. d_1
2. d_1, d_2
3. d_1, d_2, d_3
4. d_4, d_1, d_2
5. d_4, d_2, d_1

At the end of the experiment, the two search engines produced two different lists and s_1 got one relevant documents at rank 2 because ties were scrambled during the creation of the list. Consequently, s_1 is evaluated as being more effective than s_2 not because of the core retrieval algorithm but because of a simple choice, which might look marginal but still needs to be explicitly reported.

3. PROPOSED APPROACH

We propose to proceed as follows:

- we need to build a taxonomy of components, e.g. stemmers, ranking models, stop lists and so on, that are available for each off-the-shelf IR system;
- for each off-the-shelf system, a combination of its settings basically corresponds to a branch in the above taxonomy, i.e. we are using Lucene with Porter stemmer, without stop list, and scoring according to BM25 with given parameters. Note that a branch basically represents a configuration of the system under examination;
- for each branch/configuration, we run the off-the-shelf system on standard experimental collections, e.g. *Text REtrieval Conference (TREC)*[8] and *Conference and Labs of the Evaluation Forum (CLEF)*[9] ones, and we record the achieved performances according to a chosen measure, e.g. AP;
- we publish on a Web portal all the information about the branch (configuration), the tested collections, and the experimental results.

Note that, in the above steps, each component and setting needs to be uniquely identified in order to allow for exactly identifying a given configuration, experimental collection and measure. Figure 1 shows an example of the proposed approach. In this way, each researcher will be able to inspect the expected performances of an off-the-shelf system in several standard configurations. This will allow researchers to:

- know whether they have correctly configured the off-the-shelf system by checking if they are able to repeat the same performances;
- explicitly cite a given configuration, without the need of reporting all its details;
- pick the configuration with the performances which fit best to the task they are targeting.

Once the above mentioned goals are achieved, we can move one step forward. Indeed, as it emerges from the discussion in Section 2, even saying "we used BM25 as provided by Lucene" may not exactly match with the formal definition of BM25 [28] and may not match either with the implementation provided by, say, Terrier. Therefore, once we know well how the internals of each off-the-shelf IR system work, we can provide instructions on how to tune and set them to make the outputs of the systems as close as possible to the formal definition of a given model, say BM25. These, which may not be standard configurations, can then be detailed using the same mechanisms described above and become a reference point for the community. In this way, we will know better what the differences among off-the-shelf IR systems are and what causes them; moreover, we will be able to make more informed decisions about what system to use. Within this context, it would be important to have a "reference BM25 implementation" which implements the formal definition of the model. This implementation could be used to measure quantitatively how distant are off-the-shelf systems from it.

From the above discussion, it emerges that not all the components are unique to the branch of the system using them but they may actually be shared among different off-the-shelf systems. Consider, for example, the Porter stemmer: both Lucene and Terrier have their own implementations of the Porter stemmer, which are then unique to their branches, but they may also rely on the same implementation provided by the Snowball project[10], which is thus common to different branches. Therefore, the most general case, is not the one of a taxonomy but of a grid of

[8] `http://trec.nist.gov/`
[9] `http://www.clef-initiative.eu/`
[10] `https://github.com/snowballstem`

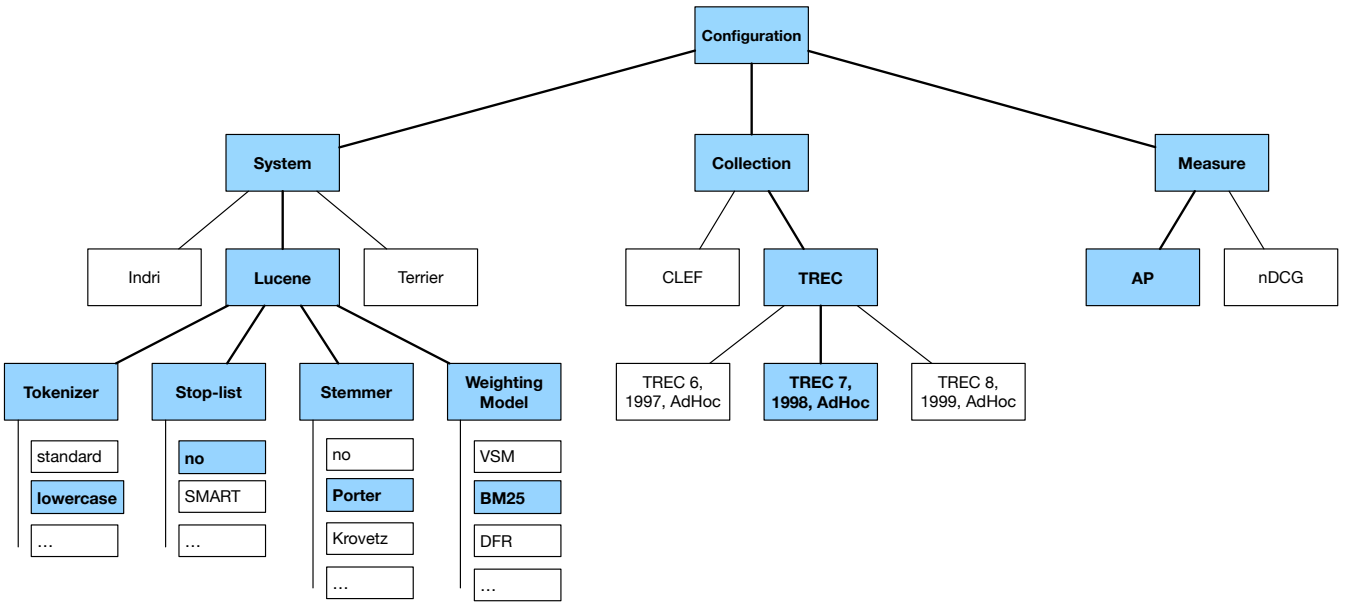

Figure 1: An example of a possible taxonomy.

components [18] which can be mixed together to instantiate a given off-the-shelf IR system. The broader topic of component-based evaluation [1, 18, 21] can help us in framing this more general situation. Indeed, it is oriented to a decompositional approach to IR experimentation where each component of an IR system can be, as far as possible, analysed and evaluated independently from the others. Isolating the single components of IR systems may allow us to better understand how they work and which are their effects on the system outcome. By employing this approach, for each off-the-shelf system, we could be able to (i) specify and implement the components concerning each retrieval method to be evaluated, (ii) formulate research hypotheses on the basis of methods viewed as alternatives to a particular retrieval operation, and (iii) test these hypotheses by measuring the performance of complete systems with each of the alternative components embedded.

4. TOWARDS IMPLEMENTING THE PROPOSED APPROACH

In accordance with the ideas about principles of a robust evaluation infrastructure [39], we will discuss the technological building blocks we plan to exploit to embody the approach discussed in the previous sections, we will introduce the off-the-shelf IR systems we plan to examine as well as initial examples of their components, and we will present the experimental collections and evaluation measures we plan to adopt.

4.1 Enabling Technologies

In order to implement the proposed approach, we plan to rely on and extend two main building blocks: the *Coordinated Information Retrieval Components Orchestration (CIRCO)* framework [17, 18], which allows us to represent the taxonomy of components and how they are connected in a specific off-the-shelf IR system, and the *Distributed Information Retrieval Evaluation Campaign Tool (DIRECT)* system [2, 3, 4, 5], which allows us to manage the experimental collections, the produced runs, and the computed performance measures.

The CIRCO framework allows for a *distributed, loosely-coupled, and asynchronous* experimental evaluation of *Information Retrieval (IR)* systems. The basic idea – and assumption – behind CIRCO is to streamline the architecture of an off-the-shelf IR system and represent it as a *pipeline* of components chained together. The processing proceeds by passing the results of the computations of a component as input to the next component in the pipeline without branches, i.e. no alternative paths are allowed in the chain. To get an intuitive idea of the overall approach adopted in CIRCO, consider the example pipeline shown in Figure 2. The example IR system is constituted by the following components:

- *tokenizer*: breaks the input documents into a sequence of tokens;
- *stop word remover*: removes stop words from the sequence of tokens;
- *stemmer*: stems the tokens;
- *indexer*: weights the tokens and stores them and the related information in an index.

Instead of directly feeding the next component as usually happens in an IR system, CIRCO operates by requiring each component to input and output from/to *eXtensible Markup Language (XML)* [38] files in a well-defined format. Each of the components of the IR pipeline provides an XML–based stream. These files can be reused by any of the components that follow in the pipeline, enabling a loosely–coupled, distributed, asynchronous evaluation of the components. The strengths of this approach lie first on its simplicity, since the only requirement is to convert the output to a given specification. Secondly, for the finest grained components, that act on a corpus basis, this is the only way of obtaining interoperability, since online computation on a whole corpus is not feasible.

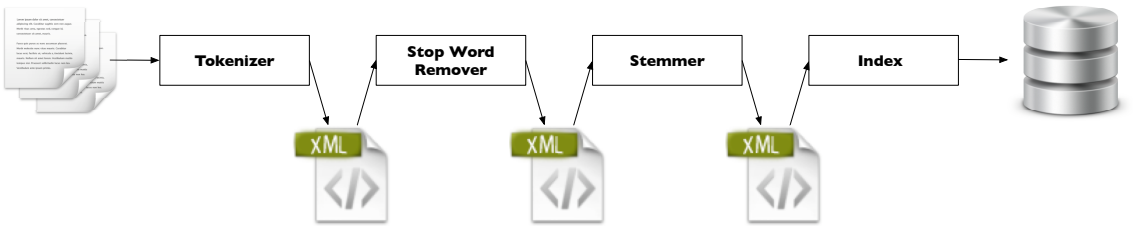

Figure 2: An example of CIRCO pipeline for an IR system.

CIRCO will give the means to achieve the main goals of our approach: we can use it for representing branches/configurations of an off-the-shelf IR system, in order to embody the first two objectives of our approach, i.e. precisely unfolding an off-the-shelf system configuration and providing a "reference implementation" for some components; but we can use it also for more advanced component-based cases, where components are shared across different off-the-shelf systems in a grid-like fashion.

Moreover, CIRCO will provide us an additional benefit by enabling us to finely trace what happens within each component for each processed token. Consider the XML fragment in the example of Figure 3: the token `Very` (line 24) is produced by a standard space-based tokenizer (line 25), it is then transformed to lower-cases `very` (line 14) by a lowercase filter (line 15) and finally stemmed `veri` (line 4) by the Lucene implementation of the Porter stemmer (line 5). These fine trace capabilities will provide researcher the possibility to follow step-by-step what happens to input documents as they pass from one component to another. If these XML dumps of the intermediate outputs of the processing are published on the Web portal together with the other information previously discussed, they will give researchers the chance to conduct a step-by-step debug of the runs they produce using off-the-shelf systems in order to understand how and why they are different from the expected ones, in case this happens.

As anticipated above and shown in Figure 4, the DIRECT system manages the scientific data produced during evaluation activities, as well as supports the archiving, access, citation, dissemination, and sharing of the experimental results. Therefore, it will be used, as a component of the Web portal, in order to publish the performance figures obtained on the used experimental collections for the selected configurations of off-the-shelf IR systems.

4.2 Inspected Off-the-shelf Systems

We choose to start from the most used systems narrowing down the employable components to those commonly integrated by standard configurations. The off-the-shelf IR systems we propose to analyse are: (i) the Indri system version 5.8; (ii) Apache Lucene version 5.1.0; and, (iii) Terrier IR Platform version 4.0. For each of these systems we propose to consider the following main components narrowing down their possible configurations to the most commonly adopted in default configurations, for example:

- stop-list: no stop-list, the SMART stop-list (571 English terms), the Lucene stop-list (33 English terms), the Indri stop-list (418 English terms), and the Terrier stop-list (733 English terms);
- stemmer: no stemmer, Porter stemmer[27] and Krovetz stemmer[23].
- weighting model: the vector space model (with diverse instantiation of the TF·IDF weighting scheme) [30], BM25 with default parameters [28] and the Divergence for Randomness model [8];
- relevance feedback and query expansion: KL-divergence with Relevance Models [24], Rocchio [29] with diverse TF·IDF instantiations, Query Expansion with Bose-Einstein Distribution or Hypergeometric Model [7].

Note that, for languages other than English, we will consider different open and shared linguistic resources, e.g. for stop lists and stemmers.

Relevance feedback approaches will be considered both in explicit and pseudo feedback settings: in the former case, the source for feedback is a set of judged documents, while in the latter the top k retrieved documents. Besides free parameters that are peculiar of the considered feedback approach, also the number of top retrieved documents, k, and the number of expansion terms or thresholds for term selection need to be explicitly documented.

4.3 Experimental Collections and Measures

Each system configuration will be evaluated against well-known and commonly used shared test collections:

- for English, the TREC 6, 1997 ad-hoc Track [35], the TREC 7, 1998 ad-hoc Track [36] and the TREC 8, 1999, ad-hoc Track [37] which adopt the TIPSTER corpus (disk 4 and 5 excluding the Congressional Record, 528K documents) and 50 topics;
- for European languages, the CLEF 2002 German monolingual ad-hoc track [10] (125K documents) and the Dutch monoligual ad-hoc track (192K documents), the CLEF 2005 French monolingual ad-hoc track [15] (177K document), the CLEF 2003 Italian monolingual ad-hoc track [11] (158K documents) and the CLEF 2006 Portuguese monolingual ad-hoc track [16] (211K documents) which are all based on corpora of newspaper articles and 50 topics;
- for non European languages, the CLEF 2009 Persian monolingual ad-hoc track [19] (166K documents) with 50 topics;
- for the Web, the TREC 21, 2012, Web Track [14] ClueWeb09 corpus (1040M documents) with 50 topics.

```
<stream identifier="c77f4b4e-2c87-4a5a-821c-84b8639be2f6">
  <resource identifier="doc1" mime-type="text/plain">
    <tokens>
      <token identifier="bdeb53fd-dacb-4fdf-91de-e5d709bf8609" value="veri">
        <component identifier="org.apache.lucene.analysis.en.PorterStemFilter" type="stemmer"/>
        <features>
          <feature name="field" value="body" type="xs:string" />
          <feature name="position" value="1" type="xs:long" />
          <feature name="start-offset" value="16" type="xs:long" />
          <feature name="end-offset" value="20" type="xs:long" />
          <feature name="type" value="&lt;ALPHANUM&gt;" type="xs:string" />
        </features>
        <trace>
          <token identifier="561e0c0a-e76c-4eaf-957a-bfac4ca2339e" value="very">
            <component identifier="org.apache.lucene.analysis.core.LowerCaseFilter" type="filter"/>
            <features>
              <feature name="field" value="body" type="xs:string" />
              <feature name="position" value="1" type="xs:long" />
              <feature name="start-offset" value="16" type="xs:long" />
              <feature name="end-offset" value="20" type="xs:long" />
              <feature name="type" value="&lt;ALPHANUM&gt;" type="xs:string" />
            </features>
            <trace>
              <token identifier="68e9a1a2-cb95-4e02-a5f1-f9da7c406a55" value="Very">
                <component identifier="org.apache.lucene.analysis.core.WhitespaceTokenizer" type="tokenizer"/>
                <features>
                  <feature name="field" value="body" type="xs:string" />
                  <feature name="position" value="4" type="xs:long" />
                  <feature name="start-offset" value="16" type="xs:long" />
                  <feature name="end-offset" value="20" type="xs:long" />
                  <feature name="type" value="&lt;ALPHANUM&gt;" type="xs:string" />
                </features>
              </token>
            </trace>
          </token>
        </trace>
      </token>
    </tokens>
  </resource>
</stream>
```

Figure 3: Example of fine tracing in CIRCO.

We can see that the proposed experimental collections span through different retrieval tasks (ad-hoc and Web), languages, corpus sizes and document types which cover a wide range of typical retrieval scenarios. We calculate the performances of each configurations in terms of *Average Precision (AP)* [12] for the collections with binary relevance judgments (TREC 6-8, CLEF collections) and *Normalized Discounted Cumulated Gain (nDCG)* [22] for the graded relevance one (TREC 21).

5. CONCLUSIONS AND FUTURE WORK

In this position paper we have pointed out the issues deriving from implicit or not-known settings in off-the-shelf open source IR systems and how these impair the replicability and reproducibility of the experimental results, as well as how they may make it difficult for researchers to understand whether they are using one of these systems as expected. We have proposed a general approach for addressing these issues, basically consisting of unfolding the taxonomy of components used in a systems, running them on standard experimental collections, and publishing all the gathered information for reference and comparison.

Future works will concern the actual implementation of this approach, which will require to carefully develop the taxonomy of components under examination, actually running them on the proposed standard collections, collecting all the performance figures and publish them on a Web portal. Moreover, even if the main required building blocks are already available, further work is needed to adapt them to the specific goal of this paper.

6. REFERENCES

[1] M. Agosti, G. Bordea, N. Ferro, A. Foncubierta, M. Imhof, B. Larsen, I. Masiero, S. Peruzzo, and G. Silvello. Deliverable D3.5 – Final Prototype of the Evaluation Infrastructure with the Distributed Evaluation Protocol. PROMISE Network of Excellence, EU 7FP, Contract N. 258191. `http://www.promise-noe.eu/documents/10156/03258cb7-1757-45d0-8da5-1b1632953a05`, September 2013.

[2] M. Agosti, E. Di Buccio, N. Ferro, I. Masiero, S. Peruzzo, and G. Silvello. DIRECTions: Design and Specification of an IR Evaluation Infrastructure. In T. Catarci, P. Forner, D. Hiemstra, A. Peñas, and G. Santucci, editors, *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics. Proceedings of the Third International Conference of the CLEF Initiative (CLEF 2012)*, pages 88–99. Lecture Notes in Computer Science (LNCS) 7488, Springer, Heidelberg, Germany, 2012.

[3] M. Agosti, G. M. Di Nunzio, M. Dussin, and N. Ferro. 10 Years of CLEF Data in DIRECT: Where We Are and Where We Can Go. In T. Sakai, M. Sanderson,

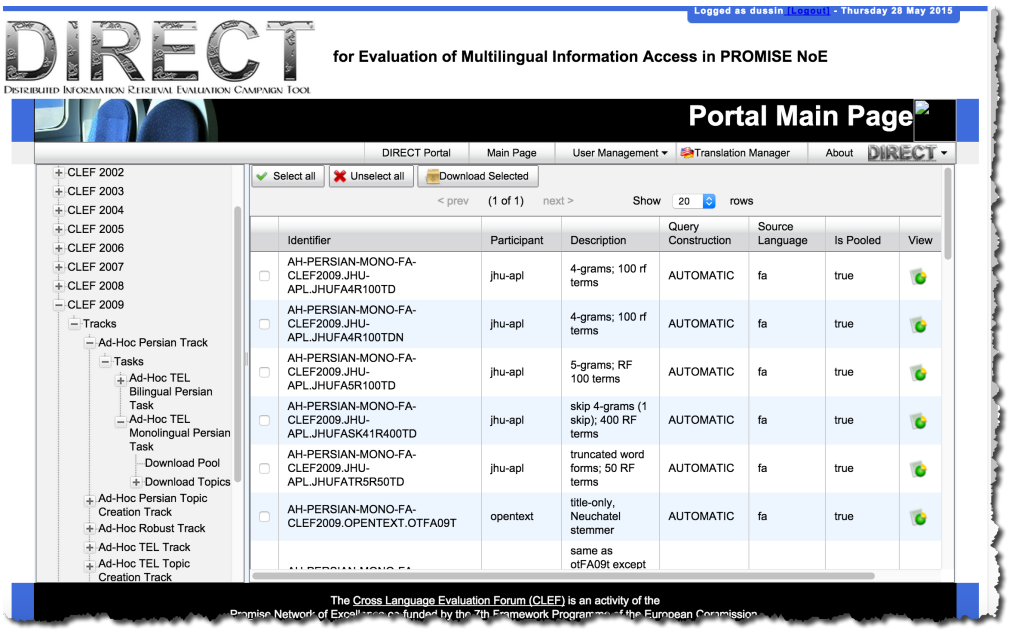

Figure 4: Example of management of experimental data in DIRECT.

and W. Webber, editors, *Proc. 3rd International Workshop on Evaluating Information Access (EVIA 2010)*, pages 16–24. National Institute of Informatics, Tokyo, Japan, 2010.

[4] M. Agosti, G. M. Di Nunzio, and N. Ferro. Scientific Data of an Evaluation Campaign: Do We Properly Deal With Them? In Peters et al. [25], pages 11–20.

[5] M. Agosti and N. Ferro. Towards an Evaluation Infrastructure for DL Performance Evaluation. In G. Tsakonas and C. Papatheodorou, editors, *Evaluation of Digital Libraries: An insight into useful applications and methods*, pages 93–120. Chandos Publishing, Oxford, UK, 2009.

[6] M. Agosti, N. Ferro, C. Peters, M. de Rijke, and A. Smeaton, editors. *Multilingual and Multimodal Information Access Evaluation. Proceedings of the International Conference of the Cross-Language Evaluation Forum (CLEF 2010)*. Lecture Notes in Computer Science (LNCS) 6360, Springer, Heidelberg, Germany, 2010.

[7] G. Amati. *Probability Information Models for Retrieval based on Divergence from Randomness*. PhD thesis, 2003.

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

[9] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Has adhoc retrieval improved since 1994? In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 692–693, New York, NY, USA, 2009. ACM.

[10] M. Braschler. CLEF 2002 – Overview of Results. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Advances in Cross-Language Information Retrieval: Third Workshop of the Cross–Language Evaluation Forum (CLEF 2002) Revised Papers*, pages 9–27. Lecture Notes in Computer Science (LNCS) 2785, Springer, Heidelberg, Germany, 2003.

[11] M. Braschler. CLEF 2003 – Overview of Results. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Comparative Evaluation of Multilingual Information Access Systems: Fourth Workshop of the Cross–Language Evaluation Forum (CLEF 2003) Revised Selected Papers*, pages 44–63. Lecture Notes in Computer Science (LNCS) 3237, Springer, Heidelberg, Germany, 2004.

[12] C. Buckley and E. M. Voorhees. Retrieval System Evaluation. In D. K. Harman and E. M. Voorhees, editors, *TREC. Experiment and Evaluation in Information Retrieval*, pages 53–78. MIT Press, Cambridge (MA), USA, 2005.

[13] G. Cabanac, G. Hubert, M. Boughanem, and C. Chrisment. Tie-Breaking Bias: Effect of an Uncontrolled Parameter on Information Retrieval Evaluation. In Agosti et al. [6], pages 112–123.

[14] C. L. A. Clarke, N. Craswell, and H. Voorhees. Overview of the TREC 2012 Web Track. In E. M. Voorhees and L. P. Buckland, editors, *The Twenty-First Text REtrieval Conference Proceedings (TREC 2012)*, pages 1–8. National Institute of Standards and Technology (NIST), Special Publication 500-298, Washington, USA., 2013.

[15] G. M. Di Nunzio, N. Ferro, G. J. F. Jones, and C. Peters. CLEF 2005: Ad Hoc Track Overview. In C. Peters, F. C. Gey, J. Gonzalo, G. J. F. Jones, M. Kluck, B. Magnini, H. Müller, and M. de Rijke, editors, *Accessing Multilingual Information Repositories: Sixth Workshop of the Cross–Language Evaluation Forum (CLEF 2005). Revised Selected Papers*, pages 11–36. Lecture Notes in Computer

Science (LNCS) 4022, Springer, Heidelberg, Germany, 2006.

[16] G. M. Di Nunzio, N. Ferro, T. Mandl, and C. Peters. CLEF 2006: Ad Hoc Track Overview. In Peters et al. [25], pages 21–34.

[17] N. Ferro. Specification of the CIRCO Framework, Version 0.10. Technical Report IMS.2009.CIRCO.0.10, Department of Information Engineering, University of Padua, Italy, 2009.

[18] N. Ferro and D. Harman. CLEF 2009: Grid@CLEF Pilot Track Overview. In Peters et al. [26], pages 552–565.

[19] N. Ferro and C. Peters. CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. In Peters et al. [26], pages 13–35.

[20] N. Ferro and G. Silvello. Rank-Biased Precision Reloaded: Reproducibility and Generalization. In N. Fuhr, A. Rauber, G. Kazai, and A. Hanbury, editors, *Advances in Information Retrieval. Proc. 37th European Conference on IR Research (ECIR 2015)*, pages 768–780. Lecture Notes in Computer Science (LNCS) 9022, Springer, Heidelberg, Germany, 2015.

[21] A. Hanbury and H. Müller. Automated Component-Level Evaluation: Present and Future. In Agosti et al. [6], pages 124–135.

[22] K. Järvelin and J. Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, October 2002.

[23] R. Krovetz. Viewing Morphology as an Inference Process. In R. Korfhage, E. Rasmussen, and P. Willett, editors, *Proc. 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993)*, pages 191–202. ACM Press, New York, USA, 1993.

[24] V. Lavrenko and W. B. Croft. Relevance based language models. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, pages 120–127, 2001.

[25] C. Peters, P. Clough, F. C. Gey, J. Karlgren, B. Magnini, D. W. Oard, M. de Rijke, and M. Stempfhuber, editors. *Evaluation of Multilingual and Multi-modal Information Retrieval : Seventh Workshop of the Cross–Language Evaluation Forum (CLEF 2006). Revised Selected Papers*. Lecture Notes in Computer Science (LNCS) 4730, Springer, Heidelberg, Germany, 2007.

[26] C. Peters, G. M. Di Nunzio, M. Kurimo, T. Mandl, D. Mostefa, A. Peñas, and G. Roda, editors. *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, 2010.

[27] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[28] S. E. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1):95–108, January 2000.

[29] J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System. Experiments in Automatic Document Processing*, pages 313–323. Prentice–Hall, Inc., Englewood Cliff, New Jersey, USA, 1971.

[30] G. Salton and C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[31] A. Trotman, C. L. A. Clarke, I. Ounis, J. S. Culpepper, M.-A. Cartright, and S. Geva. Open Source Information Retrieval: a Report on the SIGIR 2012 Workshop. *ACM SIGIR Forum*, 46(2):95–101, December 2012.

[32] A. Trotman, A. Puurula, and B. Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, ADCS '14, pages 58:58–58:65, New York, NY, USA, 2014. ACM.

[33] E. M. Voorhees. Overview of the TREC 2004 Robust Track. In E. M. Voorhees and L. P. Buckland, editors, *The Thirteenth Text REtrieval Conference Proceedings (TREC 2004)*. National Institute of Standards and Technology (NIST), Special Pubblication 500-261, Whasington, USA. `http://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf` [last visited 2007, March 23], 2004.

[34] E. M. Voorhees and D. K. Harman. Overview of the Fifth Text REtrieval Conference (TREC-5). In E. M. Voorhees and D. K. Harman, editors, *The Fifth Text REtrieval Conference (TREC-5)* , pages 1–28. National Institute of Standards and Technology (NIST), Special Publication 500-238, Washington, USA., 1996.

[35] E. M. Voorhees and D. K. Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, pages 1–24. National Institute of Standards and Technology (NIST), Special Publication 500-240, Washington, USA., 1997.

[36] E. M. Voorhees and D. K. Harman. Overview of the Seventh Text REtrieval Conference (TREC-7). In E. M. Voorhees and D. K. Harman, editors, *The Seventh Text REtrieval Conference (TREC-7)*, pages 1–24. National Institute of Standards and Technology (NIST), Special Publication 500-242, Washington, USA., 1998.

[37] E. M. Voorhees and D. K. Harman. Overview of the Eigth Text REtrieval Conference (TREC-8). In E. M. Voorhees and D. K. Harman, editors, *The Eighth Text REtrieval Conference (TREC-8)*, pages 1–24. National Institute of Standards and Technology (NIST), Special Publication 500-246, Washington, USA., 1999.

[38] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition) – W3C Recommendation 26 November 2008. `http://www.w3.org/TR/xml/`, November 2008.

[39] 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.