Replicability and Reproducibility of Automatic Routing Runs

Timo Breuer and Philipp Schaer

TH Köln (University of Applied Sciences), 50678 Cologne, Germany
firstname.lastname@th-koeln.de

Abstract. This paper reports our participation in CENTRE@CLEF19. We focus on reimplementing submissions by Grossman and Cormack to the TREC 2017 Common Core Track. Our contributions are twofold. Reimplementations are used to study the replicability as well as the reproducibility of WCRobust04 and WCRobust0405. Our results show that the replicability and reproducibility of transferring relevance judgments across different corpora are limited. It is not possible to replicate or reproduce the baseline. However, improvements in evaluation measures by enriching training data are achievable. Further experiments examine general relevance transfer and the augmentation of tfidf-features.

Keywords: Relevance Transfer · Replicability · Reproducibility.

1 Introduction

Being able to reproduce the results of scientific experiments is essential for the validity of new findings. Especially in the field of computer science, it is desirable to ensure reproducible outcomes of complex systems. In 2018 the Association for Computing Machinery (ACM) introduced publication guidelines and procedures concerned with artifact review and badging[1]. According to these definitions, the terminology of repeatability, replicability, and reproducibility is coined as follows. While repeatability is limited to the reliable repetition of experiments with the same experimental setup conducted by the original researcher, replicability expands this scenario to the conduction by a different researcher. Reproducibility expands replicability by the use of another experimental setup.

In information retrieval (IR) research evaluation is a primary driver of manifesting innovation. In order to apply new IR systems to different datasets, reproducible evaluation outcomes have to be guaranteed. This requirement led to the advent of attempts like RIGOR [1], the Open-Source IR Reproducibility Challenge [5] and most recently the CENTRE lab which has been held in 2018

[1] https://www.acm.org/publications/policies/artifact-review-badging

at the CLEF conference for the first time [3][2]. Its second iteration CENTRE@-CLEF19 [2] is devoted to the replicability, reproducibility, and generalizibility of IR systems submitted to CLEF, NTCIR and TREC in previous years.

ACM badging and CENTRE terminologies do not entirely coincide. CENTRE defines replicability and reproducibility by the use of original or experimental test collections. In the following, we adhere to the definitions used in the context of CENTRE. We chose to participate in replicating and reproducing the automatic routing runs by Grossman & Cormack [4]. Thus we are obliged to the following two tasks:

> **Task 1 - Replicability:** The reimplemented system will replicate the runs WCRobust04 and WCRobust0405 by assessing the New York Times (NYT) corpus which has also been used in the original paper by Grossman & Cormack.
>
> **Task 2 - Reproducibility:** The reimplemented system will reproduce the runs WCRobust04 and WCRobust0405 by assessing the TREC Washington Post (WaPo) corpus.

The remainder of this paper is structured as follows. In Section 2 we outline the original runs WCRobust04 and WCRobust0405. Likewise, document collections will be introduced shortly. Section 3 will give insights into our implementation. Summaries of our results follow in Section 4. The paper ends with Section 5 which concludes our findings.

2 Automatic Routing Runs & Corpora

In the context of the TREC Common Core Track in 2017, Grossman and Cormack contributed the WaterlooCormack submissions. More specifically, we will focus on the runs WCRobust04 and WCRobust0405. Both run submissions follow the principle of automatic routing runs. For a given topic a logistic regression model will be trained on relevance judgments from one (or two) collection(s). Afterwards, the model predicts relevance assessments of documents from another collection. In contrast to other retrieval procedures, no explicit query is needed for ranking documents. Training and prediction are done on a topic-wise basis.

In order to train the model, text documents are transformed into a numerical representation with the help of tfidf-weights. The qrel files are based on ternary relevance judgments and will be converted to a binary scheme. In doing so, tfidf-features can be subdivided into two classes. Training is based on features of judged documents only. The likelihood of tfidf-representations being relevant will score documents. The complete corpus is ranked by score. The 10,000 highest-scoring documents form the ranking for a single topic.

[2] Note that there have been other iterations of CENTRE at TREC in 2018 and NTCIR in 2019

The tfidf-features are derived based on a union corpus which consolidates vocabulary from all corpora. Consequently, training features are augmented by the vocabulary of the corpus whose documents will be judged. Both runs assess documents from the NYT corpus. The two runs differ in the composition of the training set. While WCRobust04 is trained on features derived from documents of Robust04 only, WCRobust0405 enriches the training set by incorporating documents from Robust05. Table 1 gives an overview of run constellations.

Task	Run name	Corpus to be classified	Relevance judgments for training	Training data
Replicability	WCRobust04	New York Times	Robust Track 2004	TREC Disks 4&5
	WCRobust0405	New York Times	Robust Track 2004 & 2005	TREC Disks 4&5 + AQUAINT
Reproducibility	WCRobust04	Washington Post	Robust Track 2004	TREC Disks 4&5
	WCRobust0405	Washington Post	Robust Track 2004 & 2005	TREC Disks 4&5 + AQUAINT

Table 1. Overview of run constellations and their respective relevance judgments and corpora. Depending on the task, a different corpus will be classified.

The corpora used in the CENTRE lab contain documents from the news domain. Relevance judgments and documents are taken from corpora of the TREC Robust Track in 2004 [7] and 2005 [8]. Relevance will be assessed for the New York Times[3] and Washington Post[4] corpora. The Robust04 collection consists of documents from TREC Disks 4&5[5] (minus Congressional Record data). Articles range from the years 1989 to 1996 and add up to approximately 500,000 single documents. AQUAINT[6] is known as the test collection of Robust05. The document collection gathers articles from the years 1996 to 2000 and holds around one million single documents. TREC Disks 4&5 as well as the AQUAINT corpus consist of SGML-tagged text data. The New York Times corpus covers articles from over 20 years starting in 1987 up to the year 2007. On the whole, the corpus contains 1,8 million documents. The NYT corpus is formatted in News Industry Text Format (NITF)[7]. The TREC Washington Post corpus comprises news

[3] https://catalog.ldc.upenn.edu/LDC2008T19

[4] https://trec.nist.gov/data/wapost/

[5] https://trec.nist.gov/data/qa/T8_QAdata/disks4_5.html

[6] https://catalog.ldc.upenn.edu/LDC2002T31

[7] https://iptc.org/standards/nitf/

articles of a time span from January 2012 to August 2017. The initial version contains duplicate documents. After removing these, the corpus contains nearly 600,000 different articles. The Washington Post corpus is provided as JSON Lines[8] file. Both corpora served as a data basis for the TREC Common Core Tracks in 2017/18.

3 Implementation

As depicted in figure 1, our interpretation of the WaterlooCormack workflow can be subdivided into three processing steps. First of all, corpora data will be prepared, resulting in single documents containing normalized text. The next step consists of deriving tfidf-features from these documents in order to perform topic-wise training and prediction. The last step will evaluate the resulting run with the help of the respective qrels and TREC evaluation measures.

For our implementation, we chose to use Python. According to the premise of CENTRE, participants are obliged to use open source tools. The Python community offers a vast variety of open and free software, thus we had no problems in finding the required components of the workflow. In the following, more detailed insights into the processing steps of the workflow will be given.

3.1 Data preparation

Specific characteristics have to be considered when preparing data of four different collections. There are differences both in compression data formats and text formatting. This circumstance has to be kept in mind when trying to implement the workflow as generic as possible. Extraction of compressed corpora files is realized with GNU tools tar[9] and gzip[10]. Within this context, the different extensions of compressed files from the TREC Disks 4&5, AQUAINT and NYT corpora (.z, .0z, .1z, .2z, .gz, .tgz) have to be handled properly. We expect the routine to start with the extracted JSON Lines file of the Washington Post corpus. We use BeautifulSoup[11] in combination with lxml[12] for parsing raw text data from the formatted document files. Embeddings and URLs to external documents were removed. The raw text will be normalized by excluding punctuation, removing stop words, and stemming words in the respective order. For this purpose, we make use of nltk[13]. Originally, documents of two corpora have to be unified into one single corpus. However, our procedure deviates from this approach. The tfidf-weights are derived solely on the basis of the corpus, which provides tfidf-features for the training of the logistic regression model.

[8] http://jsonlines.org/

[9] https://www.gnu.org/software/tar/

[10] https://www.gnu.org/software/gzip/

[11] https://www.crummy.com/software/BeautifulSoup/

[12] https://lxml.de/

[13] https://www.nltk.org/

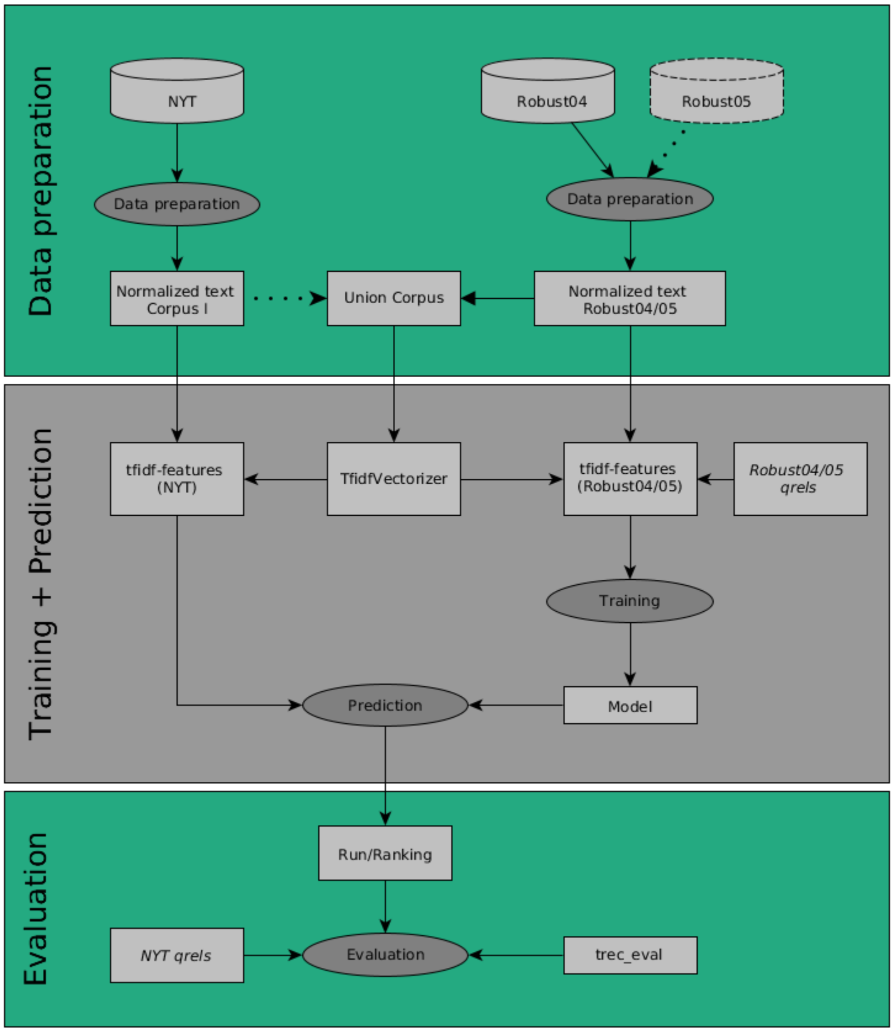

Fig. 1. Exemplary visualization of the workflow for the replication of WCRobust04 and WCRobust0405. Elliptical shapes represent processing steps and rectangular boxes their produced results. After data preparation, the TfidfVectorizer can be derived. Originally, this has to be done with a unified corpus consisting of NYT and Robust04/05 documents. Our approach deviates from this procedure, which is indicated by the dotted arrow. We derive the TfidfVectorizer solely based on Robust04/05 documents. Training data for two classes can be acquired with the help of qrel files from Robust04/05. The training step will result in a logistic regression model, which is adapted for a specific topic. Tfidf-features of the NYT corpus will be classified with this model during the prediction step. The run will be evaluated by using trec_eval in combination with the NYT qrels.

That means tfidf-features will not be augmented by the vocabulary of the corpus whose documents will be ranked. We choose this approach with respect to the results reported in 4.2

3.2 Training & Prediction

Our implementation of the training and prediction routines mainly relies on the scikit-learn package [6]. More specifically we make use of the TfidfVectorizer and the LogisticRegression classifier. As explained earlier, training and prediction will be conducted topic-wise. For both steps, a tfidf-representation of documents is required. In order to convert text documents into numerical vectors, we construct the TfidfVectorizer based on Robust04/05 documents (depending on the specific run). Yu et al. [9] pay special attention to the importance of L_2-normalization of feature vectors. The TfidfVectorizer uses the L_2-norm as a default setting. Training features will be stored on disk in SVMlight format to ensure compatibility with other machine learning frameworks. Depending on the corpora constellations, there are deviating numbers of topics for which the logistic regression classifier can be trained and used for classification. Only those topics, which are judged for the test collection as well, can be used for the training of a model. Using NYT in combination with Robust04, for instance, results in a subset of 50 intersecting topics which are judged for both corpora. Combining NYT with Robust05 gives a subset of 33 intersecting topics. For each intersecting topic of the test and training corpus, a ranking with 10,000 entries will be determined.

3.3 Evaluation

The evaluation will be done by the use of trec_eval. Besides the ranking from the previous step qrels of the corpus to be assessed have to be provided. Evaluation measures are reported in the next section.

3.4 Miscellanea

Our code contributions also incorporate other machine learning models. Originally WaterlooCormack runs were computed by the use of Sofia-ML[14]. We tried to integrate Sofia-ML in our workflow but were not able to report any experimental results due to hardware limitations. Using the CLI of Sofia-ML, predictions are done with SVMlight formatted features. Providing the tfidf-features of the entire corpus to Sofia-ML was not possible for us, since we ran out of memory on our 16GB laptop machine. Providing tfidf-features separately as single files to the CLI prolonged the classification routine to unreasonable processing times. Likewise, the use of SVM models from the scikit-learn library resulted in longer processing times. The interfaces of the models are identical and code integration was possible with little effort. However, due to the more compute-intensive nature of SVMs the processing time of a single prediction nearly multiplied by the factor of ten.

[14] `https://code.google.com/archive/p/sofia-ml/`

4 Experimental Results

Based on the workflow described in the previous section, we evaluate different combinations of test and training corpora in order to assess the characteristics of the procedure and underlying data. In section 4.1 we try out all corpora combinations beyond the envisaged constellations of WCRobust04 and WCRobust0405. In section 4.2 we investigate the necessity of augmenting training data. Section 4.3 has a special focus on the replicability and reproducibility of the WaterlooCormack runs. In this context, we have a look at the benefits of preprocessing text data before deriving tfidf-features.

4.1 Relevance transfer

Having four different corpora at hand (TREC Disk 4&5, AQUAINT, NYT, WaPo) we produce runs for all possible corpora combinations. Table 2 shows results of all simple combinations. Whereas 'simple' refers to using only one corpus for the training step and omitting the enrichment of tfidf-features by the vocabulary of the test corpus. Figure 2 shows the MAP values in decreasing order. Classifying NYT documents by relevance judgments from the Robust corpora results in the two highest MAP values. However, the reported MAP values cannot be compared directly due to the deviating number of intersecting topics across different combinations.

Test	Training	Topics	MAP	P@10
NYT	Robust04	50	0.2963	0.6860
	Robust05	33	0.3019	0.7212
	WaPo	25	0.1684	0.5120
Robust04	NYT	50	0.1183	0.2560
	Robust05	50	0.1797	0.4160
	WaPo	25	0.1068	0.3400
Robust05	NYT	33	0.1629	0.3455
	Robust04	50	0.1913	0.4360
	WaPo	15	0.1430	0.3733
WaPo	NYT	25	0.1058	0.3000
	Robust04	25	0.1373	0.3200
	Robust05	15	0.1789	0.4333

Table 2. Transferring relevance judgments across different corpora combinations

4.2 Feature augmentation

Originally tfidf-features are derived from the union corpus. That implies tfidf-weights will be determined by the vocabulary of the training and test corpus.

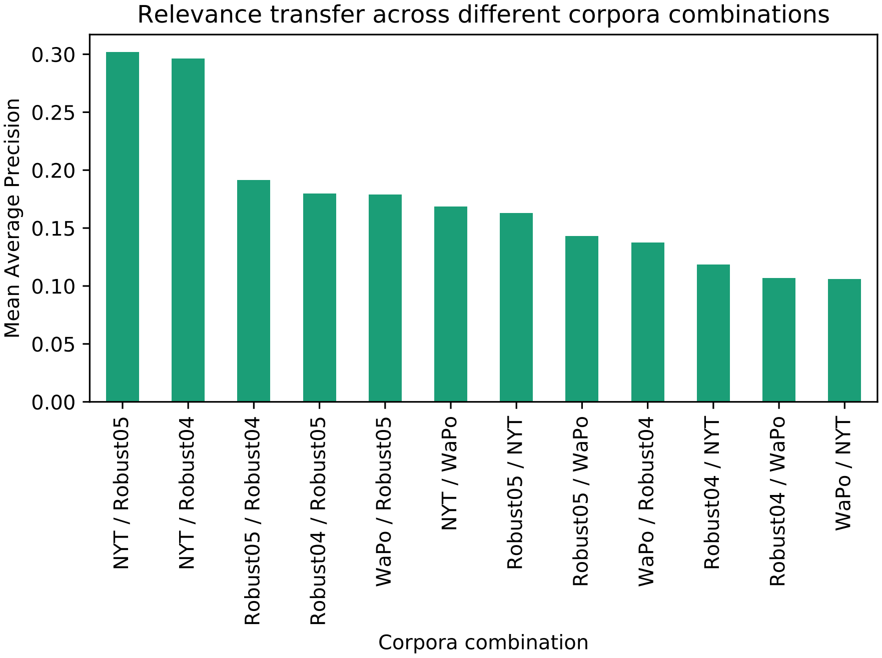

Fig. 2. MAP values for different corpora combinations beyond the envisaged training routine of the WaterlooCormack runs. The first corpus being labeled is the test corpus, whereas the second represents the training data. Direct comparison is not advised due to diverging numbers of intersecting topics. However, it can be seen, that classifying the NYT corpora with a model trained on Robust corpora results in the highest MAP values.

In their contribution to the reproducibility track of ECIR 2019 Yu et al. consider augmenting tfidf-features in this manner to be negligible, thus facilitating generalizibility [9]. Even though this assumption is reasonable, the authors do not provide evidence. The following setup compares different corpora combinations in two variants. The first variant produces runs based on training with tfidf-features derived exclusively from the training corpus. The second variant is based on training features that are augmented by the vocabulary of the corpus to be classified. Numerical representations of documents will contain more tfidf-features, and less out-of-vocabulary terms during prediction should occur. This variant complies with the procedure proposed originally for the WaterlooCormack runs. Table 3 reports evaluation results of these runs. For none of the reported combinations there are significant differences when augmenting training data. For instance, classifying NYT with training data from Robust04 results in a MAP value of 0.2963. Augmenting the training data with the NYT vocabulary results in a MAP value of 0.2924. Due to these findings, we omit augmenting training data for our final runs.

Test	Training	Topics	MAP	P@10
NYT	Robust04	50	0.2963	0.6860
	NYT+Robust04	50	0.2924	0.6660
	Robust0405	33	0.3751	0.7455
	NYT+Robust0405	33	0.3715	0.7364
Robust04	Robust05	50	0.1797	0.4160
	Robust0405	50	0.1766	0.4160
Robust05	Robust04	50	0.1913	0.4360
	Robust0405	50	0.1938	0.4320
WaPo	Robust04	25	0.1373	0.3200
	WaPo+Robust04	25	0.1360	0.3120
	Robust0405	15	0.1987	0.4333
	WaPo+Robust0405	15	0.1935	0.4200

Table 3. Feature augmentation for different corpora constellations. The first variant uses the training corpus only for deriving tfidf-weights. The second variant embodies the vocabulary of the test corpus for deriving tfidf-weights.

4.3 Replicability and Reproducibility of WCRobust04 & WCRobust0405

Table 4 reports evaluation measures of the replicated and reproduced WaterlooCormack runs. All reported MAP values stay below the baseline reported by Grossman and Cormack [4]. P@10 values of replicated runs stay slightly below those given by the original paper. For each run constellation results without our preprocessing pipeline are added. Especially WCRobust04 profits from our preprocessing proposal.

Grossman and Cormack retrieve better results when enriching training data by an additional corpus. As explained earlier, the union corpus consists of documents and relevance judgments from Robust04/05 corpora. The improvement of evaluation measures is also valid for both our replicated and reproduced results. Table 5 shows the same evaluation measures based on 15 intersecting topics across all corpora for a better comparison of both tasks. Reproduced runs yield lower measures. Figure 3 and 4 show bar plots for each of the 15 topics resulting from replication and reproduction, respectively. Improvements by enriching training data are more consistent across topics of replicated runs. 14 out of 15 topics profit from training data enrichment. Evaluation measures of reproduced runs are generally lower and fewer topics profit from training data enrichment (with regards to our sample of 15 topics).

Test	Training	Preprocessing	Topics	MAP	P@10
Baseline [4]	Robust04	-	50	0.3711	0.6460
	Robust0405	-	33	0.4307	0.7788
NYT	Robust04	yes	50	0.2963	0.6860
		no	50	0.2671	0.6380
	Robust0405	yes	33	0.3751	0.7455
		no	33	0.3784	0.7455
WaPo	Robust04	yes	25	0.1373	0.3200
		no	25	0.1003	0.2600
	Robust0405	yes	15	0.1987	0.4333
		no	15	0.2142	0.4333

Table 4. Evaluation measures of replicated and reproduced runs based on all intersecting topics for each specific corpora combination. Outcomes are compared against the baseline reported by Grossman and Cormack [4]. None of the replicated or reproduced runs can reach the baseline in terms of MAP. P@10 of WCRobust04 slightly beats the baseline. Improved measures confirm our preprocessing proposal.

Test	Training	MAP	P@10
NYT	Robust04	0.2648	0.6067
	Robust0405	0.3788	0.7133
WaPo	Robust04	0.1409	0.2933
	Robust0405	0.1987	0.4333

Table 5. Evaluation measures of replicated and reproduced runs based on 15 intersecting topics

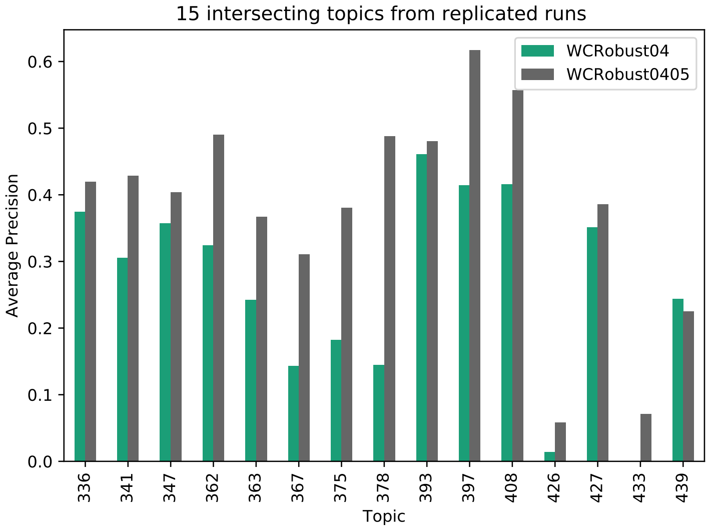

Fig. 3. Resulting AP values of the replicated WaterlooCormack runs for each of the 15 intersecting topics.

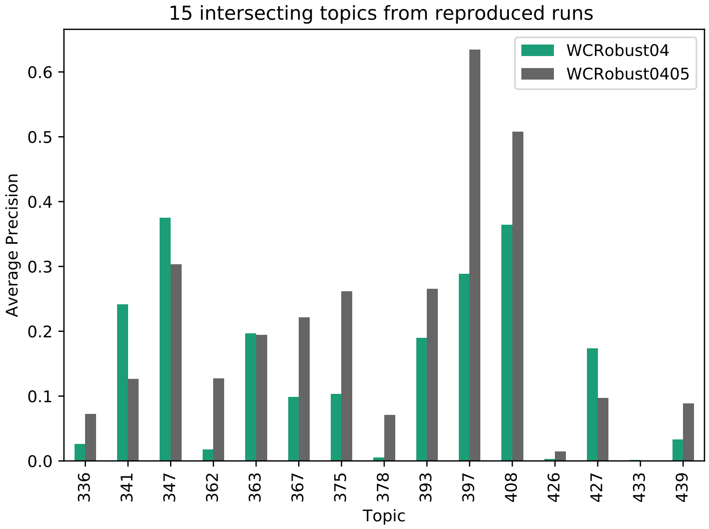

Fig. 4. Resulting AP values of the reproduced WaterlooCormack runs for each of the 15 intersecting topics.

Complementing WCRobust0405 Concerning WCRobust0405, Grossman and Cormack also report MAP and P@10 values based on 50 topics. Our previous setups derive rankings for WCRobust0405 based on 33 topics (replicability) and 15 topics (reproducibility). In our case, topic classifiers are trained on intersecting topics only, i.e., there are 33 intersecting topics between NYT and the Robust corpora and 15 intersecting topics between WaPo and the Robust corpora. With regard to the remaining topics, no details were given in the original paper. For this reason, we chose to investigate solely intersecting topics for WCRobust0405. After contacting Cormack, we came to know that for these topics, training data is taken where available. That means, when training data is only available from Robust04, the classifier will be trained with documents from one corpus only. The resulting rankings should be comparable to those from WCRobust04. Given this information, we retrieved more complete runs, which are shown in table 6.

Task	Run	Topics	MAP	P@10
Replicability	WCRobust04	50	0.2963	0.6860
	WCRobust0405	50	0.3534	0.7340
Reproducibility	WCRobust04	25	0.1373	0.3200
	WCRobust0405	25	0.1708	0.4000

Table 6. Evaluation outcomes of WCRobust04 and WCRobust0405 with equal number of topics. Depending on the topic, training data might be derived from Robust04 documents only.

Further considerations Even though the workflow proposed by Grossman and Cormack is intuitive, its description is only one paragraph long in the original paper. As we were reimplementing the workflow, many details had to be considered, which were not explicitly mentioned by the authors. For instance, our text preprocessing improved evaluation measures, but no details about such a processing step are given in the original paper. So, it is possible that there are still hidden details that are not covered by our reimplementation. Furthermore, the implementations of the logistic regression classifier by Sofia-ML and scikit-learn may differ.

Reflecting on decreasing scores of reproduced runs, it is worth considering the data basis of both replicated and reproduced runs. Replicated runs rank New York Times articles which cover a period from 1987 to 2007. The Robust corpora, used for training, contain articles that fall into this period (1989 to 2000). Opposed to this, the Washington Post collection contains more recent news articles from the years 2012 to 2017. News articles are subject to a strong time dependency, and topic coverage varies over time. This influence may affect the choice of words and consequently the vocabulary. News article collections covering the same years may be more likely to share larger amounts of the same vocabulary, which is beneficial for the reimplemented procedure based on tfidf-features.

5 Conclusion

Our participation in CENTRE@CLEF19 is motivated by replicating and reproducing automatic routing runs proposed by Grossman and Cormack [4]. For the replicability task, the New York Times corpus is used, whereas the reproducility task applies the procedures to the Washington Post corpus.

We provide a schematic overview of how we interpret the workflow description of the WaterlooCormack submissions by Grossman and Cormack. The underlying implementation is based on Python and available open source extensions.

Our experimental setups include assessments of general relevance transfer, tfidf-feature augmentation and the replicability and reproducibility of the WaterlooCormack runs. Outcomes of relevance transfer vary across corpora combinations. Ranking the New York Times corpus with the help of relevance judgments and documents from Robust corpora yields the best MAP values. Augmenting tfidf-features by the vocabulary of the corpus to be ranked is originally intended for the WaterlooCormack runs. A further setup investigates the necessity of feature augmentation. Our results conform with the assumptions by Yu et al. [9]. Augmenting tfidf-features is negligible.

We were not able to fully replicate or reproduce the baseline given by Grossman and Cormack. All MAP values stay below the baseline. P@10 values of replicated runs differ only slightly from the baseline. Our replicated results are comparable to the *classification only* approach by Yu et al. Due to missing details in the original paper, we contacted Cormack concerning WCRobust0405 and were able to complement runs which were initially limited to rankings of intersecting topics only.

Reproduced runs generally perform worse. This might be a starting point for future investigations. General corpora characteristics could be assessed by quantitative and qualitative analysis. These findings might be related to diverging evaluation measures. Likewise, it is possible to exchange the logistic regression model by more sophisticated approaches. Our code contributions provide possibilities for using other models and frameworks. Especially Python implementations should be easily integrable. The source code is available at `https://bitbucket.org/centre_eval/c2019_irc/`.

References

1. Arguello, J., Crane, M., Diaz, F., Lin, J., and Trotman, A. Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). *SIGIR Forum 49*, 2 (Jan. 2016), 107–116.
2. Ferro, N., Fuhr, N., Maistro, M., Sakai, T., and Soboroff, I. CENTRE@CLEF 2019. In *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part II* (2019), L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and

D. Hiemstra, Eds., vol. 11438 of *Lecture Notes in Computer Science*, Springer, pp. 283–290.

3. Ferro, N., Maistro, M., Sakai, T., and Soboroff, I. Overview of CENTRE@CLEF 2018: A First Tale in the Systematic Reproducibility Realm. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings* (2018), P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J. Nie, L. Soulier, E. SanJuan, L. Cappellato, and N. Ferro, Eds., vol. 11018 of *Lecture Notes in Computer Science*, Springer, pp. 239–246.
4. Grossman, M. R., and Cormack, G. V. MRG_UWaterloo and WaterlooCormack Participation in the TREC 2017 Common Core Track. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017* (2017), E. M. Voorhees and A. Ellis, Eds., vol. Special Publication 500-324, National Institute of Standards and Technology (NIST).
5. Lin, J. J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., MacDonald, C., and Vigna, S. Toward reproducible baselines: The open-source IR reproducibility challenge. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings* (2016), N. Ferro, F. Crestani, M. Moens, J. Mothe, F. Silvestri, G. M. D. Nunzio, C. Hauff, and G. Silvello, Eds., vol. 9626 of *Lecture Notes in Computer Science*, Springer, pp. 408–420.
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.
7. Voorhees, E. M. Overview of the TREC 2004 Robust Track. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004* (2004), E. M. Voorhees and L. P. Buckland, Eds., vol. Special Publication 500-261, National Institute of Standards and Technology (NIST).
8. Voorhees, E. M. Overview of the TREC 2005 Robust Retrieval Track. In *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, USA, November 15-18, 2005* (2005), E. M. Voorhees and L. P. Buckland, Eds., vol. Special Publication 500-266, National Institute of Standards and Technology (NIST).
9. Yu, R., Xie, Y., and Lin, J. Simple Techniques for Cross-Collection Relevance Feedback. In *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I* (2019), L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds., vol. 11437 of *Lecture Notes in Computer Science*, Springer, pp. 397–409.