Replicability and Reproducibility of Automatic Routing Runs

CENTRE@CLEF 2019, 10th September 2019, Lugano, Switzerland

T.Breuer and P. Schaer
Version: 2019-09-03

Technology
Arts Sciences
TH Köln

1. Automatic Routing Runs

Automatic Routing Runs

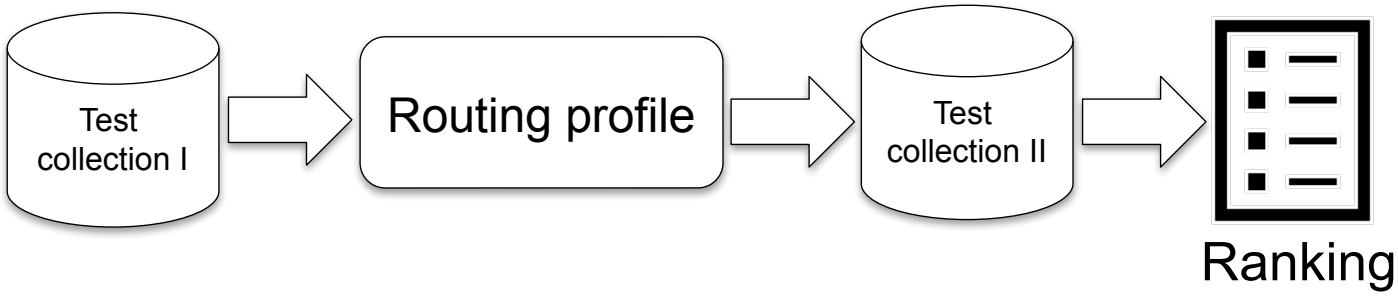
- Submission to TREC **Common Core 2017**
  by Grossman and Cormack

- **WCRobust04** and **WCRobust0405**
  rank **New York Times Annotated Corpus**

- (Automatic) **routing runs**
  Derive ranking profile from one test collection and use it to rank documents from another collection.

Automatic Routing Runs

- (Automatic) **routing runs**
  Derive ranking profile from one test collection and use it to rank documents from another collection.



Automatic Routing Runs

- Ranking profile is implemented as **logistic regression** classifier

- Training data fetched from Robust04/05 corpora
  - Numerical representation with **tfidf-weights**
  - Class assignment with **binarized relevance judgments**

- Documents are scored by their likelihood of being relevant

Automatic Routing Runs

- No explicit query
  (in contrast to conventional ad hoc rankings)

- Topic-wise training of classifiers

- 10,000 first documents form ranking for a single topic

2. Approach & Implementation

Approach & Implementation

- Replicability

  New York Times Annotated Corpus

- Reproducibility

  TREC Washington Post Corpus

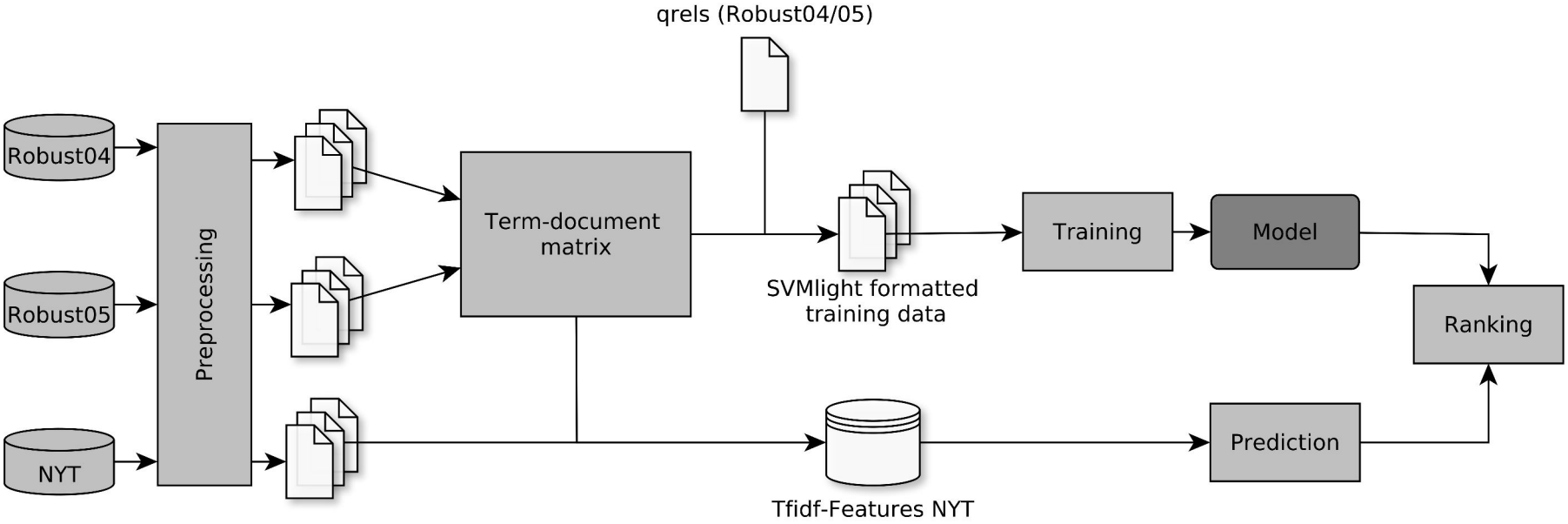
Approach & Implementation

Task	Run	Corpus	Training data
Replicability	WCRobust04	New York Times Annotated Corpus	Robust04
	WCRobust0405		Robust04/05
Reproducibility	WCRobust04	TREC Washington Post Corpus	Robust04
	WCRobust0405		Robust04/05

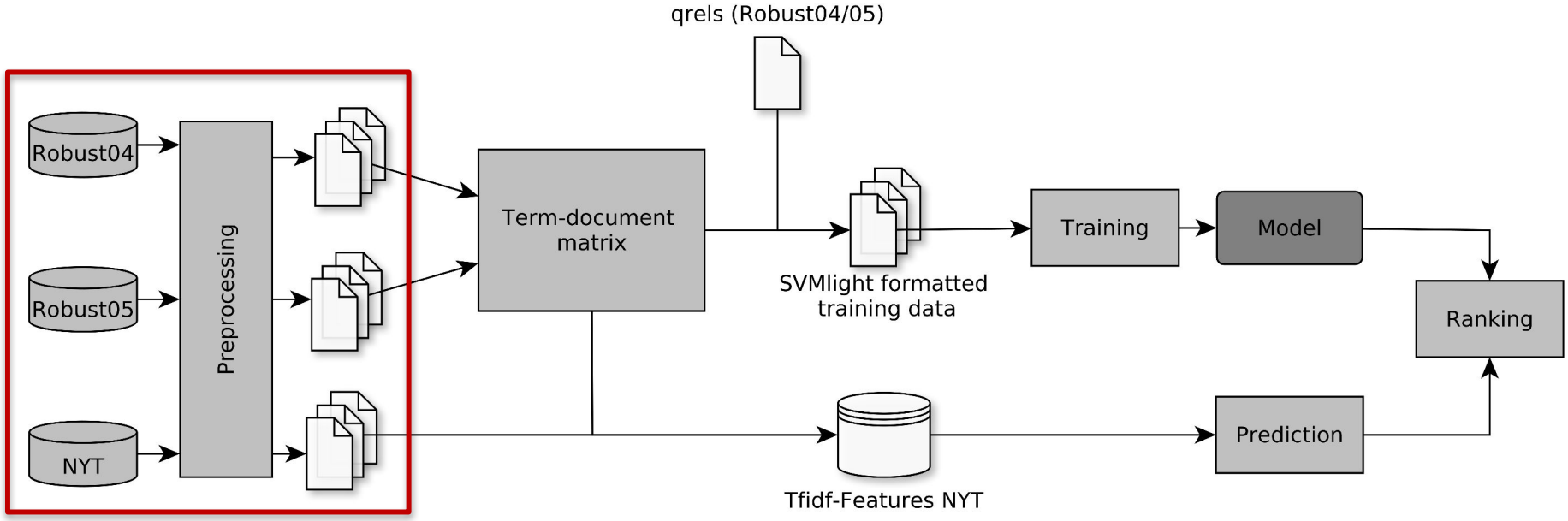
Approach & Implementation

- Implementation based on **Python3**
- scikit-learn: **TfidfVectorizer**, **LogisticRegression** classifier
- Training data: **SVMlight** format

Approach & Implementation

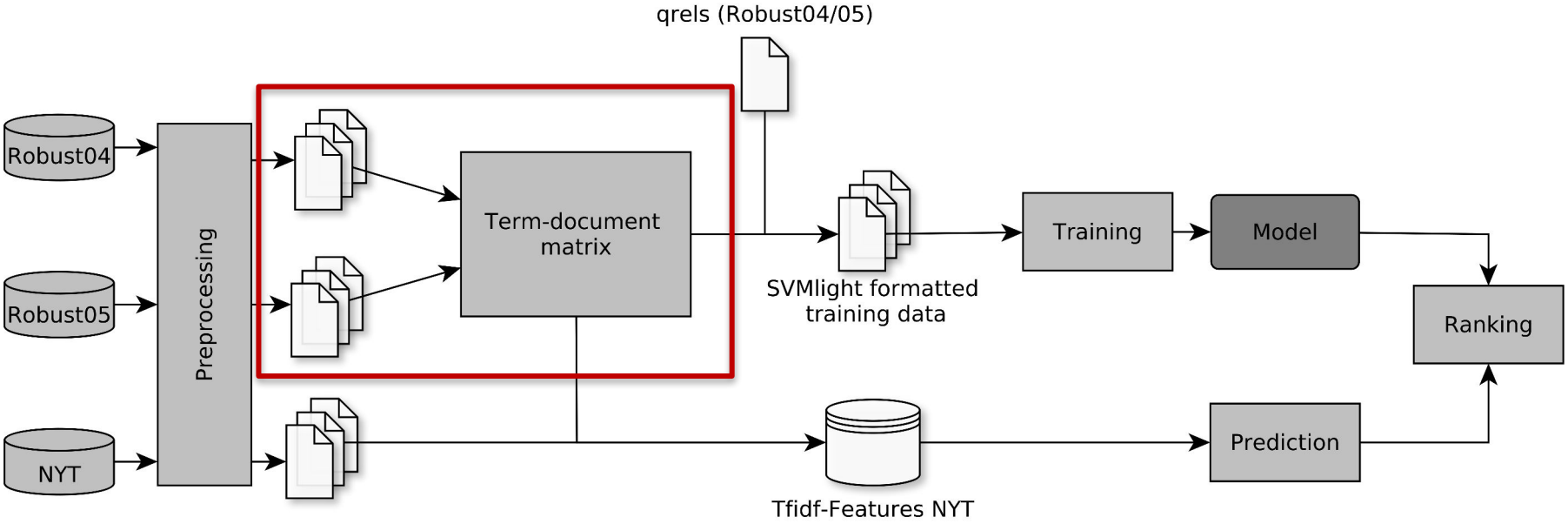


Approach & Implementation



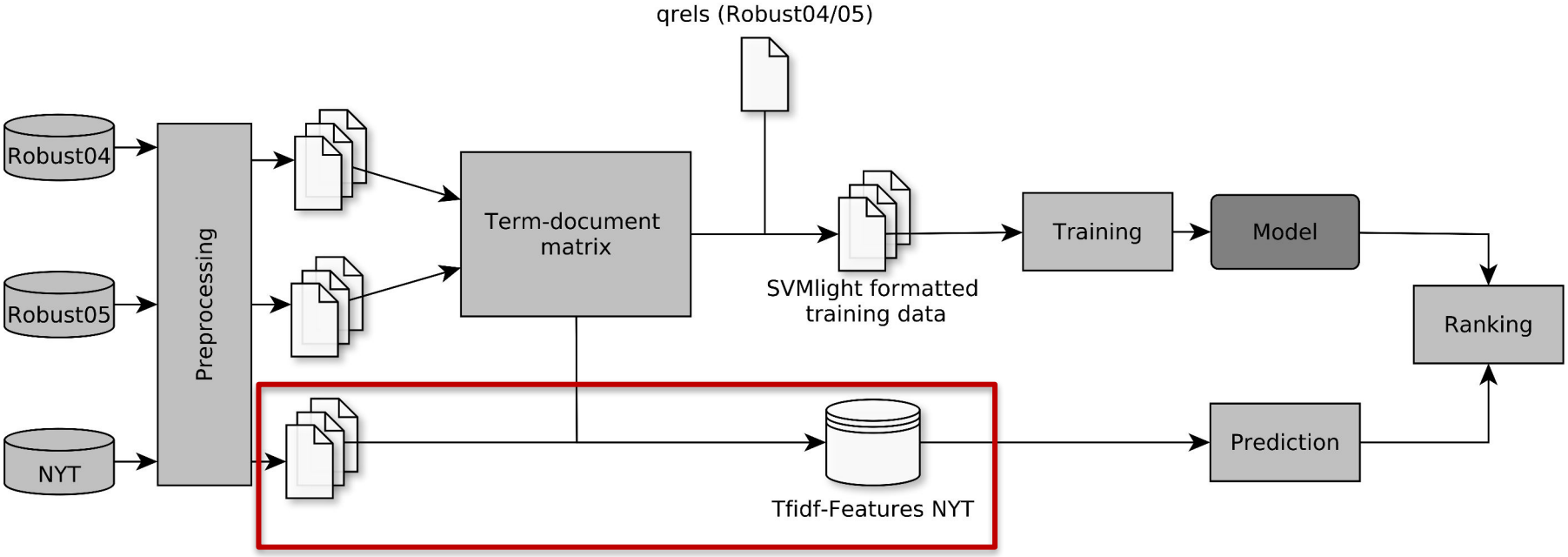
Text preprocessing

Approach & Implementation



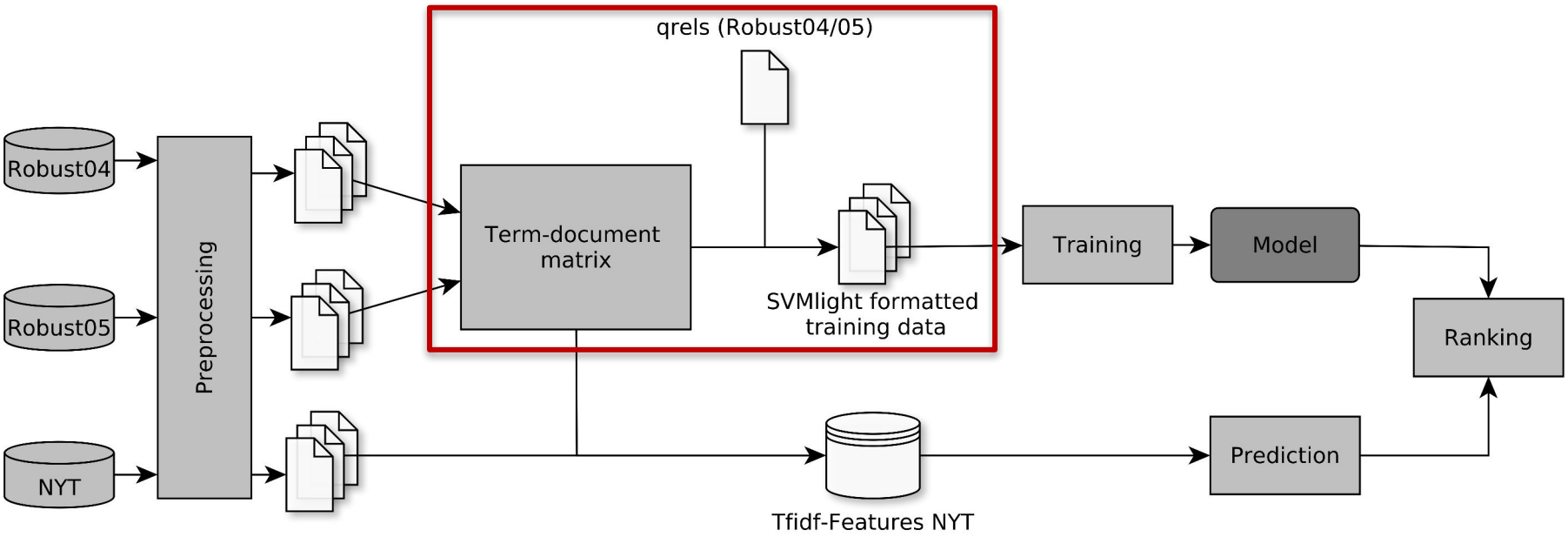
Derive tfidf-weights

Approach & Implementation



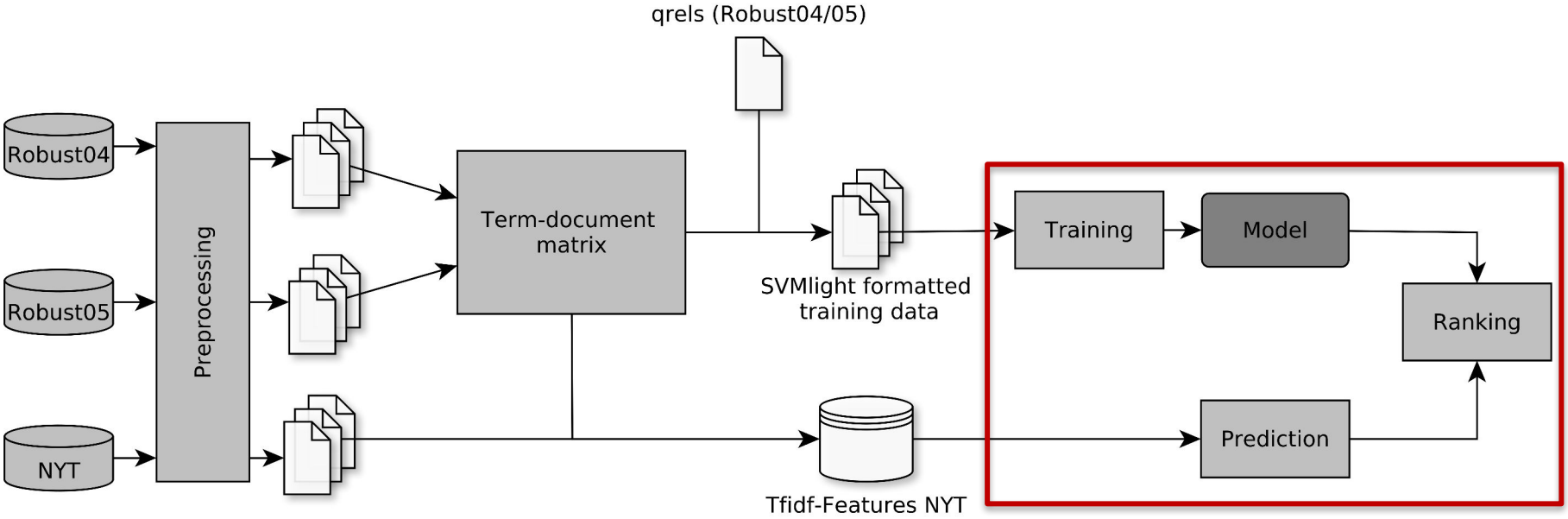
Derive corpus feature

Approach & Implementation



Prepare training data

Approach & Implementation



Train & predict

3. Experimental Results

Experimental results

- **Relevance transfer** across different corpora 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

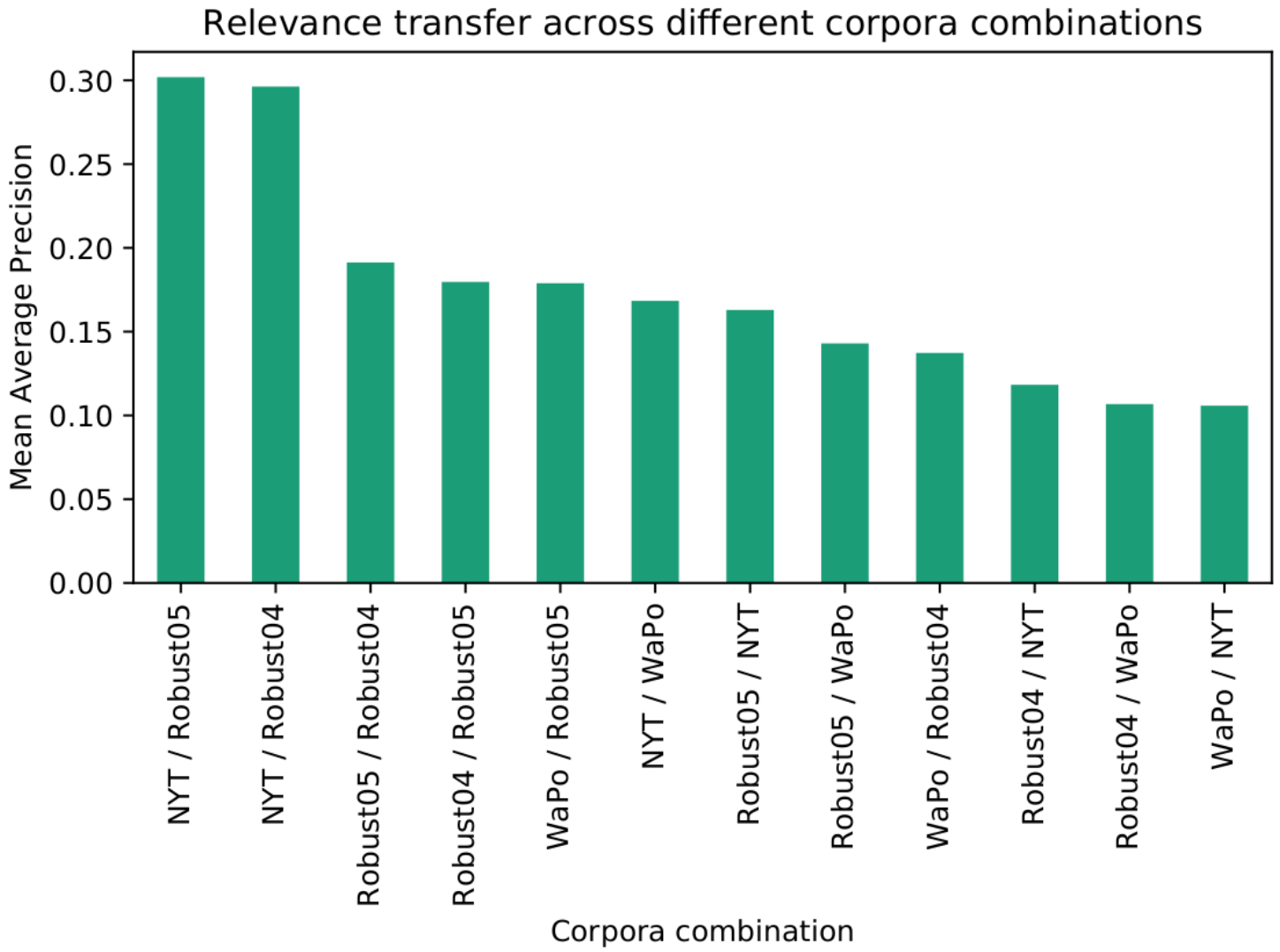
Experimental results

- **Relevance transfer** across different corpora 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

Using Robust corpora as training data for NYT yields best results (**but** consider different number of topics).

Experimental results



Experimental results

- **Feature augmentation** for different corpora combinations

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

Experimental results

- **Feature augmentation** for different corpora combinations

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

Little or no difference when including vocabulary of the test collection to the training data

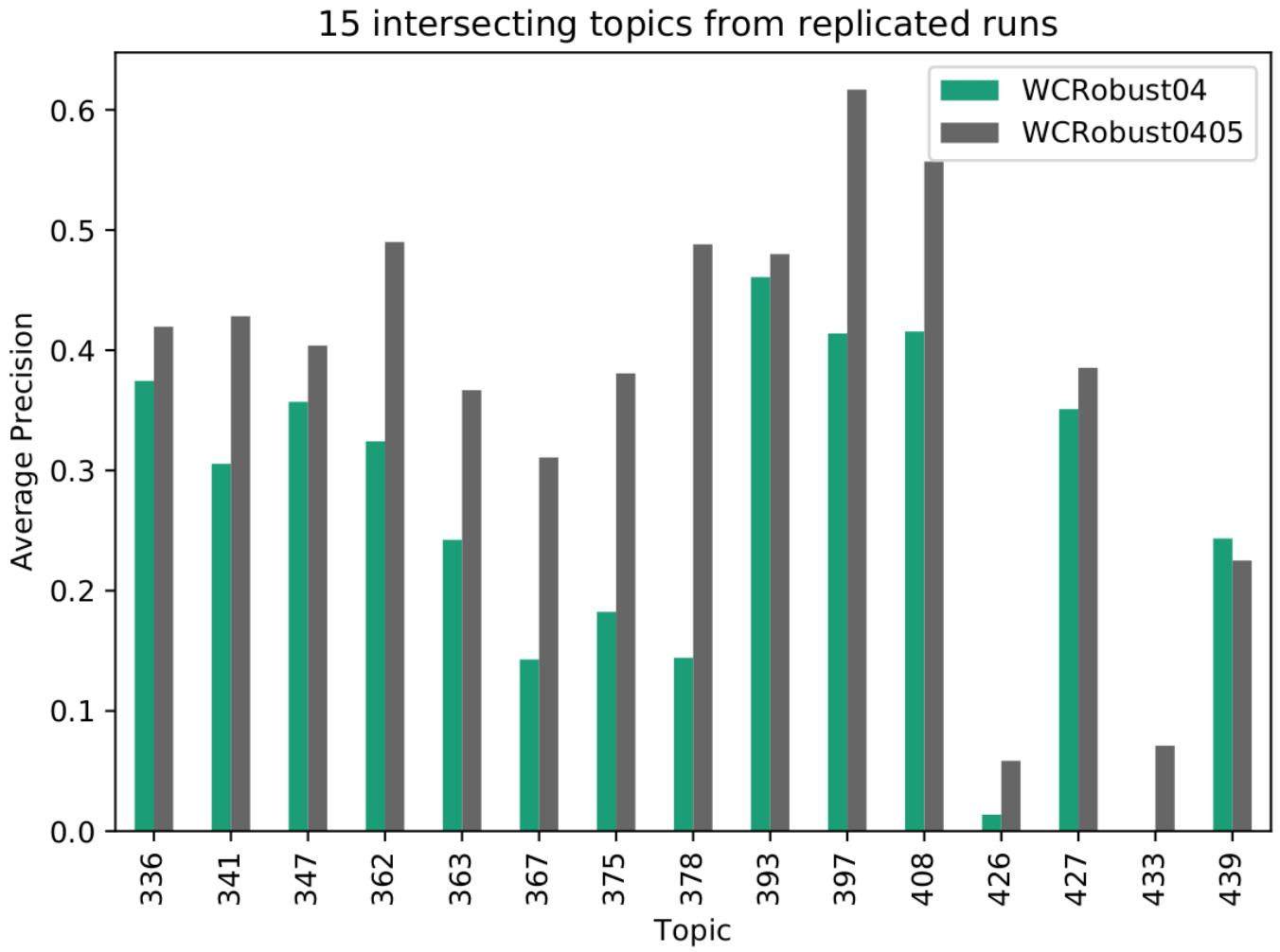
Experimental results

- **Replicated** and **reproduced** outcomes

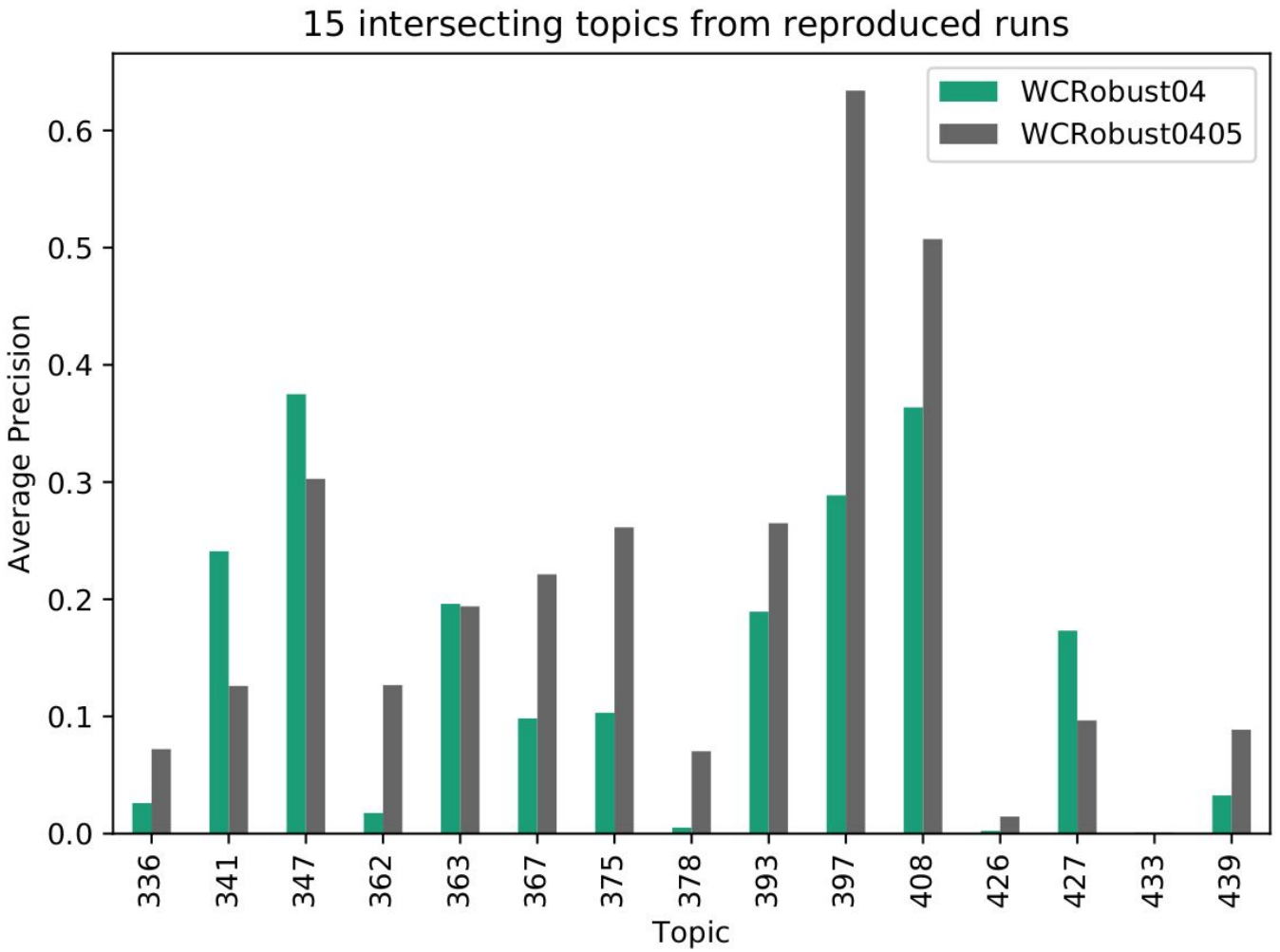
Test	Training	Preprocessing	Topics	MAP	P@10
Baseline [4]	Robust04	-	50	0.3711	0.6460
	Robust0405	-	33	0.4307	0.7788
NYT	Robust04	yes	50	0.2963	0.6860
		no	50	0.2671	0.6380
	Robust0405	yes	33	0.3751	0.7455
		no	33	0.3784	0.7455
WaPo	Robust04	yes	25	0.1373	0.3200
		no	25	0.1003	0.2600
	Robust0405	yes	15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

WCRobust04 WCRobust0405

Experimental results



Experimental results

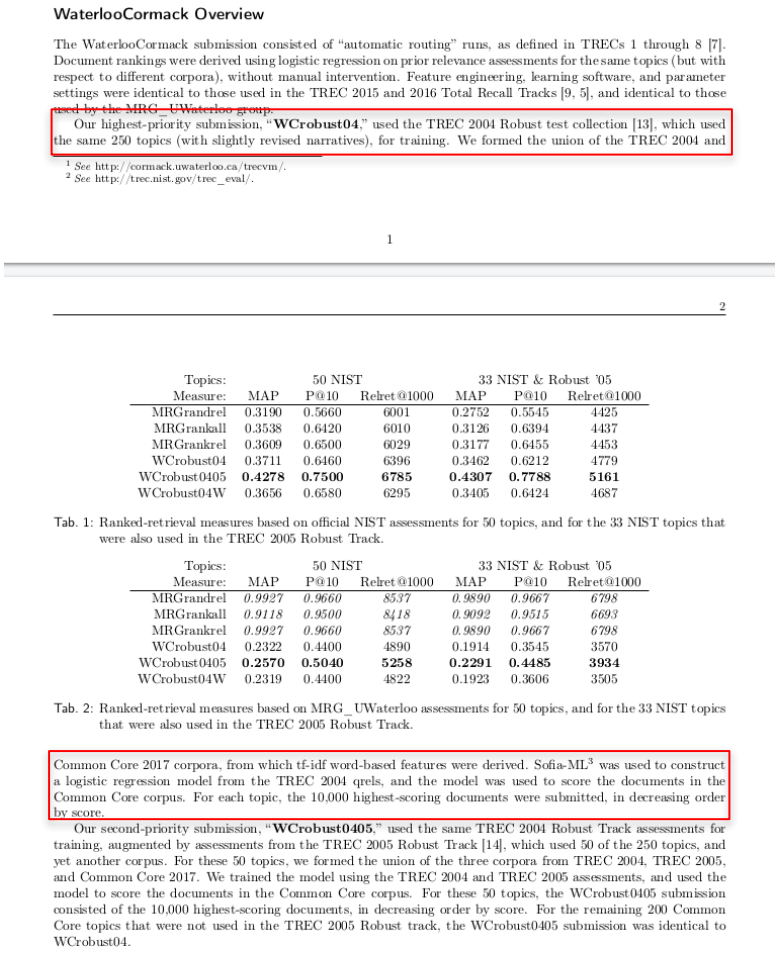


4. Obscurities & Future Work

Obscurities

- Description in original paper is approx. one paragraph long!

- No information about:
  - Text pre-processing
  - Other hidden details…
  - Formula of tfidf weights?
  - WCRobust0405:
    Training data for some topics is not available.



WaterlooCormack Overview

The WaterlooCormack submission consisted of "automatic routing" runs, as defined in TRECs 1 through 8 [7]. Document rankings were derived using logistic regression on prior relevance assessments for the same topics (but with respect to different corpora), without manual intervention. Feature engineering, learning software, and parameter settings were identical to those used in the TREC 2015 and 2016 Total Recall Tracks [9, 5], and identical to those used by the MRG_UWaterloo group.

Our highest-priority submission, "**WCrobust04**," used the TREC 2004 Robust test collection [13], which used the same 250 topics (with slightly revised narratives), for training. We formed the union of the TREC 2004 and

[1] *See* http://cormack.uwaterloo.ca/trecvm/.
[2] *See* http://trec.nist.gov/trec_eval/.

1

2

Topics:	50 NIST			33 NIST & Robust '05		
Measure:	MAP	P@10	Relret@1000	MAP	P@10	Relret@1000
MRGrandrel	0.3190	0.5660	6001	0.2752	0.5545	4425
MRGrankall	0.3538	0.6420	6010	0.3126	0.6394	4437
MRGrankrel	0.3609	0.6500	6029	0.3177	0.6455	4453
WCrobust04	0.3711	0.6460	6396	0.3462	0.6212	4779
WCrobust0405	**0.4278**	**0.7500**	**6785**	**0.4307**	**0.7788**	**5161**
WCrobust04W	0.3656	0.6580	6295	0.3405	0.6424	4687

Tab. 1: Ranked-retrieval measures based on official NIST assessments for 50 topics, and for the 33 NIST topics that were also used in the TREC 2005 Robust Track.

Topics:	50 NIST			33 NIST & Robust '05		
Measure:	MAP	P@10	Relret@1000	MAP	P@10	Relret@1000
MRGrandrel	*0.9927*	*0.9660*	*8537*	*0.9890*	*0.9667*	*6798*
MRGrankall	*0.9118*	*0.9500*	*8418*	*0.9092*	*0.9515*	*6693*
MRGrankrel	*0.9927*	*0.9660*	*8537*	*0.9890*	*0.9667*	*6798*
WCrobust04	0.2322	0.4400	4890	0.1914	0.3545	3570
WCrobust0405	**0.2570**	**0.5040**	**5258**	**0.2291**	**0.4485**	**3934**
WCrobust04W	0.2319	0.4400	4822	0.1923	0.3606	3505

Tab. 2: Ranked-retrieval measures based on MRG_UWaterloo assessments for 50 topics, and for the 33 NIST topics that were also used in the TREC 2005 Robust Track.

Common Core 2017 corpora, from which tf-idf word-based features were derived. Sofia-ML[3] was used to construct a logistic regression model from the TREC 2004 qrels, and the model was used to score the documents in the Common Core corpus. For each topic, the 10,000 highest-scoring documents were submitted, in decreasing order by score.

Our second-priority submission, "**WCrobust0405**," used the same TREC 2004 Robust Track assessments for training, augmented by assessments from the TREC 2005 Robust Track [14], which used 50 of the 250 topics, and yet another corpus. For these 50 topics, we formed the union of the three corpora from TREC 2004, TREC 2005, and Common Core 2017. We trained the model using the TREC 2004 and TREC 2005 assessments, and used the model to score the documents in the Common Core corpus. For these 50 topics, the WCrobust0405 submission consisted of the 10,000 highest-scoring documents, in decreasing order by score. For the remaining 200 Common Core topics that were not used in the TREC 2005 Robust track, the WCrobust0405 submission was identical to WCrobust04.

Experimental results

- Missing topics for WCRobust0405?

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

Future work

- Investigate on **time dependency** of news test collection:
  - Robust corpora: 1989 - 2000
  - New York Times Annotated Corpus: 1987 – 2007
  - TREC Washington Post Corpus: 2012 - 2017

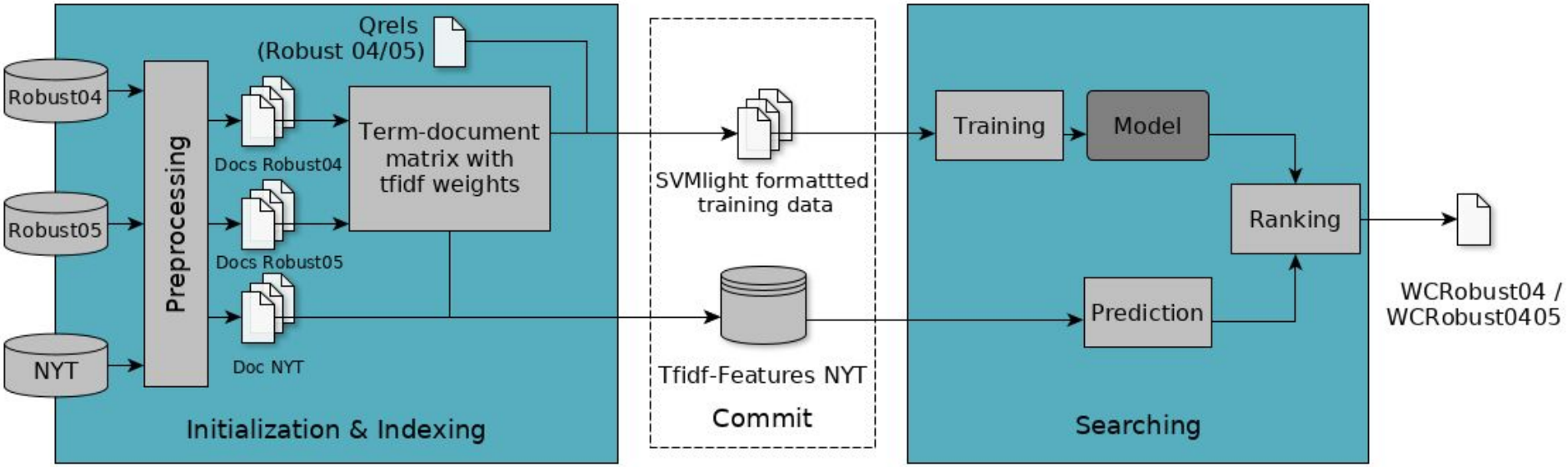
- Other classification models:
  - SVMlight compatible implementations should be easy to integrate.

5. Open-Source IR Replicability Challenge (OSIRRC) at SIGIR 2019

OSSIRC

- Workshop dedicated to the replicability of ad hoc retrieval experiments
- Docker-based framework
- Encapsulation of retrieval systems and models in Docker images
- Contribution of IRC-CENTRE2019 to image library

OSSIRC


Qrels
(Robust 04/05)
Robust04
Robust05
NYT
Preprocessing
Docs Robust04
Docs Robust05
Doc NYT
Term-document
matrix with
tfidf weights
Initialization & Indexing
SVMlight formattted
training data
Tfidf-Features NYT
Commit
Training
Model
Ranking
Prediction
Searching
WCRobust04 /
WCRobust0405

Results

- **Routing runs** as proposed by Grossman and Cormack **are replicable** (in terms of P@10)
- **Reproduced runs** result in **lower evaluation measures** compared to replicated runs
- Future directive for reproducibility study:
  „Archive“ implementation with the help of Docker

Thank you for your attention!

Many thanks to the organizers of CENTRE