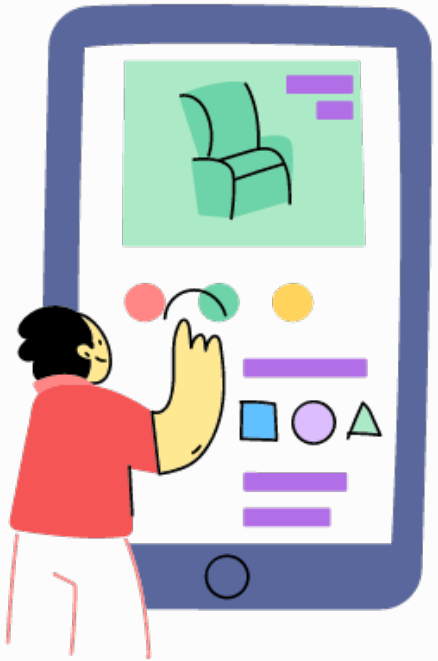
Enhancing Reproducibility in Information Retrieval

A Path Towards Scientific Integrity and Effective Implementation



47th European Conference on Information Retrieval

Sunday, April 6, 2025

Acknowledgements

- Craig Macdonald, Sean MacAvaney, Iadh Ounis
- Nicola Ferro
- Tommaso Di Noia

Reproducibility-aware conferences

Reproducibility and Replicability Tracks

- ECIR
- ACM RecSys
- ACM SIGIR
- ACM UMAP
- ACM MultiMedia
- CLEF

Challenges and Competitions

- ACM RecSys (RecSys Challenge)
- ACM SigIR (eComm WS)
- WSDM
- KDD
- CIKM

Benchmarking

- NeurIPS
- ICIP
- CVPR

Conferences

- ACM REP

*non exaustive lists

Who's in front of you

Antonio

Claudio

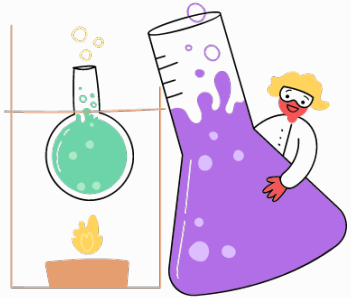
Assistant Professors at Politecnico di Bari (where we received our PhDs)
Working on recommender systems reproducibility, fairness, explanation, efficiency, and privacy

Nicola

Associate Professor at University of Pisa and formerly researcher at CNR
His main research interests include Cloud Computing, Web Search, and Information Retrieval, with focus on efficient data processing and neural information retrieval

Let’s talk about science

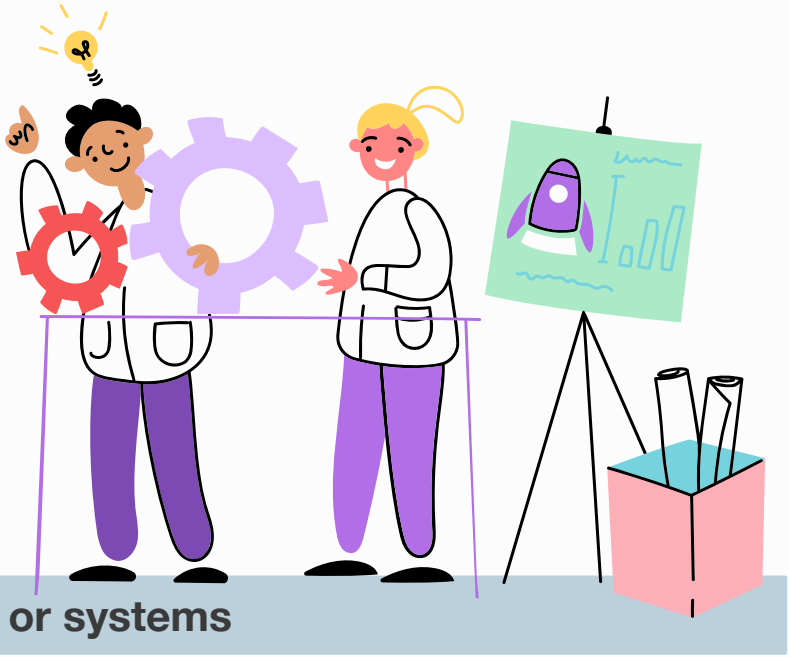
Am I a scientist?



What is science?

A mode of inquiry aiming to pose questions about the world, arriving at the answers and assessing their degree of certainty

- **Describe the world**
- **Explain the world**
- **Predict what will happen**
- **Intervene in specific processes or systems**



What is science?

How is the work of a scientist?

The scientists in the world follow a common approach

1. Introduce ideas and theories
2. Collect data
3. Analyze data and experiment
4. Communicate the results (e.g., through a scientific article)

The modern scientific method

The scientific method is an empirical method for acquiring knowledge about the world

1 Observe something

- Observe evidence systematically
- Document observations in an objective way

The modern scientific method

The scientific method is an empirical method for acquiring knowledge about the world

2 Develop a hypothesis

- Formulate a clear problem statement
- Identify the main question or investigation goal
- Pose a testable and measurable question

The modern scientific method

The scientific method is an empirical method for acquiring knowledge about the world

3 Collect data

- Gather relevant data systematically
- Use appropriate methods for data collection
- Organize data for analysis

The modern scientific method

The scientific method is an empirical method for acquiring knowledge about the world

4 Test with experiments

- Be aware of performing well-controlled experiments
- Control some parameters while manipulating others
- Collect result from the experiments

The modern scientific method

The scientific method is an empirical method for acquiring knowledge about the world

5 Analyze results

- Understand the meaning behind the results
- Establish cause-and-effect relationships
- Provide evidence to support or reject the hypothesis

The modern scientific method

The scientific method is an empirical method for acquiring knowledge about the world

6 Report conclusions

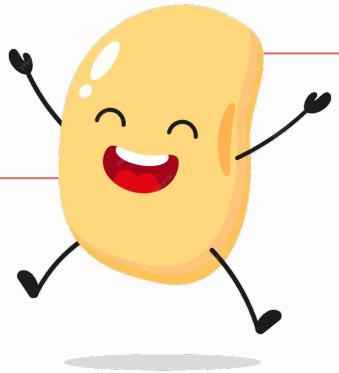
- Share experiment outcomes through conferences or journal articles
- Contribute to the body of knowledge for future research
- Don’t forget to detail the project’s design, methods, and results

Let's develop a new idea

Story of a bean: ideas from deduction

Known rule

The beans in my bag are white
(i.e., if a bean is in my bag, then it is white)



Evidence

I have in my hand a bean from my bag

KB: ∀x bag(x) ⇒ white(x)
bag(B)

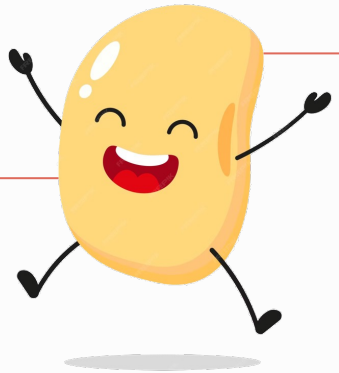
α: white(B)

New theory The bean in my hand is white

Story of a bean: ideas from induction

Facts

I have in my hand
some white beans



Knowledge

All these beans come
from my bag

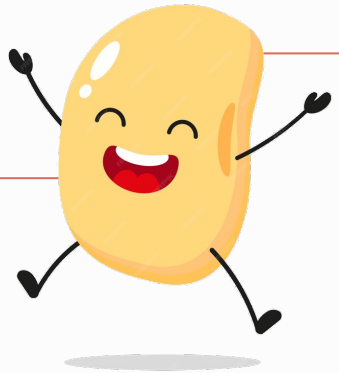
General rule

The beans in my bag are white
(i.e., if a bean is in my bag, then it is white)

Story of a bean: ideas from abduction

Known rule

The beans in my bag are white
(i.e., if a bean is in my bag, then it is white)



Evidence

I have in my hand a white bean (let's call it Fagiolino)

Supposed explanation

Fagiolino comes from my bag

Story of a bean: ideas from abduction

Let’s rewrite the last story in a more formal way

KB: ∀x bag(x) ⇒ white(x)

α: white(Fagiolino)

With deduction we cannot conclude that the bean comes from my bag
But abduction can help us explain why the bean in my hand is white!

Let’s define some potential hypotheses

Fagiolino fell from the sky

Fagiolino comes from my pocket

Fagiolino comes from my bag

Story of a bean: ideas from abduction

Now, add each hypothesis to the knowledge base and check whether it can be valid

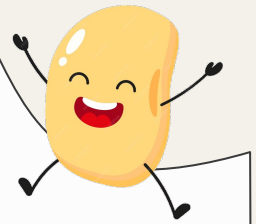
(h_1) Fagiolino fell from the sky: sky(Fagiolino)

KB ∪ h_1: [∀x bag(x) ⇒ white(x)] ∪ sky(Fagiolino)

α: white(Fagiolino)

With this hypothesis, we cannot deduce α from KB ∪ h_1

Story of a bean: ideas from abductio



Now, add each hypothesis to the knowledge base and check whether it can be valid

(h_2) Fagiolino comes from my pocket: pocket(Fagiolino)

KB ∪ h_2: [∀x bag(x) ⇒ white(x)] ∪ pocket(Fagiolino)

α: white(Fagiolino)

With this hypothesis, we cannot deduce α from KB ∪ h_2

Story of a bean: ideas from abduction

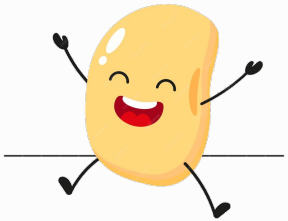
Now, add each hypothesis to the knowledge base and check whether it can be valid

(h_3) Fagiolino comes from my bag: bag(Fagiolino)

KB ∪ h_3: [∀x bag(x) ⇒ white(x)] ∪ bag(Fagiolino)

α: white(Fagiolino)

With this hypothesis, we can deduce the bean is white!
We have found a potentially valid hypothesis



Story of a bean: ideas from abduction

Now, add each hypothesis to the knowledge base and check whether it can be valid

(h_3) Fagiolino comes from my bag: bag(Fagiolino)

KB ∪ h_3: [∀x bag(x) ⇒ white(x)] ∪ bag(Fagiolino)

α: white(Fagiolino)

And… what if I have more than one potentially valid hypothesis?

Select the most simple and elegant

(see the Occam's razor)

Story of a bean: ideas from abduction

OK, but is this enough?

No.

Abuctive reasoning helps us in generating
new hypotheses that must be validated

We won't be sure about them until we are not able
to somehow prove their validity

Let's do some experiments

Experimenting with a new RS

Observation

My recommender considering only the last interaction of a user isn't working well 😢

Hypothesis

The low performance is due to the limited user representation and a new model considering a longer user history would perform better

Experimental result

The new model considering a longer history shows its effectiveness over the previous model

Experimenting with a new IR system

Observation

A traditional retrieval model (e.g., BM25) fails in retrieving relevant documents that use synonyms

Hypothesis

Leveraging techniques like embeddings would offer a more semantic-rich representation of text, leading to matching queries and documents based on meaning, not just keywords

Experimental result

Dense retrieval model shows statistically significant improvements in retrieval effectiveness

The value of the experiments

Do the experiments prove the new model is «the best»?

Do the experiments prove the new model works in any scenarios?

The scientific method never proves something with absolute certainty

Instead, the scientific method provides a structured process for testing, evaluating, and validating hypothesis and solutions through evidence

Making hypotheses and experiments reliable

How to make hypotheses (models, ideas, …) more and more reliable?

Allow others to verify our findings.

Other people should be able to:

- check the **validity and generalizability** of our results,
- or **contradict** our evidence (according to Popper, the progress does not consist in the accumulation of certainties, but in the progressive elimination of errors)

Making hypotheses and experiments reliable

But, why people want to check my findings?

Sometimes there could be **mistakes and they just want to check**

Sometimes they want to explore the limits of the findings and relationships you discovered to make other inquiry

Sometimes… a **young researcher** may be pressed to publish papers to improve their CVs
This pressure may lead to **overstate the importance of the results** and **increase the risk of bias** in data collection, analysis, and reporting

Making hypotheses and experiments reliable

Remember that nature is not capricious and follows rules that are consistent overtime and across different contexts

So… redoing an experiment, people should observe no difference between the original and the reproduction

Yes, this is what we call

reproducibility

Four questions about reliability of hypotheses and experiments

1 Are the data and the analyses laid with **sufficient transparency and clarity** that the results can be checked?

Reproducible research is research that is capable of being checked because the data, code, and methods of analysis are available to other researchers

Four questions about reliability of hypotheses and experiments

2 If checked, do the data and analysis offered in support of the result in fact **support that result**?

Research is reproducible if another researcher uses the available data and code and obtains the same results

Four questions about reliability of hypotheses and experiments

3 If the data and analysis are shown to support the original result, can the same findings be confirmed in the specific study context?

To answer this question, a researcher must redo the study, following the original methods as closely as possible and collecting new data, aimed at the same or a similar scientific question as the original research

This is no more reproducibility, but what we call

replicability

Four questions about reliability of hypotheses and experiments

4 Can the result reported or the inference drawn be found again in a broader set of study contexts?

A researcher could take a variety of paths: choose a new condition of analysis, conduct the same study in a new context, or conduct a new study aimed at the same or similar research question

And this is the notion of

generalizability

Let's tidy things up

Reproducibility

Obtaining **consistent results** using the **same** input data, computational steps, methods, code, and conditions of analysis;
a.k.a. transparency and "**computational reproducibility**"

Replicability

Obtaining **consistent results** across studies aimed at answering the same scientific question, each of which has obtained its own data

Generalizability

Exploring a similar scientific question but in **other contexts or populations** that differ from the original one and finding consistent results

Note that historically, various disciplines have used different nomenclatures, sometimes even reversing the terms "reproducibility" and "replicability" (ACM itself previously used this reverse terminology)

Reproducibility vs. Replicability

Reproducibility

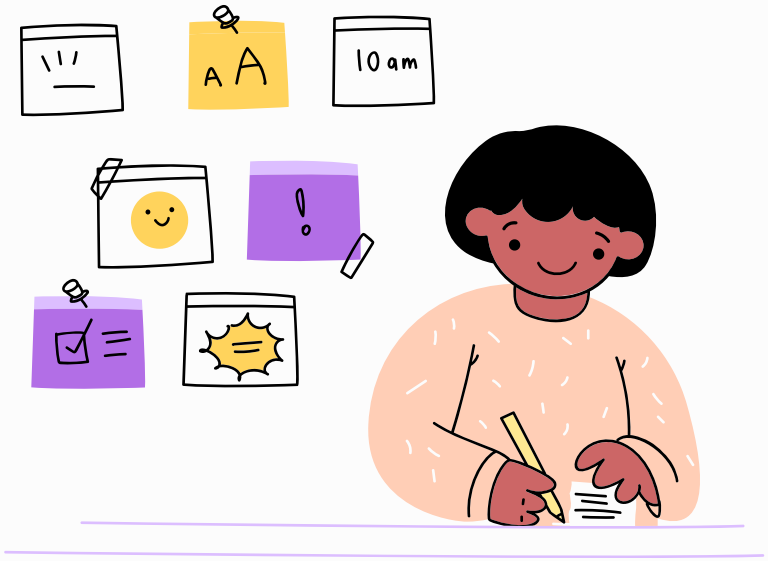
- Avoids changes
- Allows others to inspect and validate the experiment
- Expected from any well-controlled experiment, it is crucial for trasparent and accountable research

Replicability

- Requires changes
- Validates the experiment's core ideas, ensuring results aren't due to ad-hoc design choices
- Essential for corroborating findings and advancing inquiry

Let's rep recommenders

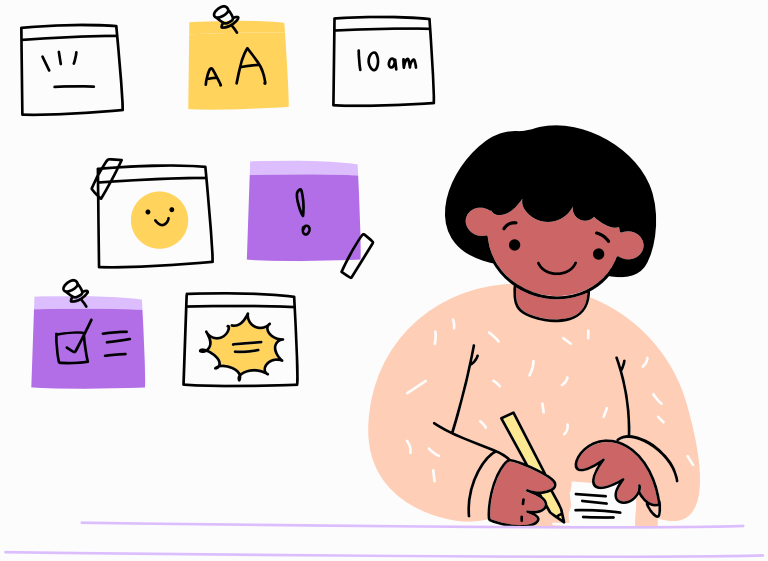
To-do list for a reproducible work



Provide a detailed description of:

- Dataset collection
- Data splitting
- Implementation details of the recommendation algorithms
- Parameters
- Candidate Item Filtering
- Evaluation
- Statistical Testing

To-do list for a replicable work



TL;DR: Try to change your environment:

- Dataset
- Parameters
- …

Are your findings still confirmed?

To-do list for a replicable work

Example

- We have created a **new graph recommender system**
- Our **hypothesis** is that the new recommender system works better than the state-of-the-art graph recommenders
- Let's create an **experimental environment** and test our algorithm:
    1. Choose a dataset
    2. Preprocess it to remove cold users and items according to a threshold
    3. Select a candidate items protocol
    4. And…

To-do list for a replicable work

Example (cont'd)

	nDCG@10
Our model	0.21

WOW! This is a very good performance 🤩

Not at all. 😏
What about the other recommender systems?

Ok. Let's read other papers and pick their results

To-do list for a replicable work

Example (cont'd)

	nDCG@10
Our model	0.21
Other graph model	0.10
Item kNN	0.06
Matrix Factorization	0.05

WOW! We are still the best! 🥳

No, these results won't be replicable 😞

Remember that the experimental environment (thus, the evaluation protocol) dramatically impacts the observed results

To-do list for a replicable work

Digression on the impact of the experimental setup

Let's have a look at two works experimenting with MovieLens 100K

	Algorithm				
Metric	k-Item	k-User	PureSVD	*Pop-item*	IMM
P@5	0.00135	0.006	0.067	0.227	0.267
NDCG@5	0.0036	0.0091	0.0566	0.216	0.245
MAP	0.013	0.041	0.061	0.119	0.156

Gorla et al, 2013

	Baseline(Test)
MAP	0.447
MRR	0.889
NDCG@10	0.720
NDCG@5	0.570
NDCG@3	0.447

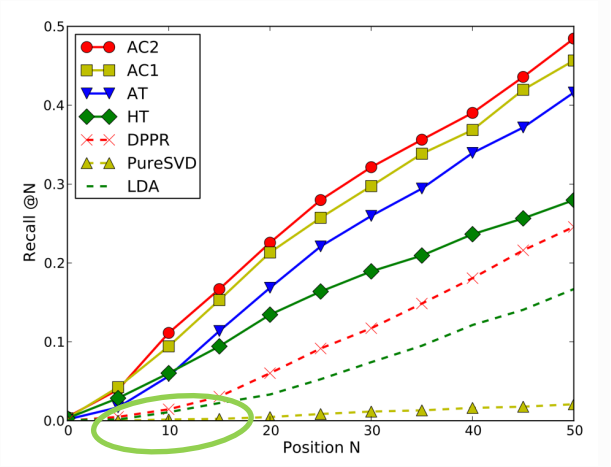
Jambor & Wang, 2010

MAP and nDCG seem ten times different!

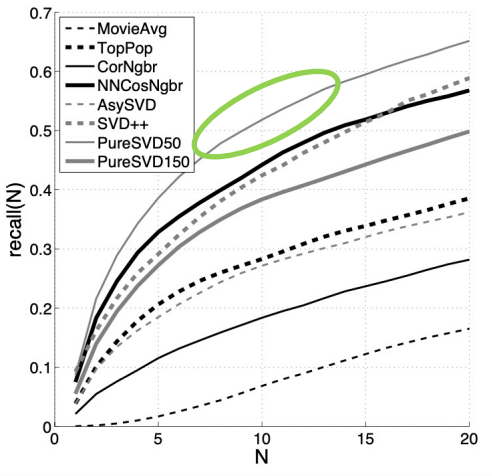
To-do list for a replicable work

Digression on the impact of the experimental setup

Both these works experiment with MovieLens 1M but report recall values that differ by one order of magnitude



Yin et al, 2012



Cremonesi et al, 2010

To-do list for a replicable work

Digression on the impact of the experimental setup

Remember that a lot of factors influence the results of an evaluation pipeline

- Splitting methods
- The selection of the items candidate to ranking (test ratings, test items, training items, all items, …)
- The use of different implementations of the same metric (e.g., normalizations, compensations, treatment of equal scores, …)
- Ability to predict for all items or users
- …

To-do list for a replicable work

Example (cont'd)

Ok, you got me! We have to **replicate** the other baselines in our environment

	nDCG@10
Our model	0.21
Other graph model	0.20
Item kNN	0.17
Matrix Factorization	0.16

The final (replicable?) finding: our graph recommender system improves the state of the art of graph recommender systems

To-do list for a replicable work

Example (cont'd)

	nDCG@10
Our model	0.21
Other graph model	0.10
Item kNN	0.06
Matrix Factorization	0.05

Still not sure about the replicability

Who is the «other graph model»?
Is it recent enough? Is it competitive enough?

Often, improved scores surpass **outdated baselines** and don't trend upwards over time, as baselines are **rarely recent or competitive** and **fail to reflect new discoveries**

To-do list for a replicable work

Example (cont'd)

	nDCG@10
Our model	0.21
Other graph model	0.10
Item kNN	0.06
Matrix Factorization	0.05

Still not sure about the replicability

Who are the other two baselines?
How our scientific findings relate to the two non-graph baselines?
Are they useful to confirm our hypothesis?

To-do list for a replicable work

Digression on Top-N Recommendation Algorithms

Algorithm	Top@10					
	nDCG	MAP	MRR	Pre	Rec	F1
EASE^R	**0.336**	0.335	**0.583**	0.274	**0.194**	0.190
SLIM	0.335	**0.337**	0.580	**0.275**	0.189	0.188
MF2020	0.329	0.327	0.563	0.272	0.190	**0.192**
UserKNN	0.315	0.314	0.554	0.256	0.183	0.179
RP$^3\beta$	0.315	0.313	0.556	0.256	0.184	0.179
iALS	0.306	0.304	0.542	0.252	0.179	0.176
MultiVAE	0.294	0.284	0.514	0.243	0.183	0.175
ItemKNN	0.292	0.293	0.518	0.242	0.163	0.163
NeuMF	0.277	0.275	0.494	0.232	0.157	0.158
BPRMF	0.275	0.271	0.502	0.226	0.166	0.161
MostPop	0.159	0.159	0.317	0.137	0.084	0.086
Random	0.008	0.007	0.020	0.007	0.004	0.004

Accuracy Results for MovieLens-1M. The tables are sorted by nDCG in descending order.

The paper *Top-N Recommendation Algorithms: A Quest for the State-of-the-Art* shows consistent performance by linear models, nearest-neighbor methods, and traditional matrix factorization on modest-sized, commonly-used datasets

Each algorithm is “competitive” in a different way w.r.t. the objective as measured by different metrics

To-do list for a replicable work

Example (cont'd)

	nDCG@10
Our model	0.21
Other graph model	0.10
Item kNN	0.06
Matrix Factorization	0.05

Still not sure about the replicability

Is the metric properly chosen for the task?
E.g., if our aim is to recommend just one item, in what helps nDCG?

To-do list for a replicable work

Example (cont'd)

	nDCG@10	
Our model	0.21	
Other graph model	0.10	
Item kNN	0.06	
Matrix Factorization	0.05	

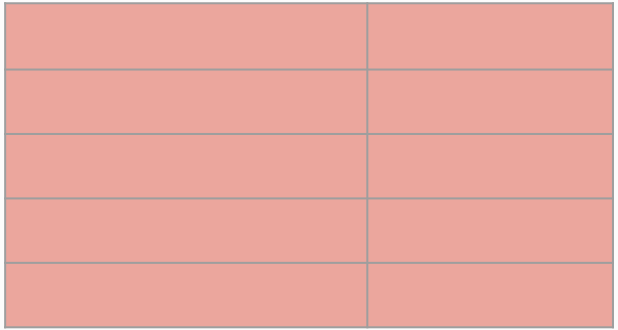
Still not sure about the replicability

Are we including all the metrics needed to analyze and justify our findings?

To-do list for a replicable work

Example (cont'd)

	nDCG@10
Our model	0.21
Other graph model	0.10
Item kNN	0.06
Matrix Factorization	0.05



Still not sure about the replicability

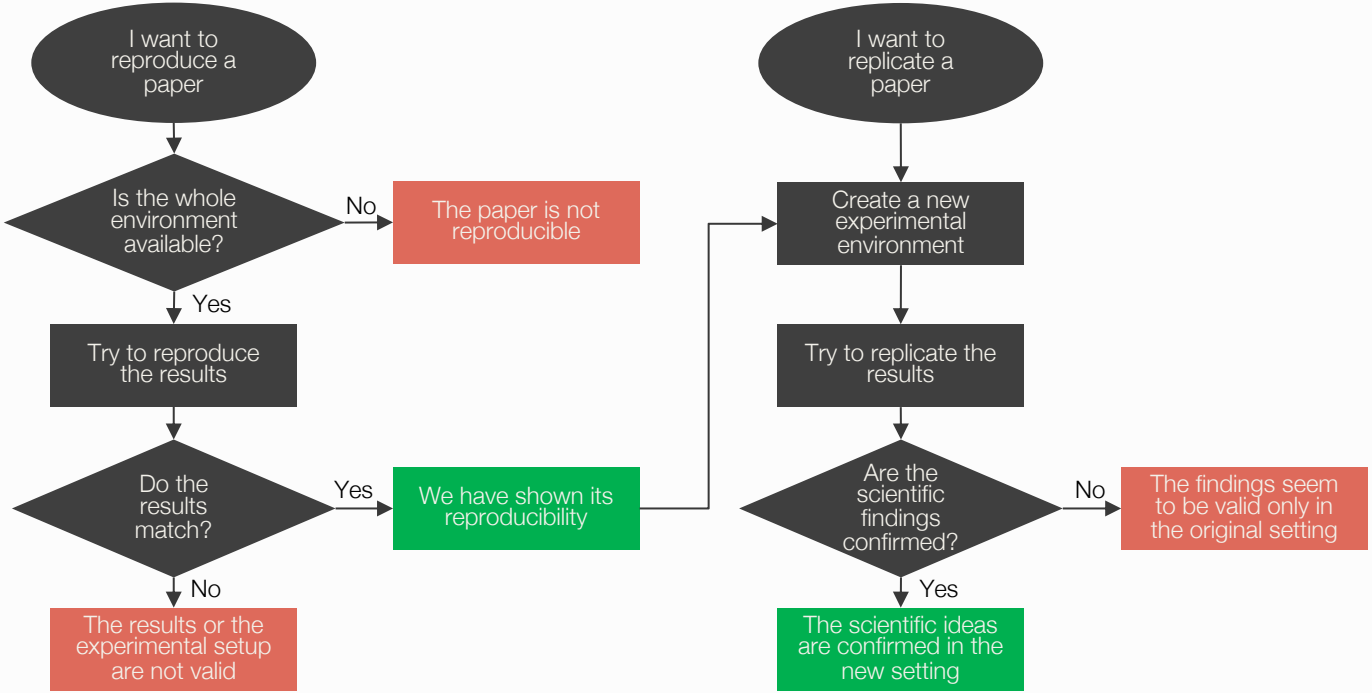
What about other datasets?
Is this (are these) dataset(s) enough to prove our findings?

To-do list for a replicable work

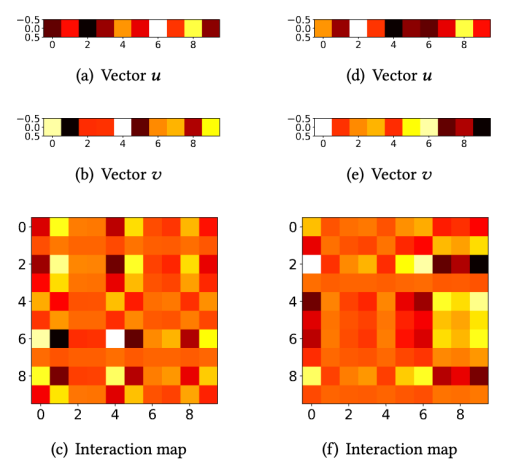
Take-home message

- Many possible mistakes can hinder the replicability of our work
- Carefully check your experimental environment to ensure your hypotheses are as strongly validated as possible within your context
- Test your findings with other experimental setups
- Make your paper (at least) reproducible to promote transparency, facilitate verification, and simplify future replicability efforts

How to reproduce or replicate another work



Convolutions over User-Item Embedding Maps?



Effects of permutating the columns of vectors u and v on their resulting outer product (the interaction map)

The paper *Critically Examining the Claimed Value of Convolutions over User-Item Embedding Maps for Recommender Systems* poses questions about CNN advantages

CNNs leverage the position of each “pixel” to discover “semantic” patterns
Does it make sense in user-item matrices?

CNN-based models cannot offer the claimed advantages (think about permutations of rows)

Convolutions over User-Item Embedding Maps?

They used the original code, data, data splits, as well as hyperparameters that were provided by the authors

	@5		@10		@20	
	HR	NDCG	HR	NDCG	HR	NDCG
TopPopular	0.0817	0.0538	0.1200	0.0661	0.1751	0.0799
UserKNN CF	**0.2068**	**0.1355**	**0.3126**	**0.1695**	0.4401	**0.2017**
ItemKNN CF	**0.2521**	**0.1686**	**0.3669**	**0.2056**	**0.4974**	**0.2385**
$P^3\alpha$	**0.2146**	**0.1395**	**0.3211**	**0.1737**	0.4442	**0.2049**
$RP^3\beta$	**0.2202**	**0.1431**	**0.3323**	**0.1793**	**0.4667**	**0.2132**
SLIM	**0.2330**	**0.1535**	**0.3475**	**0.1904**	**0.4799**	**0.2238**
PureSVD	**0.2011**	**0.1307**	0.3002	**0.1626**	0.4238	0.1938
iALS	**0.2048**	**0.1348**	**0.3080**	**0.1680**	0.4319	**0.1993**
ConvNCF	0.1947	0.1250	0.3059	0.1608	0.4446	0.1957

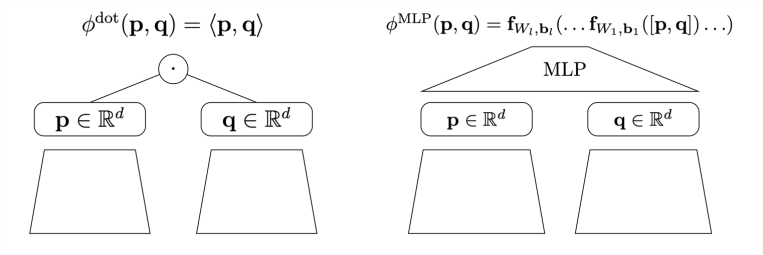
	@ 1		@ 5		@ 10	
	HR	NDCG	HR	NDCG	HR	NDCG
TopPopular	0.1593	0.1593	0.4217	0.2936	0.5813	0.3451
UserKNN CF	0.3540	0.3540	0.6884	0.5324	0.8060	0.5704
ItemKNN CF	0.3305	0.3305	0.6682	0.5080	0.7940	0.5488
$P^3\alpha$	0.3316	0.3316	0.6543	0.5031	0.7687	0.5402
$RP^3\beta$	0.3464	0.3464	0.6743	0.5198	0.7959	0.5591
SLIM	**0.3906**	**0.3906**	**0.7116**	**0.5625**	**0.8315**	**0.6014**
PureSVD	**0.3735**	**0.3735**	**0.7088**	**0.5522**	0.8132	**0.5861**
iALS	**0.3816**	**0.3816**	**0.7121**	**0.5581**	0.8200	**0.5933**
CoupledCF	0.3522	0.3522	0.7018	0.5374	0.8247	0.5775

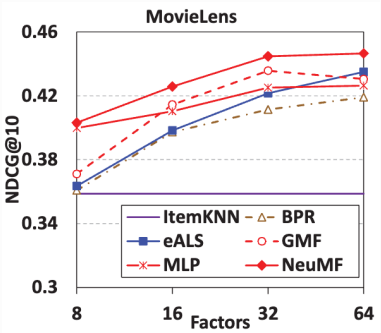
	@5		@10		@20	
	HR	NDCG	HR	NDCG	HR	NDCG
TopPopular	0.0016	0.0009	0.0023	0.0011	0.0033	0.0014
UserKNN CF	**0.5964**	**0.4527**	**0.6715**	**0.4773**	**0.7032**	**0.4855**
ItemKNN CF	**0.5975**	**0.4425**	**0.6776**	**0.4689**	**0.7070**	**0.4764**
$P^3\alpha$	**0.6327**	**0.4929**	**0.6744**	**0.5066**	**0.7014**	**0.5135**
$RP^3\beta$	**0.5896**	**0.4458**	**0.6756**	**0.4739**	**0.7071**	**0.4821**
SLIM	**0.6674**	**0.5169**	**0.6972**	**0.5267**	**0.7102**	**0.5300**
PureSVD	**0.4026**	**0.3117**	**0.4891**	**0.3397**	**0.5652**	**0.3590**
iALS	**0.6110**	**0.4811**	**0.6735**	**0.5017**	**0.7033**	**0.5093**
CFM	0.2241	0.1485	0.3338	0.1839	0.4661	0.2173

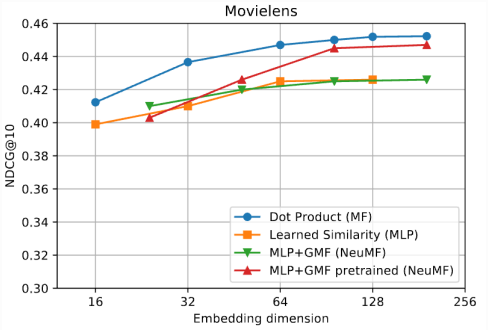
Experimental results for ConvNCF, CoupledCF, and CFM for Yelp, MovieLens1M, and Last.fm respectively

Neural Collaborative Filtering vs. Matrix Factorization Revisited

Rendle et al. show that a well-tuned simple dot product outperforms MLPs (NeuMF) in both effectiveness and efficiency for estimating the similarity between a user and an item



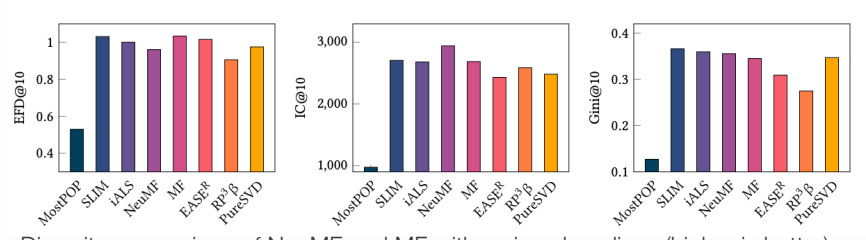




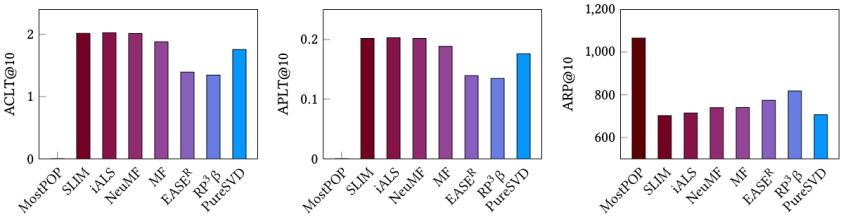
Performance of NDCG@10 w.r.t. the number of predictive factors on MovieLens1M. Comparison of the results of the two papers.

Reenvisioning Collaborative Filtering vs Matrix Factorization

Anelli et al. **reproduce and replicate** experiments from *Neural Collaborative Filtering vs Matrix Factorization* and **extend the original findings** confirming that MF provides better accuracy, especially on long-tail items, but NeuMF offers better coverage and diversification



Diversity comparison of NeuMF and MF with various baselines (higher is better)



Analysis of Bias for NeuMF, MF and various baselines considering a cutoff @10

A Troubling Analysis of Recommender Systems Research

In *A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research*, Ferrari Da Crema et al. survey papers published between 2015 and 2018 in top-conferences They identify 26 relevant papers and, among these, only 12 were considered having a reproducible experimental setup (evidencing a reproducibility crisis)

In a lot of cases, they also evidence a lack of replicability

Authors report papers showing only favorable results, thus inflating the risk of presenting only "virtual" progress

They confirm a propagation of weak baselines: relying on methods like NeuMF as state-of-the-art can mislead research, as they may not outperform simpler techniques.

Let's rep retrieval

Ingredients for reproducibility in IR

- **Source code**
  - Availability with instructions to reproduce
  - Random generators
  - Training procedures
  - Hyper-parameters
  - Ablation studies
- **Benchmark data**
  - Raw datasets
  - Derived datasets
  - Sampling procedures
- **Documentation**
  - Clear procedures to reproduce
  - Annotated code
- **Metrics**
  - Which numbers
  - How to get the numbers
  - How to perform statistical tests
- **RQs**
  - What is under investigation
  - What are pre-conditions, post-conditions, and invariants

Experimental Evaluation

Cranfield Paradigm by Cyril W. Cleverdon

- Dates back to mid 1960s
- Makes use of **experimental collections**
  - **documents** (corpora)
  - **topics**, which are a surrogate for information needs
  - **relevance judgments** (binary or graded) also called relevance assessment or ground-truth (or qrels)
- Ensures **comparability** and **repeatability** of the experiments

Cyril W. Cleverdon (UK)

Cleverdon, C. W. (1962). Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems. Aslib Cranfield Research Project, College of Aeronautics, Cranfield, UK.
Cleverdon, C. W. (1997). The Cranfield Tests on Index Languages Devices. In Spärck Jones, K. and Willett, P., editors, Readings in Information Retrieval, pages 47–60. Morgan Kaufmann Publisher, Inc., San Francisco, CA, USA.

IR reproducibility initiatives

TREC (Text REtrieval Conference)

- Running since 1992
- “An evaluation workshop series for measuring the effectiveness of search algorithms and other technologies that help us find information”
- trec_eval is the standard tool used by the TREC community for evaluating an ad hoc retrieval run (source code at https://github.com/usnistgov/trec_eval)

[https://trec.nist.gov/]

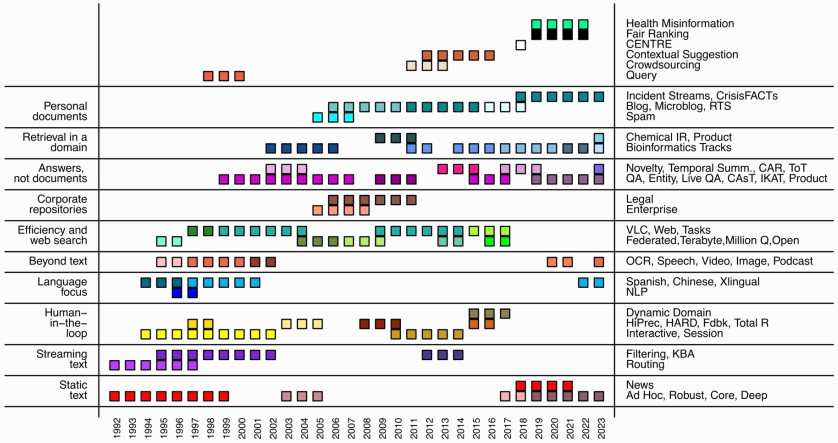
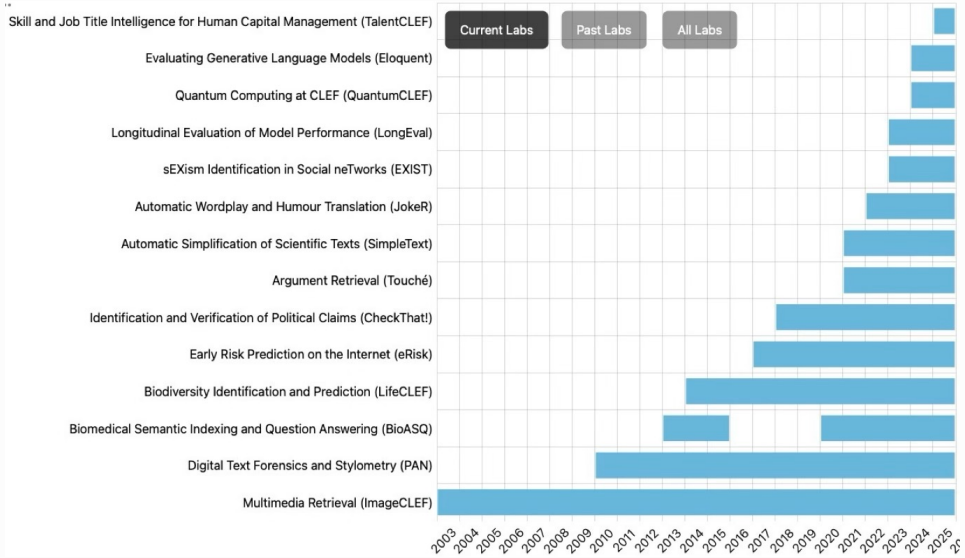


Image Source: https://pages.nist.gov/trec-browser/assets/tracks.png

IR reproducibility initiatives

CLEF (Conference and Labs of the Evaluation Forum)

- Running since 2000
- “Promotes research, innovation, and development of information access systems with an emphasis on multilingual and multimodal information with various levels of structure”



[https://www.clef-initiative.eu/]

Image Source: https://www.clef-initiative.eu/

IR reproducibility initiatives

NTCIR (NII Test Collection for IR Systems)

- Running since 1997
- “Evaluation efforts designed to enhance research on diverse information access technologies, including, but not limited to, cross-language and multimedia information access, question-answering, text mining and summarisation, with an emphasis on East Asian languages such as Chinese, Korean, and Japanese, as well as English”

[https://research.nii.ac.jp/ntcir/index-en.html]

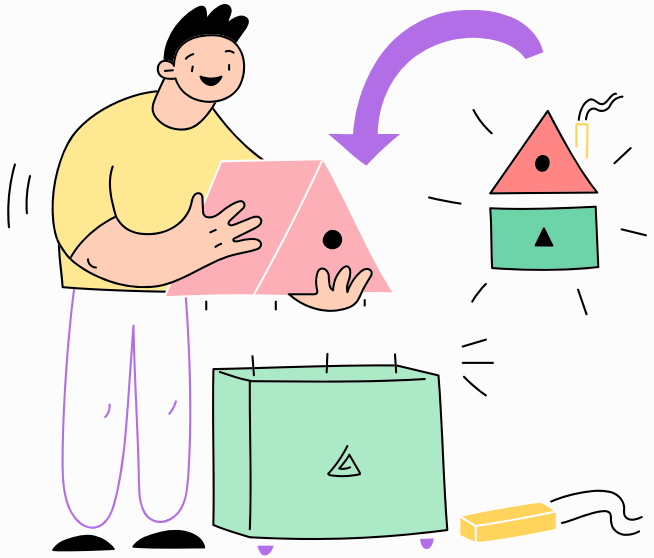
FIRE (Forum for Information Retrieval Evaluation)

- Running since 2008
- “Encourage research in Indian language Information Access technologies by providing reusable large-scale test collections for Indian language IR experiments; Provide a common evaluation infrastructure for comparing the performance of different IR system; Investigate evaluation methods for Information Access techniques and methods for constructing a reusable large-scale data set for Indian language IR experiments”

IR reproducibility initiatives

Organized in tracks

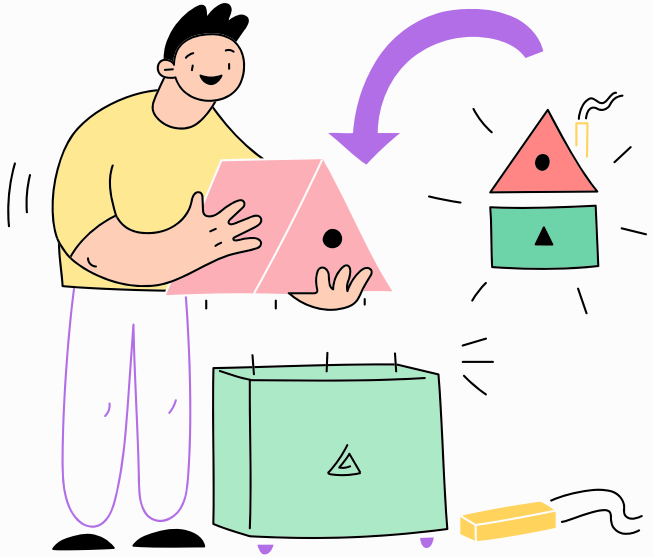
- Each track is dedicated to a particular IR problem, such as web search, legal document retrieval, question answering, or biomedical IR
- Tracks evolve based on research needs, with new tracks emerging as technology and user behavior change



IR reproducibility initiatives

Same benchmark datasets

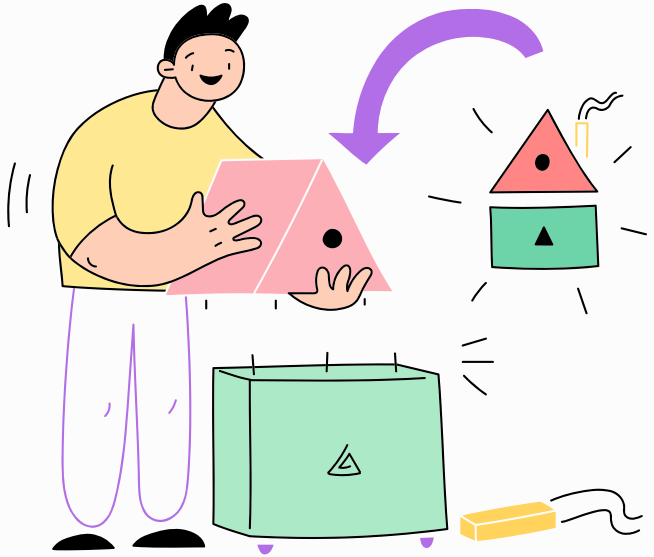
- Participants are given the same benchmark dataset to experiment with, ensuring fair comparisons
- Each participant runs topics against documents using their retrieval system, and returns a ranked list of the top documents per topic



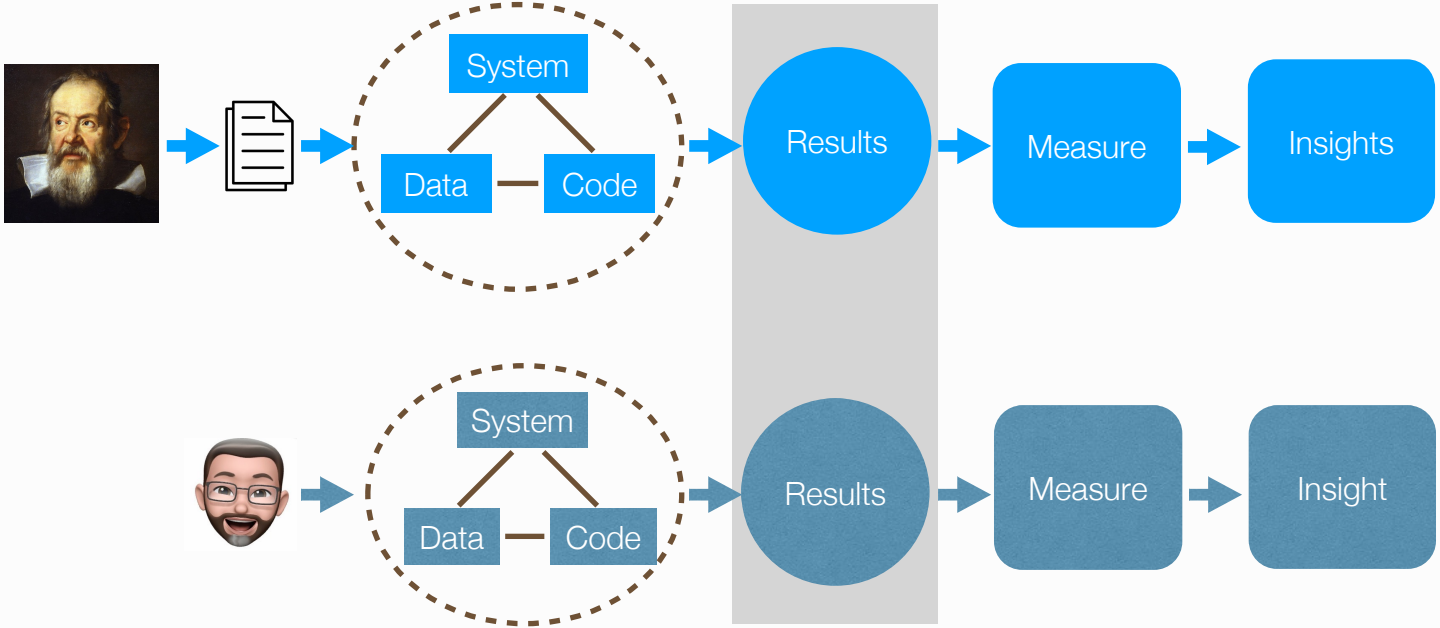
IR reproducibility initiatives

Evaluation

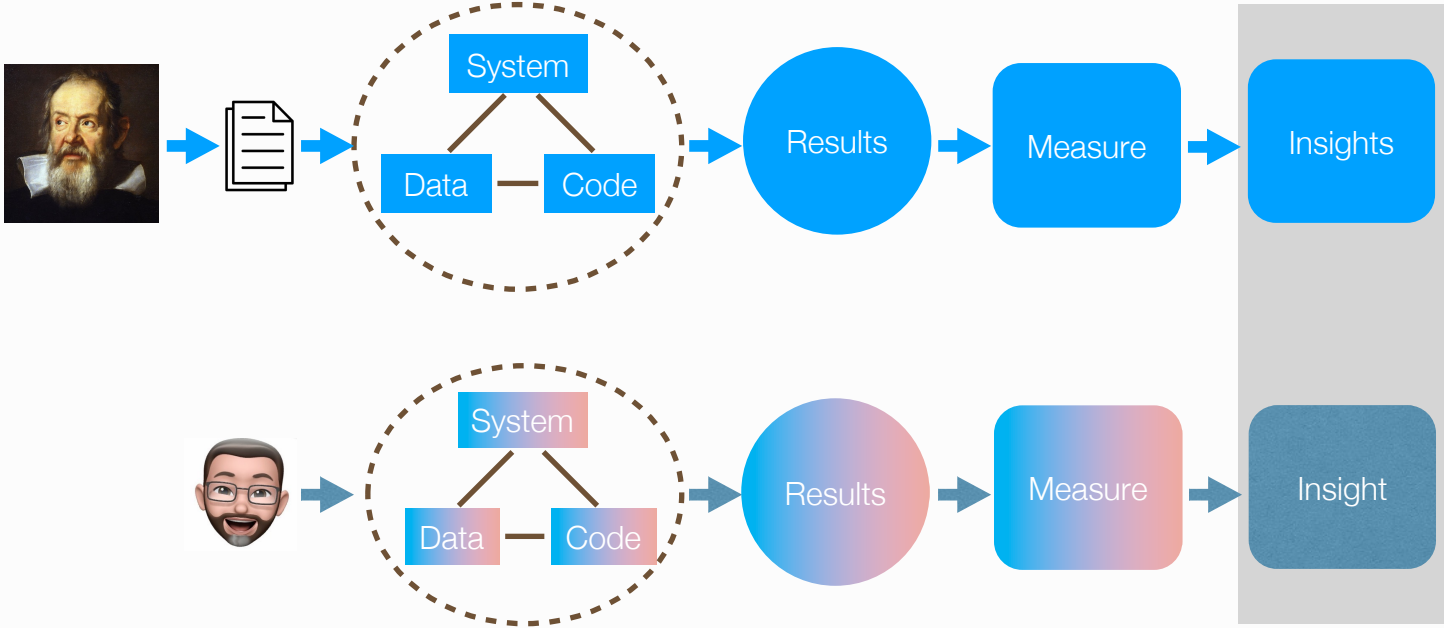
- Standardised evaluation metrics and procedures through a set of provided relevance judgments
- Submissions by participants are evaluated using resulting relevance judgments; evaluation results returned to participants



Replication Scenario



Reproducibility Scenario



Rep-* Research Questions

Can we replicate the runs of the original system?

Same code and same data and same system

Can we replicate the measures of the original system?

Same code and same data and same system

Rep-* Research Questions

Can we replicate the insights of the original system?

Same code and same data and same system

Can we reproduce the insights of the original system?

Different code and/or data and/or system

Rep-* Research Questions

Do we learn something new?

How to measure Rep-*

Some **guidelines**:

- Compare results, e.g., lists of documents
  - (Modified) Kendall's τ
  - Rank-Biased Overlap (RBO)
- Compare measures, e.g., mAP, nDCG
  - Win-Tie-Loss (WTL)
  - Root Mean Square Error
  - Effect Ratio
- Compare insights

Source: Breur et al., How to Measure the Reproducibility of System-oriented IR Experiments, SIGIR 2020, https://github.com/irgroup/repro_eval

PLAID: An Efficient Engine for Late Interaction Retrieval

Keshav Santhanam*
keshav2@cs.stanford.edu
Stanford University
United States

Omar Khattab*
okhattab@stanford.edu
Stanford University
United States

Christopher Potts
cgpotts@stanford.edu
Stanford University
United States

Matei Zaharia
matei@cs.stanford.edu
Stanford University
United States

ABSTRACT

Pre-trained language models are increasingly important components across multiple information retrieval (IR) paradigms. Late interaction, introduced with the ColBERT model and recently refined in ColBERTv2, is a popular paradigm that holds state-of-the-art status across many benchmarks. To dramatically speed up the search latency of late interaction, we introduce the Performance-optimized Late Interaction Driver (PLAID) engine. Without impacting quality, PLAID swiftly eliminates low-scoring passages using a novel centroid interaction mechanism that treats every passage as a lightweight bag of centroids. PLAID uses centroid interaction as well as centroid pruning, a mechanism for sparsifying the bag of centroids, within a highly-optimized engine to reduce late interaction search latency by up to 7× on a GPU and 45× on a CPU against vanilla ColBERTv2, while continuing to deliver state-of-the-art retrieval quality. This allows the PLAID engine with ColBERTv2 to achieve latency of tens of milliseconds on a GPU and tens or just few hundreds of milliseconds on a CPU at large scale, even at the largest scales we evaluate with 140M passages.

CCS CONCEPTS

• **Information systems** → **Top-k retrieval in databases**; **Document filtering**.

KEYWORDS

neural information retrieval, late interaction, efficient search, colbert, dynamic pruning

ACM Reference Format:
Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. PLAID: An Efficient Engine for Late Interaction Retrieval. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557325

*Equal contribution.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557325

1 INTRODUCTION

Recent advances in neural information retrieval (IR) have led to notable gains on retrieval benchmarks and retrieval-based NLP tasks. Late interaction, introduced in ColBERT [22], is a paradigm that delivers state-of-the-art quality in many of these settings, including passage ranking [14, 44, 50], open-domain question answering [21, 24], conversational tasks [37, 40], and beyond [20, 56]. ColBERT and its variants encode queries and documents into *token-level* vectors and conduct scoring via scalable yet *fine-grained* interactions at the level of tokens (Figure 1), alleviating the dot-product bottleneck of single-vector representations. The recent ColBERTv2 [44] model demonstrates that late interaction models often considerably outperform recent single-vector and sparse representations within and outside the training domain, a finding echoed in several recent studies [26, 29, 45, 46, 53, 55].

Despite its strong retrieval quality, late interaction requires special infrastructure [22, 25] for low-latency retrieval as it encodes each query and each document as a full *matrix*. Most IR models represent documents as a single vector, either sparse (e.g., BM25 [43]; SPLADE [11]) or dense (e.g., DPR [18]; ANCE [51]), and thus mature sparse retrieval strategies like WAND [5] or dense kNN methods like HNSW [31] cannot be applied directly or optimally to late interaction. While recent work [28, 44, 47] has explored optimizing individual components of ColBERT's pipeline, an end-to-end optimized engine has never been studied to our knowledge.

We study how to optimize late-interaction search latency at a large scale, taking all steps of retrieval into account. We build on the state-of-the-art ColBERTv2 model. Besides improving quality with denoised supervision, ColBERTv2 aggressively compresses the storage footprint of late interaction. It reduces the index size by up to an order of magnitude using *residual representations* (§3.1). In those, each vector in a passage is encoded using the ID of its nearest *centroid* that approximates its token semantics—among tens or hundreds of thousands of centroids obtained through k-means clustering—and a *quantized* residual vector.

We introduce the **P**erformance-optimized **L**ate **I**nteraction **D**river (PLAID) engine,[1] an efficient retrieval engine that reduces late interaction search latency by 2.5–7× on GPU and 9–45× on CPU against vanilla ColBERTv2 while retaining high quality. This allows the PLAID implementation of ColBERTv2, PLAID ColBERTv2, to achieve CPU-only latency of tens or just few hundreds of milliseconds and GPU latency of few tens of milliseconds at very large

[1]Code is available at https://github.com/stanford-futuredata/ColBERT.

A Reproducibility Study of PLAID

Sean MacAvaney
University of Glasgow
Glasgow, United Kingdom
sean.macavaney@glasgow.ac.uk

Nicola Tonellotto
University of Pisa
Pisa, Italy
nicola.tonellotto@unipi.it

ABSTRACT

The PLAID (Performance-optimized Late Interaction Driver) algorithm for ColBERTv2 uses clustered term representations to retrieve and progressively prune documents for final (exact) document scoring. In this paper, we reproduce and fill in missing gaps from the original work. By studying the parameters PLAID introduces, we find that its Pareto frontier is formed of a careful balance among its three parameters; deviations beyond the suggested settings can substantially increase latency without necessarily improving its effectiveness. We then compare PLAID with an important baseline missing from the paper: re-ranking a lexical system. We find that applying ColBERTv2 as a re-ranker atop an initial pool of BM25 results provides better efficiency-effectiveness trade-offs in low-latency settings. However, re-ranking cannot reach peak effectiveness at higher latency settings due to limitations in recall of lexical matching and provides a poor approximation of an exhaustive ColBERTv2 search. We find that recently proposed modifications to re-ranking that pull in the neighbors of top-scoring documents overcome this limitation, providing a Pareto frontier across all operational points for ColBERTv2 when evaluated using a well-annotated dataset. Curious about why re-ranking methods are highly competitive with PLAID, we analyze the token representation clusters PLAID uses for retrieval and find that most clusters are predominantly aligned with a single token and vice versa. Given the competitive trade-offs that re-ranking baselines exhibit, this work highlights the importance of carefully selecting pertinent baselines when evaluating the efficiency of retrieval engines.

https://github.com/seanmacavaney/plaidrepro

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking.**

KEYWORDS

Late Interaction, Efficiency, Reproducibility

ACM Reference Format:
Sean MacAvaney and Nicola Tonellotto. 2024. A Reproducibility Study of PLAID. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3626772.3657856

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657856

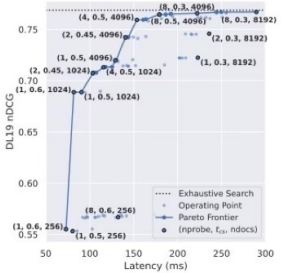


Figure 1: The Pareto frontier of PLAID for ColBERTv2 on TREC DL 2019 over the three parameters it introduces (nprobe, t_{cs}, and ndocs). Several operational points are labeled to highlight the interdependence of PLAID's parameters.

1 INTRODUCTION

Relevance ranking is a central task in information retrieval. Numerous classes of models exist for the task, including lexical [21], dense [10], learned sparse [18], and late interaction [11]. While efficient exact top-k retrieval algorithms exist for lexical and learned sparse retrieval systems (e.g., BlockMaxWAND [6]), dense and late interaction methods either need to perform expensive exhaustive scoring over the entire collection or resort to an approximation of top-k retrieval. A myriad of approximate k-nearest-neighbor approaches are available for (single-representation) dense models (e.g., HNSW [16]). However, these approaches generally do not apply directly to late interaction scoring mechanisms, so bespoke retrieval algorithms for late interaction models have been proposed.

PLAID (Performance-optimized Late Interaction Driver) [22] is one such retrieval algorithm. It is designed to efficiently retrieve and score documents for ColBERTv2 [23], a prominent late interaction model. PLAID first performs coarse-grained retrieval by matching the closest ColBERTv2 centroids (used for compressing term embeddings) to the query term embeddings. It then progressively filters the candidate documents by performing finer-grained estimations of a document's final relevance score. These filtering steps are controlled by three new parameters, which are discussed in more detail in Section 2.

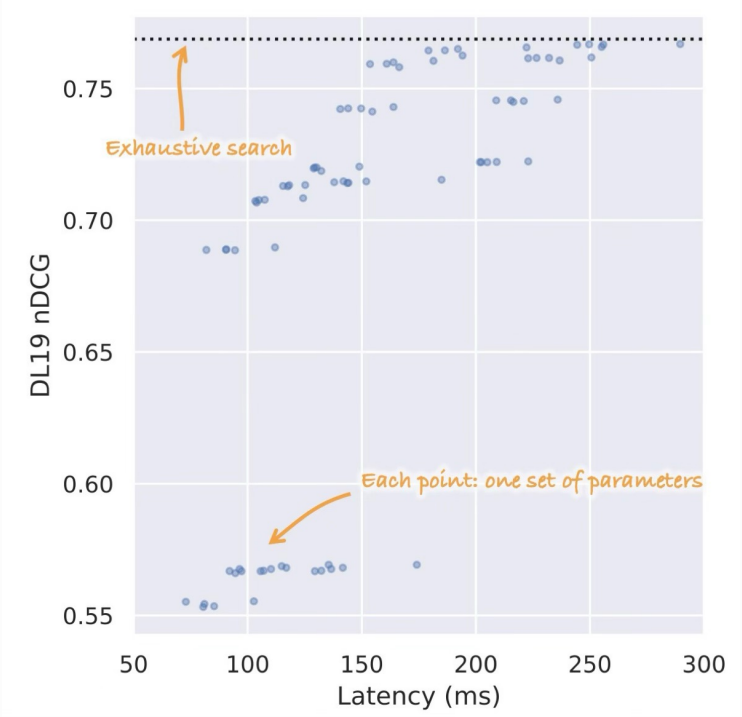
Source: MacAvaney & Tonellotto, "A Reproducibility Study of PLAID", SIGIR 2024.

Replication Results

	MS MARCO Dev				TREC DL 2019		
	RR@10	R@1k	RBO	ms/q	nDCG@10	nDCG@1k	R@1k
PLAID Reproduction							
(a)	0.394	0.833	0.612	80.5	0.739	0.553	0.555
(b)	0.397	0.933	0.890	103.4	0.745	0.707	0.786
(c)	0.397	0.975	0.983	163.9	0.745	0.760	0.871
Original PLAID Results							
(a)	0.394	NR	NR	185.5	NR	NR	NR
(b)	0.398	NR	NR	222.3	NR	NR	NR
(c)	0.398	0.975	NR	352.3	NR	NR	NR
Exhaustive ColBERTv2 Search							
-	0.397	0.984	1.000	N/A	0.745	0.769	0.894

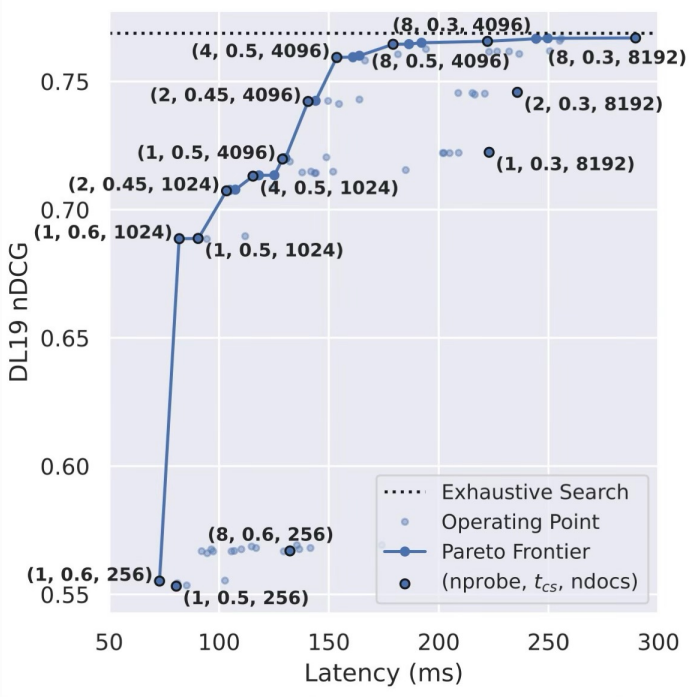
Source: MacAvaney & Tonellotto, “A Reproducibility Study of PLAID”, SIGIR 2024.

Hyperparameter Configuration



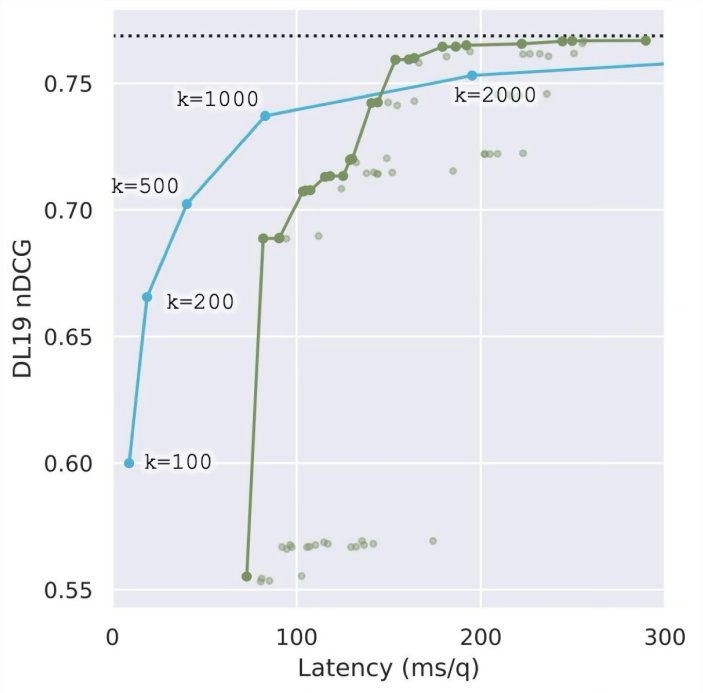
Source: MacAvaney & Tonellotto, “A Reproducibility Study of PLAID”, SIGIR 2024.

Pareto Frontier



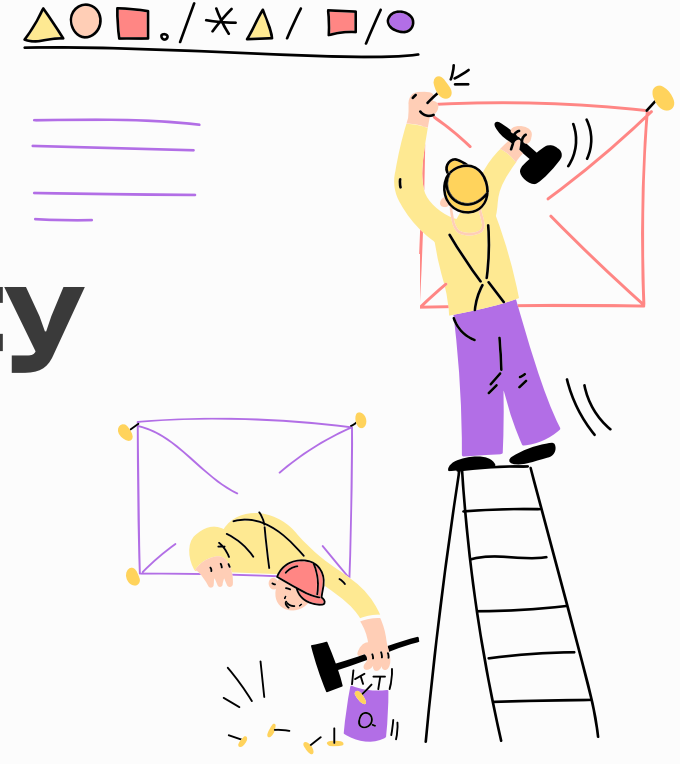
Source: MacAvaney & Tonellotto, “A Reproducibility Study of PLAID”, SIGIR 2024.

BM25 Reranking Insights



Source: MacAvaney & Tonellotto, “A Reproducibility Study of PLAID”, SIGIR 2024.

Let's make reproducibility easier



Towards a easier reproducibility

We have seen how replicability is strictly related with good hypotheses and evaluation methodologies properly chosen to confirm the hypotheses

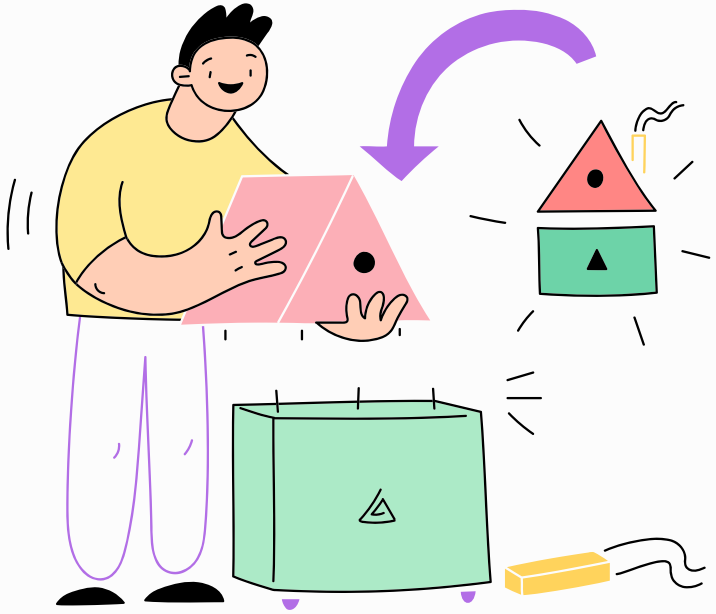
Reproducibility, instead, is related to rigourously provide code, data, and artifacts that lead to the same experimental results

But how hard can be guarantee (at least) reproducibility without any errors? Remember that in our works we should «reimplement» the baselines (and the metrics), so that this «reimplementation» and the experimental settings are in turn reproducible

Towards a easier reproducibility

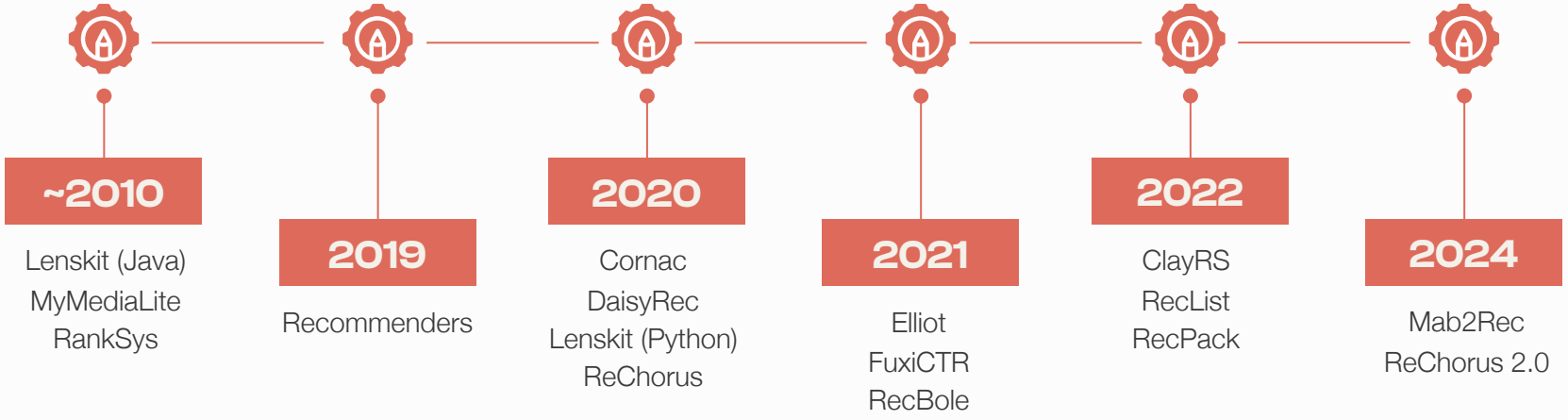
Reproducibility would be for sure easier with…

1. Artifact sharing
2. Shared implementations
3. Shared metrics implementations
4. Common practices



In **recommendation research**, these efforts have been supported by

Reproducibility frameworks



The RecSys CfP suggests using one of the frameworks above for the submitted papers and sharing the used *experimental environment*

Reproducibility frameworks for recommendation

Data-pipeline	Item selection	Models
Metrics	Tuning	Statistical tests
Configuration	APIs and UIs	Results (CSV/LaTeX)

On the other hand…

Initiatives in Information Retrieval

Search Engines

- MG4J
- Terrier
- PISA
- Lucene
- Anserini
- OWS

Experimental Platforms

- Pyserini
- Tirex
- Pyterrier

Pyserini

Python toolkit for reproducible information retrieval research:

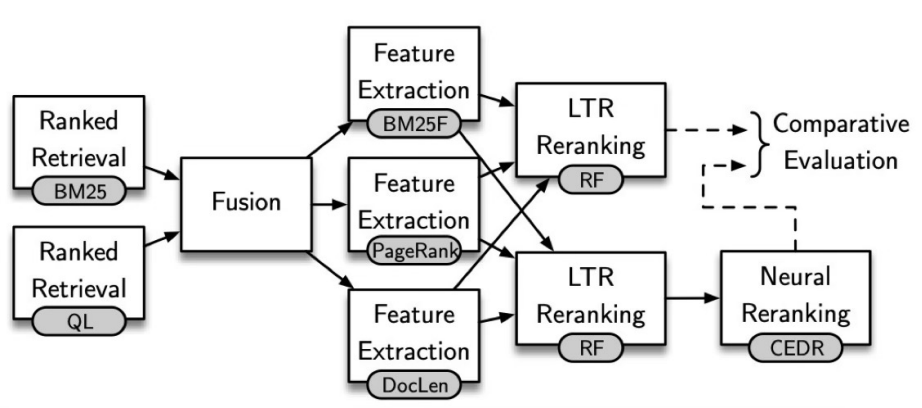
- integrated with Anserini
- sparse (classical & learned), dense and hybrid retrieval
- multi-stage ranking architectures
- out of the box support for the entire IR research lifecycle
- “two-click reproductions”

Source: Lin et al., Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations, SIGIR 2021, https://github.com/castorini/pyserini

PyTerrier

Python toolkit for declarative IR pipelines:

- integrated with Terrier
- powerful formalism to express advanced retrieval pipelines to be expressed
- declarative in nature, close to conceptual experiment design
- sparse (classical & learned), learning-to-rank and dense retrieval
- vibrant ecosystem of plugins for state-of-the-art IR research

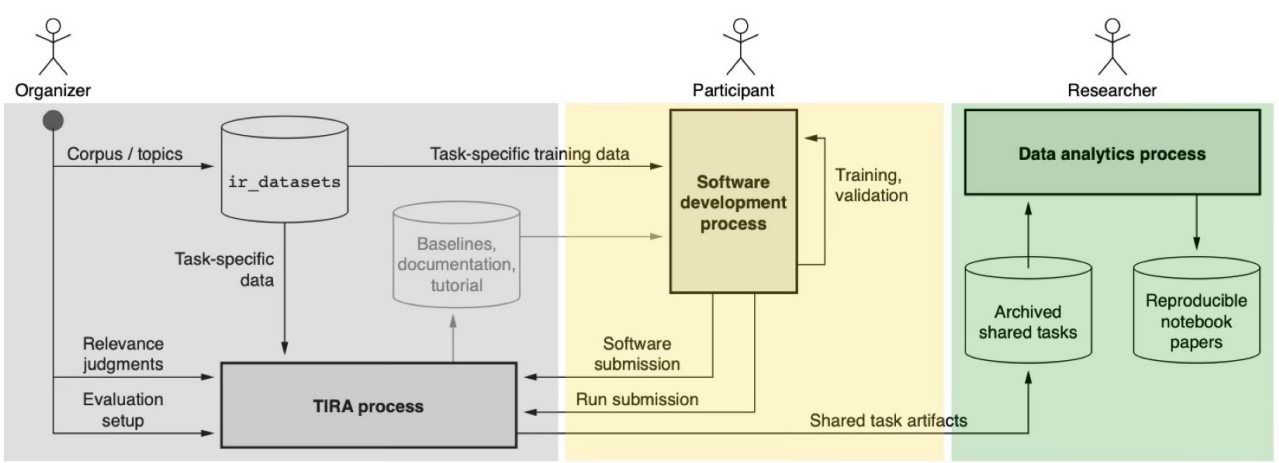


Source: Macdonald et al., Declarative Experimentation in Information Retrieval using PyTerrier, ICTIR 2020, https://github.com/terrier-org/pyterrier

Tirex

Integration of ir_datasets, ir_measures, and PyTerrier with TIRA to promote more standardized, reproducible, and scalable retrieval experiments:

- to conduct (blinded) IR experiments
- to organise “always-on” reproducible shared tasks on the basis of software submissions



Source: Fröbe et al., The Information Retrieval Experiment Platform, SIGIR 2023, https://github.com/tira-io/ir-experiment-platform

Experimaestro

A versatile tool for designing and managing complex workflows
Maintains comprehensive records of experiments, including parameters and environments, supporting the essential research principle of reproducibility

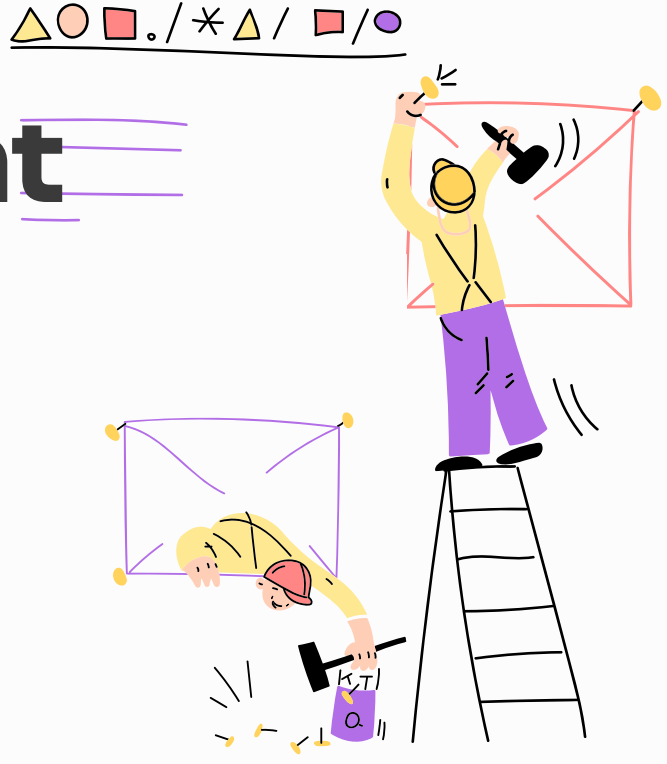
Experimaestro's distinct features include:

Comprehensive Task Composition
Parameter Monitoring
Automated Output Organization
Imperative Experiment Definition

See the reproduction of the CoSPLADE paper where the full experiment is included in one experimental file (https://github.com/xpmir/cosplade/blob/main/first-stage/experiment.py)

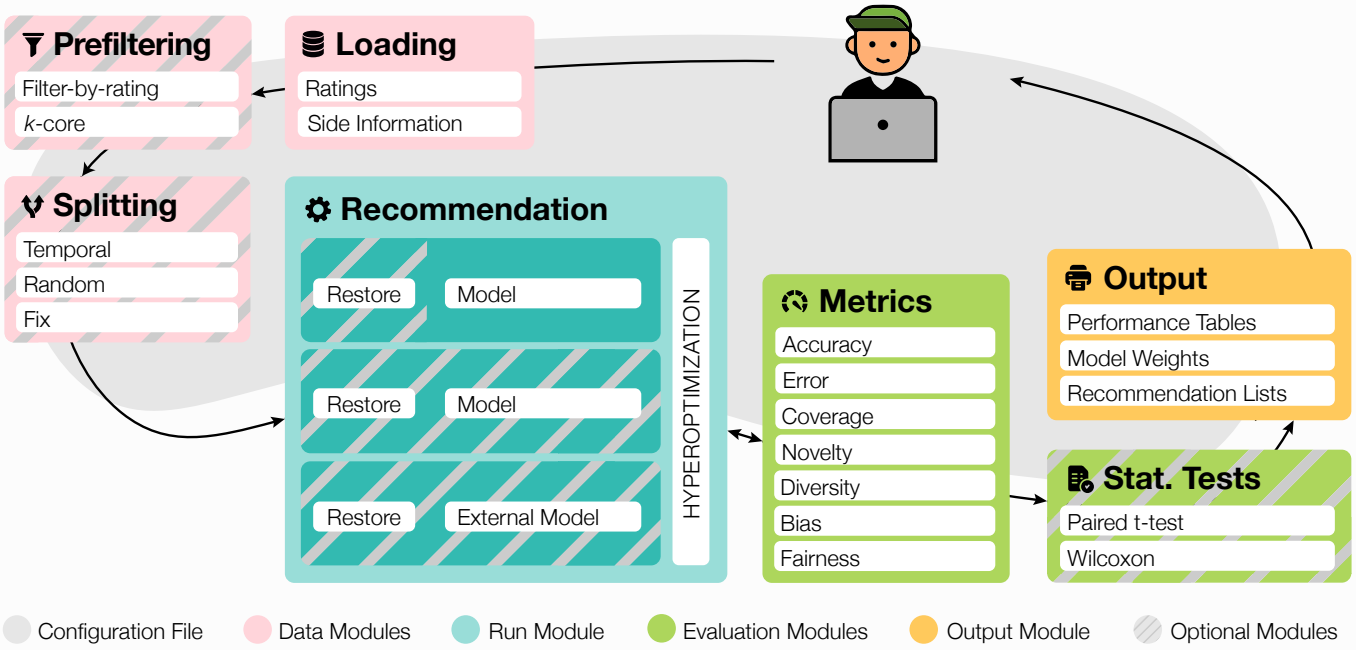
https://experimaestro-python.readthedocs.io/en/latest/
https://experimaestro-ir.readthedocs.io/

Let’s dive right into the hands-on session



Elliot

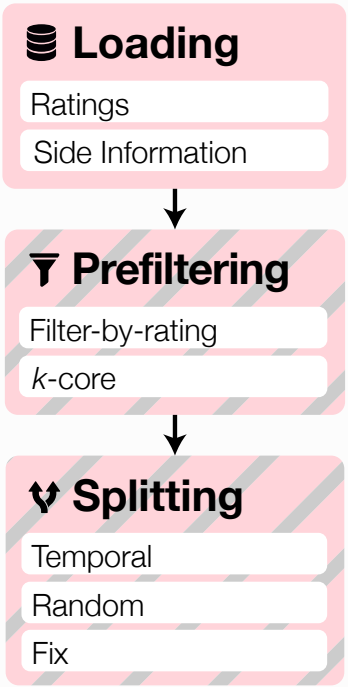
A flexible recommendation framework driven by a simple configuration file



Elliot

```
experiment:
  dataset: movielens_1m
  data_config:
    strategy: dataset
    dataset_path: ../data/movielens_1m/dataset.tsv
  splitting:
    test_splitting:
        strategy: random_subsampling
        test_ratio: 0.2
  models:
    ItemKNN:
      meta:
        hyper_opt_alg: grid
        save_recs: True
      neighbors: [50, 100]
      similarity: cosine
  evaluation:
    simple_metrics: [nDCG]
  top_k: 10
```

Elliot



The Data module is responsible for handling and managing the experiment input, supporting various additional information, e.g., item features, visual embeddings, and images

1. Loading

user-item interaction, (knowledge) side information, visual embeddings

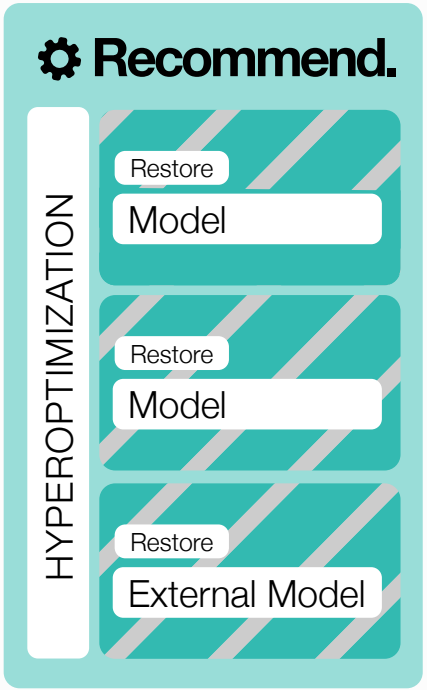
2. Prefiltering

Threshold-based (Global Threshold, User Average) and K-Core strategy (User/Item k-core, iterative k-core, n-round k-core, cold users)

3. Splitting

Time-aware, K-fold Validation, K-random subsampling

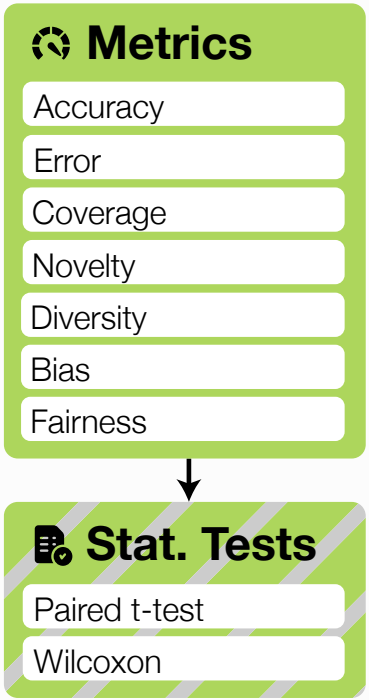
Elliot



Recommendation module provides the functionalities to train (and restore) the ELLIOT recommendation models and the new ones integrated by users
ELLIOT integrates more than 50 recommendation models belonging to different families

- Unpersonalized
- Neighborhood-Based
- Algebric
- Latent Factors
- Content-Based
- Visual Models
- Autoencoders
- Adversarial Learning
- GAN-Based
- Neural Networks-Based

Elliot



ELLIOT is the framework that exposes the largest number of metrics and it considers bias and fairness measures.
It provides 36 evaluation metrics partitioned into seven families.

ELLIOT was been designed for multi-recommender evaluation and handling the fine-grained results, and it brings the opportunity to compute statistical hypothesis tests.

PyTerrier

A common data model

A rich library of transformation functions

e.g., classical retrieval, dense retrieval, LTR, neural re-ranking

Operators to combine transformers

Experimentation API for measurement and comparison

Rich ecosystem of plugins

measures, bi-encoders, cross-encoders, LSR, RAG
easy access to common benchmarks and data (automatically download available data, pre-built indices available)

PyTerrier

Data model

Q

qid	query
0	gold coast temperature november
1	one way flight to gold coast price
…	…

D

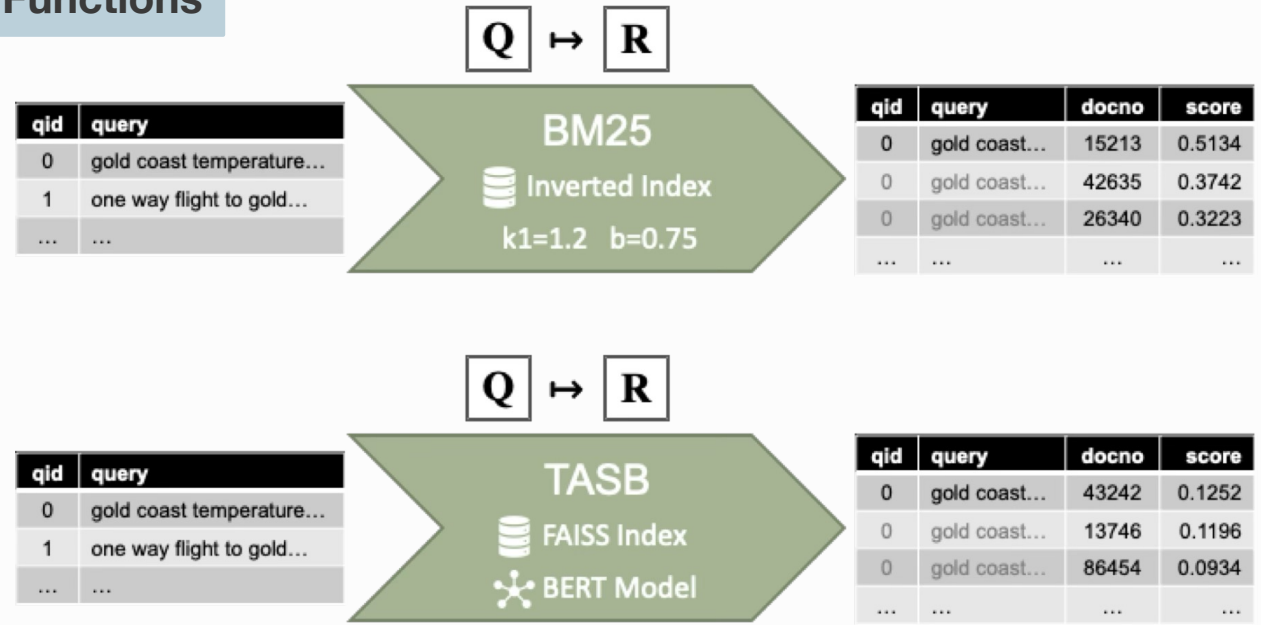
docno	text	title
0	Science & Mathematics Physi…	The hot glowing…
1	School-Age Kids Growth &…	Developmental…
…	…	…

R

qid	query	docno	score
0	gold coast temperature november	15213	0.5134
0	gold coast temperature november	42635	0.3742
0	gold coast temperature november	26340	0.3223
…	…	…	…

PyTerrier

Transformation Functions



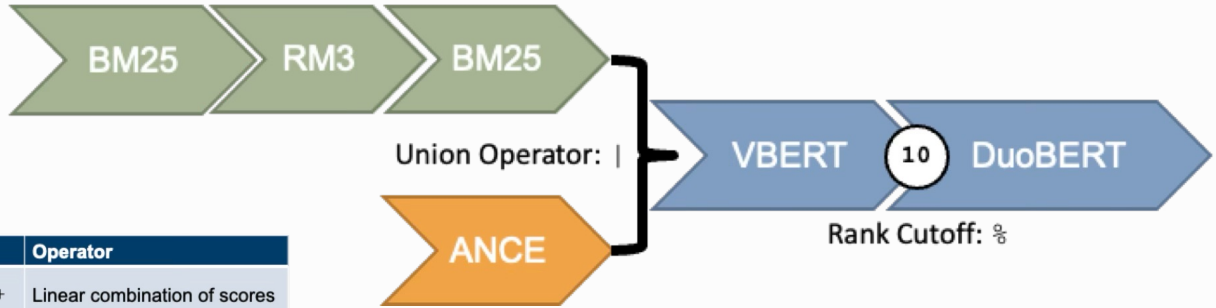
PyTerrier

Transformation Functions

Transformer Class	Examples
Q ↦ R Retrieval	BM25, ColBERT, ANCE, etc.
R ↦ Q Query Expansion	RM3, BO1, etc.
R ↦ R Re-ranking	Vanilla BERT, monoT5, etc.
Q ↦ Q Query Re-writing	SDM, IntenT5, etc.

PyTerrier

Operators

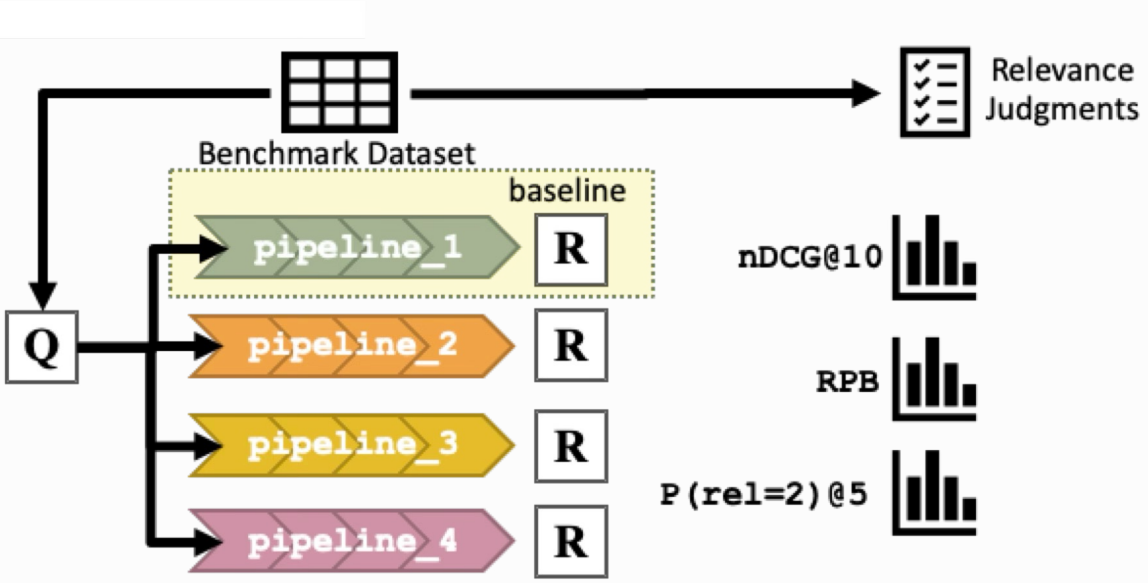


	Operator
>>	Then (chaining)
|	Result union
&	Result intersection
^	Result concatenation
%	Rank cutoff

	Operator
+	Linear combination of scores
*	Scale scores by factor
**	LTR Feature Union
~	Cache result

PyTerrier

Experimentation



Thanks!

Do you have any questions?

 antonio.ferrara@poliba.it

 claudio.pomo@poliba.it

 nicola.tonellotto@unipi.it

Let's (really) dive right into the hands-on session!

Thanks!

Do you have any questions?

 antonio.ferrara@poliba.it

 claudio.pomo@poliba.it

 nicola.tonellotto@unipi.it

Let's (really) dive right into the hands-on session!

References

- Anelli, Vito Walter, et al. "Top-n recommendation algorithms: A quest for the state-of-the-art." Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization. 2022.
- Anelli, Vito Walter, et al. "Reenvisioning the comparison between neural collaborative filtering and matrix factorization." Proceedings of the 15th ACM Conference on Recommender Systems. 2021.
- Rendle, Steffen, et al. "Neural collaborative filtering vs. matrix factorization revisited." Proceedings of the 14th ACM Conference on Recommender Systems. 2020.
- Ferrari Dacrema, Maurizio, et al. "A troubling analysis of reproducibility and progress in recommender systems research." ACM Transactions on Information Systems (TOIS) 39.2 (2021): 1-49.
- Ferrari Dacrema, Maurizio, et al. "Critically examining the claimed value of convolutions over user-item embedding maps for recommender systems." Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020.
- Gorla, Jagadeesh, et al. "Probabilistic group recommendation via information matching." Proceedings of the 22nd international conference on World Wide Web. 2013
- Jambor, Tamas, and Jun Wang. "Goal-driven collaborative filtering–a directional error based approach." European Conference on Information Retrieval. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- Cremonesi, Paolo, Yehuda Koren, and Roberto Turrin. "Performance of recommender algorithms on top-n recommendation tasks." Proceedings of the fourth ACM conference on Recommender systems. 2010.
- Yin, Hongzhi, et al. "Challenging the long tail recommendation." arXiv preprint arXiv:1205.6700 (2012).
- Armstrong, Timothy G., et al. "Improvements that don't add up: ad-hoc retrieval results since 1998." Proceedings of the 18th ACM conference on Information and knowledge management. 2009
- Cockburn, Andy, et al. "Threats of a replication crisis in empirical computer science." Communications of the ACM 63.8 (2020): 70-79.
- Popper, K. All Life Is Problem Solving. Routledge, 1999.
- Burch, Robert. "Charles sanders peirce." (2001).