Opportunities and Challenges of LLMs in Information Retrieval

Chuan Meng
IRLab
University of Amsterdam

17th April 2024

About me

Chuan Meng

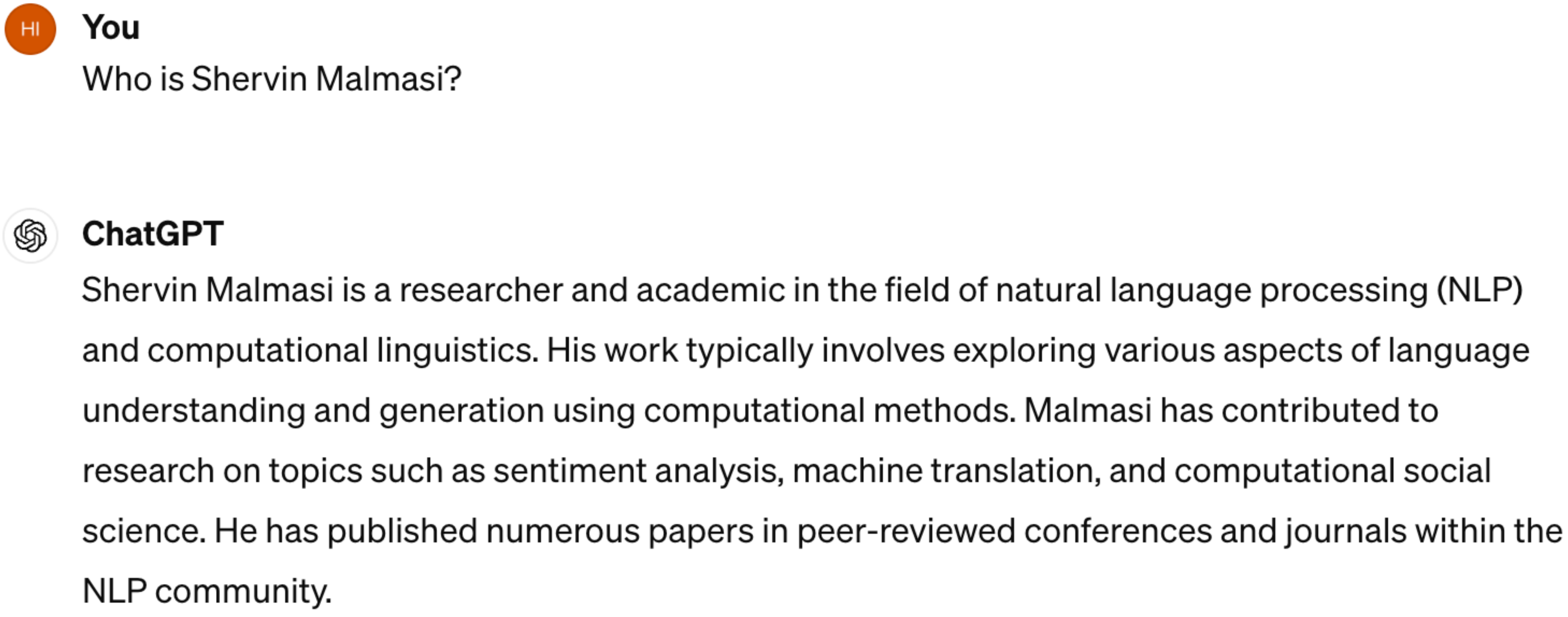
- Third-year PhD student at the University of Amsterdam
- Supervised by Maarten de Rijke, Mohammad Aliannejadi
- Interested in LLM-based
  - mixed-initiative conversational search
  - query performance prediction (QPP)
  - re-ranking/data augmentation
- As of April 2024,
  - 13 papers
  - 176 citations (Google Scholar) with an H-index of 5

Personal website

𝕏 @ChuanMg

Background

- Large language models (LLMs) have remarkable language understanding, generation, generalization, and reasoning abilities



You

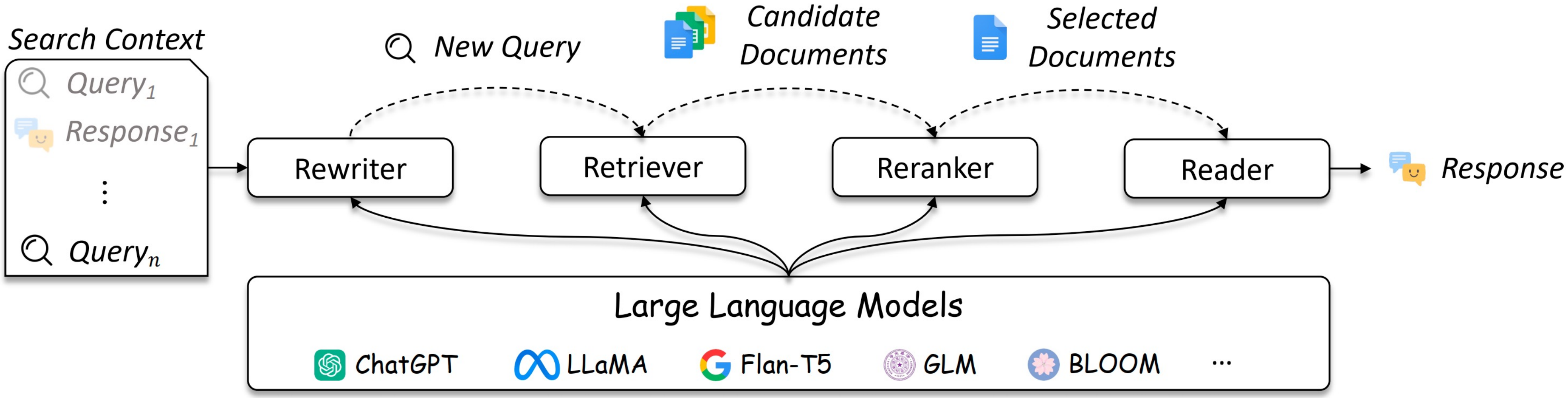
Who is Shervin Malmasi?

ChatGPT

Shervin Malmasi is a researcher and academic in the field of natural language processing (NLP) and computational linguistics. His work typically involves exploring various aspects of language understanding and generation using computational methods. Malmasi has contributed to research on topics such as sentiment analysis, machine translation, and computational social science. He has published numerous papers in peer-reviewed conferences and journals within the NLP community.

Background

- Large language models (LLMs) in information retrieval (IR)
  - Opportunities
    - LLMs directly as task solvers
    - LLMs for data augmentation (e.g., training retrievers/re-rankers)
    - LLMs for evaluation (e.g., generating relevance judgments)
  - Challenges
    - Low efficiency
    - Unfaithful generation
    - …



Background

- Large language models (LLMs) in information retrieval (IR)
  - Opportunities
    - LLMs directly as task solvers [3,6]
    - LLMs for data augmentation (e.g., training retrievers/re-rankers) [4,5]
    - LLMs for evaluation (e.g., generating relevance judgments) [2]
  - Challenges
    - Low efficiency [1]
    - Unfaithful generation
    - …

[1] Ranked List Truncation for Large Language Model-based Re-Ranking. SIGIR 2024
[2] Query Performance Prediction using Relevance Judgments Generated by Large Language Models. arXiv 2024.
[3] LLM-based Retrieval and Generation Pipelines for TREC Interactive Knowledge Assistance Track (iKAT) 2023. TREC 2023.
[4] Expand, Highlight, Generate: RL-driven Document Generation for Passage Reranking. EMNLP 2023.
[5] Self-seeding and Multi-intent Self-instructing LLMs for Generating Intent-aware Information-Seeking dialogs. arXiv 2024.
[6] System Initiative Prediction for Multi-turn Conversational Information Seeking. CIKM 2023

Background

- Large language models (LLMs) in information retrieval (IR)
  - Opportunities
    - LLMs directly as task solvers [3,6]
    - LLMs for data augmentation (e.g., training retrievers/re-rankers) [4,5]
    - **LLMs for evaluation (e.g., generating relevance judgments) [2]**
  - Challenges
    - **Low efficiency [1]**
    - Unfaithful generation
    - …

[1] Ranked List Truncation for Large Language Model-based Re-Ranking. SIGIR 2024
[2] Query Performance Prediction using Relevance Judgments Generated by Large Language Models. arXiv 2024.
[3] LLM-based Retrieval and Generation Pipelines for TREC Interactive Knowledge Assistance Track (iKAT) 2023. TREC 2023.
[4] Expand, Highlight, Generate: RL-driven Document Generation for Passage Reranking. EMNLP 2023.
[5] Self-seeding and Multi-intent Self-instructing LLMs for Generating Intent-aware Information-Seeking dialogs. arXiv 2024.
[6] System Initiative Prediction for Multi-turn Conversational Information Seeking. CIKM 2023

Outline

- ☐ Study 1: Ranked List Truncation for Large Language Model-based Re-Ranking [15 min]
- ☐ Study 2: Query Performance Prediction using Relevance Judgments Generated by Large Language Models [15 min]
- ☐ Conclusion [5 min]

Outline

- ☐ **Study 1: Ranked List Truncation for Large Language Model-based Re-Ranking** [15 min]
- ☐ Study 2: Query Performance Prediction using Relevance Judgments Generated by Large Language Models [15 min]
- ☐ Conclusion [5 min]

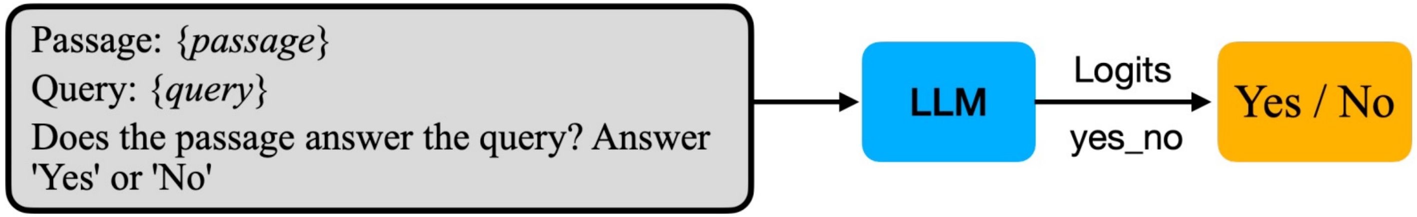
Ranked List Truncation for Large Language Model-based Re-Ranking

Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi,
Maarten de Rijke

SIGIR 2024

Background

- Large language models (LLMs) as text re-rankers
    - achieve state-of-the-art performance
    - hard to be applied in practice due to significant computational overhead
        - the average query latency (re-ranking 100 items per query) for Flan-t5-xxl (11B) of is around 4 seconds, on a NVIDIA RTX A6000 GPU [1]



Passage: {*passage*}
Query: {*query*}
Does the passage answer the query? Answer 'Yes' or 'No'

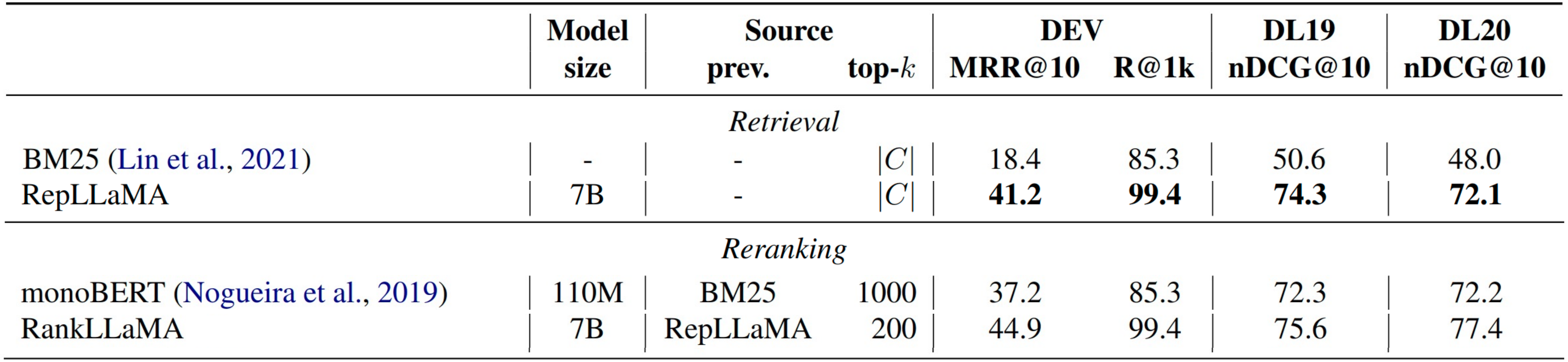
LLM-based re-ranker

---

[1] Zhuang et al. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. SIGIR 2024.

Motivation

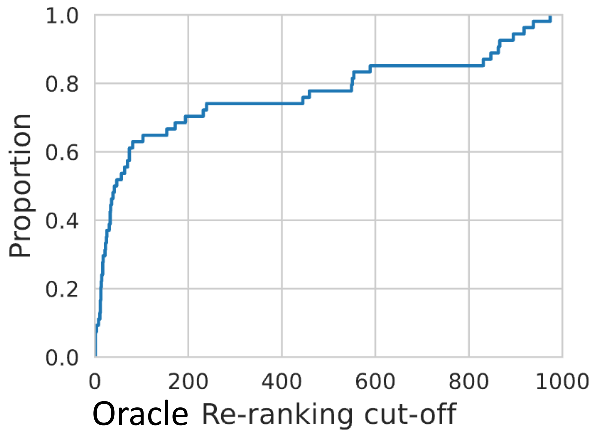
- Limitation
  - Applying a fixed re-ranking cut-off (e.g., 200, 1000) to all queries



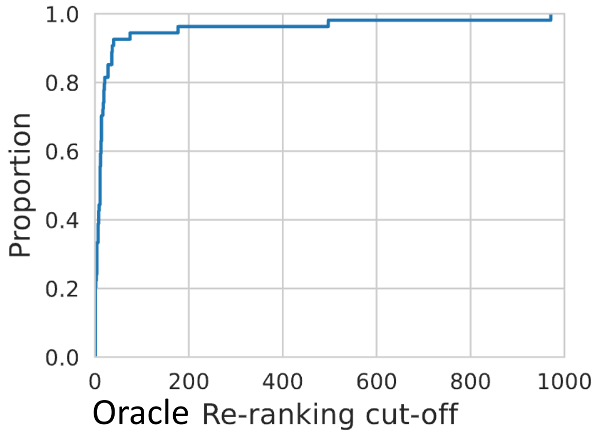
	Model	**Source**		**DEV**		**DL19**	**DL20**
	size	**prev.**	**top-k**	**MRR@10**	**R@1k**	**nDCG@10**	**nDCG@10**
		Retrieval					
BM25 (Lin et al., 2021)	-	-	$\|C\|$	18.4	85.3	50.6	48.0
RepLLaMA	7B	-	$\|C\|$	**41.2**	**99.4**	**74.3**	**72.1**
		Reranking					
monoBERT (Nogueira et al., 2019)	110M	BM25	1000	37.2	85.3	72.3	72.2
RankLLaMA	7B	RepLLaMA	200	44.9	99.4	75.6	77.4

Motivation (fixed cut-offs vs. query-specific cut-offs)

- Query-specific re-ranking cut-offs improve *efficiency*
  - Individual queries have different oracle cut-offs with a wide range
  - A deeper fixed cut-off wastes computational resources
  - A shallower fixed cut-off hurts re-ranking quality for queries needing a deeper cut-off



Cumulative distribution function of oracle cut-offs for
BM25–RankLLaMA
TREC-DL 20

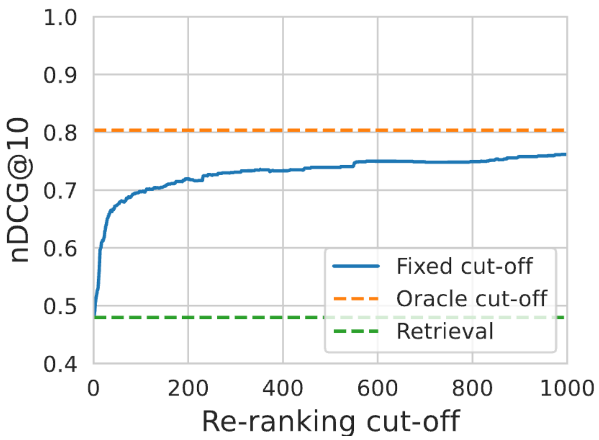


Cumulative distribution function of oracle cut-offs for
RepLLaMA–RankLLaMA
TREC-DL 20

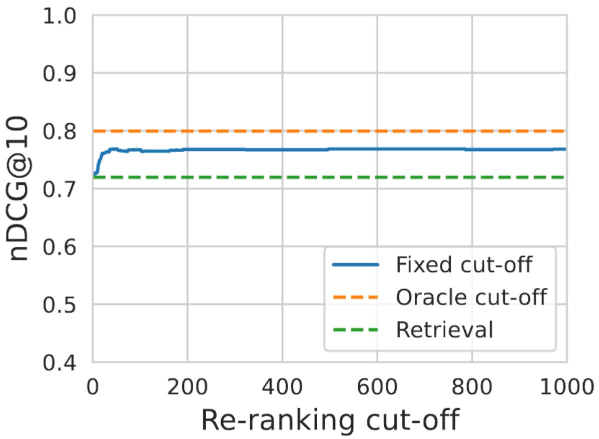
For a query, an oracle cut-off is the minimum re-ranking cutoff producing the highest nDCG@10 value

Motivation (fixed cut-offs vs. query-specific cut-offs)

- Query-specific re-ranking cut-offs improve *effectiveness*
  - Oracle cut-offs show statistically significant improvements over all fixed cut-offs
  - A deeper fixed cut-off
    - does not always result in improvement (consistent with [1])
    - even is detrimental to re-ranking quality (consistent with [1])



BM25–RankLLaMA
TREC-DL 20



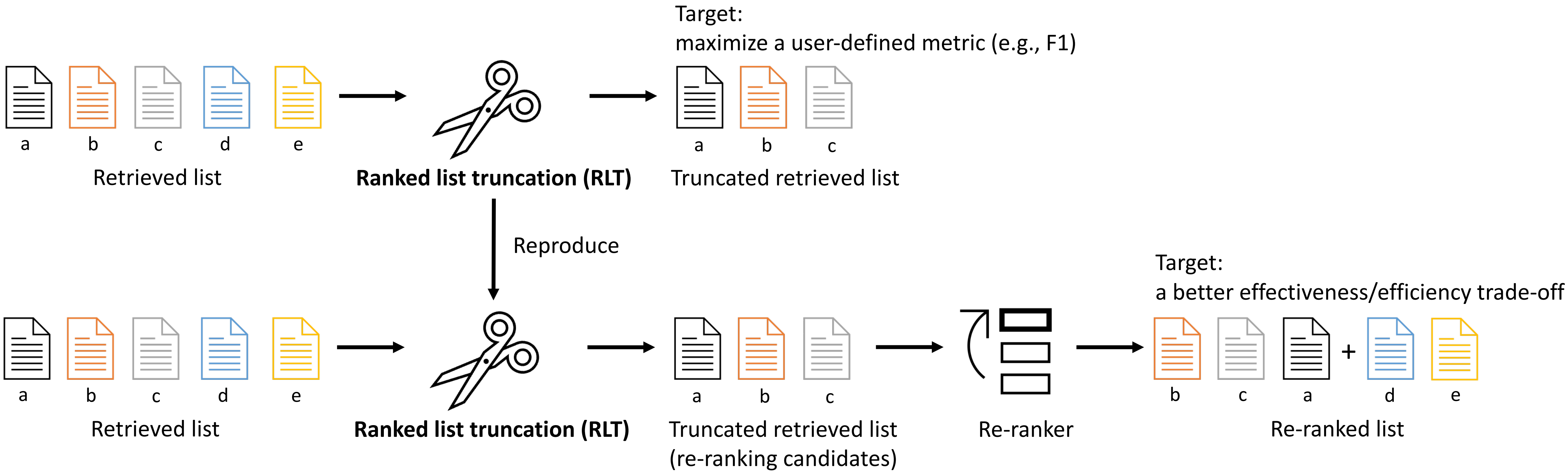
RepLLaMA–RankLLaMA
TREC-DL 20

[1] Zamani et al. Stochastic Retrieval-Conditioned Reranking. In ICTIR 2022.

Motivation (How to predict query-specific cut-offs)

- Ranked list truncation (RLT)
  - predicts how many items in a ranked list should be returned
  - optimizes the truncated ranked list regarding a user-defined metric (e.g., F1)
  - aids applications where reviewing returned items is costly, e.g., patent or legal search

- **We reproduce exiting RLT methods in the context of re-ranking**



Target:
maximize a user-defined metric (e.g., F1)

Retrieved list

Ranked list truncation (RLT)

Truncated retrieved list

Reproduce

Target:
a better effectiveness/efficiency trade-off

Retrieved list

Ranked list truncation (RLT)

Truncated retrieved list
(re-ranking candidates)

Re-ranker

Re-ranked list

Reproducibility methodology

- Verify 4 findings on RLT:
  - *Finding 1: Supervised RLT methods generally perform better than their unsupervised counterparts (e.g., set a fixed cut-off)*
  - *Finding 2: Distribution-based supervised RLT methods perform better than their sequential labeling-based counterpart*
  - *Finding 3: Jointly learning RLT with other tasks results in better RLT quality*
  - *Finding 4: When truncating a retrieved list returned by a neural-based retriever, incorporating its embeddings improves RLT quality*

Reproducibility methodology

- *Do RLT methods generalize to the context of*
  - *(RQ1) LLM-based re-ranking with a lexical first-stage retriever?*
  - *(RQ2) LLM-based re-ranking with learned sparse or dense first-stage retrievers?*
  - *(RQ3) pre-trained language model-based re-ranking?*

Reproducibility methodology

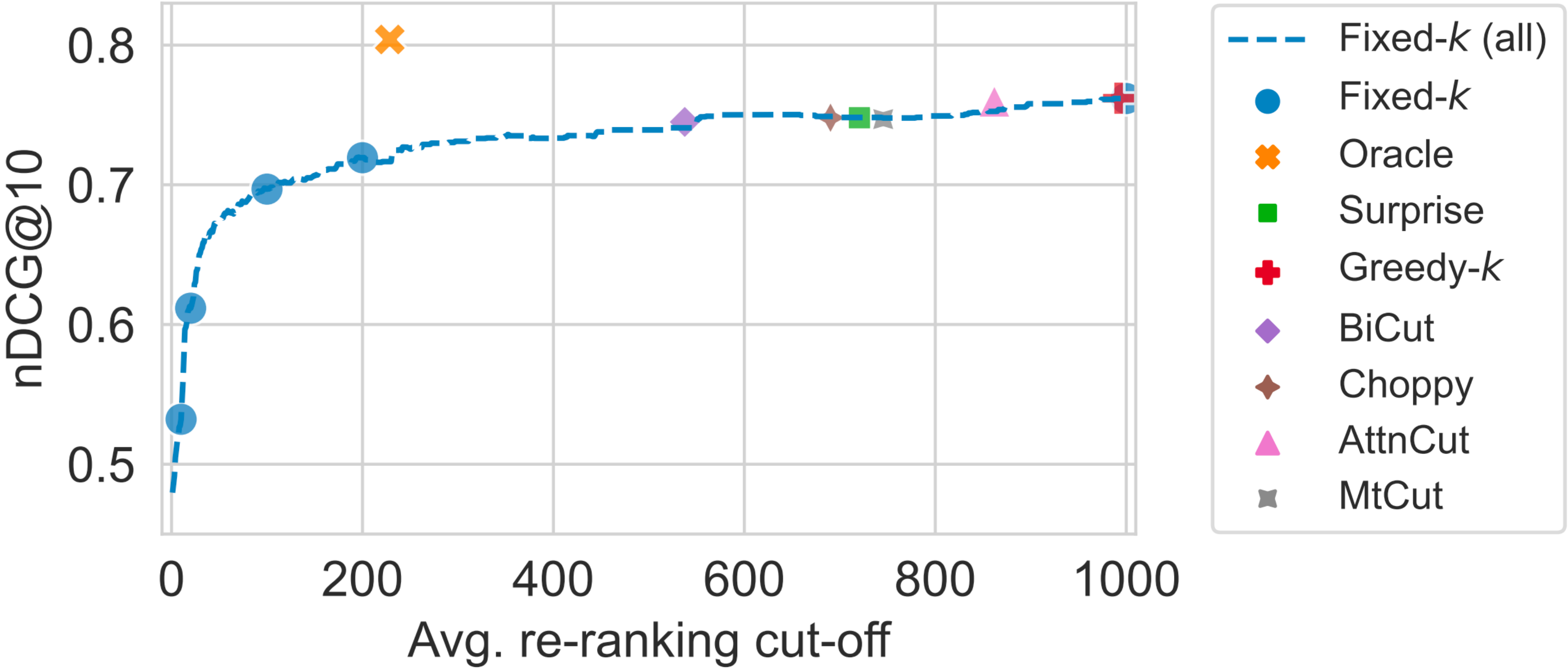
- Experimental settings:
  - 8 RLT methods

Method	Attribute 1	Attribute 2	Attribute 3
Fixed-k (10, 20, 100, 200, 1000)	Unsupervised	-	-
Greedy-k	Unsupervised	-	-
Surprise	Unsupervised	-	-
BiCut	Supervised	Sequential labeling-based	-
Choppy	Supervised	Distribution-based	-
AttnCut	Supervised	Distribution-based	-
MtCut	Supervised	Distribution-based	Jointly learning with other tasks
LeCut	Supervised	Distribution-based	Use retriever embeddings

  - Datasets:
    - TREC-DL 19, TREC-DL 20

Experiments

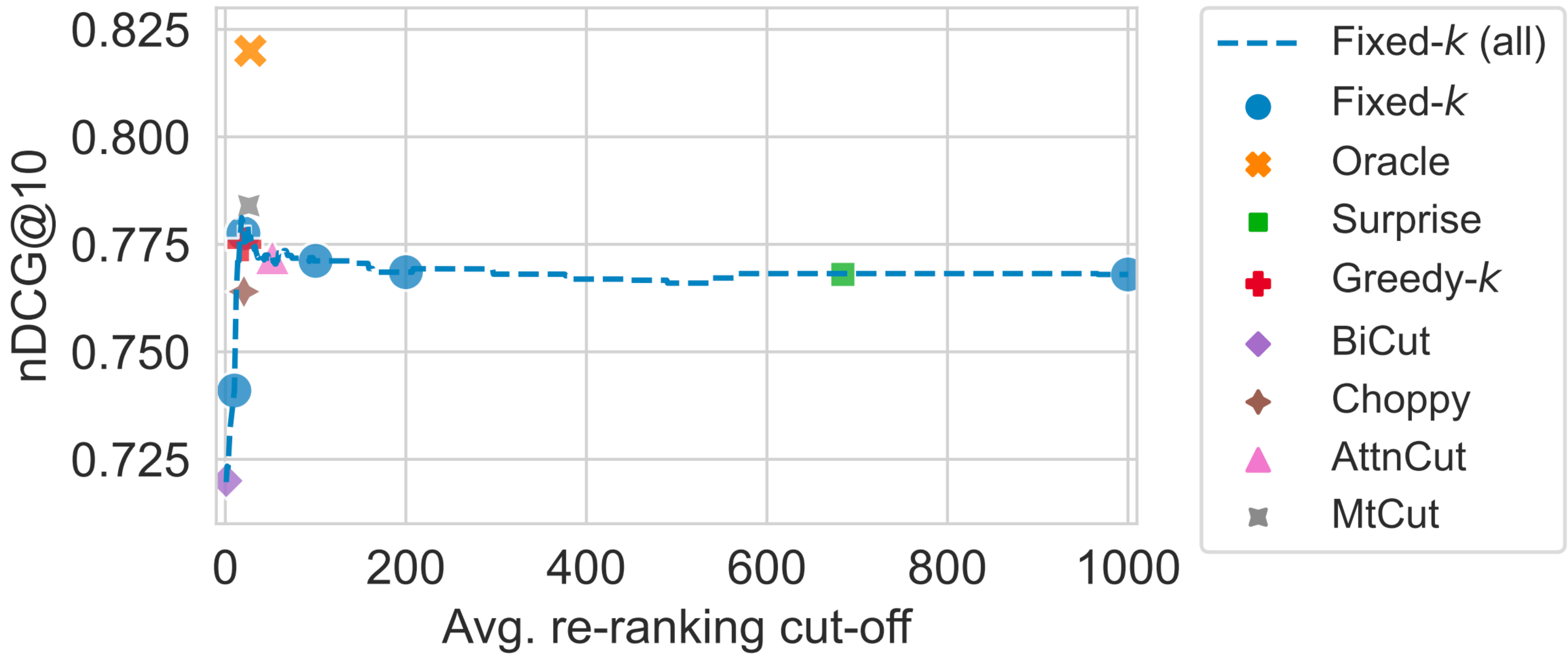
- RQ1: Do RLT methods generalize to the context of LLM-based re-ranking with a lexical first-stage retriever?
  - Fixed re-ranking depths can closely approximate the results of supervised methods
  - Supervised RLT methods do not show a clear advantage over fixed re-ranking depths



BM25–RankLLaMA
TREC-DL 20

Experiments

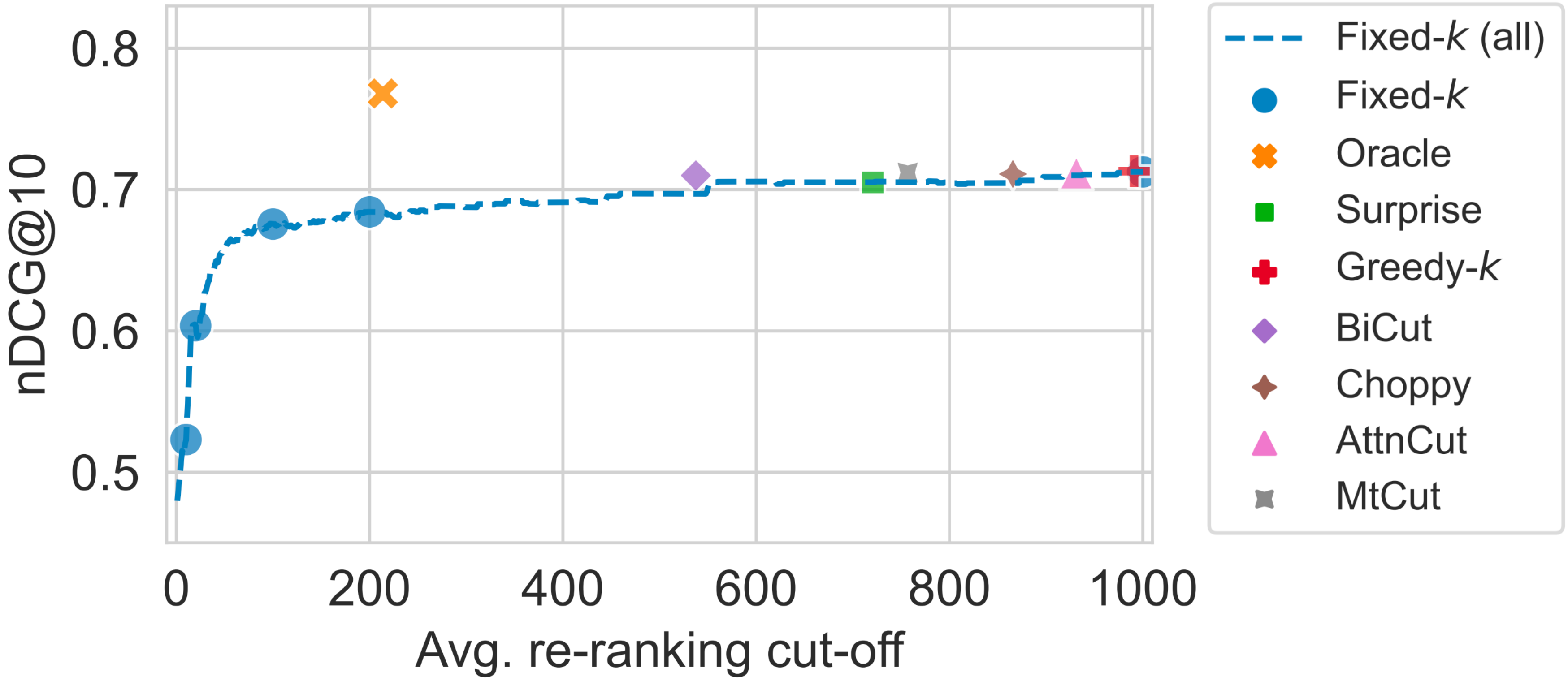
- RQ2: Do RLT methods generalize to the context of LLM-based re-ranking with learned sparse or dense first-stage retriever?
  - Supervised methods do not lead to significant improvement in terms nDCG@10
  - A fixed re-ranking depth of 20 achieves the best effectiveness/efficiency trade-off



Splade++–RankLLaMA
TREC-DL 20

Experiments

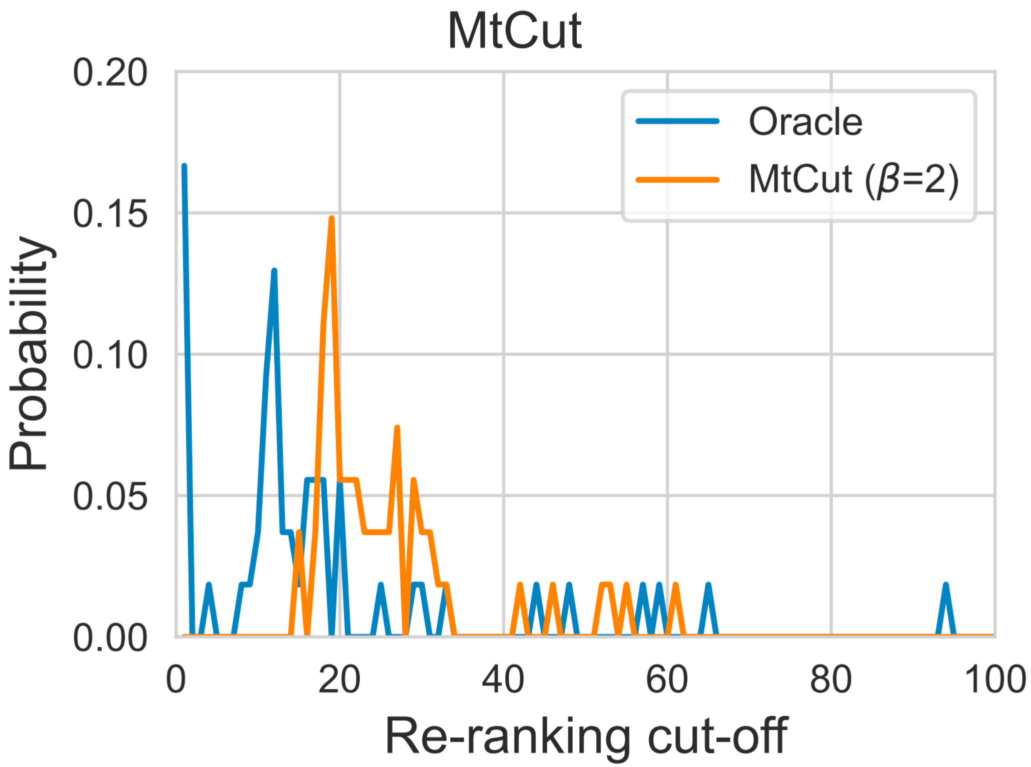
- RQ3: Do RLT methods generalize to the context of pre-trained language model-based re-ranking?
  - Results are similar to RQ1



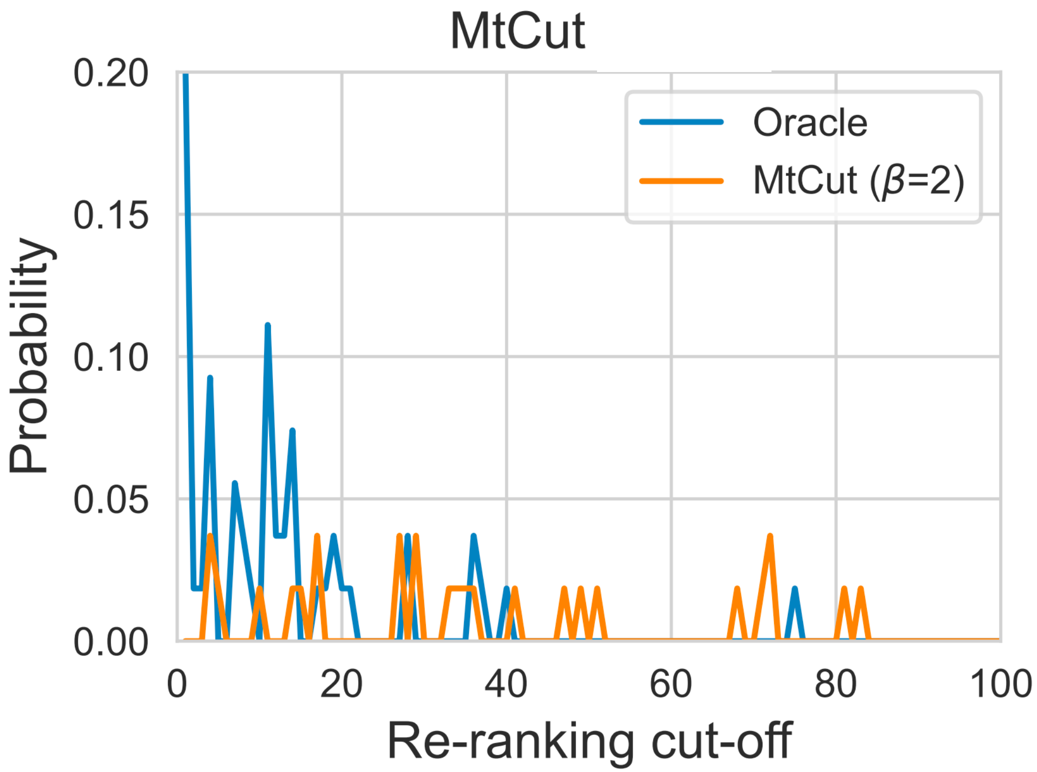
BM25–monoT5
TREC-DL 20

Experiments

- Error analysis for supervised RLT methods
  - They fail to predict a re-ranking cut-off of zero
  - They perform worse when truncating RepLLaMA's retrieved lists



Splade++–RankLLaMA
TREC-DL 20



RepLLaMA–RankLLaMA
TREC-DL 20

Takeaways

- We showed that findings on RLT do not generalize well to this new setup:
  - *Finding 1: Supervised RLT methods generally perform better than their unsupervised counterparts (e.g., set a fixed cut-off)*
  - *Finding 2: Distribution-based supervised RLT methods perform better than their sequential labeling-based counterpart*
  - *Finding 3: Jointly learning RLT with other tasks results in better RLT quality*
  - *Finding 4: When truncating a retrieved list returned by a neural-based retriever, incorporating its embeddings improves RLT quality*

Takeaways

- The type of retriever makes a difference
  - With an effective retriever (e.g., SPLADE++/RepLLaMA)
    - A fixed re-ranking depth of 20 yields an excellent effectiveness/efficiency trade-off
    - A fixed depth>20 does not significantly improve re-ranking quality

- The type of re-ranker (LLM or pre-trained LM-based) does not appear to influence the findings

- Supervised RLT methods need to improve their ability to predict “0”

Conclusion

- Contributions
  - An empirical analysis in the context of LLM-based re-ranking, shows that
    - Effective query-specific re-ranking depths can improve re-ranking efficiency and effectiveness
  - We reproduce RLT methods in the context of LLM-based re-ranking
  - The data and code are open-source https://github.com/ChuanMeng/RLT4Reranking
- Future work
  - Explore RLT for pairwise and listwise LLM-based re-rankers
  - Develop new RLT methods for LLM-based re-ranking

QR code for the repo

Outline

- ☐ Study 1: Ranked List Truncation for Large Language Model-based Re-Ranking [15 min]
- ☐ **Study 2: Query Performance Prediction using Relevance Judgments Generated by Large Language Models** [15 min]
- ☐ Conclusion [5 min]

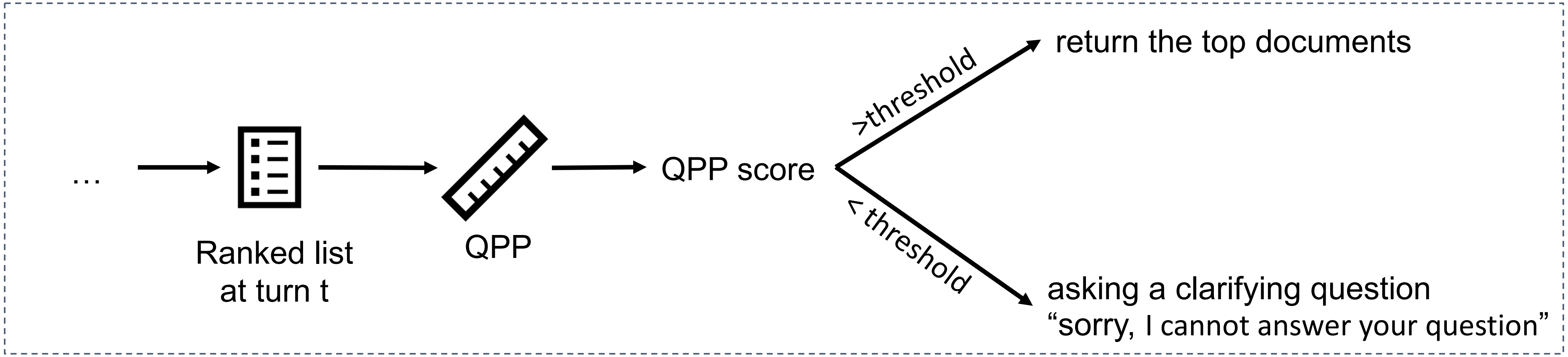
Query Performance Prediction using Relevance Judgments Generated by Large Language Models

Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi,
Maarten de Rijke

Background—Query performance prediction

- Query performance prediction (QPP)
  - Predicts retrieval quality of search system for query without human-labeled relevance judgments

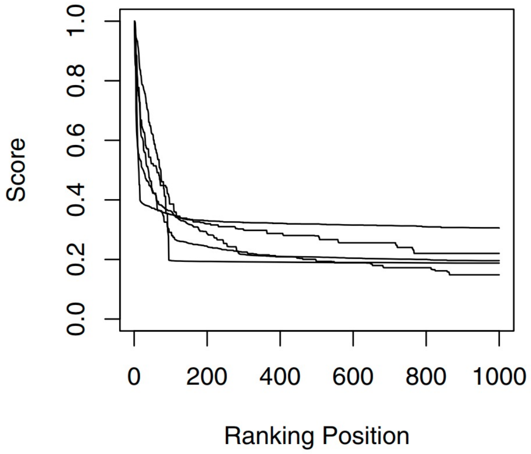
- QPP benefits a variety of applications, e.g., action prediction in conversational search

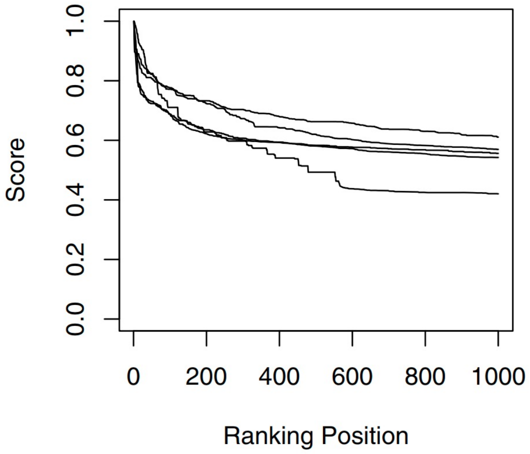


Background—Query Performance Prediction

- There are two types of QPP methods
  - Pre-retrieval QPP methods
    - $f(query) \rightarrow QPP$ score
  - Post-retrieval QPP methods
    - $f(query, a\ ranked\ list) \rightarrow QPP\ score$

- Post-retrieval QPP methods
  - Unsupervised post-retrieval QPP methods





  - Supervised post-retrieval QPP methods
    - BERT $(query, a\ ranked\ list) \rightarrow\ QPP\ score$

Motivation

- Existing QPP methods typically
  - return only a single real-valued score that indicates the retrieval quality for a query
  - do not require the predicted score to approximate a specific IR evaluation metric

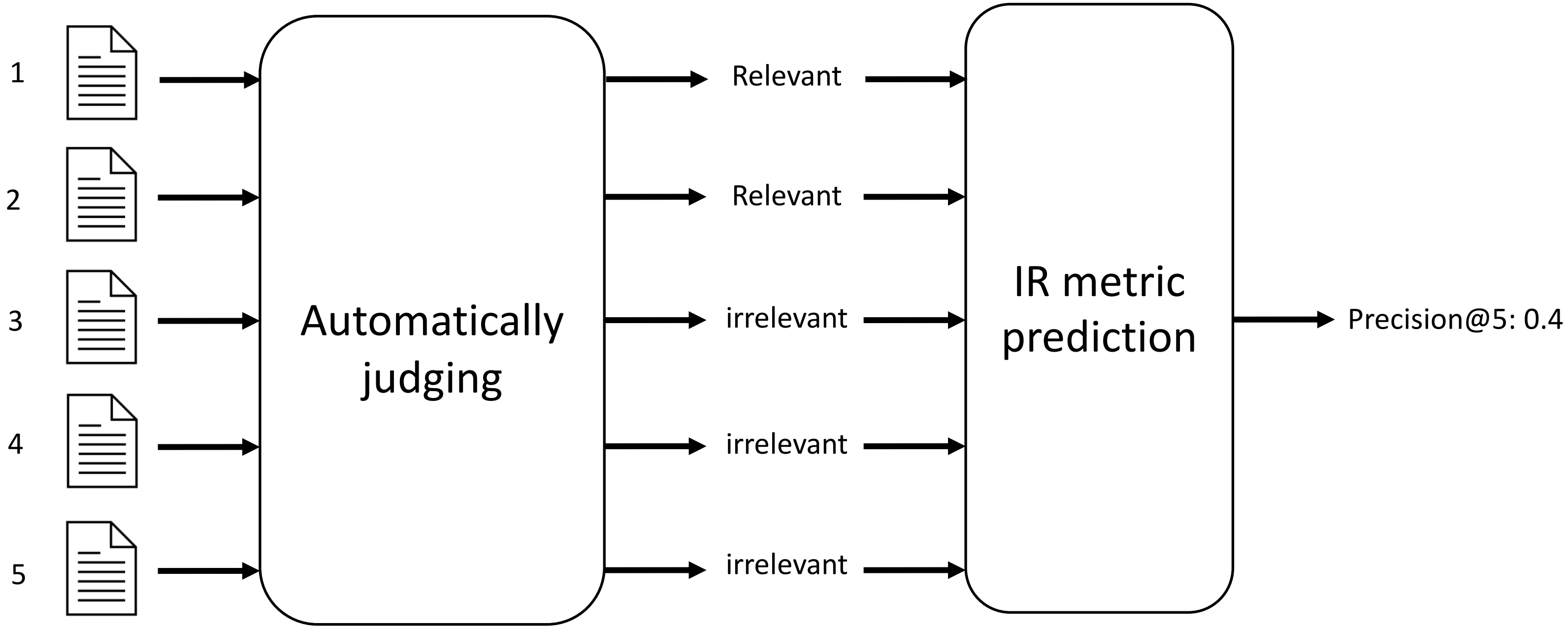
- Limitations:
  - Using a single value to represent different IR evaluation metrics leads to a "one size fits all" issue; some IR metrics do not correlate well [1]

  - Single-score prediction limits the interpretability of QPP

[1] Gupta et al. Correlation, Prediction and Ranking of Evaluation Metrics in Information Retrieval. ECIR 2019.

Methodology

- Propose a **QPP** framework using automatically **Gen**erated **RE**levance judgments (QPP-GenRE
  - Decompose QPP into independent subtasks of automatically judging the relevance of each item in a ranked list to a given query



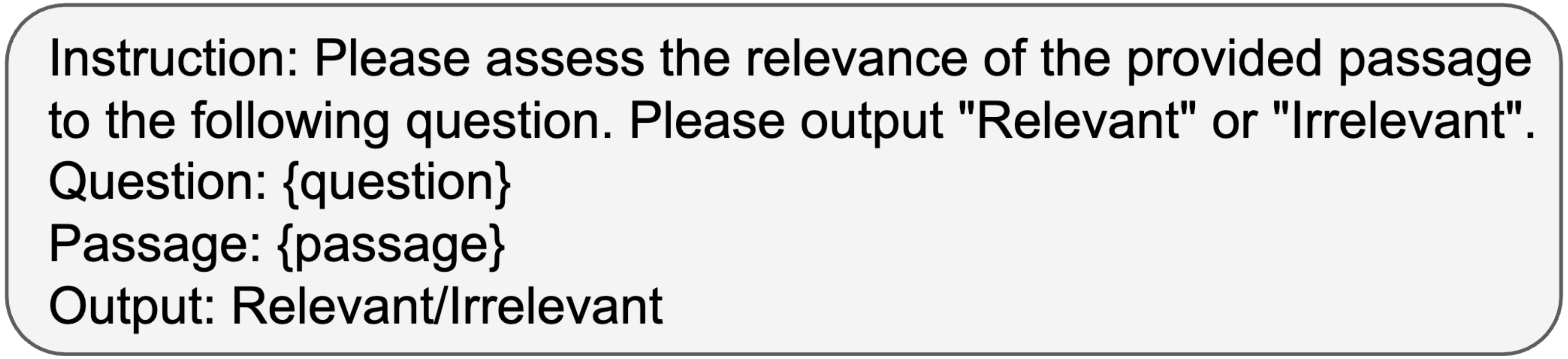
Methodology

- Challenges
  - Unlike prompting commercial LLMs [1,2], prompting open-source LLMs in a zero-/few-shot way results in limited performance of relevance judgments
  - Predicting recall-oriented metrics requires seeking all relevant items in the corpus for a query, leading to high computational costs

[1] Faggioli et al. Perspectives on Large Language Models for Relevance Judgment. In ICTIR 2023.
[2] Thomas et al. Large Language Models Can Accurately Predict Searcher Preferences. In arXiv 2023.

Methodology

- Solutions
  - Train an open-source LLM (LLaMA) on human-labeled relevance judgments
    - Use a parameter-efficient fine-tuning method, QLoRA



Instruction: Please assess the relevance of the provided passage to the following question. Please output "Relevant" or "Irrelevant".
Question: {question}
Passage: {passage}
Output: Relevant/Irrelevant

  - Devise an approximation strategy for predicting recall-oriented metrics
    - Only judge a few items in the ranked list for a query and use them to estimate the metric

Experiments

- Experimental settings:
  - QPP baselines
    - 10 unsupervised QPP ones
    - 4 supervised QPP ones
  - Datasets:
    - TREC-DL 19, 20, 21, 22
  - Rankers:
    - BM25
    - ANCE
  - Target metrics
    - RR@10
    - nDCG@10
  - Evaluation metrics
    - Pearson's ρ and Kendall's τ correlation between actual IR metric values and predicted metric values

Experiments

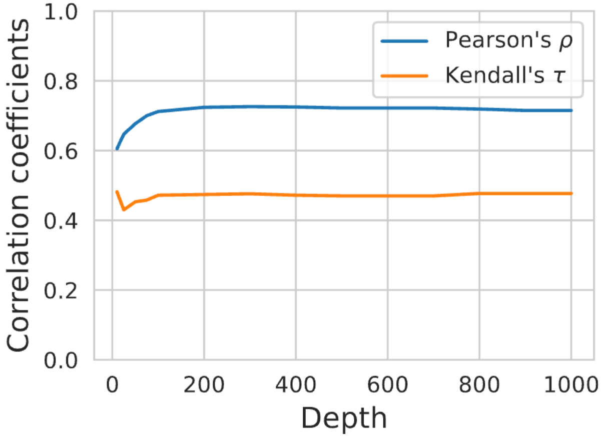
- 4 Research questions
  - *RQ1: To what extent does QPP-GenRE improve QPP effectiveness for lexical and neural rankers in terms of RR@10 compared to state-of-the-art baselines*
  - *RQ2 To what extent does QPP-GenRE improve QPP effectiveness for lexical and neural rankers in terms of nDCG@10 compared to state-of-the-art baselines?*
  - *RQ3: How deep do we need to automatically judge in a ranked list to effectively predict nDCG@10?*
  - *RQ4: To what extent does fine-tuning LLaMA impact the quality of the generated relevance judgments and QPP effectiveness?*

Experiments

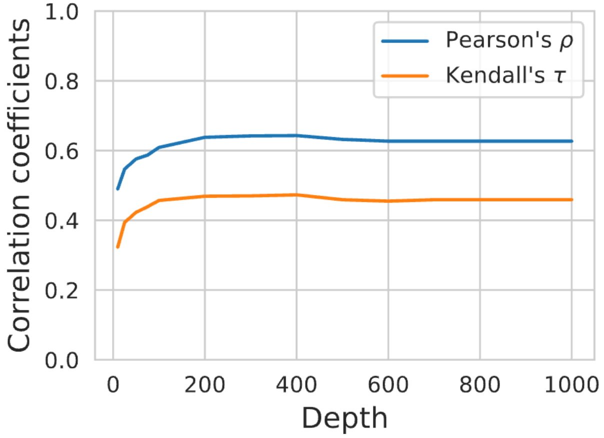
- RQ1 & 2
  - QPP-GenRE achieves state-of-the-art QPP quality
    - in estimating the retrieval quality of BM25 (lexical) and ANCE (dense)
    - in terms of RR@10 (precision) and nDCG@10 (recall)

Experiments

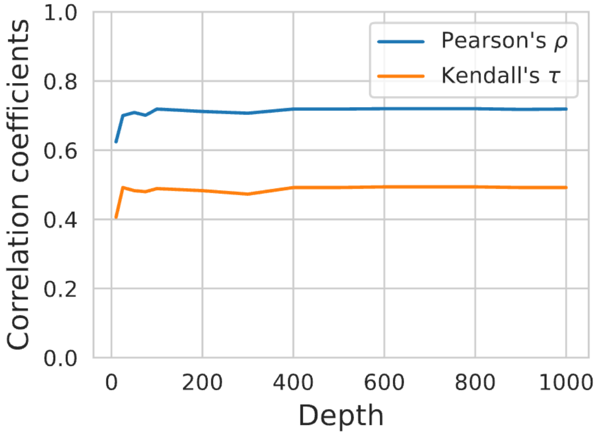
- RQ3: How deep do we need to automatically judge in a ranked list to effectively predict nDCG@10?
  - Judging up to 100–200 retrieved items in a ranked list can reach saturation
  - QPP-GenRE can achieve state-of-the-art QPP at shallow judging depth 10



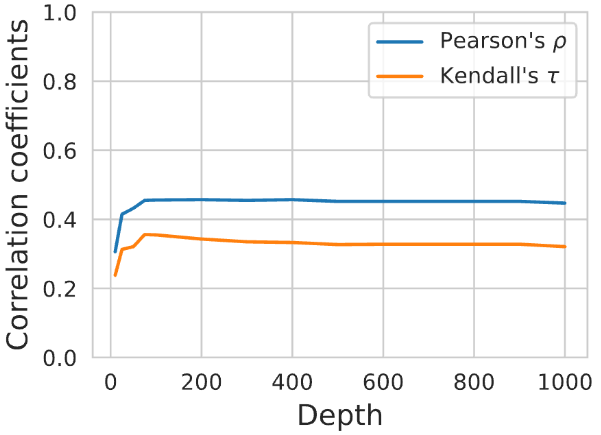
BM25
TREC-DL 19



BM25
TREC-DL 20



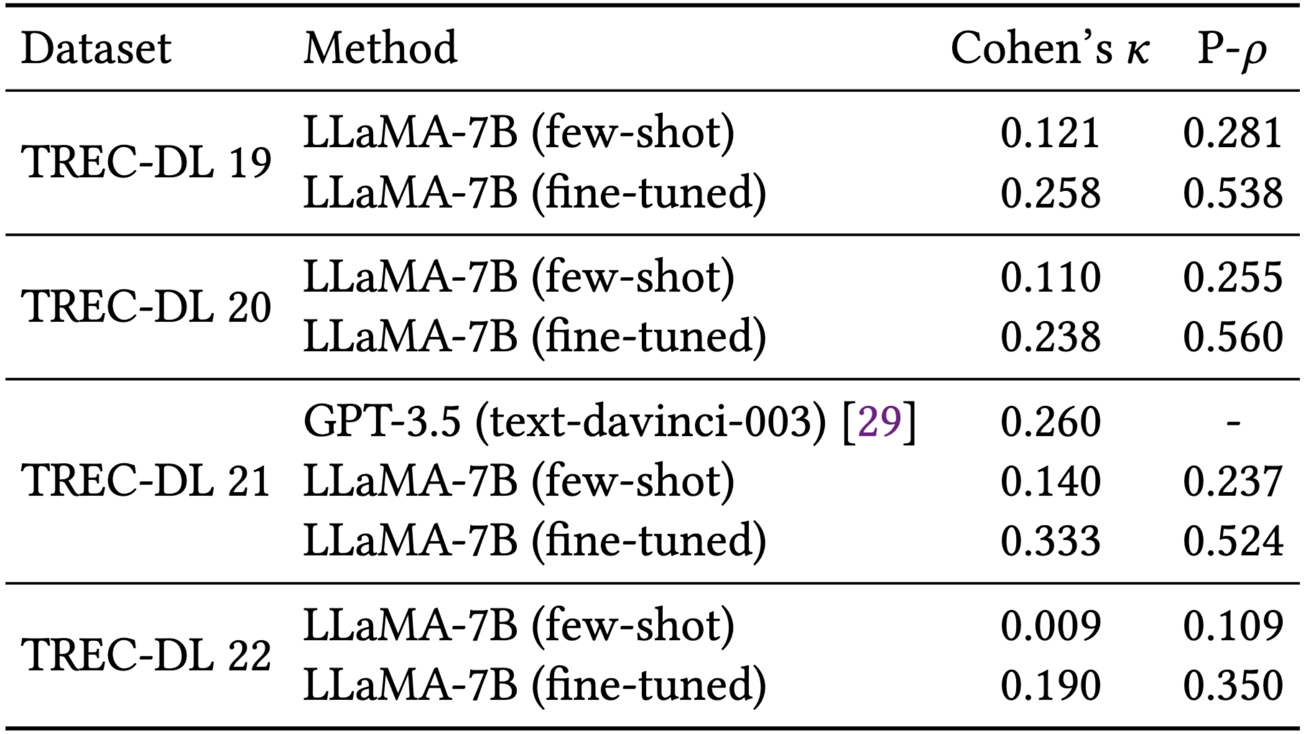
ANCE
TREC-DL 19



ANCE
TREC-DL20

Experiments

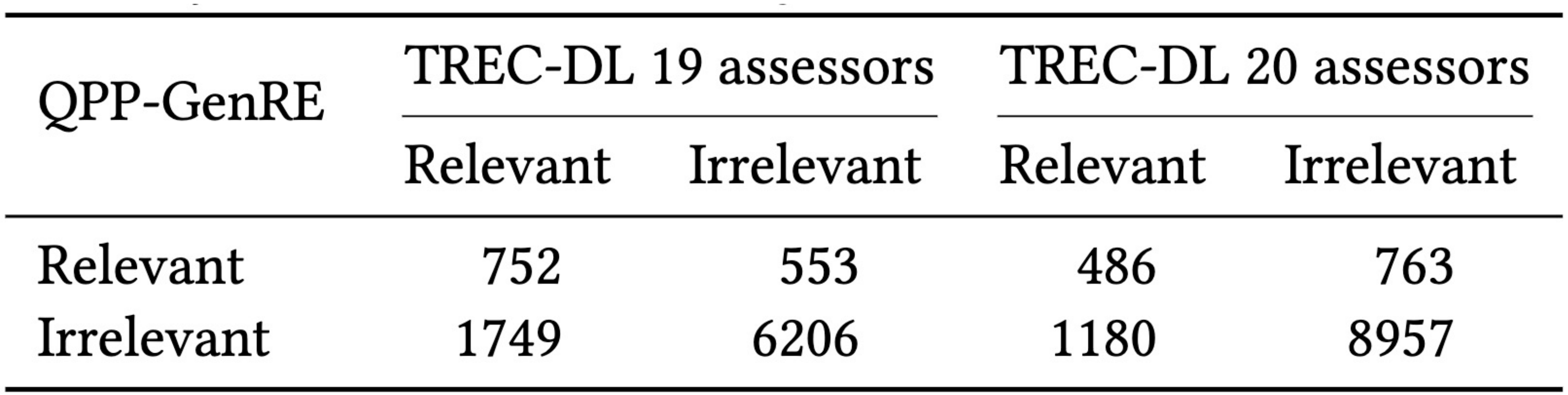
- RQ4: To what extent does fine-tuning LLaMA impact the quality of the generated relevance judgments and QPP effectiveness?
  - Fine-tuned LLaMA performs better few-shot LLaMA and GPT-3.5 regarding relevance prediction
  - Better quality in generating relevance judgments translates into better QPP quality



Dataset	Method	Cohen's κ	P-ρ
TREC-DL 19	LLaMA-7B (few-shot)	0.121	0.281
	LLaMA-7B (fine-tuned)	0.258	0.538
TREC-DL 20	LLaMA-7B (few-shot)	0.110	0.255
	LLaMA-7B (fine-tuned)	0.238	0.560
TREC-DL 21	GPT-3.5 (text-davinci-003) [29]	0.260	-
	LLaMA-7B (few-shot)	0.140	0.237
	LLaMA-7B (fine-tuned)	0.333	0.524
TREC-DL 22	LLaMA-7B (few-shot)	0.009	0.109
	LLaMA-7B (fine-tuned)	0.190	0.350

Experiments

- Error analysis
  - QPP-GenRE tends to wrongly predict some relevant items as irrelevant (false negatives)



QPP-GenRE	TREC-DL 19 assessors		TREC-DL 20 assessors	
	Relevant	Irrelevant	Relevant	Irrelevant
Relevant	752	553	486	763
Irrelevant	1749	6206	1180	8957

Conclusion

- Contributions
  - Propose a new QPP framework, QPP-GenRE, which predicts IR metrics based on automatically generated relevance judgments
  - Fine-tune open-source LLMs for generating relevance judgments
  - Devise an approximation strategy for predicting a recall-oriented IR measure

  - QPP-GenRE achieves state-of-the-art QPP quality

  - The data and code are open-sourced https://github.com/ChuanMeng/QPP-GenRE

QR code for the repo

Outline

- ❑ Study 1: Ranked List Truncation for Large Language Model-based Re-Ranking [15 min]
- ❑ Study 2: Query Performance Prediction using Relevance Judgments Generated by Large Language Models [15 min]
- ❑ **Conclusion** **[5 min]**

Conclusion and Future Work

- Contributions
  - The challenge of low efficiency:
    - Improve the efficiency of LLM-based re-ranking by using query-specific re-ranking cut-offs
  - The opportunity for LLMs for evaluation
    - A new QPP framework using LLM-based generated relevance judgments
    - Fine-tune open-source LLMs to generate relevance judgments

- Future work
  - Propose new RLT methods for LLM-based re-ranking
  - Investigate the performance of other open-source LLMs
  - Domain-specific scenarios

Thank you!

Chuan Meng

✉ c.meng@uva.nl
𝕏 @ChuanMg
https://chuanmeng.github.io