



System Initiative Prediction and Query Performance Prediction for Conversational Information Seeking

Chuan Meng

IRLab

University of Amsterdam

3rd November 2023

Background

- Conversational information seeking (CIS) is concerned with sequences of interactions between one or more users and a system, in which the system's goal is to satisfy the users' information needs



Apple price?



The price of an Apple product, such as an iPhone, iPad, Macbook, or other Apple devices, can vary widely depending on the specific model, configuration, and where you are purchasing it. Apple offers a range of products with different features and price points.



To get the current price for a specific Apple product, I recommend visiting the official Apple website or checking with authorized Apple retailers or online stores. Prices may change over time due to factors like product updates, promotions, and regional variations.

Background

- Mixed-initiative conversational information seeking
 - User and system can both take initiative at different times in conversation
 - System initiative-taking has the potential to offend users
- When to take the initiative in a conversation?
 - System initiative prediction
 - Query performance prediction (QPP)

Outline

- ❑ Study 1: System initiative prediction for CIS (CIKM 2023) [12 min]
- ❑ Study 2: QPP for CIS: reproducing existing QPP methods in CIS (SIGIR 2023) [12 min]
- ❑ Study 3: QPP for CIS: improve QPP for CIS using query rewriting quality (QPP++2023) [6 min]
- ❑ Conclusion [5 min]

Outline

- ❑ **Study 1: System initiative prediction for CIS (CIKM 2023) [12 min]**
- ❑ Study 2: QPP for CIS: reproducing existing QPP methods in CIS (SIGIR 2023) [12 min]
- ❑ Study 3: QPP for CIS: improve QPP for CIS using query rewriting quality (ECIR 2023) [6 min]
- ❑ Conclusion [5 min]

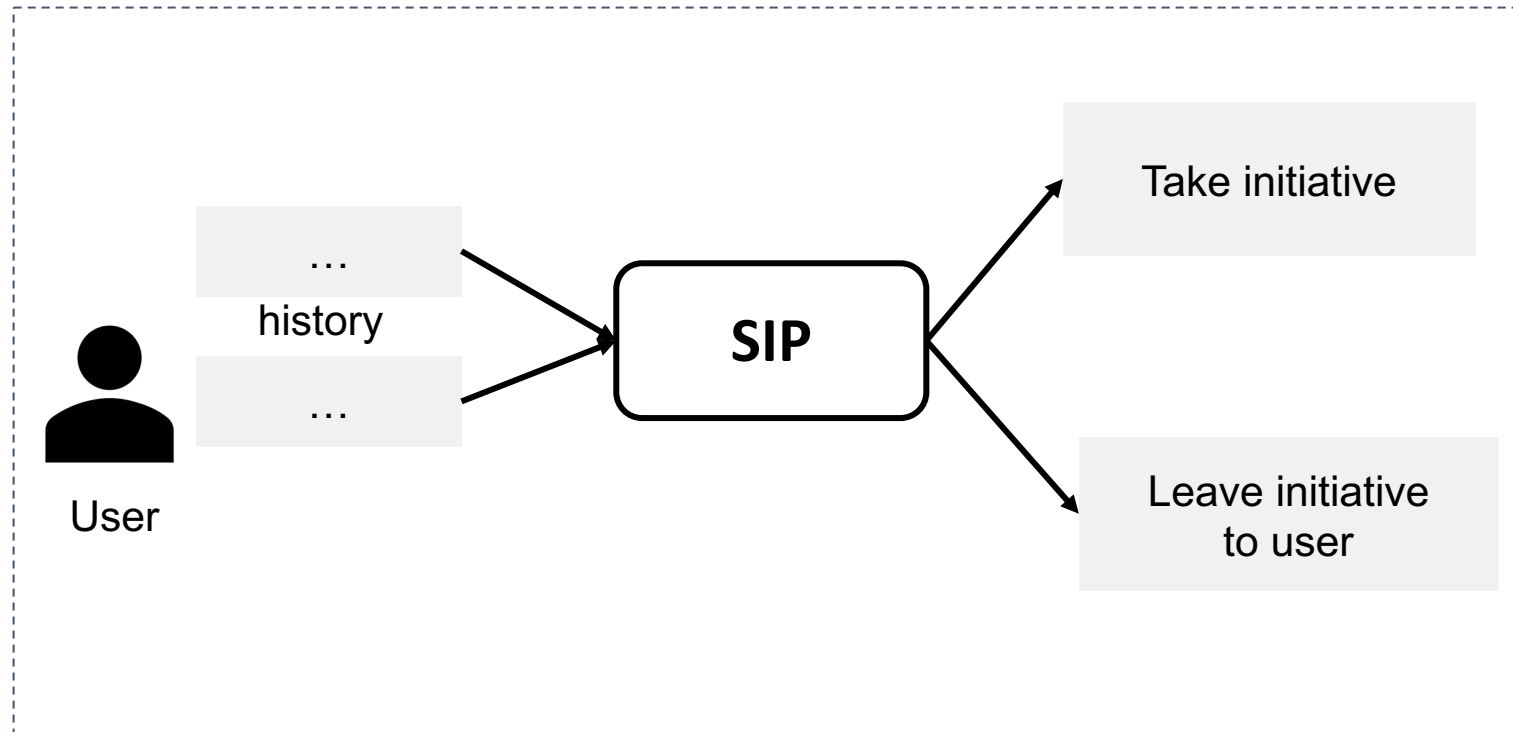


System Initiative Prediction for Multi-turn Conversational Information Seeking

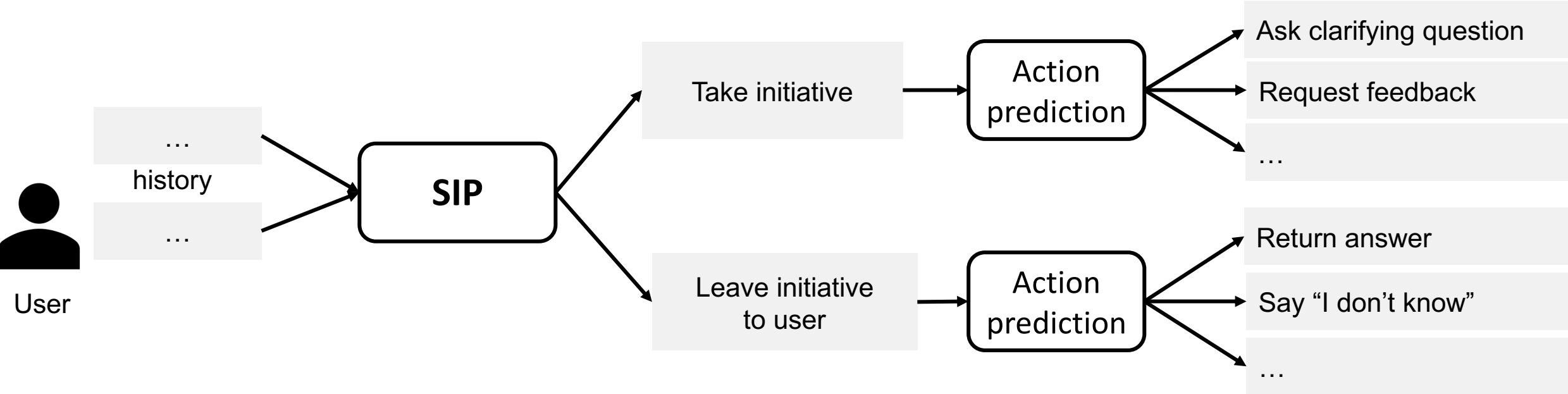
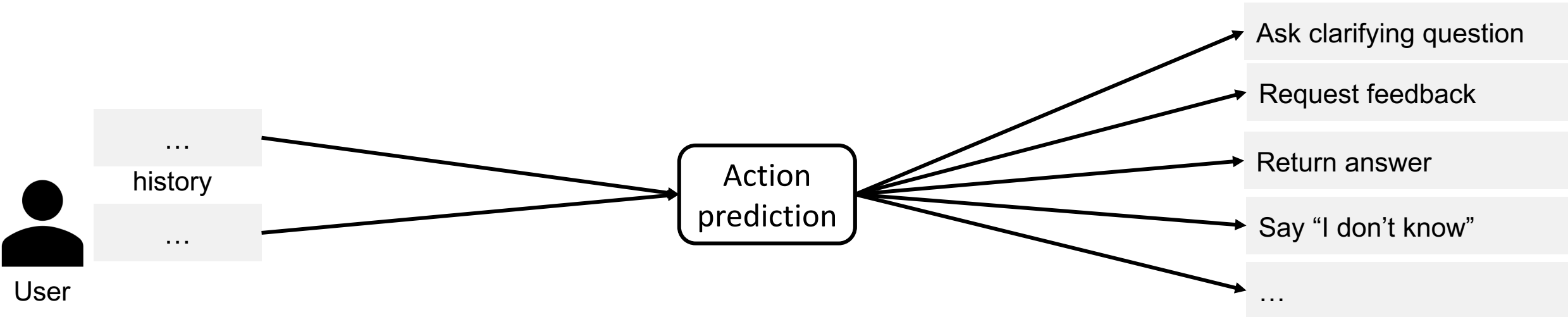
Chuan Meng, Mohammad Aliannejadi, Maarten de Rijke
CIKM 2023

Task definition

- System initiative prediction (SIP)
 - predicts **whether system should take initiative at next turn** in information-seeking conversation



Task definition



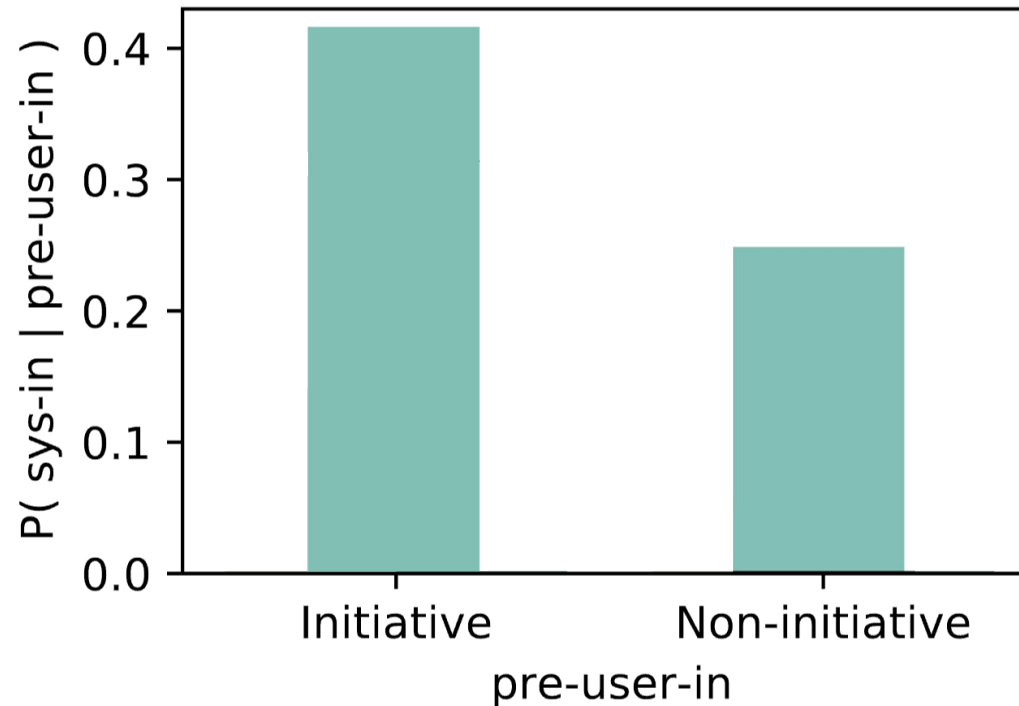
How well do LLMs perform on SIP?

- Preliminary experiments show:
 - performance of LLMs comparable to that of BERT
 - LLMs lack interpretability and transparency

Methods	MSDialog (%)			
	F1	Precision	Recall	Accuracy
LLaMA-7B	60.22	60.40	60.13	62.15
LLaMA-13B	62.54	62.73	63.21	62.99
LLaMA-33B	58.11	58.24	58.53	58.76
LLaMA-65B	55.30	62.33	60.44	55.93
BERT	60.17	60.25	60.12	61.86

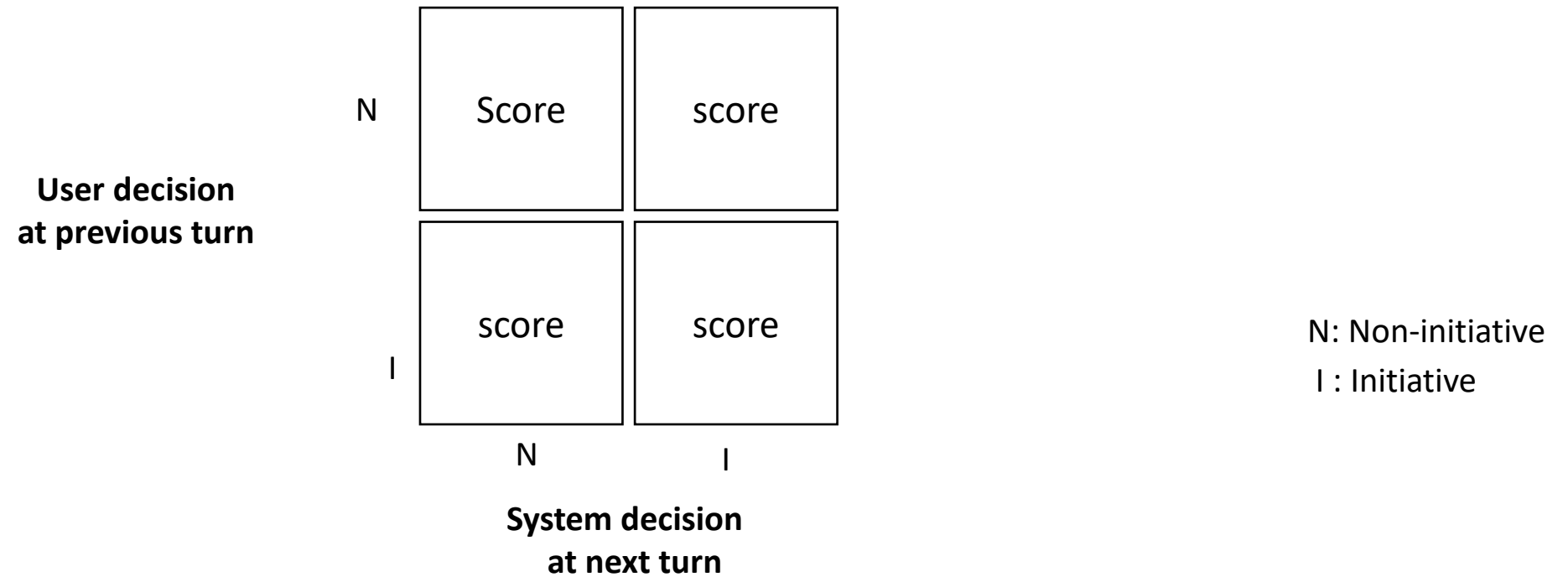
Why do we need a probabilistic graphical model for SIP

- Empirical analysis shows:
 - dependencies between adjacent user–system initiative-taking decisions



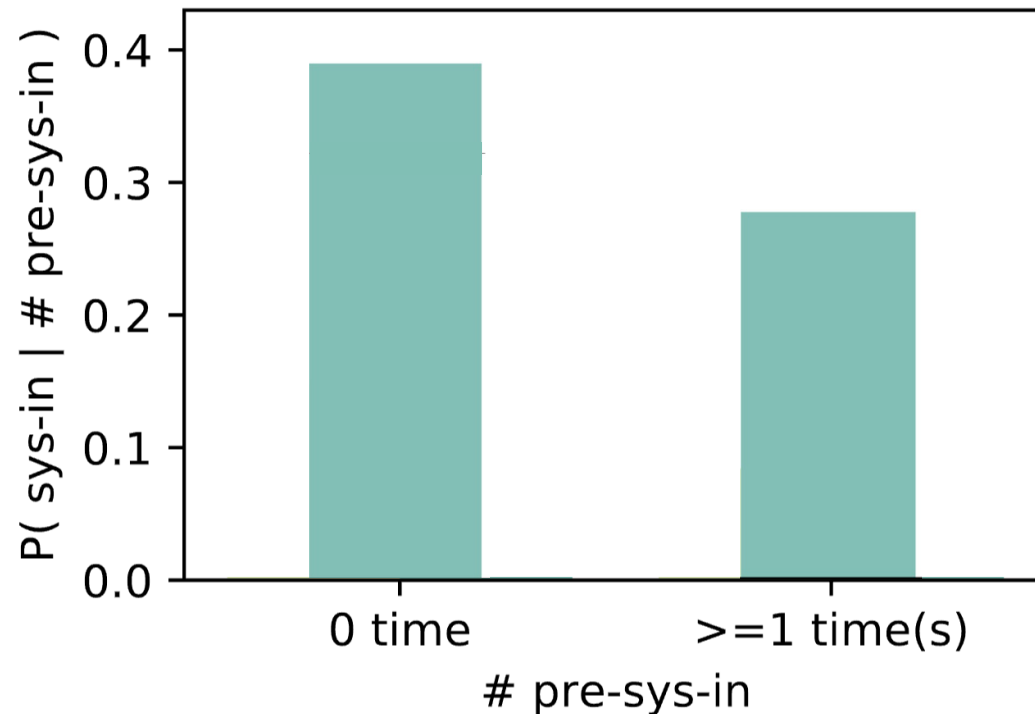
Why we need a probabilistic graphical model for SIP

- **Our proposal:** model SIP by conditional random fields (CRFs)
 - CRFs are effective in capturing **dependencies between adjacent decisions**
 - CRFs have greater transparency



Why we need a probabilistic graphical model for SIP

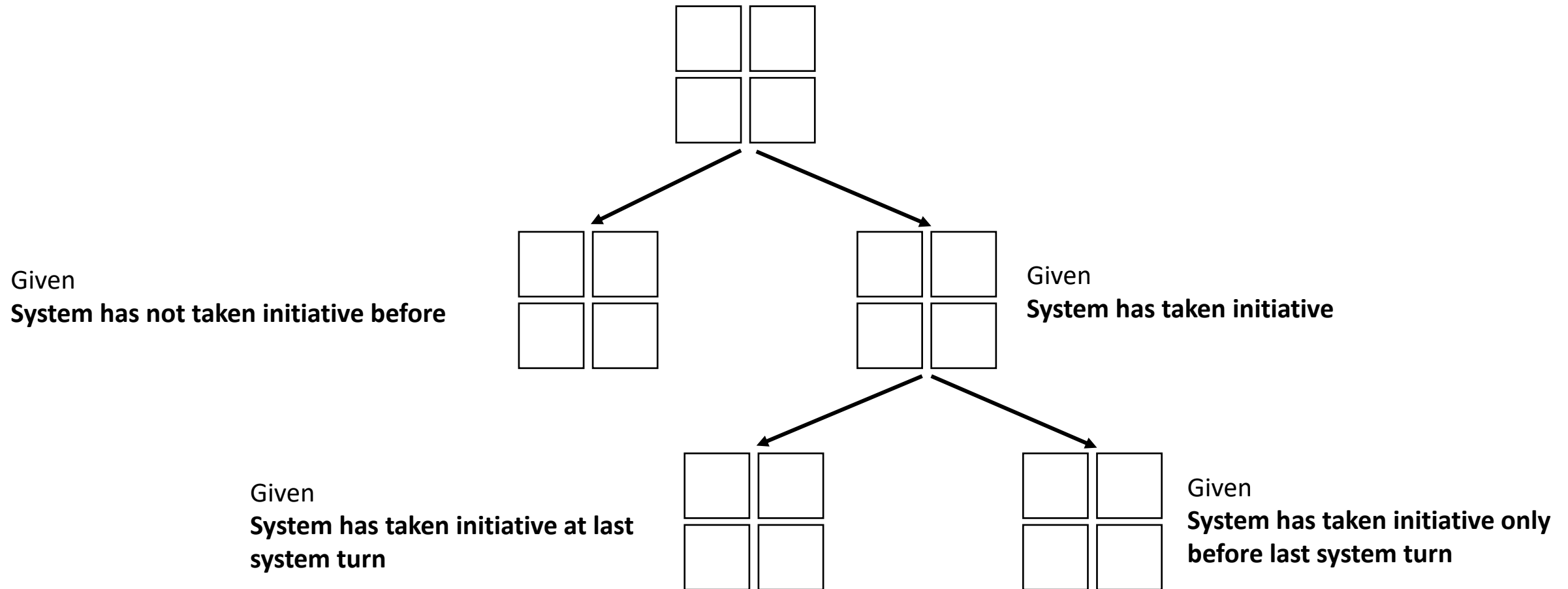
- Empirical analysis shows:
 - Dependencies between an initiative-taking decision and multi-turn features



- Challenge:
 - Vanilla CRFs cannot explicitly model multi-turn features

Why we need a probabilistic graphical model for SIP

- Propose **multi-turn feature-aware CRF**
 - conditions transition matrix between adjacent initiative-taking decisions on multi-turn features



Experimental results

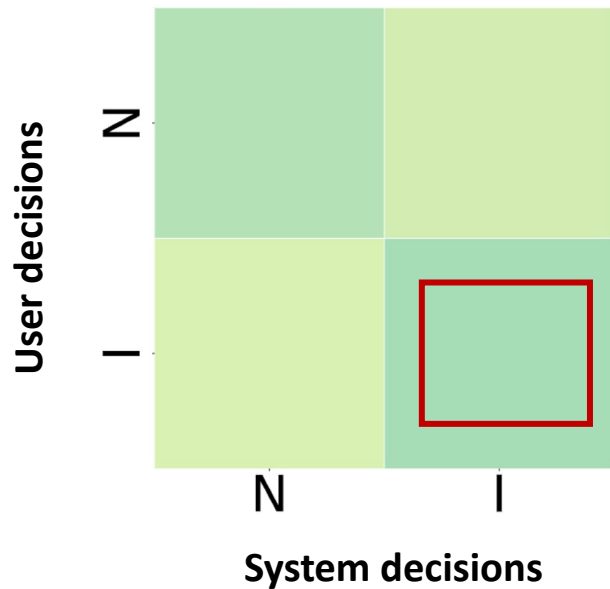
- Multi-turn feature-aware CRF achieves SOTA performance on SIP

Methods	MSDialog (%)			
	F1	Precision	Recall	Accuracy
LLaMA-7B	60.22	60.40	60.13	62.15
LLaMA-13B	62.54	62.73	63.21	62.99
LLaMA-33B	58.11	58.24	58.53	58.76
LLaMA-65B	55.30	62.33	60.44	55.93
BERT	60.17	60.25	60.12	61.86
VanillaCRF	62.31	63.24	62.17	64.97
Ours	65.37	65.79	65.19	67.23*

Experimental results

- Multi-turn feature-aware CRF exhibits great transparency

Given
System **has not** taken initiative before

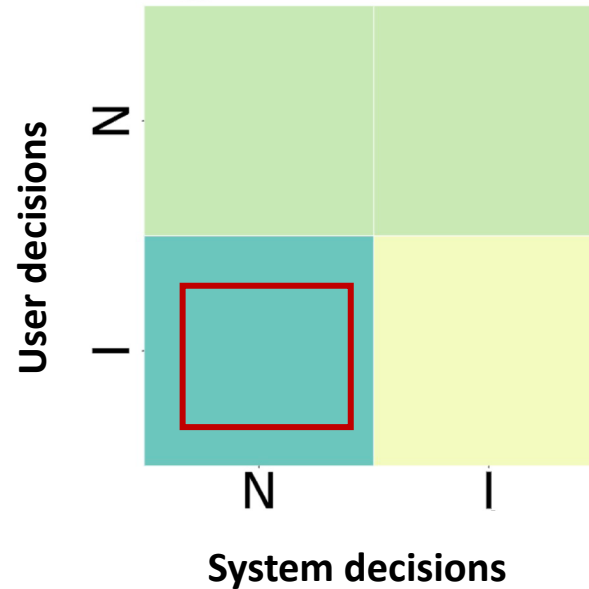


Example:

Turn 1: user asks a question

Turn 2 : system asks a clarifying question

Given
System has taken initiative **at last system turn**



Example:

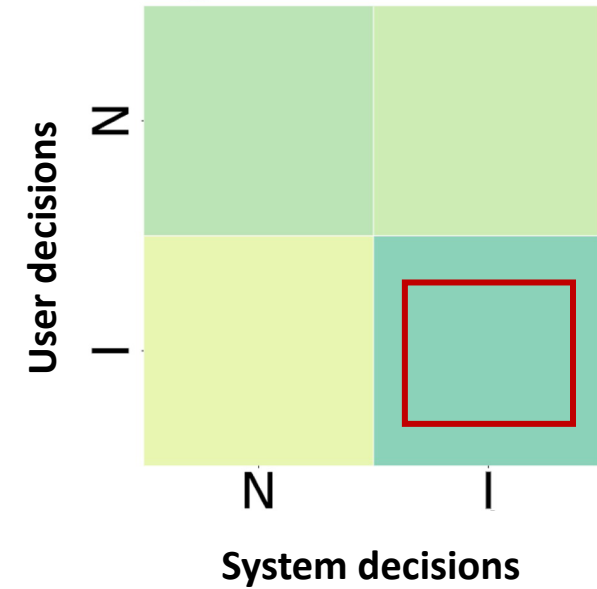
Turn 1: user asks a question

Turn 2: system asks a clarifying question

Turn 3: user rephrases a new question

Turn 4: system returns an answer

Given
System has taken initiative **only before last system turn**



Example:

Turn 1: user asks a question

Turn 2: system asks a clarifying question

Turn 3: user answers the clarifying question

Turn 4: system returns an answer

Turn 5: user asks a follow-up question

Turn 6 : system requests information

Conclusion

- Contributions
 - Introduce **system initiative prediction (SIP)**
 - Propose **multi-turn feature-aware CRF** to capture two types of dependencies
 - between **adjacent user–system initiative-taking decisions**
 - between **initiative-taking decision and multi-turn features**
- Our method
 - achieves SOTA performance on SIP
 - exhibits great transparency
 - improves downstream action prediction task
- Data and code open-sourced at <https://github.com/ChuanMeng/SIP>



QR code for the repo

Appendix

Given the user utterance at current turn and the conversational history at previous turns, predict whether the system should take the initiative or not at the current turn. Please output "yes" or "no". "yes" means the system should take the initiative at the current turn by asking a clarifying question or requesting feedback and so on; "no" means the system should not take the initiative at the current turn, e.g., giving an answer to the user.

Turn: 1

User utterance: {}

Should the system take the initiative at the current turn? {}

System utterance: {}

...

Turn: t

User utterance: {}

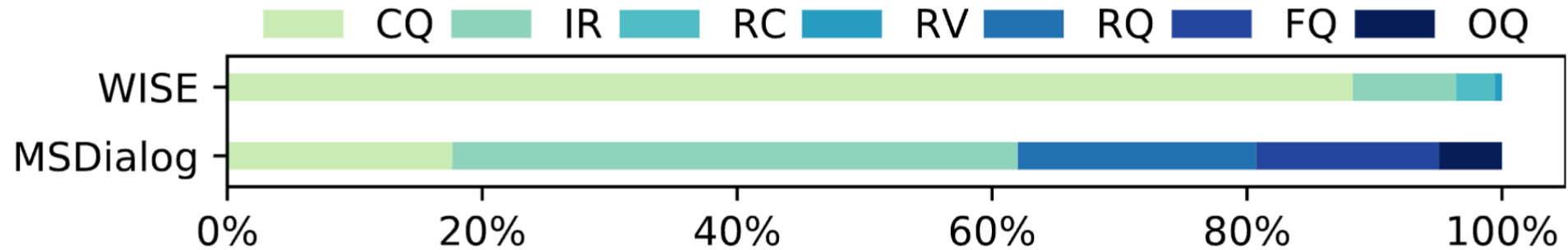
Should the system take the initiative at the current turn? yes/no

Appendix

- **WISE** consists of conversations collected through crowdsourcing. Each conversation contains mixed-initiative interactions between two workers playing the role of the user and system
- **MSDialog** consists of conversations that contain mixed-initiative interactions between users who ask for technical help and Microsoft staff or experienced product users (i.e., system) who help users solve their problems.

	WISE			MSDialog		
	train	valid	test	train	valid	test
#conversations	705	200	1,000	1,760	220	219
#utterances	12,184	3,811	18,828	6,305	752	747
Max. #turns/conversation	38	38	42	10	10	10
Avg. #turns/conversation	17.28	19.06	18.83	3.58	3.42	3.41
Max. #actions/system turn	3	2	3	6	6	7
Avg. #actions/system turn	1.02	1.02	1.02	1.67	1.77	1.80
Avg. #words/utterance	30.25	31.79	29.23	90.07	88.54	89.13
Avg. #system-initiatives/conv.	0.98	1.62	1.46	0.62	0.60	0.65
Avg. #claryfying questions/conv.	0.87	1.23	1.17	0.15	0.18	0.15

Appendix

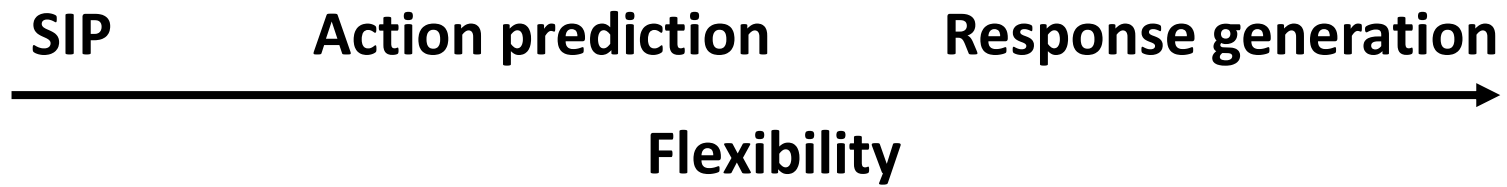


- **CQ**: clarifying question
- **IR**: information request
- **RV**: revise;
- **RC**: recommendation
- **OQ**: original question
- **RQ**: repeat question
- **FQ**: Follow-up question

Inverse Scaling: When Bigger Isn't Better

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, Ethan Perez

Work on scaling laws has found that large language models (LMs) show predictable improvements to overall loss with increased scale (model size, training data, and compute). Here, we present evidence for the claim that LMs may show inverse scaling, or worse task performance with increased scale, e.g., due to flaws in the training objective and data. We present empirical evidence of inverse scaling on 11 datasets collected by running a public contest, the Inverse Scaling Prize, with a substantial prize pool. Through analysis of the datasets, along with other examples found in the literature, we identify four potential causes of inverse scaling: (i) preference to repeat memorized sequences over following in-context instructions, (ii) imitation of undesirable patterns in the training data, (iii) tasks containing an easy distractor task which LMs could focus on, rather than the harder real task, and (iv) correct but misleading few-shot demonstrations of the task. We release the winning datasets at [this https URL](#) to allow for further investigation of inverse scaling. Our tasks have helped drive the discovery of U-shaped and inverted-U scaling trends, where an initial trend reverses, suggesting that scaling trends are less reliable at predicting the behavior of larger-scale models than previously understood. Overall, our results suggest that there are tasks for which increased model scale alone may not lead to progress, and that more careful thought needs to go into the data and objectives for training language models.



Outline

- ❑ Study 1: System initiative prediction for CIS (CIKM 2023) [12 min]
- ❑ **Study 2: QPP for CIS: reproducing existing QPP methods in CIS (SIGIR 2023) [12 min]**
- ❑ Study 3: QPP for CIS: improve QPP for CIS using query rewriting quality (ECIR 2023) [6 min]
- ❑ Conclusion [5 min]

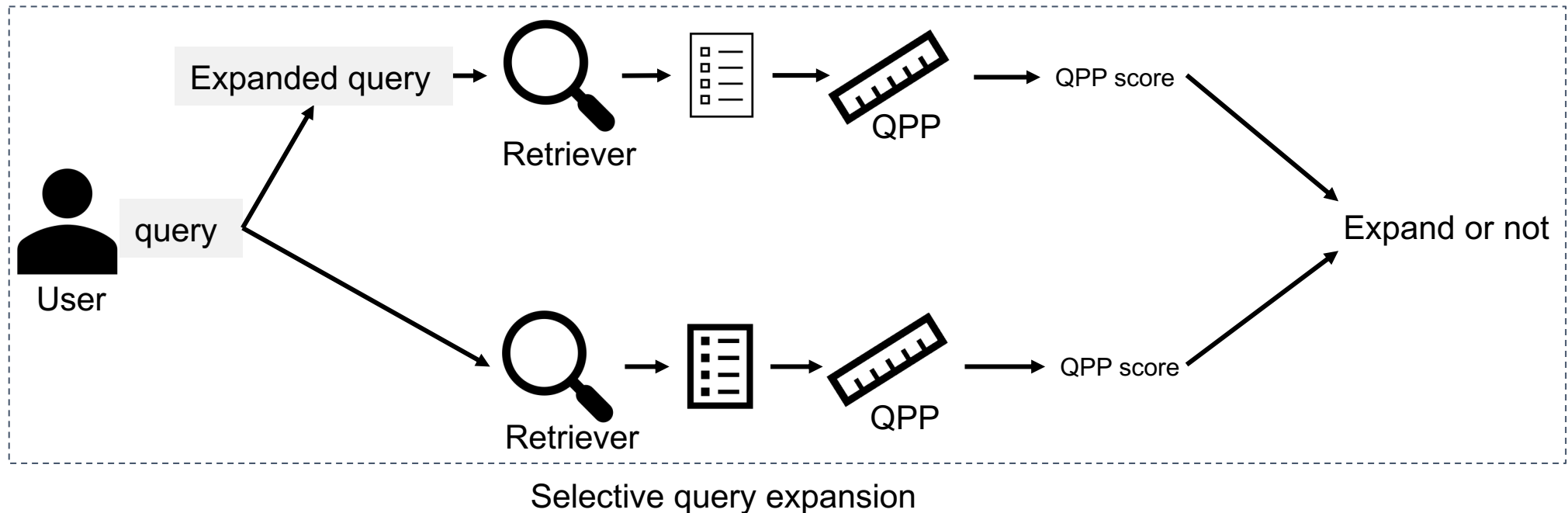


Query Performance Prediction: From Ad-hoc to Conversational Search

Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi and Maarten de Rijke
SIGIR 2023

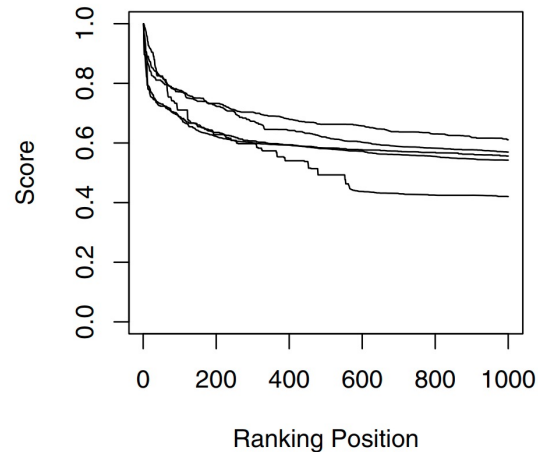
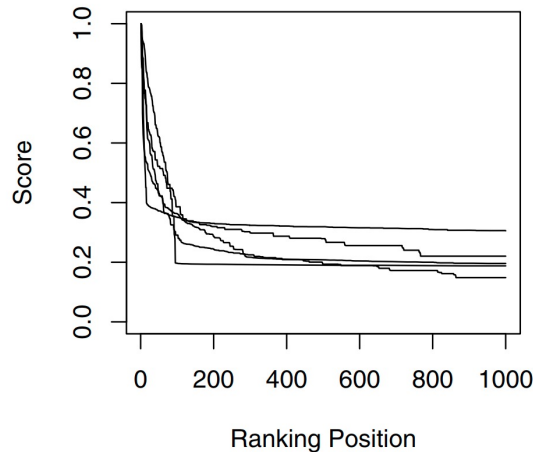
Background—Query performance prediction

- Query performance prediction (QPP)
 - Predicts retrieval quality of search system for query without relevance judgments
 - Widely studied in ad-hoc search
- QPP benefits a variety of applications, e.g., selective query expansion, query rewrite selection



Background—Query Performance Prediction

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

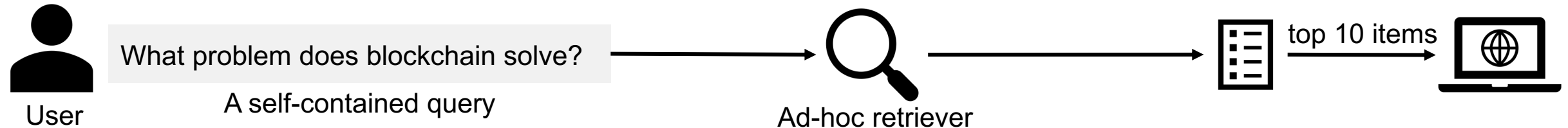


- Supervised post-retrieval QPP methods
 - BERT ($query, a \text{ ranked list}$) $\rightarrow QPP \text{ score}$

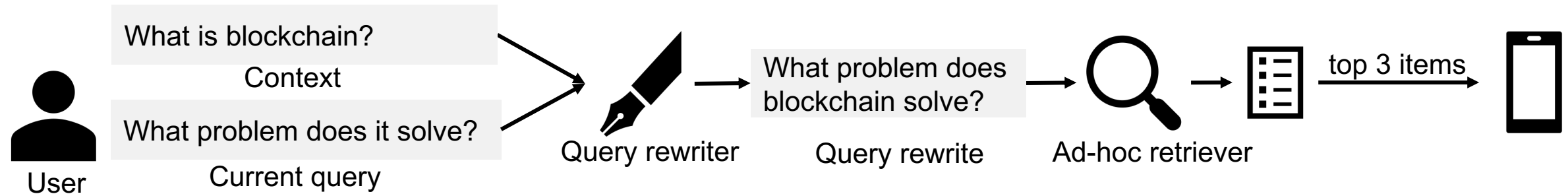
Background—Conversational search (CS)

- Ad-hoc search vs. CS
 - Self-contained vs. context-dependent queries
 - Deeper ranked list vs. only top of the ranked list

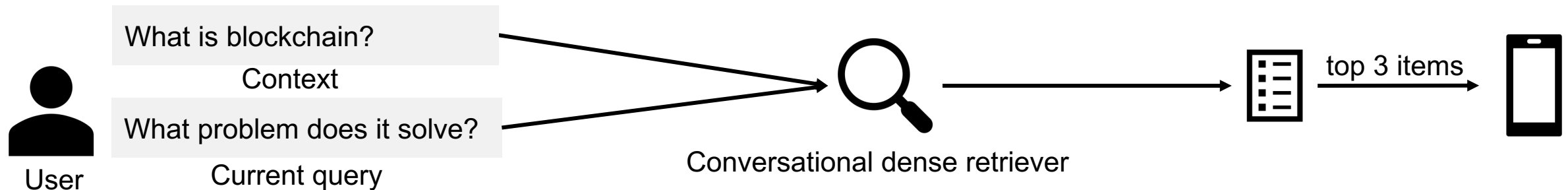
Ad-hoc search



Query rewriting-based retrieval

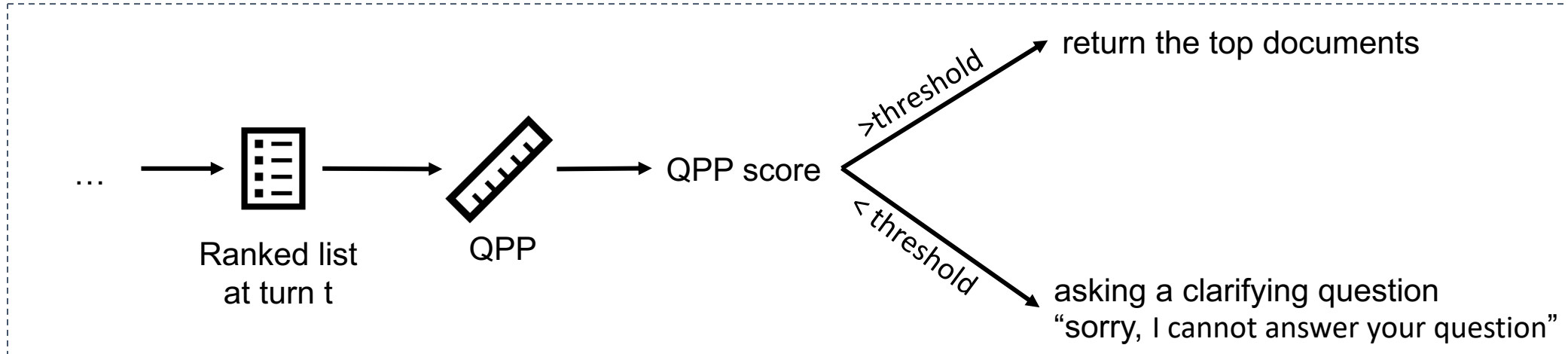


Conversational dense retrieval



Motivation

- Why do we need QPP for CS?
 - QPP can benefit CS regarding, e.g., action prediction



- **To what extent do findings from QPP methods for ad-hoc search generalize to CS?**
 - Motivate a reproducibility study

Reproducibility methodology

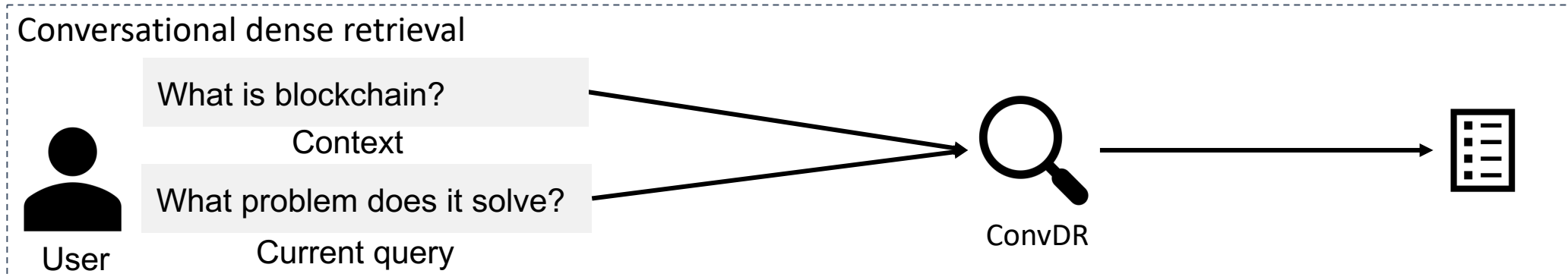
- Verify 3 findings on QPP for ad-hoc search:
 - ***Finding 1: Supervised QPP methods outperform unsupervised ones***
 - (Datta et al., 2022; Chen et al., 2022; Arabzadeh et al., 2021; Hashemi et al., 2019, Zamani et al., 2019)
 - ***Finding 2: List-wise supervised QPP methods outperform point-wise ones***
 - (Datta et al., 2022; Chen et al., 2022)
 - ***Finding 3: Retrieval score-based unsupervised QPP methods perform badly in estimating the retrieval quality of neural-based retrievers***
 - (Datta et al., 2022; Hashemi et al., 2019)

Reproducibility methodology

- To what extent do the previous findings from ad-hoc search generalize to CS ...
 - *(RQ1) ... when estimating the retrieval quality of (for top-ranked items) different query rewriting-based retrieval methods?*
 - ***(RQ2) ... when estimating the retrieval quality (for top-ranked items) of a conversational dense retrieval method?***
 - *(RQ3) ... when predicting the retrieval quality for longer-ranked lists?*

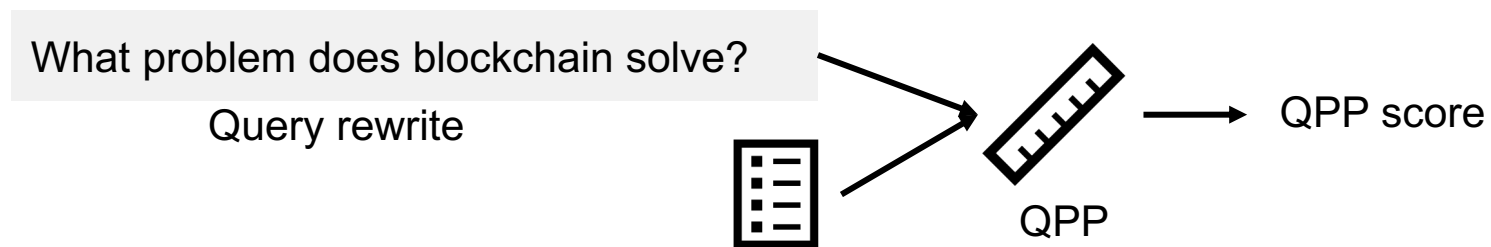
Experiments

- Experimental design for RQ2:
 - Predict the retrieval quality of conversational dense retriever, ConvDR (Yu et al., 2021)



- Feed **self-contained query rewrite** into QPP
- Study the effect of feeding different query rewrites
 - Generative query rewriting
 - Term expansion-based query rewriting
 - Human query rewriting

QPP for CS



Experiments

- Experimental settings:
 - QPP methods
 - 6 unsupervised QPP ones (5 score-based)
 - 3 supervised QPP ones (2 point-wise, 1 list-wise)
 - Datasets:
 - CAsT-19, CAsT-20, OR-QuAC
 - Evaluation metrics
 - Pearson's ρ , Kendall's τ and Spearman's ρ correlation between actual nDCG@3 score and performance predicted by QPP methods

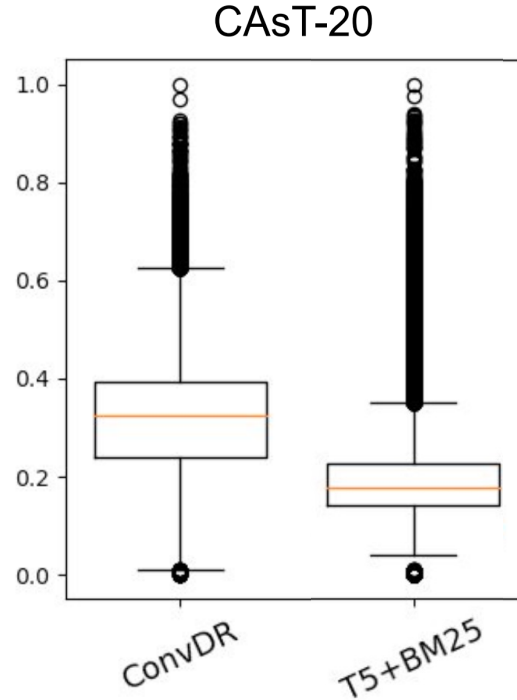
	CAsT-19	CAsT-20	OR-QuAC		
	test	test	train	valid	test
#conversations	50	25	4,383	490	771
#conversations (judged)	20	25	–	–	–
#questions	479	216	31,526	3,430	5,571
#questions (judged)	173	208	–	–	–
#documents	38M			11M	

Experiments for RQ2

- Results for RQ2:
 - Supervised QPP methods vs. unsupervised ones
 - Supervised QPP methods NQA-QPP (Hashemi et al., 2019), BERTQPP (Arabzadeh et al., 2021) achieve SOTA **only** when having large-scale training data
 - Unsupervised score-based QPP ones WIG (Zhou et al., 2007), NQA (Shtok et al., 2012) are still competitive, achieving SOTA in the few shot setting
 - Point-wise vs. list-wise
 - Point-wise supervised QPP methods NQA-QPP (Hashemi et al., 2019), BERTQPP (Arabzadeh et al., 2021) outperform the list-wise one qppBERT-PL (Datta et al., 2022) in most cases
 - Supervised QPP methods tend to perform better when fed with human-rewritten queries, especially when query rewriting is harder (CAsT-20)

Experiments for RQ2

- *Previous finding (Datta et al., 2022) found that the short range of retrieval scores returned by neural-based retrievers, such as ColBERT, would limit the performance of score-based unsupervised QPP methods*
- Why score-based methods exhibit a good performance:
 - The retrieval score distribution of ConvDR displays a higher variance than BM25
 - Score-based methods are less impacted by the query understanding challenge



Takeaway

- Takeaway
 - *Previous finding 1: Supervised QPP methods outperform unsupervised ones*
- We found
 - Supervised QPP ones distinctly outperform unsupervised ones only when a large amount of training data is available
 - Unsupervised QPP ones show strong performance
 - In cases of insufficient training data
 - When predicting the retrieval quality for deeper-ranked lists

Takeaway

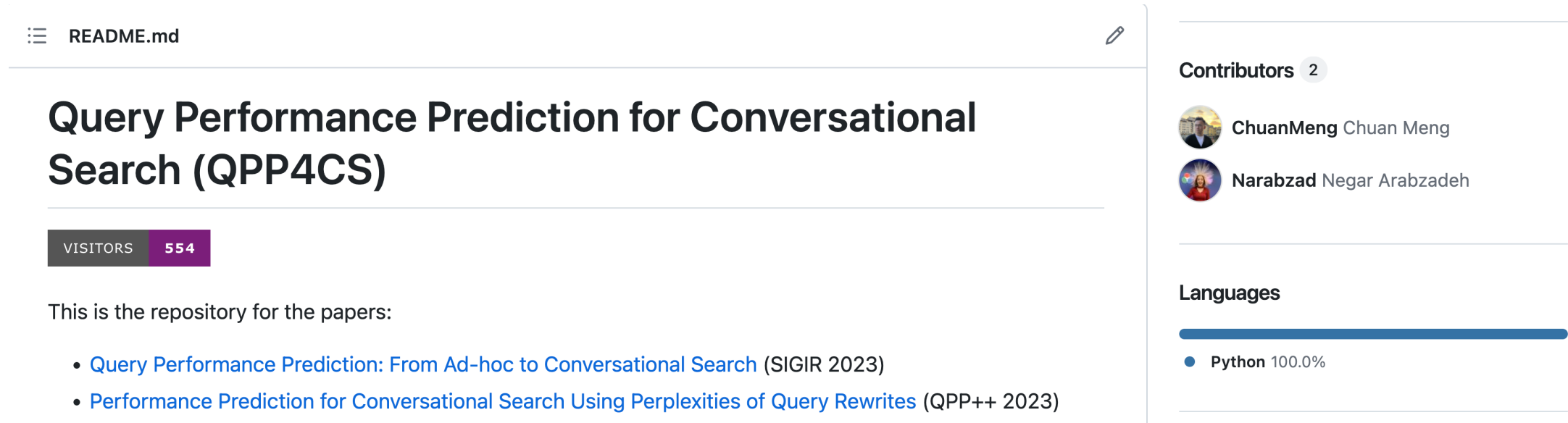
- Takeaway
 - *Previous finding 2: List-wise supervised QPP methods outperform point-wise ones*
- We found
 - Point-wise QPP ones outperform list-wise ones in most cases
 - List-wise QPP ones
 - Are more data-efficient
 - Show a slight advantage for deeper-ranked lists

Takeaway

- Takeaway
 - *Previous finding 3: Retrieval score-based unsupervised QPP methods perform badly in estimating the retrieval quality of neural-based retrievers*
- We found
 - Score-based QPP methods are still competitive when assessing a conversational neural-based retriever, either for top ranks or deeper-ranked lists
 - The effectiveness of score-based QPP methods relies on the retrieval score distribution of a specific retriever
 - A neural-based retriever can have a higher variance than lexical-based one
 - The greater variance in the retrieval score distribution, the better performance observed in score-based QPP methods

Conclusion

- Contributions
 - A comprehensive reproducibility study into ad-hoc QPP methods in CS
 - The data and code are open-sourced <https://github.com/ChuanMeng/QPP4CS>



The screenshot shows the GitHub repository page for 'Query Performance Prediction for Conversational Search (QPP4CS)'. The repository title is 'Query Performance Prediction for Conversational Search (QPP4CS)'. The repository has 554 visitors. The repository description is 'This is the repository for the papers:'. The repository contains two papers: 'Query Performance Prediction: From Ad-hoc to Conversational Search (SIGIR 2023)' and 'Performance Prediction for Conversational Search Using Perplexities of Query Rewrites (QPP++ 2023)'. The repository has two contributors: ChuanMeng Chuan Meng and Narabzad Negar Arabzadeh. The repository is written in Python 100.0%.

☰ README.md

Query Performance Prediction for Conversational Search (QPP4CS)

VISITORS 554

This is the repository for the papers:

- [Query Performance Prediction: From Ad-hoc to Conversational Search \(SIGIR 2023\)](#)
- [Performance Prediction for Conversational Search Using Perplexities of Query Rewrites \(QPP++ 2023\)](#)

Contributors 2

- ChuanMeng Chuan Meng
- Narabzad Negar Arabzadeh

Languages

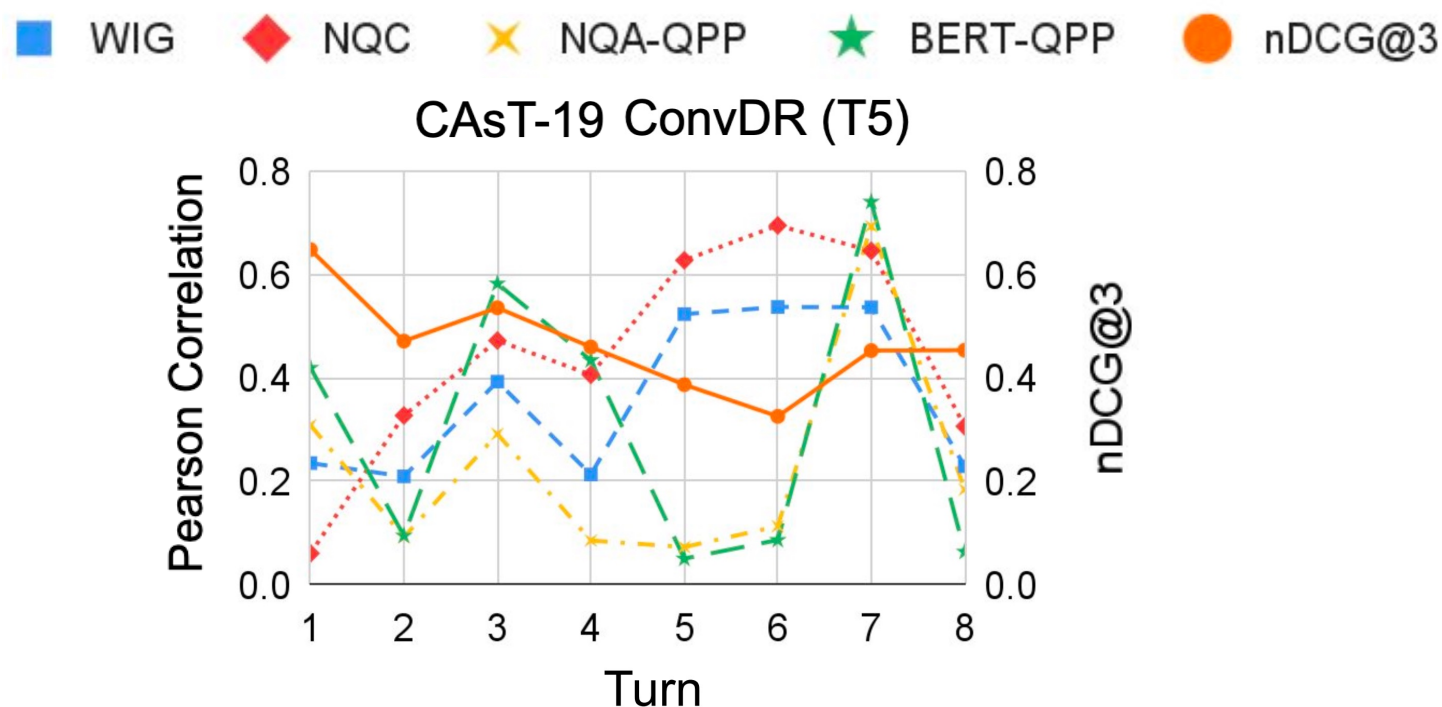
- Python 100.0%



QR code for the repo

Experiments for RQ2

- Turn-wise QPP effectiveness
 - Supervised QPP methods are more sensitive to the actual retrieval quality
 - QPP effectiveness goes up/down as nDCG@3 scores go up/down



Suggestions for future work

- Solve the query understanding challenge
 - Improve query rewriting quality
 - Develop a QPP-specific conversational context understanding method
- Utilize few-shot learning techniques
- Improve supervised QPP methods Leverage unsupervised QPP methods

Appendix

- Mao et al. Learning Denoised and Interpretable Session Representation for Conversational Search. In WWW 2023.
- Datta et al. A 'Pointwise-Query, Listwise-Document based Query Performance Prediction Approach. In SIGIR 2022.
- Arabzadeh et al. BERT-QPP: Contextualized Pre-trained Transformers for Query Performance Prediction. In CIKM, 2021.
- Hashemi et al. Performance Prediction for Non-Factoid Question Answering. In ICTIR 2019.
- Mackenzie et al. Query-Performance Prediction: Setting the Expectations Straight. In SIGIR 2014.
- Amati et al. Query Difficulty, Robustness, and Selective Application of Query Expansion. In ECIR 2014.
- Qian et al. Explicit Query Rewriting for Conversational Dense Retrieval. In EMNLP, 2022.
- Dalton et al. Cast-19: A Dataset for Conversational Information Seeking. In SIGIR 2020.
- Arabzadeh et al. Unsupervised Question Clarity Prediction Through Retrieved Item Coherency. In CIKM 2022.
- Roitman et al. A Study of Query Performance Prediction for Answer Quality Determination. In ICTIR 2019.
- Lin et al. Multi-Stage Conversational Passage Retrieval: An Approach to Fusing Term Importance Estimation and Neural Query Rewriting. In TOIS 2021.
- Al-Thani, et al. Improving Conversational Search with Query Reformulation Using Selective Contextual History. DIM 2022.
- Aliannejadi et al. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In SIGIR 2019.
- Dipasree et al. Effective Query Formulation in Conversation Contextualization : A Query Specificity-based Approach. In ICTIR 2021.
- Dalton et al. Cast-19: A Dataset for Conversational Information Seeking. In SIGIR 2020.
- Dalton et al. CAsT 2020: The Conversational Assistance Track Overview. In Text Retrieval Conference 2020.
- Qu et al. Open-retrieval Conversational Question Answering. In SIGIR 2020.
- Cronen-Townsend et al, Predicting Query Performance. In SIGIR 2002
- Zhou et al. Query Performance Prediction in Web Search Environments. In SIGIR 2007.
- Shtok et al. Predicting Query Performance by Query-Drift Estimation. In TOIS 2012.
- Pérez-Iglesias et al. Standard Deviation as a Query Hardness Estimator. In SPIRE 2010.
- Cummins et al. Improved Query Performance Prediction Using Standard Deviation. SIGIR 2010.
- Tao et al. Query Performance Prediction by Considering Score Magnitude and Variance Together. In CIKM 2014.
- Hashemi et al. Performance Prediction for Non-Factoid Question Answering. In ICTIR 2019.
- Arabzadeh et al. BERT-QPP: Contextualized Pre-trained Transformers for Query Performance Prediction. In CIKM 2021.
- Datta et al. A Relative Information Gain-based Query Performance Prediction Framework with Generated Query Variants. In TOIS 2022.
- Yu et al. Few-Shot Conversational Dense Retrieval. In SIGIR 2021.
- Voskarides et al. Query Resolution for Conversational Search with Limited Supervision. In SIGIR 2020.

Appendix



$$\text{Clarity}(q, D_{q;M}^k, D) = \sum_{w \in V} P(w|D_{q;M}^k) \log \frac{P(w|D_{q;M}^k)}{P(w|D)},$$


$$\text{WIG}(q, D_{q;M}^k, D) = \frac{1}{k} \sum_{d \in D_{q;M}^k} \frac{1}{\sqrt{|q|}} (\text{Score}(q; d) - \text{Score}(q; D)),$$

$$\text{NQC}(q, D_{q;M}^k, D) = \frac{1}{\text{Score}(q; D)} \sqrt{\frac{1}{k} \sum_{d \in D_{q;M}^k} (\text{Score}(q; d) - \mu)^2},$$

$$\text{SMV}(q, D_{q;M}^k, D) = \frac{\frac{1}{k} \sum_{d \in D_{q;M}^k} (\text{Score}(q; d) |\ln \frac{\text{Score}(q; d)}{\mu}|)}{\text{Score}(q; D)},$$

Appendix

castorini/ **t5-base-canard**   like 0

 Text2Text Generation  PyTorch  JAX  Transformers t5  AutoTrain Compatible  text-generation-inference

 Model card  Files and versions  Community **2**

 Edit model card

YAML Metadata Warning: empty or missing yaml metadata in repo card (<https://huggingface.co/docs/hub/model-cards#model-card-metadata>)

This model is trained for conversational question rewriting.

Usage:

Source text format: `${HISTORY} ||| ${CURRENT_QUESTION}`

example from CANARD: Frank Zappa ||| Disbandment ||| What group disbanded ||| Zappa and the Mothers of Invention ||| When did they disband?

Target text: When did Zappa and the Mothers of Invention disband?

You can find our guide to reproduce the training in this [repo](#).

CANARD

A Dataset for Question-in-Context Rewriting

[EMNLP'19 Paper \(Elgohary et al.\)](#)

[Download Dataset](#)

CANARD is a dataset for question-in-context rewriting that consists of questions each given in a dialog context together with a context-independent rewriting of the question. The context of each question is the dialog utterances that precede the question. CANARD can be used to evaluate question rewriting models that handle important linguistic phenomena such as coreference and ellipsis resolution.

CANARD is based on [QuAC \(Choi et al., 2018\)](#)---a conversational reading comprehension dataset in which answers are selected spans from a given section in a Wikipedia article. Some questions in QuAC are unanswerable with their given sections. We use the answer 'I don't know.' for such questions.

CANARD is constructed by crowdsourcing question rewritings using Amazon Mechanical Turk. We apply several automatic and manual quality controls to ensure the quality of the data collection process. The dataset consists of 40,527 questions with different context lengths. More details are available in our [EMNLP 2019 paper](#). An example is provided below. The dataset is distributed under the [CC BY-SA 4.0](#) license.

Outline

- ❑ Study 1: System initiative prediction for CIS (CIKM 2023) [12 min]
- ❑ Study 2: QPP for CIS: reproducing existing QPP methods in CIS (SIGIR 2023) [12 min]
- ❑ **Study 3: QPP for CIS: improve QPP for CIS using query rewriting quality (ECIR 2023) [6 min]**
- ❑ Conclusion [5 min]



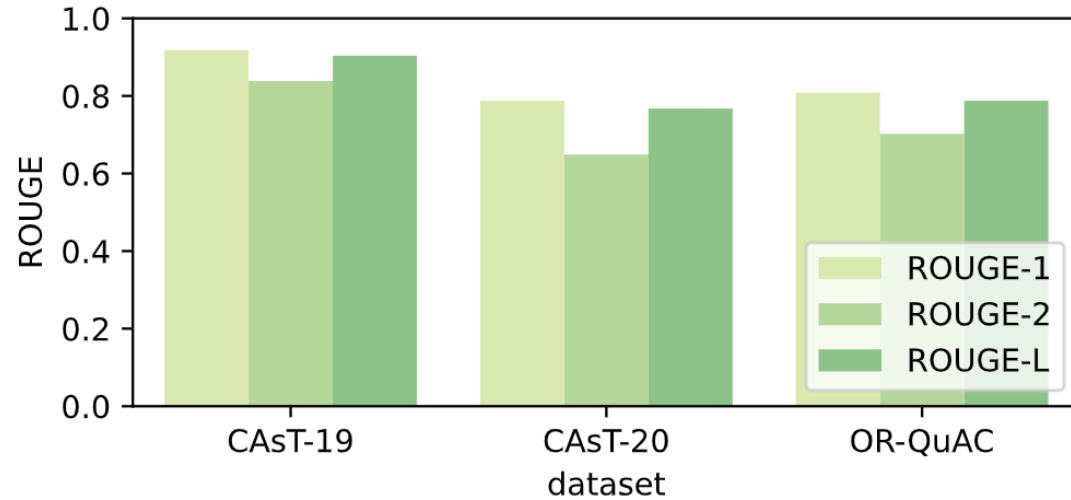
Performance Prediction for Conversational Search Using Perplexities of Query Rewrites

Chuan Meng, Mohammad Aliannejadi and Maarten de Rijke

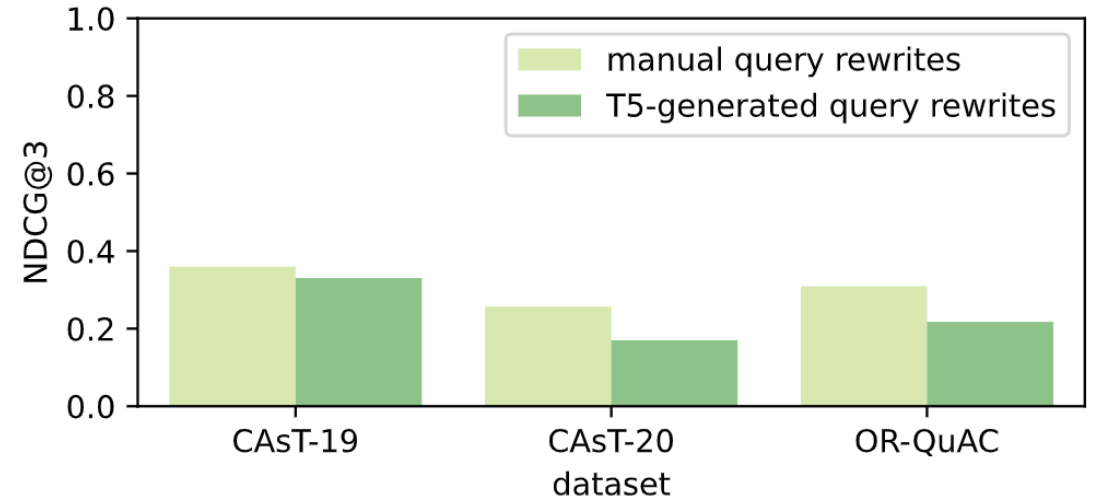
QPP++2023 (ECIR 2023)

Motivation

- Lower query rewriting quality tends to result in lower retrieval quality
- Query rewriting quality provides evidence for QPP



(a)



(b)

Figure 1: The similarity between manual and T5-generated query rewrites in terms of ROUGE (a) and the retrieval quality of BM25 for manual/T5-generated query rewrites in terms of NDCG@3 (b).

Methodology

- How?
 - evaluate the query rewriting quality
 - perplexity
 - inject the quality into the QPP
 - linear interpolation
 - $final\ QPP\ score = \alpha \cdot \frac{1}{perplexity} + (1 - \alpha) \cdot QPP\ score$

Experiments

- Experimental settings:
 - baselines: QS, SCS, avgICTF, IDF, PMI, SCQ, VAR
 - retriever: T5 query rewriter [1] + BM25
 - target metric: nDCG@3
 - perplexity measurer: GPT-2 XL (1.5B parameters) [2]

[1] <https://huggingface.co/castorini/t5-base-canard>

[2] <https://huggingface.co/gpt2-xl>

Experiments

- Observations:
 - lower quality tends to lead to worse QPP effectiveness
 - PPL-QPP improves QPP effectiveness on CAsT-19 and, in particular, CAsT-20

Methods	CAsT-19			CAsT-20		
	P- ρ	K- τ	S- ρ	P- ρ	K- τ	S- ρ
QS	-0.054	-0.011	-0.017	0.125	0.086	0.118
SCS	0.191	0.134	0.191	0.173	0.102	0.140
avgICTF	0.266	0.180	0.257	0.142	0.107	0.144
IDF (avg, avg, sum)	0.271	0.187	0.267	0.149	0.114	0.152
PMI (max, avg, max)	0.320	0.208	0.293	0.136	0.113	0.155
SCQ (avg, avg, max)	0.174	0.127	0.178	0.224	0.167	0.226
VAR (sum, avg, sum)	0.321	0.221	0.310	0.210	0.162	0.221
PPL-QPP	0.324	0.225	0.315	0.231	0.191	0.256

Conclusion and Future Work

- Contributions
 - propose PPL-QPP that incorporates query rewriting quality into QPP methods.
 - PPL-QPP improves QPP effectiveness if the query rewriting quality is limited.
 - The data and code are open-sourced <https://github.com/ChuanMeng/QPP4CS>
- Future work
 - incorporate query rewriting quality into post-retrieval QPP methods
 - the choice of evaluator (LLMs) for measuring the quality of query rewrites



QR code for the repo

Outline

- ❑ Study 1: System initiative prediction for CIS (CIKM 2023) [12 min]
- ❑ Study 2: QPP for CIS: reproducing existing QPP methods in CIS (SIGIR 2023) [12 min]
- ❑ Study 3: QPP for CIS: improve QPP for CIS using query rewriting quality (ECIR 2023) [6 min]
- ❑ **Conclusion** [5 min]

Conclusion and Future Work

- Contributions
 - System initiative prediction (SIP) for CIS
 - Query performance prediction (QPP) for CIS
- Future work
 - Apply QPP to SIP
 - Modeling SIP and response generation jointly
 - Enhancing retrieval-augmented generation using QPP

Thank you!

Chuan Meng
c.meng@uva.nl