# Predicting the Right Moment for System Initiative in Mixed-Initiative Conversational Search

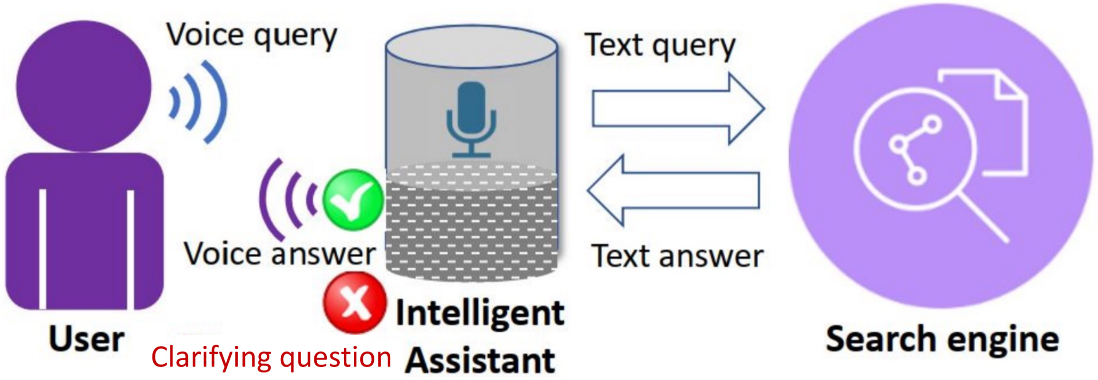
Chuan Meng

27th August 2024

# Background

- Mixed-initiative conversational search (CS)
  - 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?
  - Structural dependency modelling [1]
  - Query performance prediction (QPP) [2,3]




[1] Meng et al. System Initiative Prediction for Multi-turn Conversational Information Seeking. CIKM 2023
[2] Meng et al. Query Performance Prediction: From Ad-hoc to Conversational Search. SIGIR 2023.
[3] Meng et al. Performance Prediction for Conversational Search Using Perplexities of Query Rewrites. ECIR 2023.

# Outline

- ❑ Study 1: Structural dependency modelling for CS (CIKM 2023) [12 min]
- ❑ Study 2: Query performance prediction for CS (SIGIR 2023 & ECIR 2023) [6 min]
- ❑ Conclusion and future work [2 min]

# Outline

- **Study 1: Structural dependency modelling for CS (CIKM 2023)** [12 min]
- Study 2: Query performance prediction for CS (SIGIR 2023 & ECIR 2023) [6 min]
- Conclusion and future work [2 min]

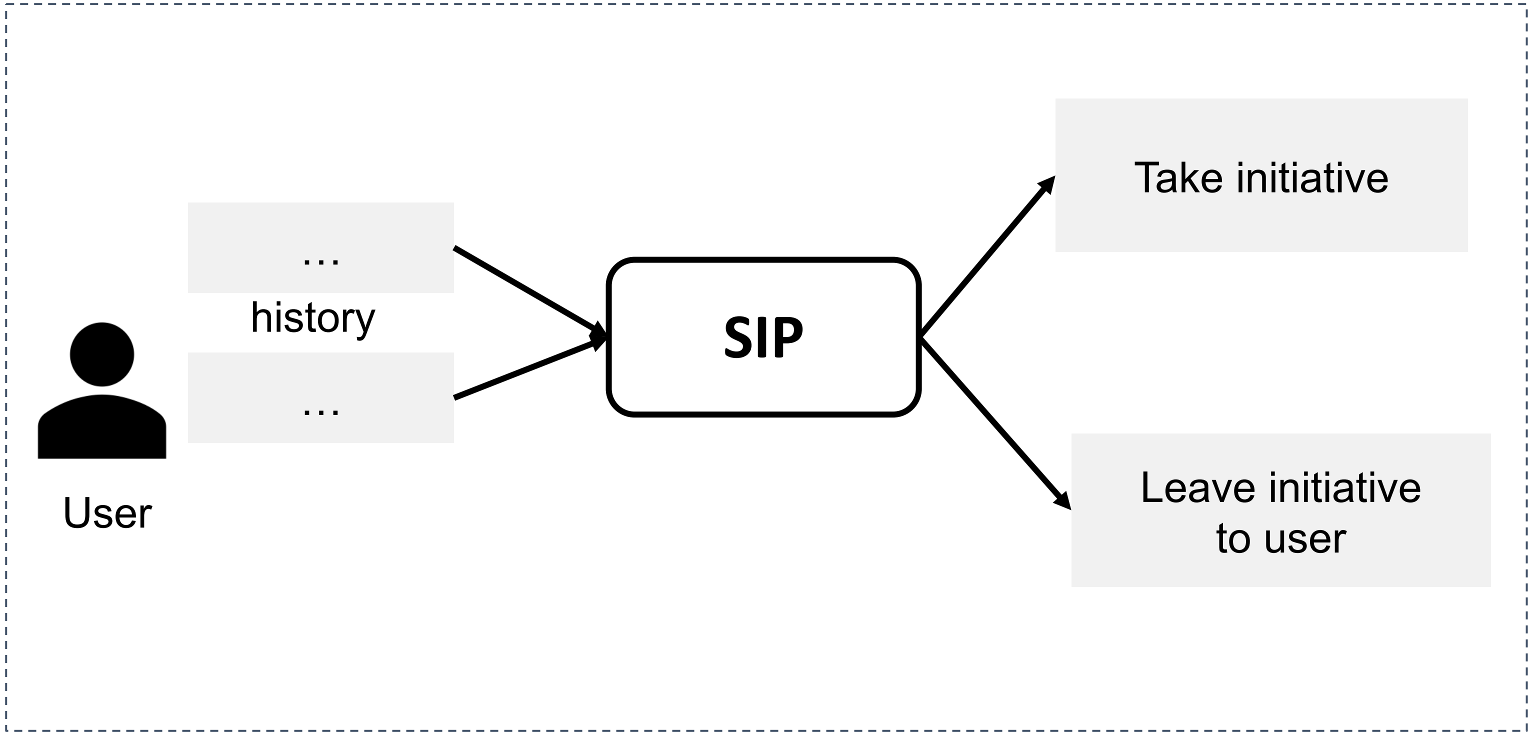
# System Initiative Prediction for Multi-turn Conversational Information Seeking

**Chuan Meng**, Mohammad Aliannejadi, Maarten de Rijke

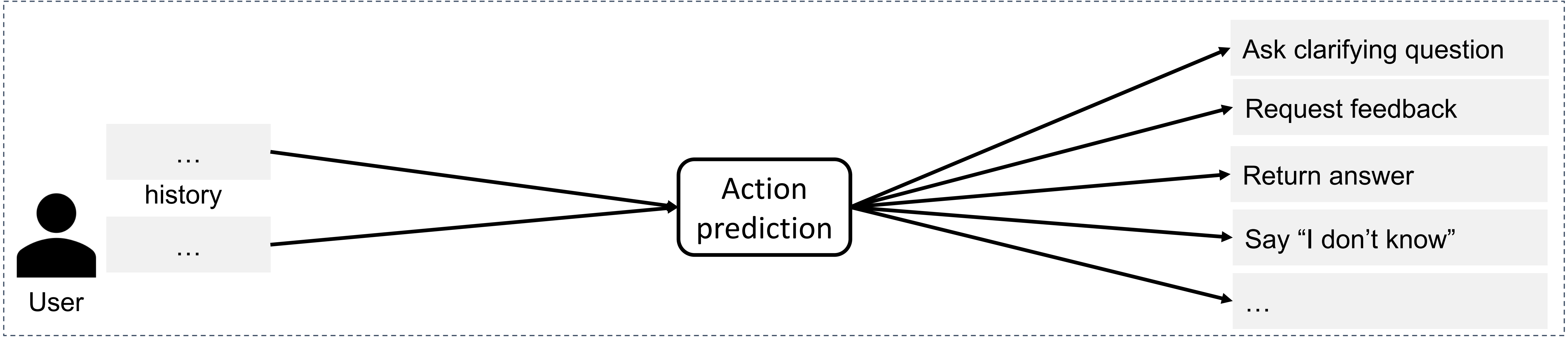
CIKM 2023

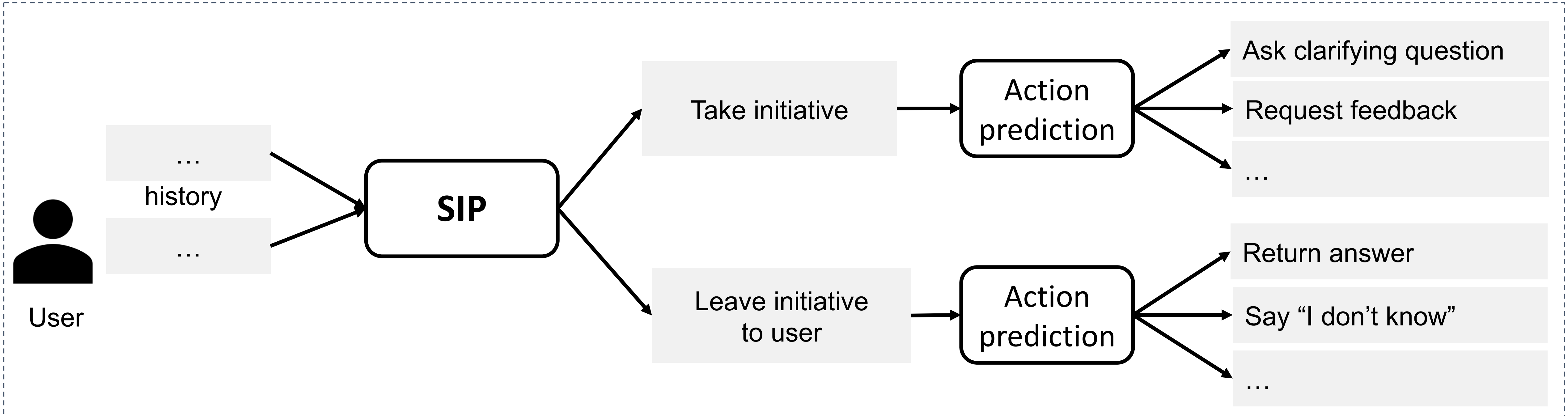
# Task definition

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



# Task definition



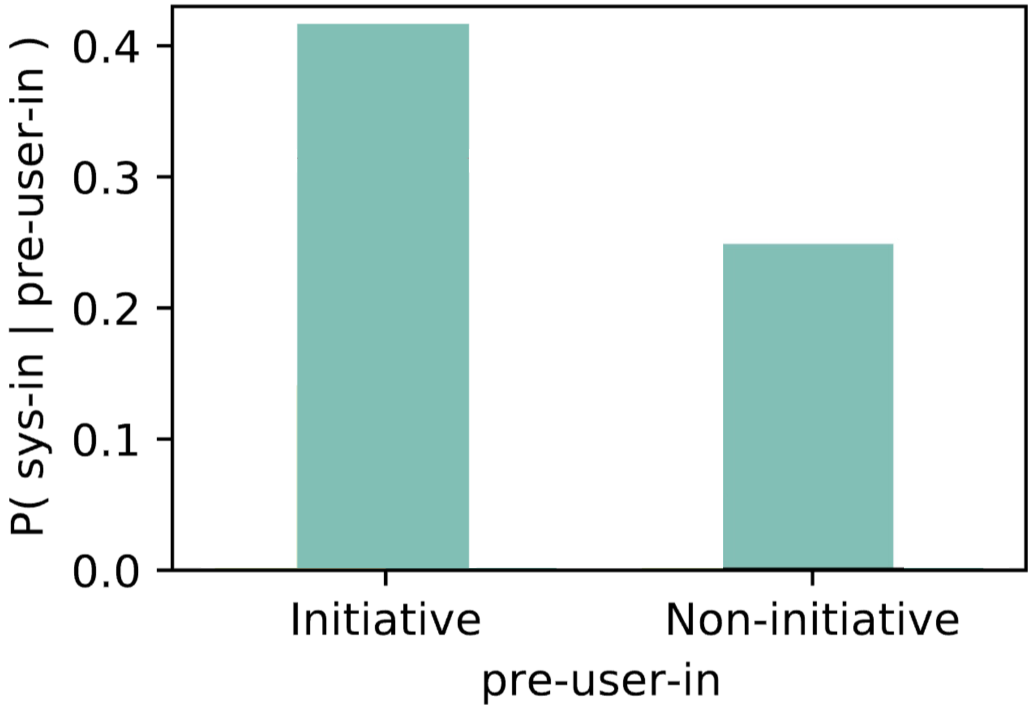
# How well do LLMs perform on SIP?

- 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

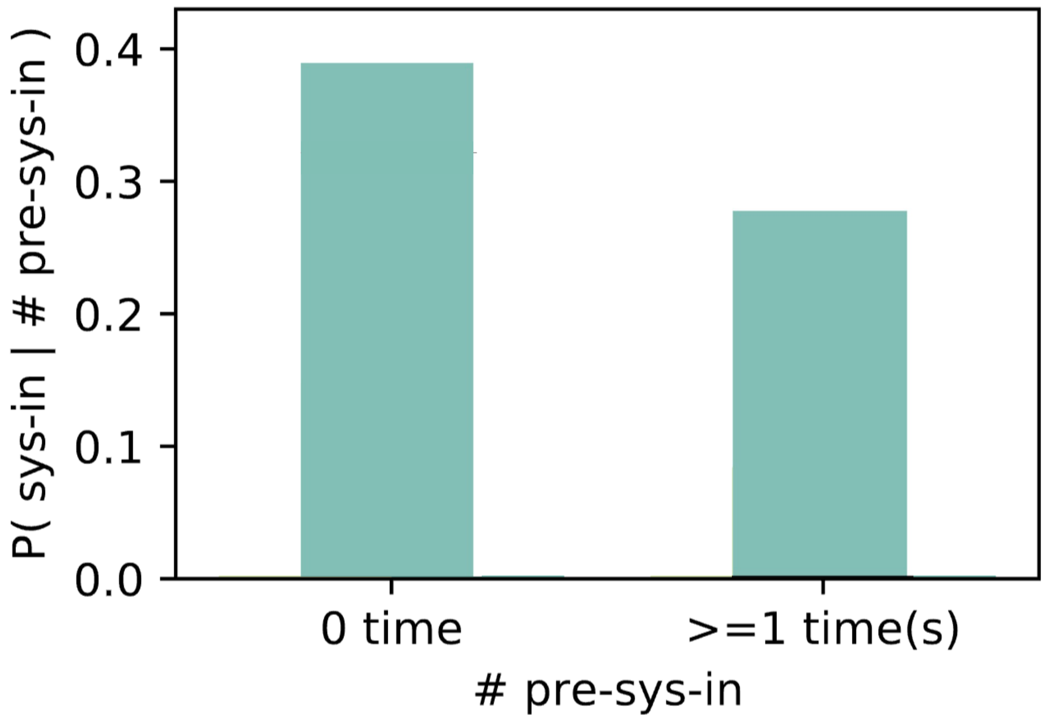
- **Our proposal**: model SIP by conditional random fields (CRFs)
  - CRFs are effective in capturing **dependencies between adjacent decisions**
  - CRFs have greater transparency

| **User decision at previous turn** | | | |
|---|---|---|---|
| | N | Score | score |
| | I | score | score |
| | | N | I |
| | | **System decision at next turn** | |

N: Non-initiative
I : Initiative

# Why we need a probabilistic graphical model for SIP

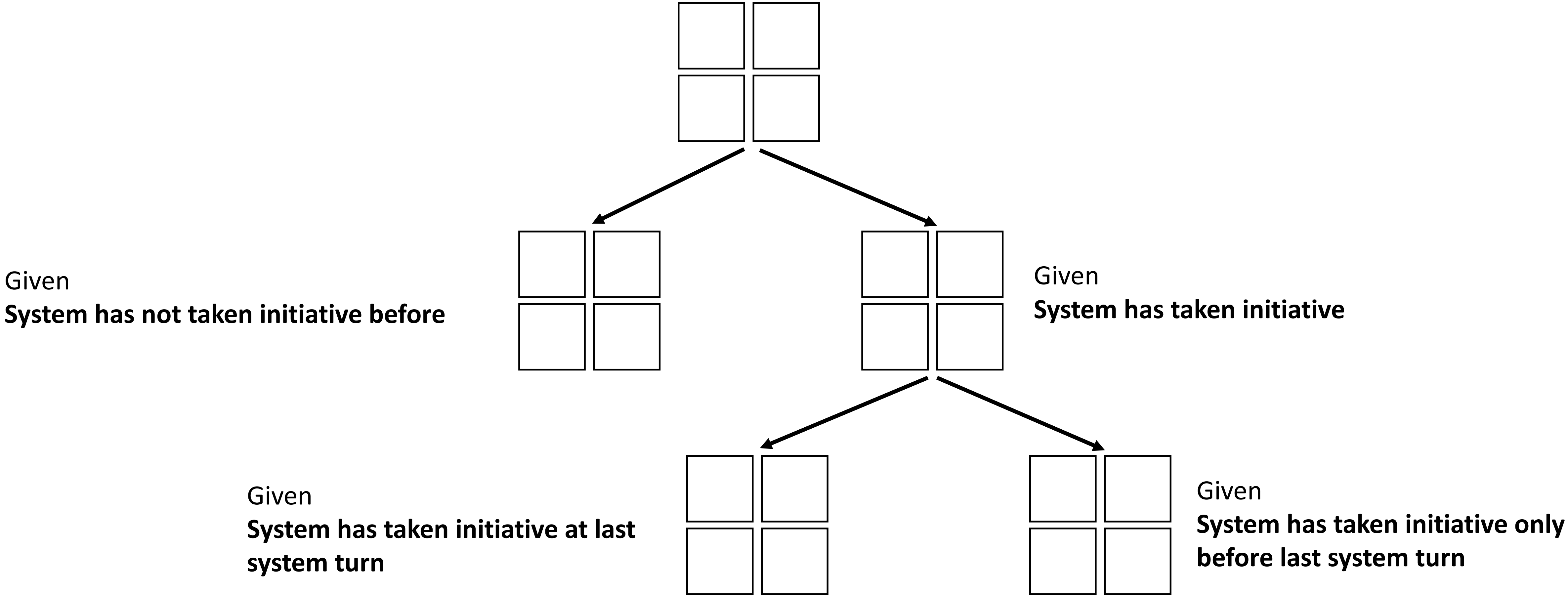
- Empirical analysis shows:
  - Dependencies between an initiative-taking decision and multi-turn features




- 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

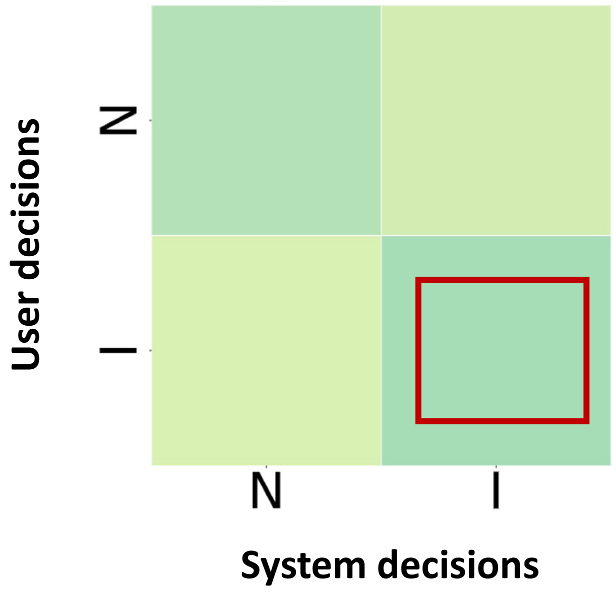
- Multi-turn feature-aware CRF achieves SOTA performance on SIP

| 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

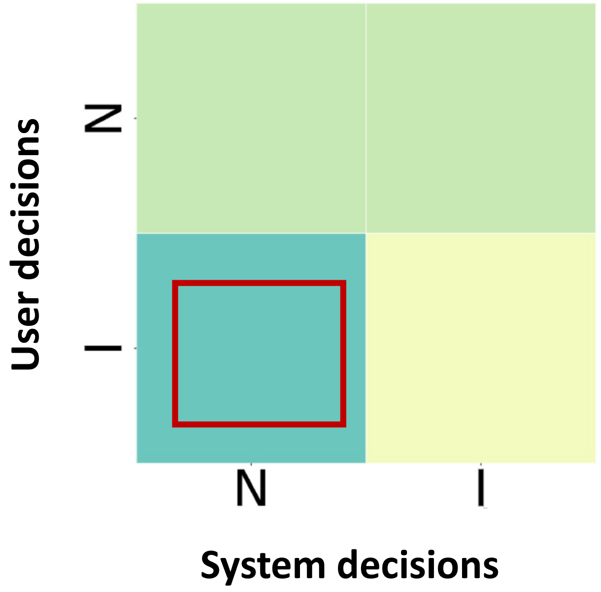
- Multi-turn feature-aware CRF exhibits great transparency

Given
**System has not taken initiative before**



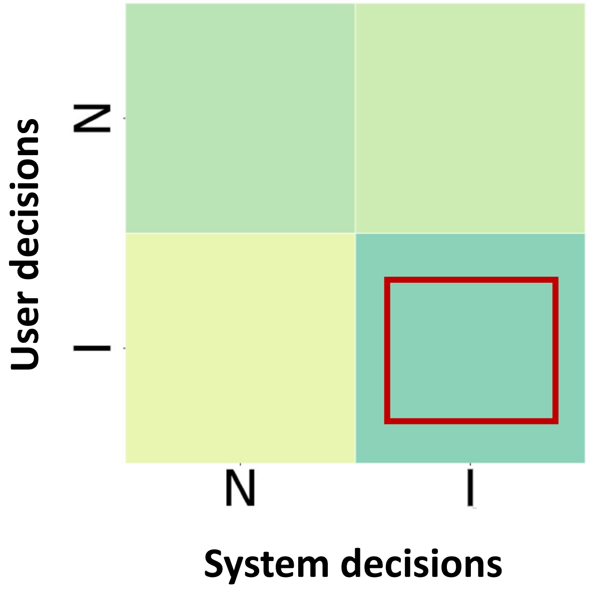

**Example:**
Turn 1: user asks a question
Turn 2 : system asks a clarifying question

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

# Outline

- ❑ Study 1: Structural dependency modelling for CS (CIKM 2023) [12 min]
- ❑ **Study 2: Query performance prediction for CS (SIGIR 2023 & ECIR 2023)** [6 min]
- ❑ Conclusion and future work [2 min]

# Query Performance Prediction for Conversational Search

**Chuan Meng**, Negar Arabzadeh, Mohammad Aliannejadi and Maarten de Rijke

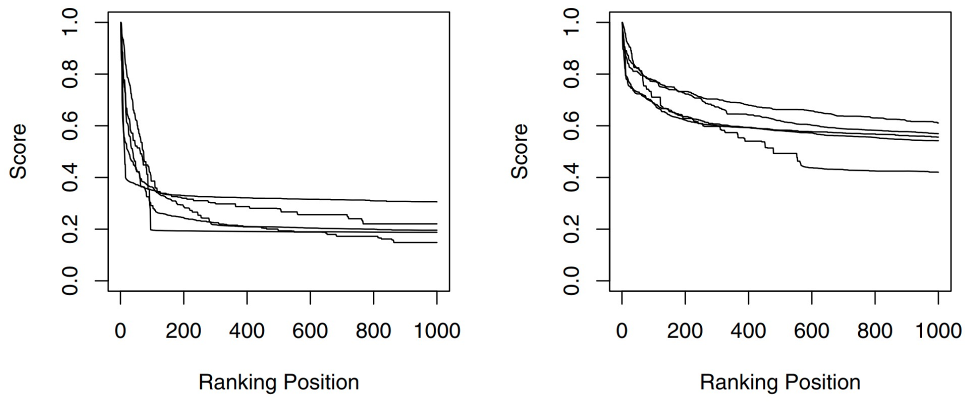
SIGIR 2023 & ECIR 2023

# 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 variant selection, ranker selection, and query routing

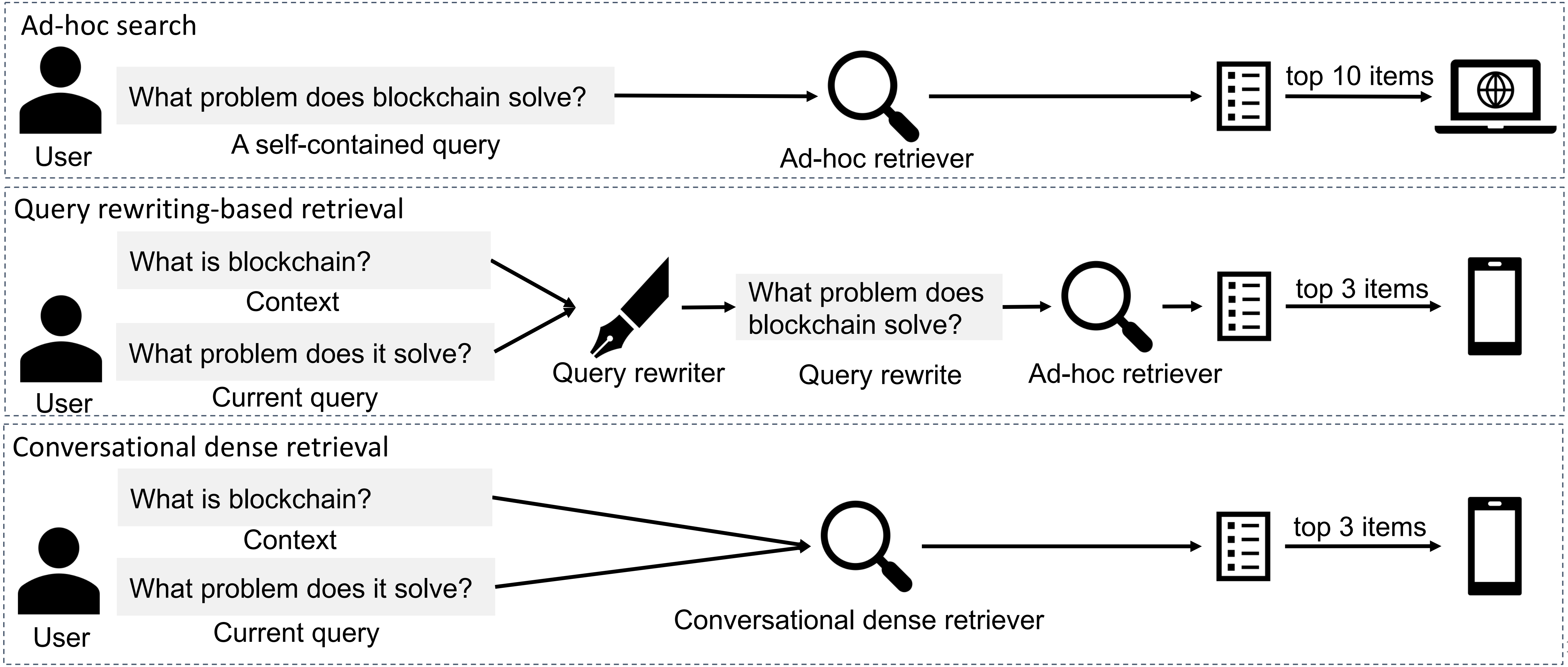
- QPP modelling
  - Unsupervised QPP methods




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

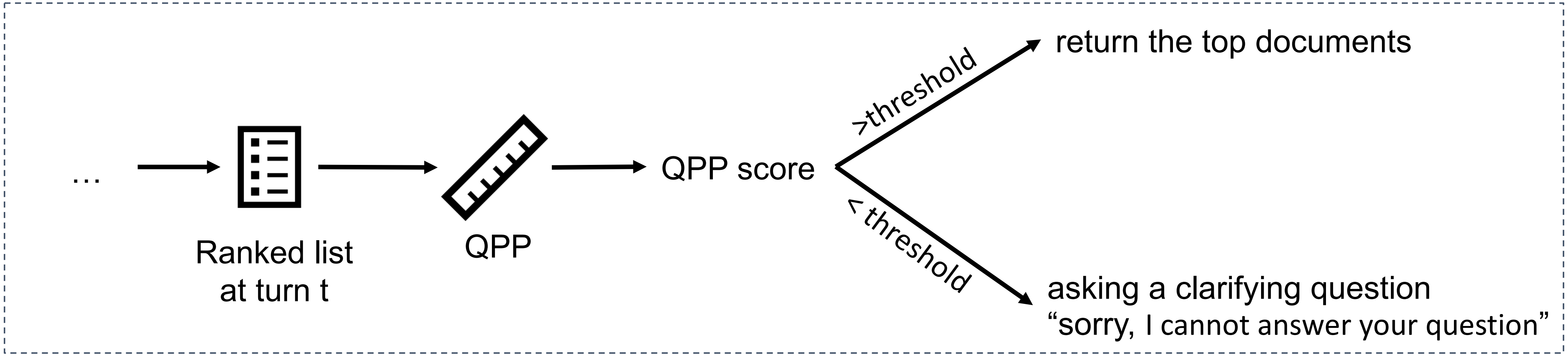
# Background—Conversational search (CS)

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



# Motivation

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




  - Unsupervised QPP methods perform on par with fine-tuned BERT models [1]

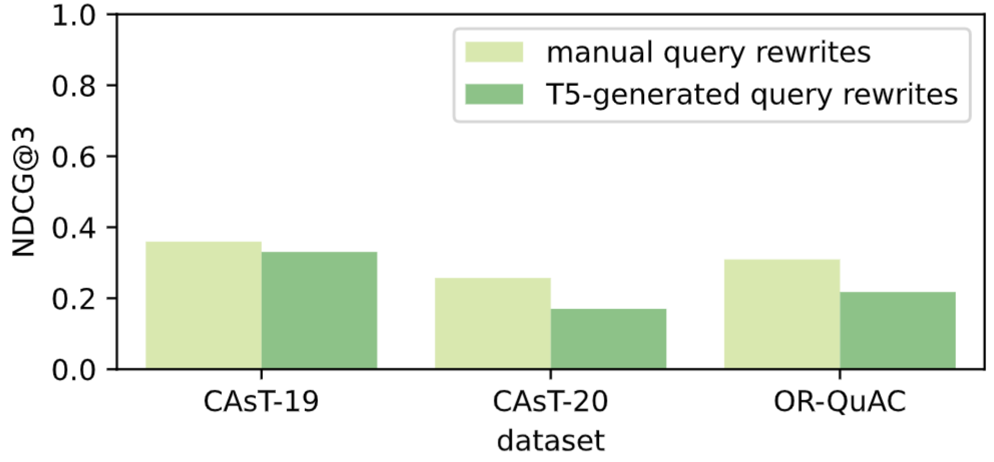
| Fine-tuned PLM | | QPP Methods | |
|---|---|---|---|
| BERT | 0.724 | WIG | 0.552 |
| BART | 0.739 | NQC | 0.690 |
| RoBERTa | 0.662 | SMV | 0.680 |
| | | $n(\sigma_{\%})$ | 0.643 |

AUC-ROC on ClariQ test set

[1] Arabzadeh et al. Unsupervised Question Clarity Prediction Through Retrieved Item Coherency. In CIKM 2022.

# Methodology

- How well QPP methods designed for ad-hoc search generalise in CS?
  - Reproduce various QPP methods in CS
  - They generalise well in CS




- How to improve QPP for CS?
  - Empirical analysis
    - Lower query rewriting quality yields lower retrieval quality
    - Query rewriting quality provides evidence for QPP
  - Propose perplexity-based QPP framework (PPL-QPP)
    - Evaluate the query rewriting quality via perplexity
    - Inject the quality into the QPP via linear interpolation
    - $final\ QPP\ score = \alpha \cdot \frac{1}{perplexity} + (1-\alpha) \cdot QPP\ score$
  - PPL-QPP results in higher QPP quality, especially on datasets where query rewriting is challenging

# Conclusion

- Contributions
  - A comprehensive reproducibility study that reproduces existing QPP methods in CS
  - A new QPP framework that improves QPP for CS using query rewriting quality
  - The data and code are open-sourced https://github.com/ChuanMeng/QPP4CS

README

## Query Performance Prediction for Conversational Search (QPP4CS)

VISITORS 2,048

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)

The repository offers the implementation of a comprehensive collection of pre- and post-retrieval query performance prediction (QPP) methods, all integrated within a unified Python/Pytorch framework. It would be an ideal package for anyone interested in conducting research into QPP for ad-hoc or conversational search.

QR code for the repo

# Outline

- ❑ Study 1: Structural dependency modelling for CS (CIKM 2023) [12 min]
- ❑ Study 2: Query performance prediction for CS (SIGIR 2023 & ECIR 2023) [6 min]
- ❑ **Conclusion and future work** [2 min]

# Conclusion and Future Work

- Contributions
  - Structural dependency modelling for CS
  - Query performance prediction (QPP) for CS

# Thank you!

Chuan Meng
chuanmen@amazon.com
c.meng@uva.nl