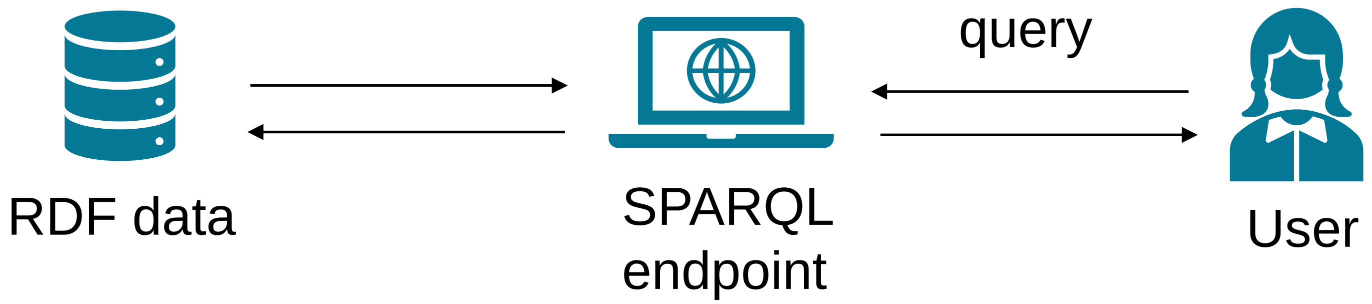
Revealing Secrets in SPARQL Session Level

Xinyue Zhang, Meng Wang, Muhammad Saleem, Axel-Cyrille Ngonga Ngomo, Guilin Qi, and Haofen Wang.

1. Background

- **Knowledge graphs** help users to discover **information of interest** by using **live SPARQL services**.



1. Background

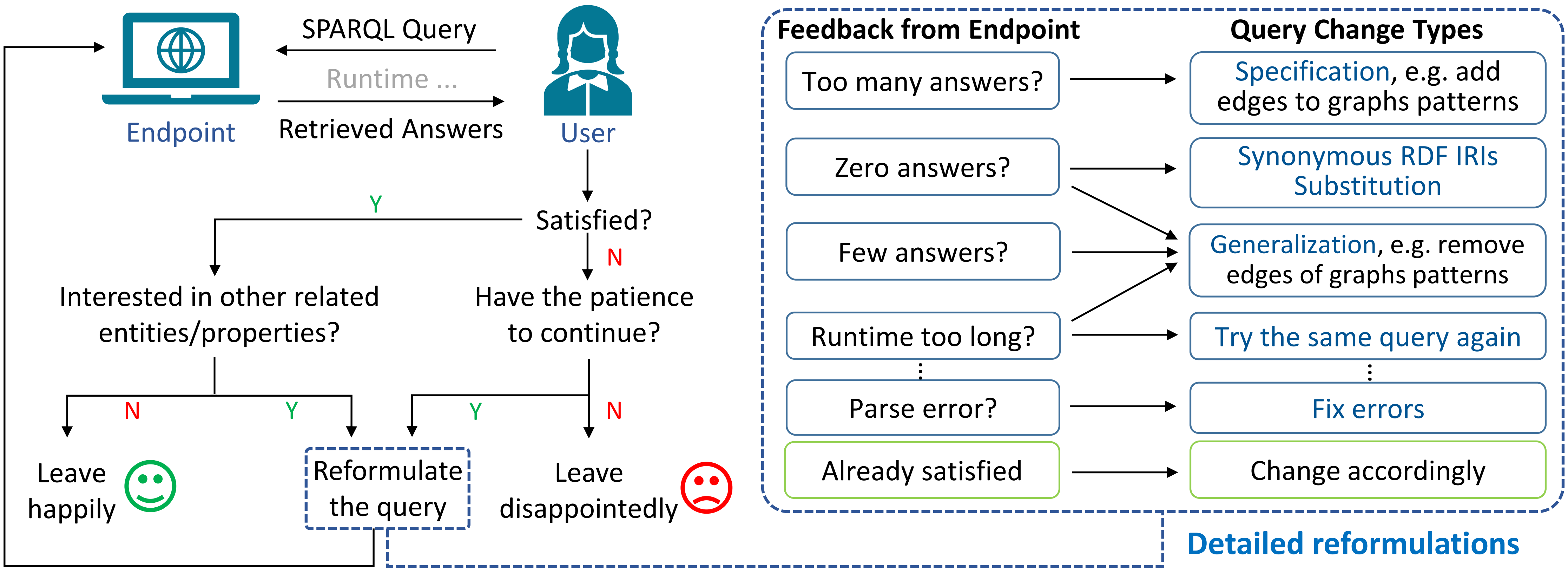
- Users often **fail** to express their information needs in **one succinct query**.

  Because of their **unfamiliarity** with:

  - the ontology underlying the endpoints,
  - SPARQL syntax.

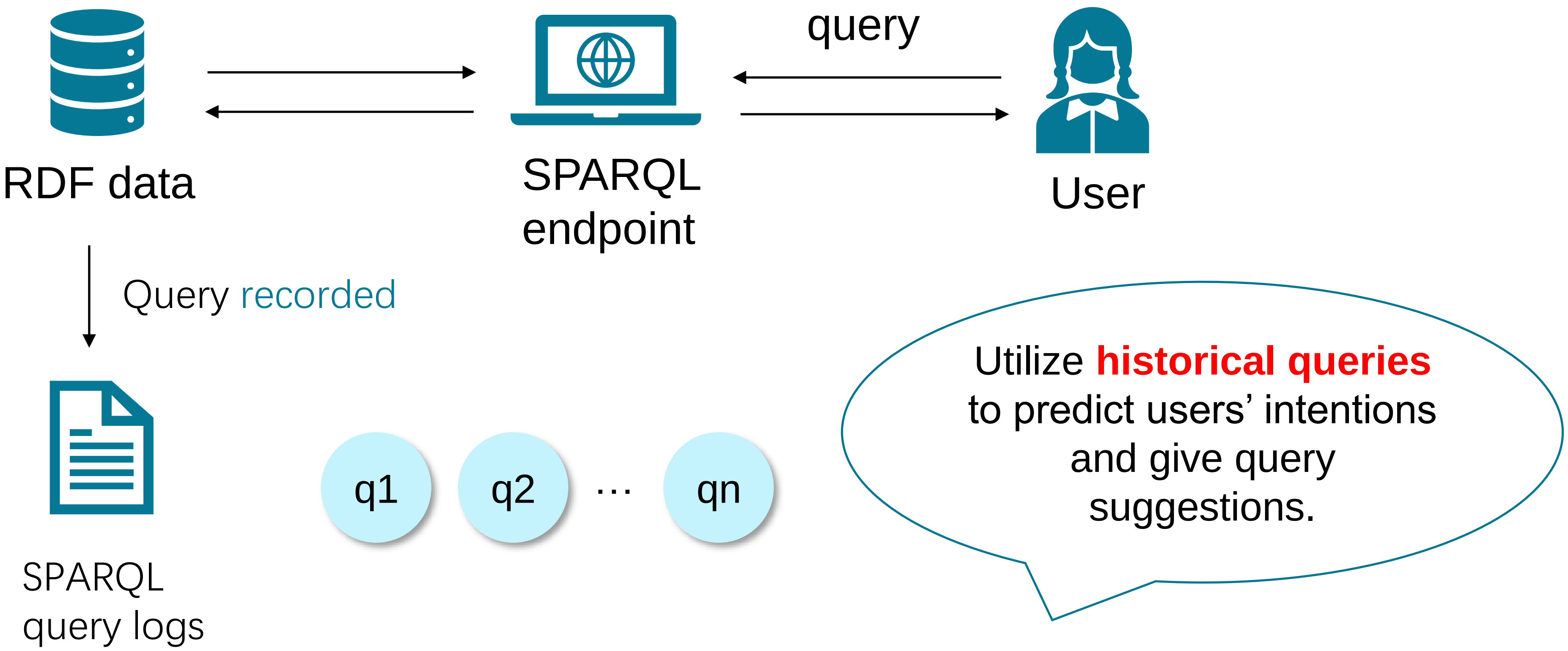
1. Background

- Therefore, SPARQL queries are **continuous refined** to retrieve satisfying results.



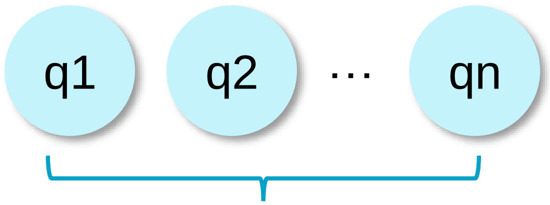
1. Background

- How to help users?

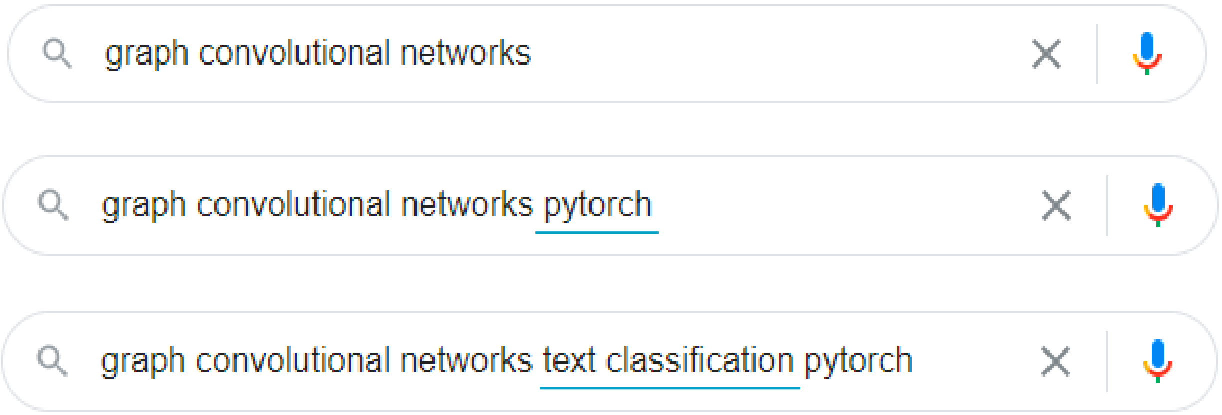


2. Our work: SPARQL query log analysis in session level

- Research Object
  - SPARQL Search Sessions



1. Executed by **one user**.
2. Within a fixed time threshold (1h).
3. Continuous query pairs $(q_i, q_{\{i+1\}})$ share **at least one** common variable or IRI.



2. Our work: SPARQL query log analysis in session level

Challenges:

- The information we could utilize in sessions.

Implicit Feedback	Traditional IR Field	SPARQL Scenario
Dwell Time in webpages	√	X
Clicked URLs	√	X
Query Reformulations	√	√

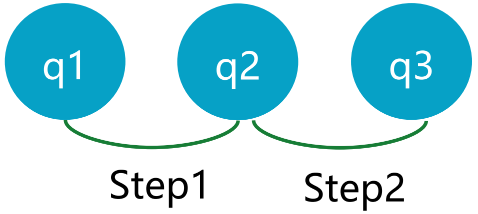
➡ Focus on the **query changes!**

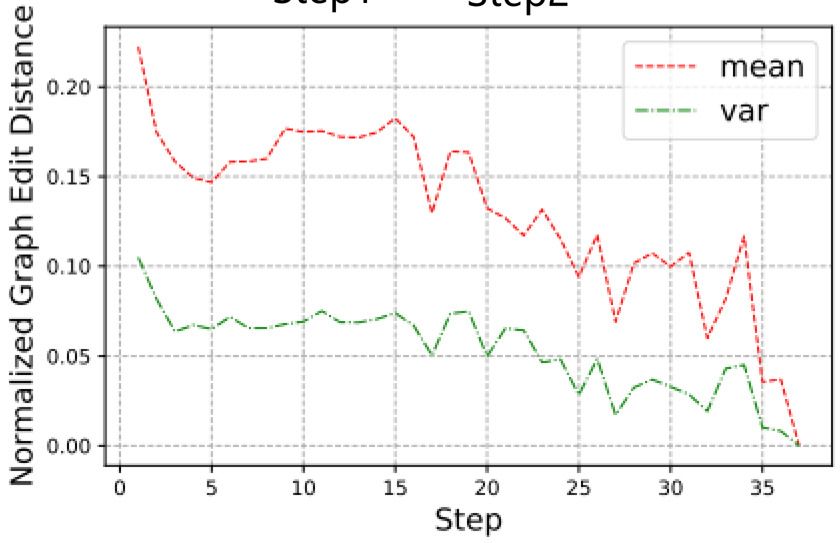
2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Structural** changes evaluated by **Graph Edit Distance (GED).**

    - q_{i}, q_{i+1}





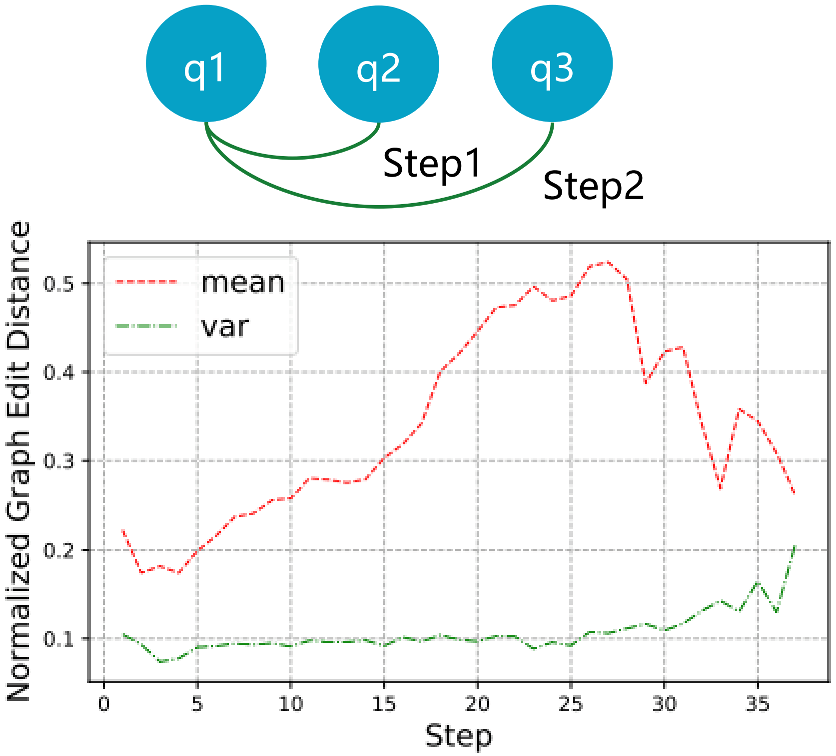
Query changes between two continuous queries is **increasingly indistinct** as users **getting closer to their information needs.**

2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Structural** changes evaluated by **Graph Edit Distance (GED).**

  - q_1, q_i



GEDs between (q_1, q_i) **increase consistently at first, then decrease**. Users may use **prior query structures** with different IRIs to explore other related information.

2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Structural** changes evaluated by graph pattern similarity.

$$\begin{aligned} V \ = [\ &\#triplePatterns,\ \#BGP,\ \#Projection,\ \#SinkJoinVertex, \\ &\#StarJoinVertex,\ \#HybridJoinVertex,\ \#PathJoinVertex, \\ &MaxJoinDegree,\ MinJoinDegree,\ MeanJoinDegree\ ]. \end{aligned}$$

2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Structural** changes evaluated by graph pattern similarity.

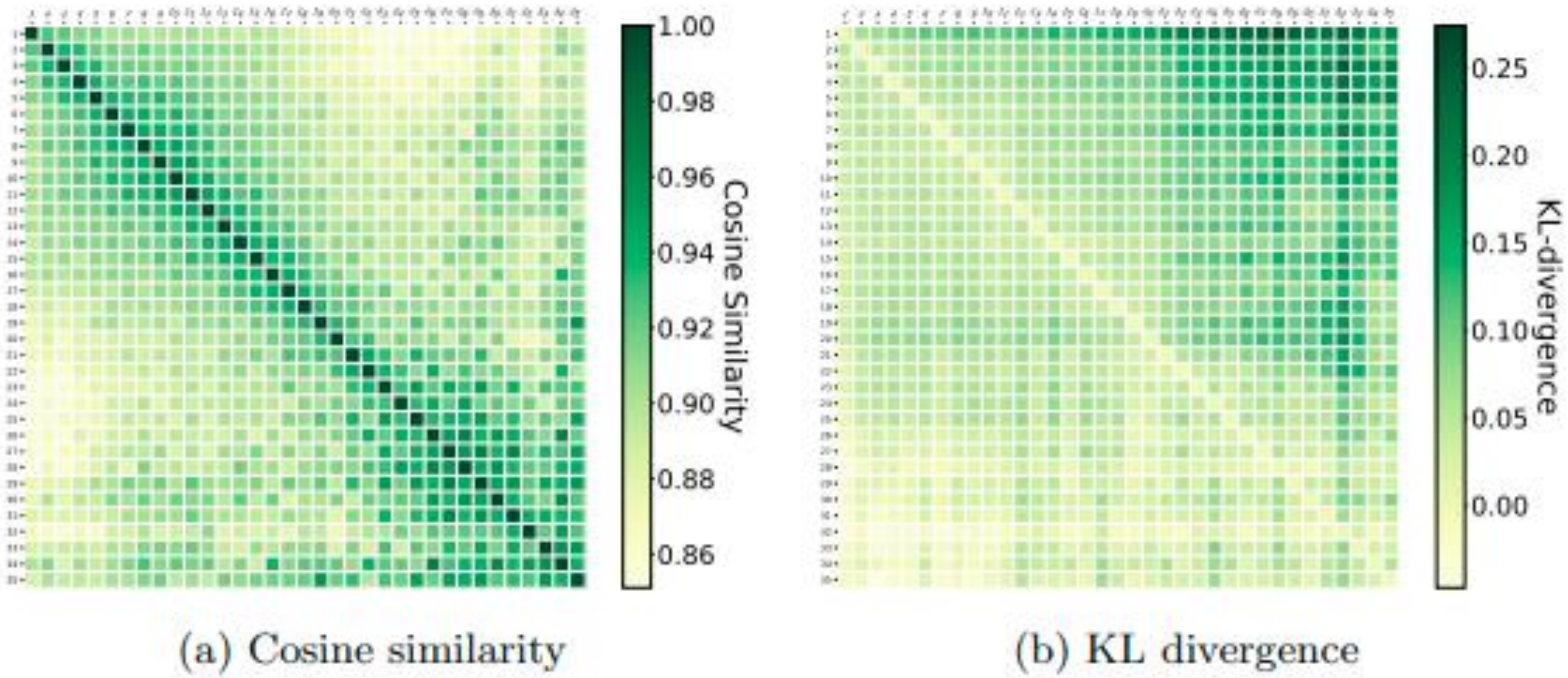


Fig. 6: Graph pattern similarity of query sequence Q in sessions.

Users usually change the **structure of graph patterns** **slightly**.

2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Structural** changes evaluated by **IRI term similarity**.

Q1: SELECT ?who

WHERE {<http://dbpedia.org/resource/Minecraft> <http://dbpedia.org/ontology/designer> ?who. }

Bags of Words are used to construct a IRI-term vector to present each SPARQL query.

<http://dbpedia.org/resource/Minecraft>	**<http://dbpedia.org/ontology/designer>**	**Other IRIs …**	
1	1	…0…	1

1 if this IRI present else 0

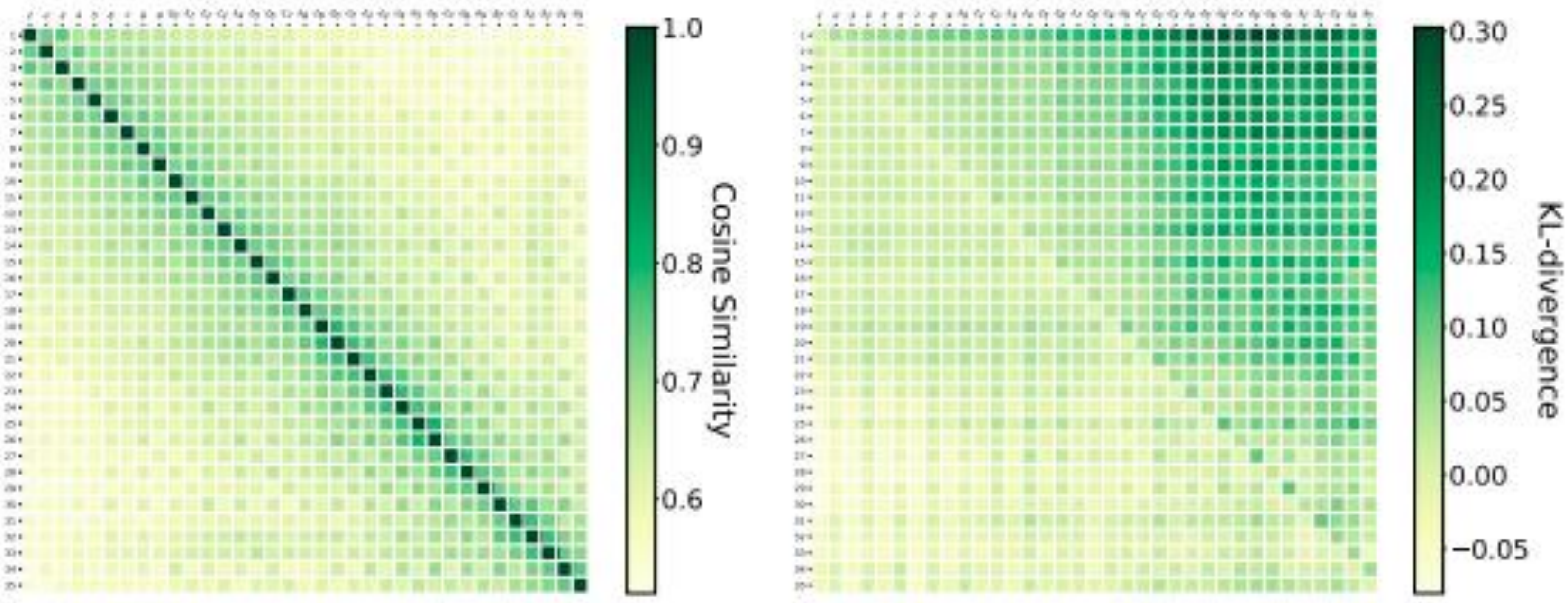
All 0

This column is used to avoid all-zero vectors.

2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Structural** changes evaluated by **IRI term similarity**.



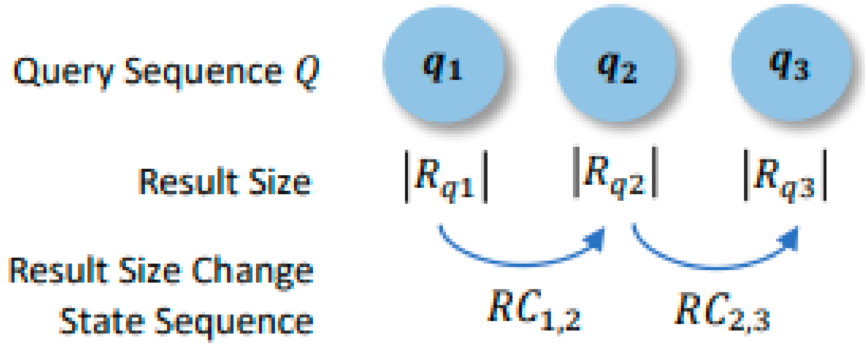
(a) Cosine similarity (b) KL divergence

Users tend to **include more IRI terms** and express a clearer intention **as the session move forward**.

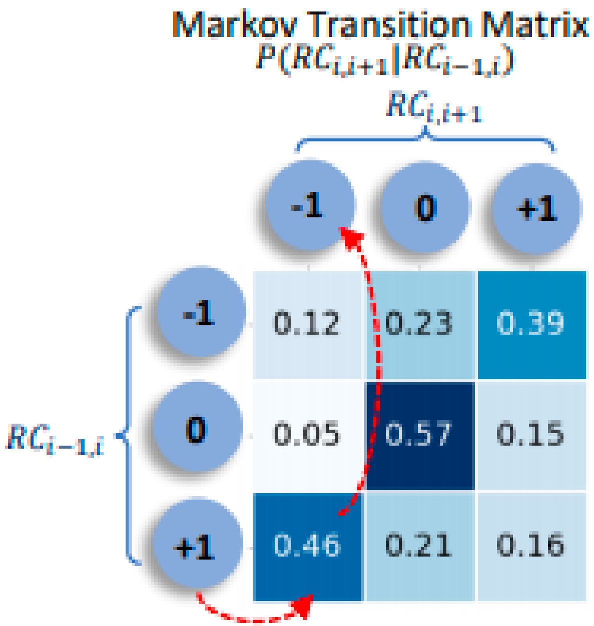
2. Our work: SPARQL query log analysis in session level

Query changes in SPARQL search sessions:

- **Intention** changes evaluated by **changes of result set size**.



$$RC_{i,i+1} = \begin{cases} -1 & if\ |R_{qi+1}| < |R_{qi}| \\ 0 & if\ |R_{qi+1}| = |R_{qi}| \\ +1 & if\ |R_{qi+1}| > |R_{qi}| \end{cases}$$



Previous result size change does **influence** the **intention of the current query.**

2. Our work: SPARQL query log analysis in session level

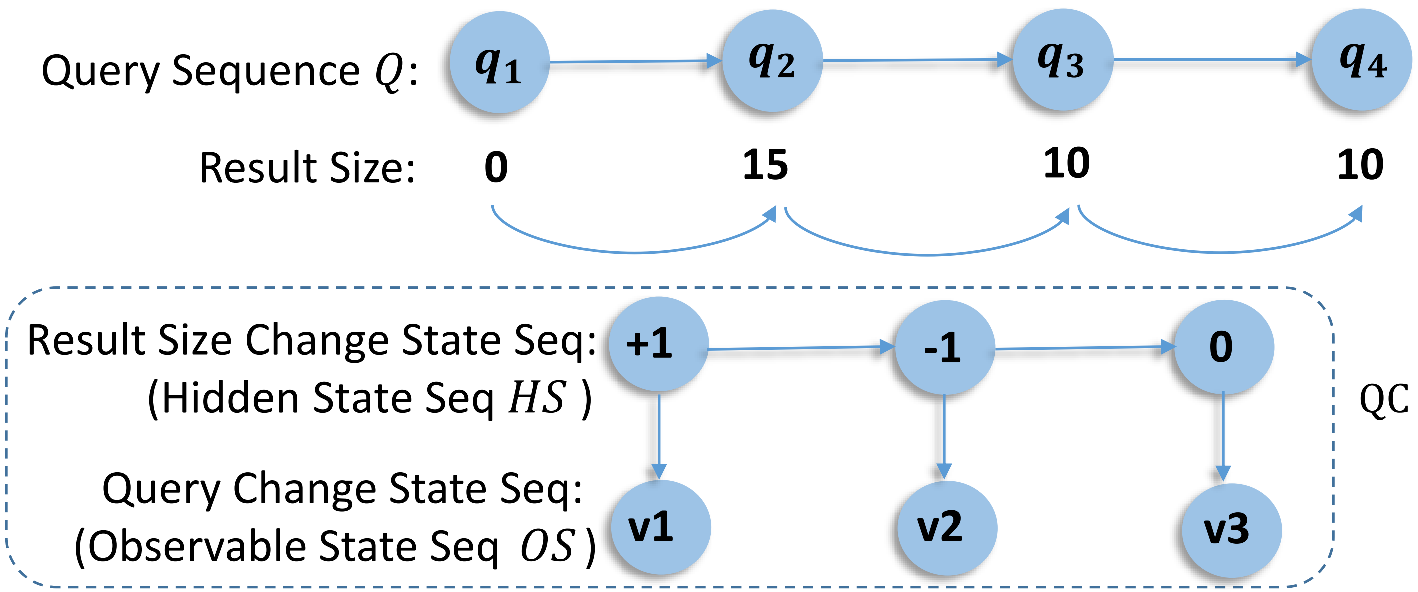
Query changes in SPARQL search sessions:

- Distribution of detailed reformulation strategies.
  - ➢ **Additions/deletions of operators.**
  - ➢ **Additions/deletions of triple patterns.**
  - ➢ **Substitutions of individual elements of triple patterns.**
  - ➢ **Substitutions on different join vertex types and their neighbors.**
  - ➢ **Additions/deletions/substitutions of FILTER constraints.**

3. An application example

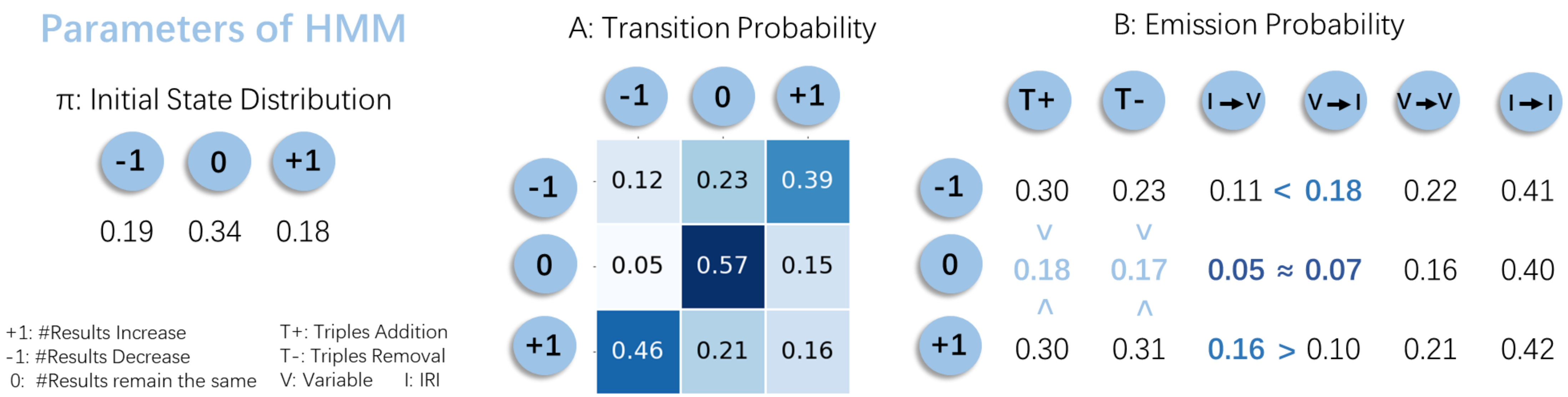
Hidden Markov Model:

	Hidden states H	Observable states U
Human users	Abstract intentions	Query reformulations
Our application	Result Size Change States RC	Query Change States QC



$$QC = \begin{cases} \text{Triple patterns Addition } T_+ \\ \text{Triple patterns Removal } T_- \\ \text{Triple patterns Substitution} \begin{cases} IRI \to VAR \\ IRI \to IRI \\ VAR \to VAR \\ VAR \to IRI \end{cases} \end{cases}$$

3. An application example



The simple demo we present here is capable of:

- **predicting user behavior** (evaluation problem: given parameters of HMM λ = (A, B, π), calculate p(O|λ))
- **understanding user intentions** (decoding problem: given λ = (A, B, π) and O = (o1, o2 · · · ot), calculate a sequence I = (i1, i2 · · ·it) that maximize P(I|O))

4. Conclusions

- **Define** the SPARQL search **session**.
- Investigate **potential relations between queries** in single search **sessions**.
- Conduct a comprehensive analysis of **query reformulations.**
- Provide an **example application** to illustrate the potentiality of utilizing user behaviors in a search session.

Thanks!