Personalized Knowledge Graph Summarization: From the Cloud to Your Pocket

Safavi, T., Belth, C., Faber, L., Mottin, D., Müller, E., & Koutra, D.
University of Michigan, Google, Aarhus University, B-IT Center

Presenter Eleni Ilkou

Introductory Terms

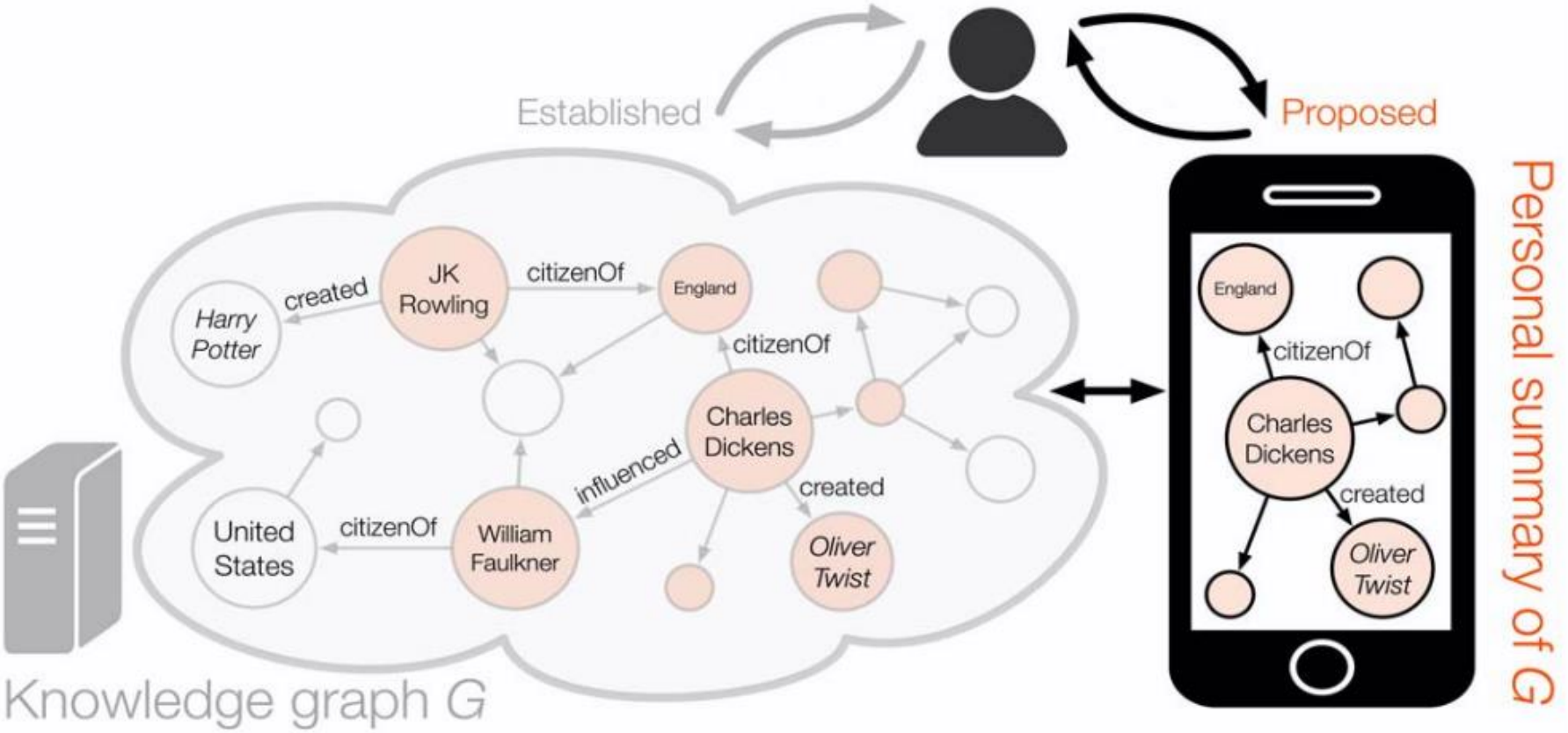


Fig. 1: Personalized KG summarization for a user interested in books and authors. Given seed information about the user's interests over G, GLIMPSE constructs an on-device personal summary of G (i.e., a mini-KG) for anytime information access.

Paper notes

PKGS notes

- Goal: construct compact “personal summaries” of KGs containing only the facts most relevant to individuals’ interests
- Problem: Mathematically formulate the problem of personalized KG summarization
- Framework: GLIMPSE, a flexible summarization framework that combines strong theoretical guarantees with the scalability necessary for large KGs

PKGS notes

- Evaluation: Analysis in GLIMPSE in a direct query answering task using real queries to KGs of up to one billion triples. GLIMPSE personal summaries outperform summaries created by strong baselines by up to 19% in query answering F1 score across various simulated user models. They demonstrate GLIMPSE's consistency across datasets, and provide in-depth analysis of their results.

PKGS notations

TABLE I: Table of main symbols.

Symbol	Meaning
G	Knowledge graph $G = (E, R, T)$ with entity set E, relation set R, and triple set T
e_i	i-th entity in entity set E
r_k	k-th relation in relation set R
x_{ijk}	Triple $(e_i, r_k, e_j) \in T$ with entities $e_i, e_j \in E$, relation r_k
G_Q	Query graph $G_Q = (E_Q, R_Q, T_Q)$ to G
Q_u	Query log $Q_u = (G_Q^1, \ldots, G_Q^n)$ of user u on G
S_u	Personal summary $S_u = (E_u, R_u, T_u) \subseteq G$ of user u
K	Number of triples in personal summary S_u

Introductory Terms

- Query graph $G_Q=(E_Q,R_Q,T_Q)$,
  - may be a subgraph of G or may contain elements not in G,
  - is directed, acyclic, and fully connected
- Query log $Q_u=(G^1_Q,...,G^n_Q)$, sequence of queries

PKGS notes

- Problem
  - Given a knowledge graph G, a user u's past queries to G, and a user-specific resource (device or application) constraint, efficiently infer a personal summary $S_u \subseteq G$ under the given constraint that best captures the user's preferred facts in G, as expressed by her past queries

GLIMPSE framework

GLIMPSE

1) User preferences

Infer entities and relations of potential interest to the user based on the historical queries

1) Conduct a summary

- maximizing a user-specific utility function drawn from these inferred preferences

GLIMPSE

Step 1: User preferences

- Entity preference
  - An interest in a single entity (e.g.,Charles Dickens) may signal interest in connected entities in the KG (e.g.,Oliver Twist,Great Expectations,England, etc)

$$\Pr(e_i|Q_u) \propto \underbrace{\sum_{G_Q \in Q_u} \mathbb{1}_{E_Q}(e_i)}_{\text{historical pref.}} + \gamma \underbrace{\sum_{e_j \in N(e_i)} \mathbb{1}_{E_Q}(e_j)}_{\text{graph structure}},$$

GLIMPSE

Step 1: User preferences

- Triple/facts preference
  - To capture the user's preference for triple $x_{ijk}=(e_i,r_k,e_j)\in T$. They follow the standard conditional independence assumption in graph mining and KG learning

$$\Pr(x_{ijk}|Q_u) \propto \Pr(e_i|Q_u)\Pr(r_k|Q_u)\Pr(e_j|Q_u)$$

GLIMPSE

Step 2: Conduct a summary

- Constructing the summary
  - Given user preference model, let $Pr(S_u|Q_u)$ be the estimate of how well a constructed summary $S_u=(E_u,R_u,T_u)$ captures the user's inferred preferences, conditioned on Q_u

$$\Pr(S_u|Q_u) \propto \prod_{e \in E_u} \underbrace{\Pr(e|Q_u)}_{\text{“topic” pref.}} \prod_{x_{ijk} \in T_u} \underbrace{\Pr(x_{ijk}|Q_u)}_{\text{fact pref.}}.$$

GLIMPSE

- Utility of personal over non-personalized
  - Utility maximization problem, where the utility function to be maximized is non negative. They exploit this non negativity to show that our utility function is submodular which allows us to devise a near-optimal approximation algorithm

Data
and
Evaluation

Data

- Real queries: WebQuestionsSP (Freebase)
- Synthetic queries: based on WebQuestionsSP structure (Dbpedia, YAGO)
  - Steps in path without self-loops <3
  - Number of query's answer <relation, argument> < 5

Evaluation

Evaluation focuses on the following questions:

- Q1: How well do GLIMPSE personal summaries answer user queries under various conditions and constraints?
- Q2: Can GLIMPSE handle large real knowledge graphs?
- Q3: How do changes in parameters affect GLIMPSE?

Baselines

- PPR: personalized PageRank
- PPR-n: PPR with walk length n
- TCM: Graph stream summarization
- CACHE: frequency-based "caching" strategy
  - devised a method that sorts all entities in the user's query history Q_u by their query frequency

Evaluation

TABLE III: GLIMPSE consistently outperforms competitors across the two user models defined in § V-B: Average F1 score for all methods, knowledge graphs, and user querying models following the settings in § V-A. All averages are over 15 simulated users per KG and querying model. Top performer per experiment in bold. In the GLIMPSE column, the value in parentheses denotes the number of percentage points improvement over the best baseline. ▲: significant improvement by GLIMPSE over the best baseline for a two-sided t-test at $p < 0.01$.

User model	*Dataset*	TCM	CACHE	PPR-1	PPR-2	PPR-5	PPR-10	GLIMPSE (+ improve.)
Few topics ($t \in 2 \dots 5$)	DBPedia	0.687 ± 0.09	0.684 ± 0.09	0.693 ± 0.09	0.846 ± 0.09	0.824 ± 0.09	0.819 ± 0.09	$\mathbf{0.980 \pm 0.02}^{\blacktriangle}$ (+0.134)
	YAGO	0.539 ± 0.11	0.558 ± 0.10	0.549 ± 0.08	0.672 ± 0.08	0.659 ± 0.08	0.653 ± 0.08	$\mathbf{0.814 \pm 0.11}^{\blacktriangle}$ (+0.142)
	Freebase	0.678 ± 0.06	0.707 ± 0.05	0.469 ± 0.05	0.486 ± 0.05	0.499 ± 0.04	0.499 ± 0.04	$\mathbf{0.724 \pm 0.06}$ (+0.017)
Many topics ($t \in 5 \dots 10$)	DBPedia	0.585 ± 0.08	0.603 ± 0.08	0.650 ± 0.08	0.782 ± 0.07	0.765 ± 0.08	0.764 ± 0.08	$\mathbf{0.971 \pm 0.03}^{\blacktriangle}$ (+0.189)
	YAGO	0.526 ± 0.07	0.546 ± 0.07	0.552 ± 0.08	0.685 ± 0.07	0.673 ± 0.07	0.670 ± 0.07	$\mathbf{0.768 \pm 0.11}^{\blacktriangle}$ (+0.082)
	Freebase	0.542 ± 0.07	0.577 ± 0.05	0.345 ± 0.05	0.339 ± 0.05	0.350 ± 0.05	0.354 ± 0.05	$\mathbf{0.593 \pm 0.06}$ (+0.016)

Evaluation

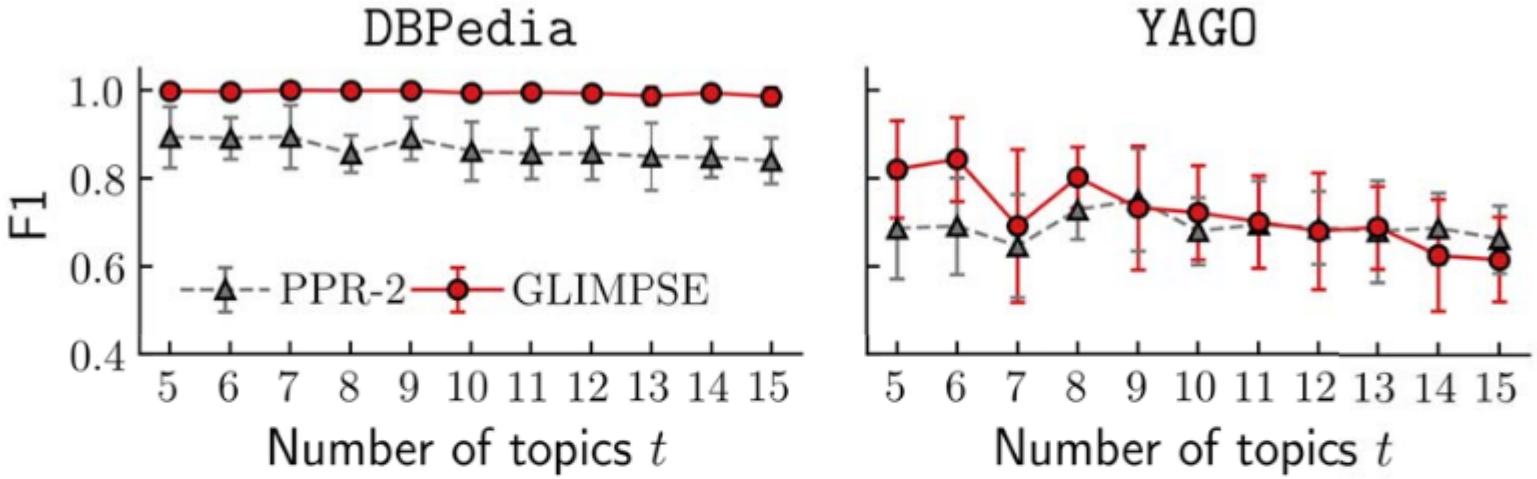
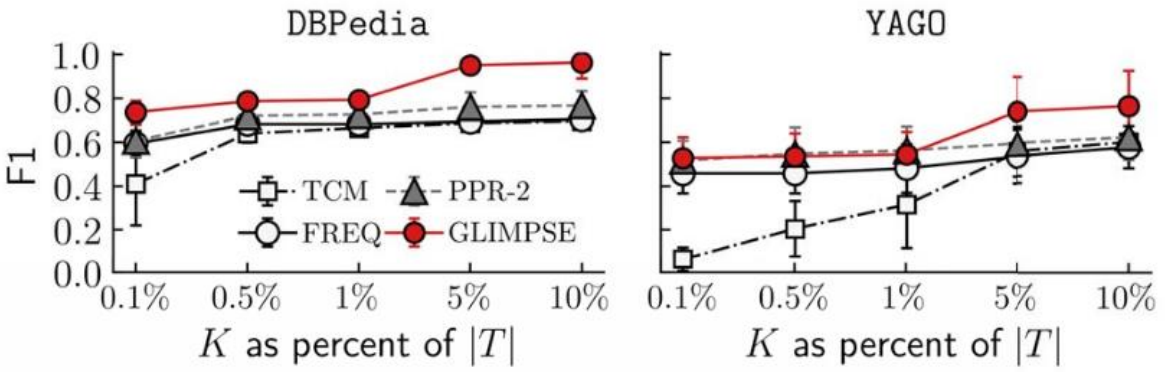
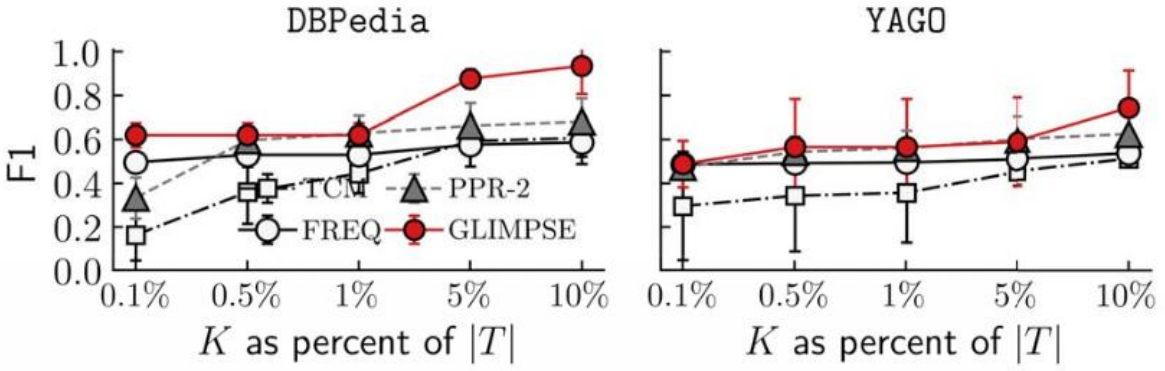


Fig. 3: Comparing GLIMPSE and its closest competitor PPR-2 by varying the number of topics of interest, averaged over 15 simulated users each. GLIMPSE consistently outperforms PPR-2 on `DBPedia`, significant at $p < 0.01$. It is also comparable to or better than PPR-2 on `YAGO` for 10-15 topics.

Evaluation



(a) Few topics model



(b) Many topics model

Fig. 4: GLIMPSE consistently outperforms baselines across constraints: Performance comparison varying K as a percentage of the number of triples $|T|$ in the original KG across user models.

Scalability

Evaluation

TABLE IV: Comparison of GLIMPSE runtime on a subset of `Freebase` with and without the optimizations discussed in § III-E. Evidently, the optimizations are necessary for GLIMPSE to be feasible on encyclopedic knowledge graphs.

	GLIMPSE	With OPT1 only	With OPT2+3 only
Runtime (seconds)	2.11 ± 0.08	15487.93 ± 978.05	28980.46 ± 416.38
Relative to GLIMPSE	$1\times$	$7340\times$	$13734\times$

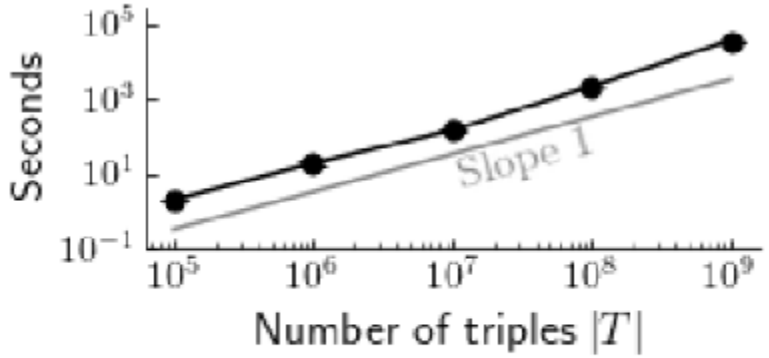


Fig. 5: GLIMPSE scalability (seconds) on `Freebase`.

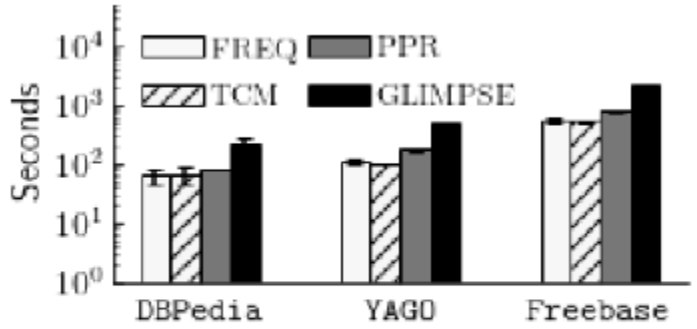
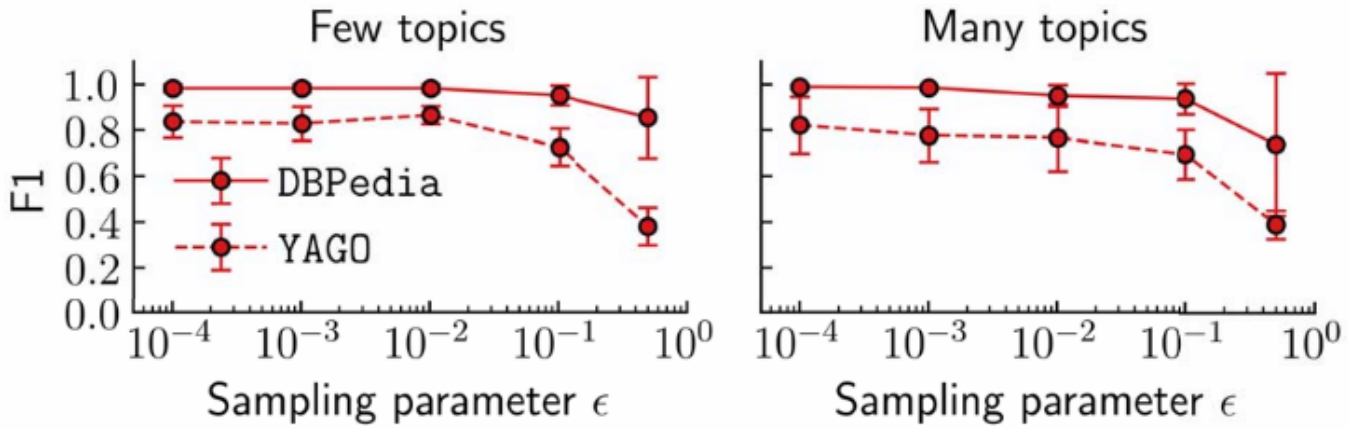
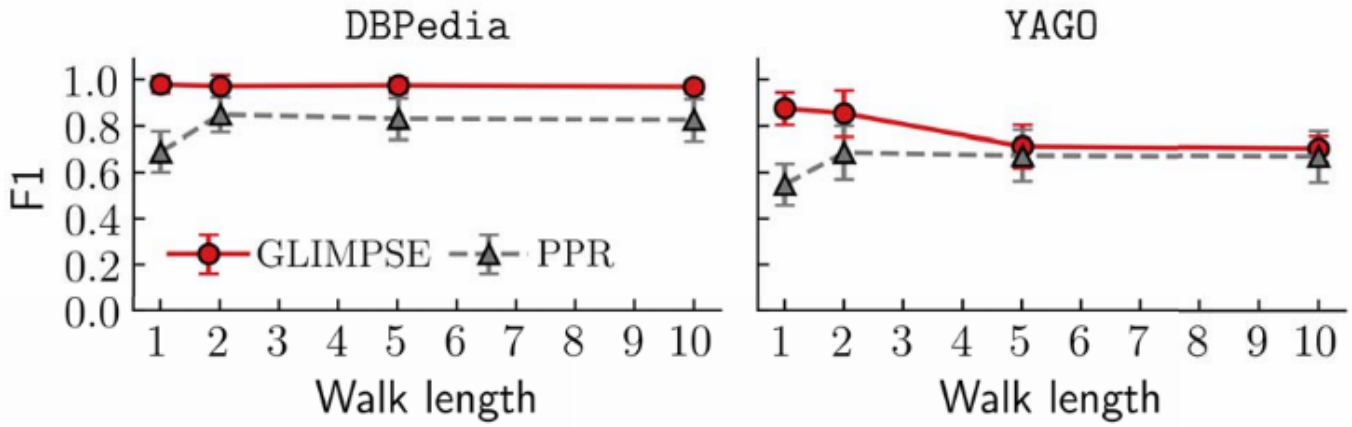


Fig. 6: Summarization time on all KGs.

Evaluation



(a) Varying the sampling parameter ϵ from OPT2.



(b) Varying the random walk length on GLIMPSE and PPR.

Fig. 7: Parameter analysis of GLIMPSE and competitors.

Conclusion

Evaluation

- This paper proposes personalized knowledge graph summarization
- Motivation: Limited information needs of individuals compared to information KGs' facts
- Approach: GLIMPSE, empirical and theoretical strengths
- Future Work: make use of the semantics provided by ontologies, and contextual user cues (e.g., location, preferred language), as is common intraditional ad-hoc web search