APEX^2: Adaptive and Extreme Summarization for Personalized Knowledge Graphs

August 3-7, 2025
KDD2025

Zihao Li[1], Dongqi Fu[2], Mengting Ai[1], Jingrui He[1]

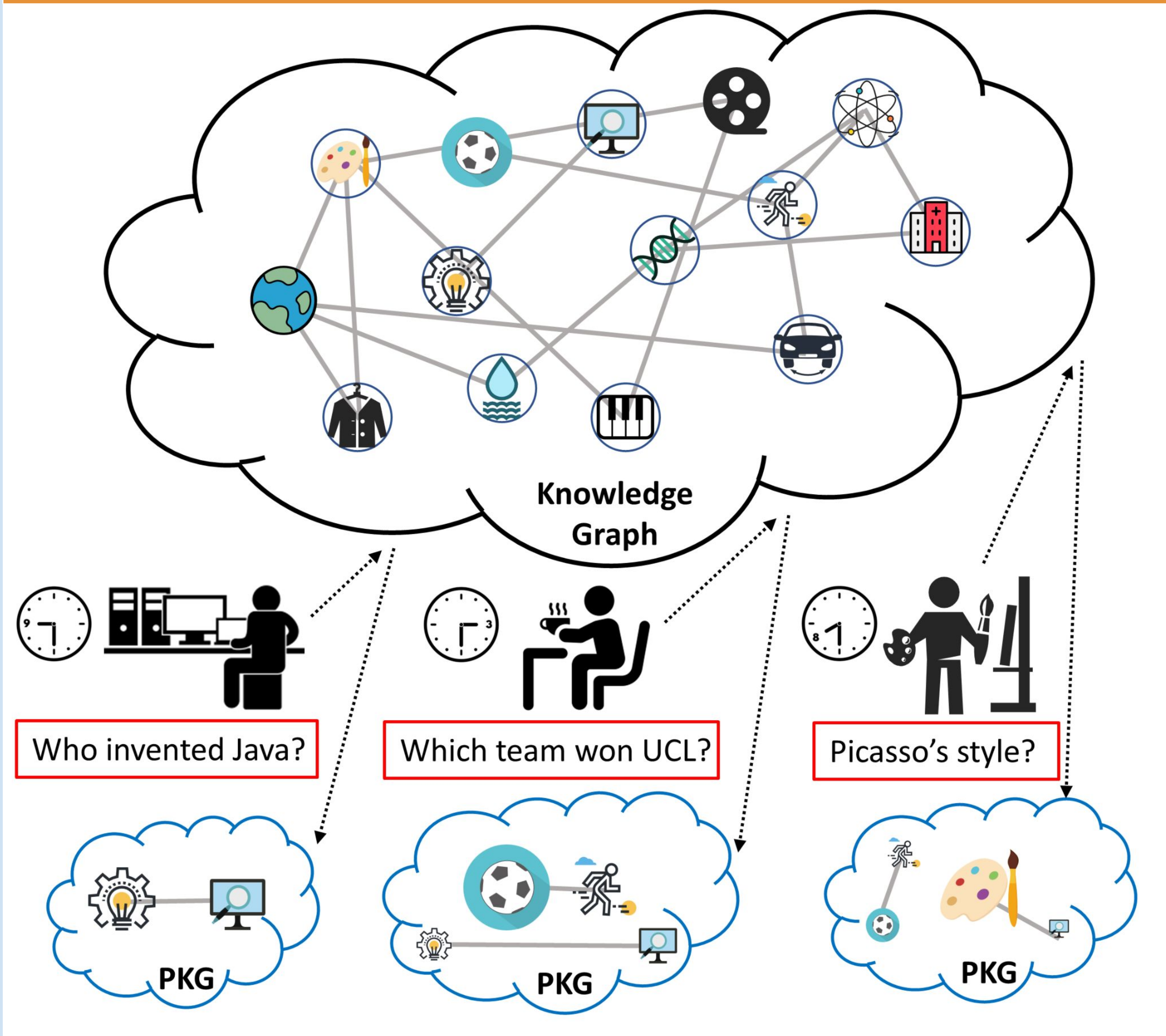
[1]University of Illinois Urbana-Champaign [2]Meta AI contact: zihaoli5@illinois.edu

paper

Open for U.S.A. internship
prev. MSR; now@Amazon

lab page

Vision: Evolving Knowledge Graphs according to Evolving User Interests



Evolving User Interests and Corresponding Adaptive PKGs that could Answer Them

Motivations

- KGs are large and even becoming larger
- But each user only cares a small portion of it

We can make KGs more

- efficient (in terms of knowledge retrieval)
- private for users

by creating *personalized KG (PKG) for each user*

- adaptively, as user's interest shifts over time
- extremely, with less than 0.1% size of original KG.

The entire KG is stored on a cloud server,
the PKG is stored on the user's device.

What can be the "users"

- human-users of KGs
- KG-powered applications/processes
- Retrieval-augmented generation systems
- AI agents with knowledge access

In a nutshell…

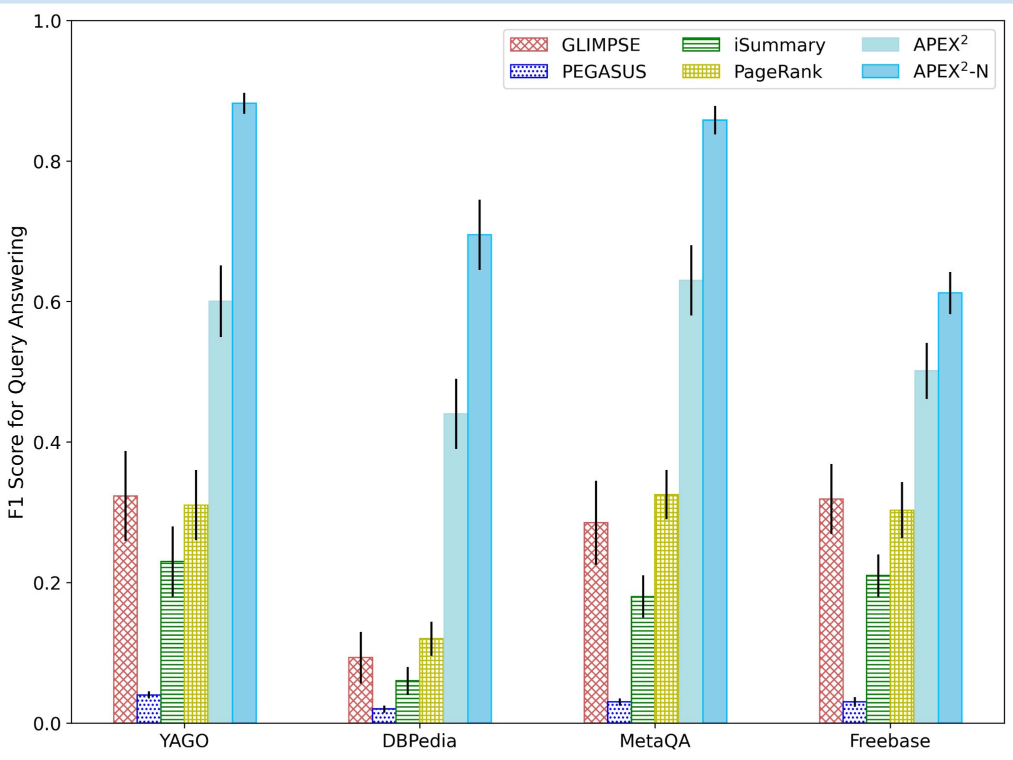
Our solution:

(1) Model the user interest by heat diffusion
(2) Incrementally maintain user's interest on entities, relations and triples
(3) Pick top-k ranked items and construct the PKG.

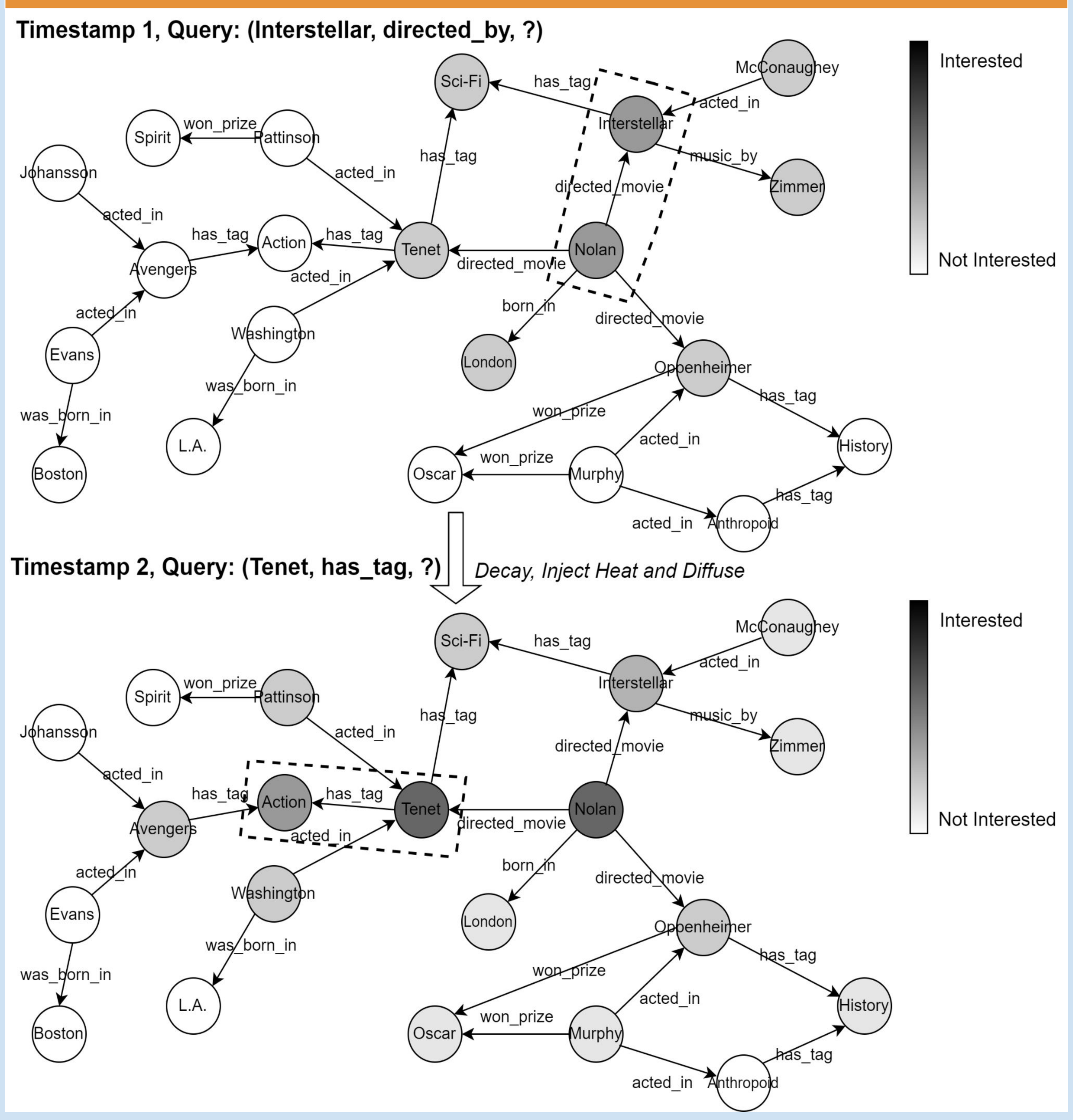
$$\boldsymbol{q}_{\text{total}}^{(T)} = \sum_{t=0}^{T} \gamma^{T-t}\boldsymbol{q}^{(t)} = \gamma \sum_{t=0}^{T-1} \gamma^{T-1-t}\boldsymbol{q}^{(t)} + \boldsymbol{q}^{(T)} = \gamma \boldsymbol{q}_{\text{total}}^{(T-1)} + \boldsymbol{q}^{(T)} \tag{16}$$

$$\begin{aligned}\boldsymbol{e}^{(T)} &= \sum_{l=0}^{d} \alpha^l \boldsymbol{A}^l \boldsymbol{q}_{\text{total}}^{(T)} = \sum_{l=0}^{d} \alpha^l \boldsymbol{A}^l \gamma \boldsymbol{q}_{\text{total}}^{(T-1)} + \sum_{l=0}^{d} \alpha^l \boldsymbol{A}^l \boldsymbol{q}^{(T)} \\ &= \gamma \boldsymbol{e}^{(T-1)} + \sum_{l=0}^{d} \alpha^l \boldsymbol{A}^l \boldsymbol{q}^{(T)}\end{aligned} \tag{17}$$

$$\boldsymbol{r}^{(T)} = \sum_{t=0}^{T} \gamma^{T-t}\boldsymbol{q}_r^{(t)} = \gamma \sum_{t=0}^{T-1} \gamma^{T-1-t}\boldsymbol{q}_r^{(t)} + \boldsymbol{q}_r^{(T)} = \gamma \boldsymbol{r}^{(T-1)} + \boldsymbol{q}_r^{(T)} \tag{18}$$



Adaptive KG summarization with incremental heat diffusion



Theoretical guarantees on both effectiveness and efficiency

Theorem 4.1 (Effectiveness of APEX^2). *Assume two areas (topics) $\mathcal{U}$ and $\mathcal{V}$ with connectivity c_u and c_v are sub-KGs of $\mathcal{G}$, and the user initially queries $\mathcal{U}$ for a times and starts to query $\mathcal{V}$, then APEX^2 takes $\log_\gamma \frac{1}{\frac{A}{B}(1-\gamma^a)+1}$ queries to adapt from $\mathcal{U}$ to $\mathcal{V}$, where $A = (\frac{1-(\alpha c_u)^{d+1}}{1-\alpha c_u})^{\frac{|\mathcal{E}_u|+2|\mathcal{T}_u|}{|\mathcal{E}_u|+3|\mathcal{T}_u|}}$ and $B = (\frac{1-(\alpha c_v)^{d+1}}{1-\alpha c_v})^{\frac{|\mathcal{E}_v|+2|\mathcal{T}_v|}{|\mathcal{E}_v|+2|\mathcal{T}_v|}}$.*

Theorem 4.2 (Time Complexity of APEX^2). *The incremental time complexity for APEX^2's adapting phase is $O(c\cdot|Q|^2\cdot\log_2(c\cdot|Q|))$, where Q is the query log decomposed into sub-queries (each query in Q has only one query entity, one query relation and one answer). $c = \frac{nnz(\sum_{l=0}^{d}(\alpha \mathbf{A})^l)}{|\mathcal{E}|}$, where nnz is the operator outputting the number of non-zero elements in a matrix, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix of $\mathcal{G}$, $\mathcal{E}$ is the set of entities of $\mathcal{G}$. (Proof in Appendix E.1)*

If set a threshold ϵ_{ths} to eliminate small-enough entries to 0, then after $log_\gamma \epsilon_{ths}$ timestamps, entries introduced by previous queries will be decayed to 0. In this case, the effective number of queries $|Q|$ above can be bounded by a constant $log_\gamma \epsilon_{ths}$. By this operation, **the incremental time complexity is further optimized to** $O(c\cdot\log_2(c))$, where $c = \frac{nnz(\sum_{l=0}^{d}(\alpha \mathbf{A})^l)}{|\mathcal{E}|}$ is the average number of neighbors within d-hops. In other words, the time complexity of APEX^2 updating is only related to the connectivity of KG.

Table 3: Efficiency Comparison (↓) (unit: seconds)

Methods	YAGO	DBPedia	MetaQA	Freebase
GLIMPSE	192.1±27.92	148.4±114.8	1.366±0.089	1.581±0.093
PageRank	22.81±259.7	2.615±0.136	0.032±0.003	0.144±0.011
ParallelPR	1.947±2.061	1.442±0.031	0.016±0.002	0.019±0.002
APEX^2	6.354±5.388	4.655±1.108	0.055±0.035	0.112±0.048
APEX-N^2	2.528±0.502	3.305±0.041	0.018±0.002	0.024±0.003

Demo: previously interested in "Stevan Riley", "The Disappearance of Haruhi Suzumiya", "LOL"; new query: (Chad Michael Murray, movie_to_actor, ?). MetaQA Movie dataset.