SummaryGPT: Leveraging ChatGPT for Summarizing Knowledge Graphs

Giannis Vassiliou[1], Nikolaos Papadakis[1], and Haridimos Kondylakis[1,2]

[1] ECE, Hellenic Mediterranean University
[2] FORTH-ICS
giannisvas@ics.forth.gr, npapadak@cs.hmu.gr, kondylak@ics.forth.gr

Abstract. Semantic summaries try to extract compact information from the original knowledge graph (KG) while reducing its size for various purposes such as query answering, indexing, or visualization. Although so far several techniques have been exploited for summarizing individual KGs, to the best of our knowledge, there is no approach summarizing the interests of the users in exploring those KGs, capturing also how these evolve. SummaryGPT fills this gap by enabling the exploration of users' interests as captured from their queries over time. For generating these summaries we first extract the nodes appearing in query logs, captured from a specific time period, and then we classify them into different categories in order to generate quotient summaries on top. For the classification, we explore both the KG type hierarchy (if existing) and also a large language model, i.e. ChatGPT. Exploring different time periods enables us to identify shifts in user interests and capture their evolution through time. In this demonstration we use WikiData KG in order to enable active exploration of the corresponding user interests, allowing end-users to visualize how these evolve over time.

1 Introduction

The explosion of the information now available in big KGs requires effective and efficient methods for quickly understanding their content, enabling the exploitation of the information they contain.

Semantic summaries have been proposed as methods for condensing information available in such KGs [4], [5]. According to our survey [1] a semantic summary is a piece of compact information, extracted from the original KG that can be used instead of the original graph for performing certain tasks more efficiently such as query answering, indexing or visualization.

Recent approaches (e.g., [7], [6]) have generated efficient summaries over large KGs such as DBpedia, WikiData, and Bio2RDF by capturing users' interests as they appear in their queries. The idea in those approaches is to identify the most frequently queried nodes in large query logs as the most important ones, and extract and link them exploiting paths from the original graph. Those summaries have nice properties regarding query answering, however, they fail to provide an overview of the overall graph as they only focus on the few most

queried nodes, pretty much ignoring the information contained in the remaining graph.

SummaryGPT, demonstrated in this paper, focuses on presenting an overview of the *entire KG for visualization* purposes by constructing *quotient* summaries. Quotient summaries establish a notion of "equivalence" for identifying node's representatives, and presenting those representatives instead of the original graph, enabling users to quickly understand the main groups of the underlying graph.

For the classification of the extracted nodes into the various groups we exploit and compare two methods: a) one based on the existing type hierarchy and b) one identifying (and labeling) generic categories generated by ChatGPT. ChatGPT is a large language model (LLM) that is trained on a substantial corpus of text data to perform natural language processing tasks such as text summarization and question answering. The ability of LLMs to pack large amounts of knowledge into their large parameter set has raised the possibility of using their implicit knowledge without domain training.

In this demonstration, we move beyond single quotient summaries, considering the fact that ontologies and KGs evolve over time [3], [2] along with user interests. As such instead of a single summary, we present a *series* of summaries, summarizing how users' interests evolve over time, allowing KG curators to appropriately visualize and explore shifts in users' interests. To the best of our knowledge, this is the first time that the notion of a *series of summaries* appears in the bibliography for enabling the exploration of KGs.

The system is available online[3] however we have disabled the configuration screen.

2 Architecture

Figure 1(left) depicts the high-level architecture of SummaryGPT. It consists of three layers, the GUI layer, the Service layer and the Data layer.

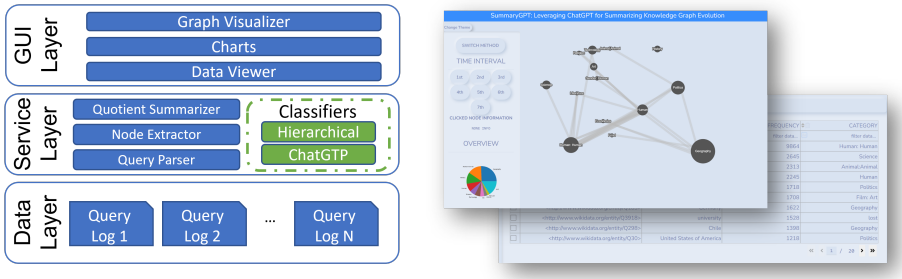

Fig. 1. SummaryGPT high-level architecture (left) and example screenshots of the system (right).

[3] `https://giannisergo.pythonanywhere.com/`

Using the GUI layer the user is able to upload query logs to be processed for a specific KG (SPARQL endpoint) and visualize the result summaries. The user is able to explore how those summaries evolve through time identifying the main categories targeting those queries. The more nodes in a specific category the bigger the node that appears in the result summary. The category nodes are linked based on the links in the processed query logs using the same philosophy (bigger lines as more queries include this link). Further, additional statistics are presented on the distribution of the queried nodes in the various groups. The main graph is interactive allowing users to click on the nodes and get further information on the nodes that appear in a specific group.

The service layer includes initially the query parser, extracting and storing the various nodes and edges in user queries, and also it implements the algorithm for constructing the quotient summaries. The algorithm summarizes a graph by assigning a representative to each class of equivalence of the nodes in the original graph.

Definition 1 (Quotient graph). *Let $G = (V, E)$ be a KG graph and $\equiv \subseteq V \times V$ be an* equivalence relation *over the nodes of V. The* quotient graph of G using $\equiv$*, denoted $G_{/\equiv}$, is a graph having:*

- *a node u_S for each set S of $\equiv$-equivalent V nodes;*
- *an edge (v_{S_1}, l, v_{S_2}) iff there exists an E edge (v_1, l, v_2) such that v_{S_1} (resp. v_{S_2}) represents the set of V nodes $\equiv$-equivalent to v_1 (resp. v_2).*

A particular feature of the quotient methods is that each graph node is represented by exactly one summary node, given that one node can only belong to one equivalence class. For determining the equivalence classes the service layer implements two classifiers.

The first classifier is based on ChatGPT, grouping query nodes into common-sense categories returned by the large language model. For each node (i.e., its label) appearing in user queries, we requested a one-word generic category that this node belongs to (i.e. "for each item in the list return a word that defines under which broad category it belongs"), and then we grouped the ones with the same description, building as such a classifier for the nodes appearing in each query log.

The second classifier queries the KG directly in order to retrieve the types of the queries' nodes, ignoring the ones that have no type, and then grouping the queried nodes based on their types.

Finally, the data layer includes the various query logs initially uploaded by the end user. SummaryGPT was implemented in Python Dash, whereas Cytoscape was used for the interactive graphs. ChatGPT API was used for querying ChatGPT and standard SPARQL was for querying the KGs.

3 Demonstration

To demonstrate the functionalities of SummaryGPT (an instance is shown in Figure 1(right)), we will use the Wikidata KG along with the query logs available

online[4]. The query logs are already split into seven batches, each one covering user queries for around a month, starting from June 2017. We used organic queries ranging between 200K and 800K queries per batch.

The demonstration will proceed in six phases:

1. **Configuration.** The summarization process starts by selecting the KG to be summarized along with the corresponding query logs. In the configuration menu, the user should provide a SPARQL endpoint for each KG and also a set of files, each one containing a distinct query log.
2. **Summary over a single query batch using existing hierarchy.** In this phase we will select a single query batch and we will demonstrate the quotient summary returned by the system. We will explain that the size of each group depends on the frequency of the nodes of the corresponding type that appears in the query batch and we will demonstrate the various statistics available.
3. **Summary over a single query batch using ChatGPT.** Then we will demonstrate the summary over the same query batch using the ChatGPT constructed classifier. The classifier automatically suggests groups and labels for these groups.
4. **Mini-Game.** In this phase we will discuss the quality of the summaries as perceived by conference participants. We will play a mini-game with the conference participants by examining a few nodes of the query logs and trying to assign those nodes in groups.
5. **Assessment.** Then we will discuss also that the results using the ChatGPT classifier seem to be more natural and very close to what a human classifier would do if s/he was asked to do the same process.
6. **Evolution of Summaries in Time.** Finally, we will demonstrate how those summaries evolve over time based on users' interests as they are captured by the query logs. We will identify that the 1st interval has a significant portion of nodes related to politics, mostly related to US elections, retaining traction for the following months, whereas after elections this portion is practically eliminated.

4 Conclusions

In this demonstration, we generate quotient summaries for KG based on user queries. We exploit query batches to explore how those summaries evolve through time making several interesting observations and comparing two classification methods for assigning queried entities into summary groups. To our knowledge, no other system today is available, enabling the rapid summarization of big KGs and no other system is able to explore how those summaries evolve over time.

[4] `https://iccl.inf.tu-dresden.de/web/Wikidata_SPARQL_Logs/en`

Acknowledgements Work reported in this paper has been partially supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Post-Doctoral Researchers" (iQARuS Project No 1147). Further is has been partially supported by the Horizon SafePolyMed EU project under grant agreement No 101057639.

References

1. Cebiric, S., Goasdoué, F., Kondylakis, H., Kotzinos, D., Manolescu, I., Troullinou, G., Zneika, M.: Summarizing semantic graphs: a survey. VLDB J. **28**(3), 295–327 (2019)
2. Kondylakis, H., Flouris, G., Plexousakis, D.: Ontology and schema evolution in data integration: Review and assessment. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) On the Move to Meaningful Internet Systems: OTM 2009, Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009, Vilamoura, Portugal, November 1-6, 2009, Proceedings, Part II. Lecture Notes in Computer Science, vol. 5871, pp. 932–947. Springer (2009). https://doi.org/10.1007/978-3-642-05151-7_14, https://doi.org/10.1007/978-3-642-05151-7_14
3. Kondylakis, H., Plexousakis, D.: Ontology evolution in data integration: Query rewriting to the rescue. In: Jeusfeld, M.A., Delcambre, L.M.L., Ling, T.W. (eds.) Conceptual Modeling - ER 2011, 30th International Conference, ER 2011, Brussels, Belgium, October 31 - November 3, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6998, pp. 393–401. Springer (2011). https://doi.org/10.1007/978-3-642-24606-7_29, https://doi.org/10.1007/978-3-642-24606-7_29
4. Trouli, G.E., Pappas, A., Troullinou, G., Koumakis, L., Papadakis, N., Kondylakis, H.: Summer: Structural summarization for RDF/S kgs. Algorithms **16**(1), 18 (2023). https://doi.org/10.3390/a16010018, https://doi.org/10.3390/a16010018
5. Troullinou, G., Kondylakis, H., Stefanidis, K., Plexousakis, D.: Exploring RDFS kbs using summaries. In: Vrandecic, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L., Simperl, E. (eds.) The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11136, pp. 268–284. Springer (2018). https://doi.org/10.1007/978-3-030-00671-6_16, https://doi.org/10.1007/978-3-030-00671-6_16
6. Vassiliou, G., Alevizakis, F., Papadakis, N., Kondylakis, H.: iSummary: Workload-based, personalized summaries for knowledge graphs. In: ESWC (2023)
7. Vassiliou, G., Troullinou, G., Papadakis, N., Kondylakis, H.: WBSum: Workload-based summaries for RDF/S kbs. In: SSDBM. pp. 248–252. ACM (2021)