HDAG-Explorer: A System for Hierarchical DAG Summarization and Exploration

Xuliang Zhu, Xin Huang, Jinbin Huang, Byron Choi, Jianliang Xu
Hong Kong Baptist University
Hong Kong, China
{csxlzhu, xinhuang, jbhuang, bchoi, xujl}@comp.hkbu.edu.hk

ABSTRACT

Hierarchical directed acyclic graph (HDAG) is an essential graph model to represent terminology relationships in a hierarchy, such as Disease Ontology, Gene Ontology, and Wikipedia. However, due to massive terminologies and complex structures in a HDAG, an end user might feel difficult to explore and summarize the whole graph, which is practically useful but less studied in the literature. In this demo, we develop an interactive system of HDAG-Explorer to help users summarize HDAG with highly important and diverse vertices. Our HDAG-Explorer system exhibits several useful features including summarized visualization, interactive exploration, and structural statistics report. All these features facilitate in-depth understanding of the HDAG data. We showcase the usability of the HDAG-Explorer through two real-world applications of summarized topic recommendation and visual data exploration.

PVLDB Reference Format:
Xuliang Zhu, Xin Huang, Jinbin Huang, Byron Choi, Jianliang Xu. HDAG-Explorer: A System for Hierarchical DAG Summarization and Exploration. *PVLDB*, 13(12): 2973-2976, 2020.
DOI: https://doi.org/10.14778/3415478.3415522

1. INTRODUCTION

Hierarchical directed acyclic graphs (HDAG) that represent hierarchical terminologies and their relation structure, are widely existed in real-world applications, such as Disease Ontology, Gene Ontology, Wikipedia, Image-net, Medical Entity Directory [8, 5]. Besides the topology structure in HDAG, vertices usually have *importance weights*. For example, Figure 1(a) shows a disease HDAG where each vertex represents a disease terminology and the associated weight indicates the disease occurrence frequency. A directed edge from one vertex to another vertex represents the concept instance relationship, e.g., "pneumonia" is a general concept of two instances "SARS" and "COVID-19". However, the massive size of terminologies and complex structure bring significant challenges to analyze and understand a HDAG dataset. It is essential to reduce a large HDAG dataset to a manageable size for visualization, which could give a direct and human-friendly data overview. More importantly, the summarized dataset within human cognitive capacity also helps guide users to further explore data. Figure 1(b) shows our expected solution of a disease HDAG in Figure 1(a). It provides a summarized view: "disease" is a general concept representing all vertices in general; "cancer" and "flu" represent two categories of important instances and "COVID-19" is the most important disease with the largest weight.

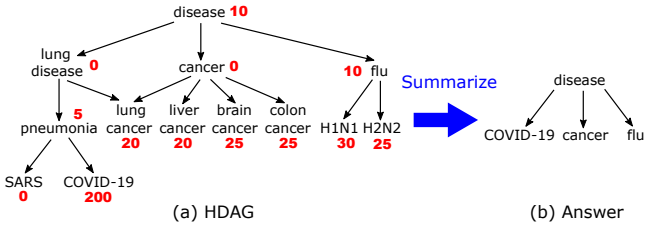


Figure 1: A disease HDAG summarization for $k = 4$.

In this demonstration, we present a novel graph analytic system, HDAG-Explorer, to summarize and analyze the HDAG datasets. The system addresses an important problem of graph summarization for large-scale hierarchical DAGs. Specifically, given a HDAG with node weights, the problem is finding a small set of k representative vertices to summarize the whole HDAG. HDAG-Explorer leverages the techniques of GVDO approach for tree summarization [5], which is a special instance of HDAG. We then extend the algorithms from trees to HDAGs and develop optimization techniques for fast summarization. An efficient approach is developed to find k representative vertices with quality guarantee. The summarization results of HDAG-Explorer are validated to capture the diversity coverage and structure correlation.

HDAG-Explorer has several useful features. First, it offers real-time analytic and interactive exploration on HDAGs. It implements a user-friendly visual interface to define the parameter k and a powerful query processing engine to efficiently generate k-summarized snippet in an online manner. Second, it features an informative output module where the visualization of summarized results is depicted in concise snippets to guide users for further visual exploration. It also offers multiple ways of interactive exploration and structural statistics reporting, which facilitate in-depth understanding of the data. Third, it allows users to upload their own datasets for analysis and exploration. HDAG-Explorer demonstrates two applications of summarized topic recommendation in SIGMOD'19 and visual data exploration on real-world datasets.

2. SYSTEM ARCHITECTURE

The system architecture of HDAG-Explorer is illustrated in Figure 2. It employs a client-server architecture and mainly consists of three components: input module, HDAG summarization, and out-

Proceedings of the VLDB Endowment, Vol. 13, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3415478.3415522

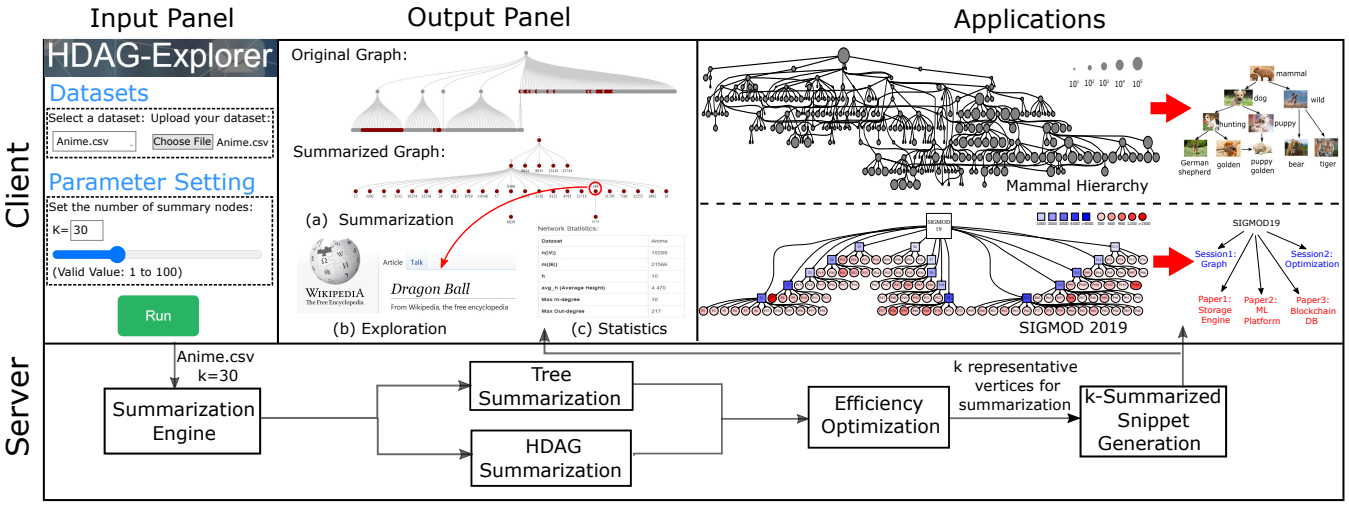


Figure 2: HDAG-Explorer **System Architecture**

put module. The input module specifics the explored HDAG data and parameter setting. The output module displays visualized results and structural statistics report, and also offers opportunities for HDAG exploration.

2.1 Input Module

The input module consists of two parts: data source selection and parameter k setting, which is shown in Figure 2. For the data source selection, HDAG-Explorer allows two input ways. One is to select an existing dataset for exploration in HDAG-Explorer. The other method is to upload user's own HDAG dataset. In addition, users need to assign the parameter value of k, which determines the size of summarized answer. Besides the text input, the input module enables users to drag the scroll bar to modify the value of k.

2.2 HDAG Summarization Algorithms

To provide an effective summarization of HDAGs, this module develops an algorithm GVDO for ontology-based graph visualization for summarized view [5]. Given an input parameter k, the summary result contains k representative vertices to give a hierarchical summarization. However, GVDO is developed to support the summarization on a hierarchical tree, which is one special instance of HDAG. We extend the techniques to general HDAGs. Specifically, to support graph summarization over trees and HDAGs, this module is composed of three components: tree summarization, HDAG summarization, and k-summarized snippet generation.

Tree summarization. Upon selecting an input number of k, the tree summarization works to select k representative vertices that diversely cover important vertices in a tree $T(V, E)$. The process of GVDO is briefly presented as follow. First, we formulate the summarized visualization of tree with ontology concepts as an optimized problem, i.e., finding a vertex set S with $|S| = k$ to maximize

$$\mathsf{g}(S) = \sum_{v \in V} \max_{u \in S \cap \mathsf{anc}(v)} (\mathsf{feq}(v) \cdot \mathsf{correlation}\langle u, v\rangle),$$

where $\mathsf{anc}(v)$ represents the set of ancestors for v, $\mathsf{feq}(v)$ is the weight of v, and $\mathsf{correlation}(u, v)$ is the depth difference between u and v in a tree T. The smaller their distance between u and v, the larger $\mathsf{correlation}(u, v)$. The objective is to find a set of representative S that achieves the largest score of summary impact.

To tackle it, the algorithm GVDO starts from an empty set of S and iteratively adds vertices with the largest marginal gain into the answer S until $|S| = k$. Due to the monotonity and submodularity of objective function $g(S)$, the greedy answer achieves the $(1 - 1/e) \approx 62\%$-approximation guarantee.

HDAG Summarization. This component performs the graph summarization of hierarchical DAGs, where involve massive terminologies and complex structures. The data structure of HDAG has a more powerful representation of complex relations than a tree, e.g., the relations of "disease $\rightarrow$ cancer $\rightarrow$ lung cancer" and "disease $\rightarrow$ lung disease $\rightarrow$ lung cancer" in Figure 1(a). Different from a single path from u to v in a tree, there exists multiple paths in a HDAG. Based on the shortest distance, we define the correlation as follows:

$$\mathsf{correlation}\langle u, v\rangle = \frac{1}{1 + \mathsf{dist}\langle u, v\rangle},$$

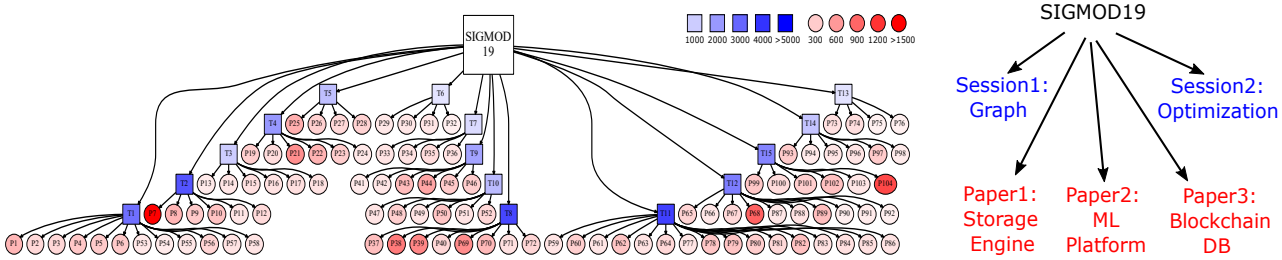
where $\mathsf{dist}\langle u, v\rangle$ is the shortest distance from u to v. The smaller their shortest distance between u and v, the larger $\mathsf{correlation}(u, v)$. Due to the complex structure, one vertex may have a large number of candidate representatives, which brings the significant computational difficulty. Actually, if we apply the objective function $g(S)$ on HDAGs, the graph summarization can be theoretically proven as a NP-hard problem.

THEOREM 1. *The problem of graph summarization on HDAGs by maximizing $g(S)$ with $|S| = k$ is NP-hard.*

Proof Sketch: We can reduce the well-known NP-complete 3-SAT problem to our problem of graph summarization on HDAGs.

To effectively resolve it, we extend the existing greedy algorithm in trees to HDAGs. The greedy algorithm runs in the time complexity of $O(nkm)$, where n, m are respectively the number of vertices and edges in a HDAG. Furthermore, we also develop a few advanced techniques of extracting tree from HDAGs and tree-based optimal summarization using dynamic programming, which are integrated to achieve high-quality summarized results in HDAGs rigorously.

k-Summarized Snippet Generation. Based on the obtained answer of S, it is ready to generate k-summarized snippet. We first create a virtual root r. We then start from each vertex $v \in S$ and add an edge path between v to the lowest ancestor in S; If such an ancestor does not exist, we add an edge path between v and the virtual root.



(a) SIGMOD'19 HDAG

(b) k-Summarized Snippet for $k = 6$

Figure 3: Summarized topic recommendation in SIGMOD'19.

2.3 Output Module

The output module displays an exploration panel to view and analyze HDAGs in a user-friendly manner. We illustrate HDAG-Explorer using a new HDAG dataset of Anime, which is a catalog ontology of Japanese animations crawled from the Wikipedia website using python library.[1] Anime has 15,134 vertices and 36,525 edges. Each vertex presents an animation webpage with a weight of monthly pageview. The larger the vertex weight, the more popular the animation. A directed edge is added from a general concept of animation to an instance. Specifically, output module consists of three components: summarized visualization, interactive exploration, and structural statistics report.

Summarized Visualization. To offer direct and human-friendly summarization for helping users understand the overview of query results, HDAG-Explorer applies graph visualization tool to depict the summarized results in Figure 2(a). The visualization uses the tool of d3.js. Besides our default algorithms of tree summarization and HDAG summarization, HDAG-Explorer also implements three competitive methods for graph summarization: FEQ [8], CAGG [8], and LASP [3]. HDAG-Explorer provides several interesting features to compare different methods in terms of summarization quality metrics, e.g., diverse coverage and importance correlation.

Interactive Exploration. HDAG-Explorer allows users to explore data in an interactive manner. If users click a label of vertex x, it will trigger the visualization that all descendants of this x will be explored and displayed. Moreover, the visualized graph allows to adjust the graph layout of HDAG. In addition, HDAG-Explorer embeds the profile information into vertices. Once clicking a vertex (e.g., 'Dragon Ball'), it triggers the our embedded API of the Wikipedia website and display the corresponding information about this vertex as shown in Figure 2(b).

Structural Statistics Report. This component reports the structural statistics in the analyzed HDAG data. Figure 2(c) shows the global graph statistics including the number of vertices and edges, the maximum length of longest path, the maximum in-degree and out-degree, and the average in-degree and out-degree.

In summary, the exploration panel facilitates users to explore results toward their search goals via various ways of summarized visualization, interactive exploration, and structural statistics report.

3. RELATED SYSTEMS AND NOVELTY

Graph summarization and visualization. A large body of studies on graph summarization and visualization has been proposed in the literature [1, 10, 3, 8, 5]. A semi-structured data summarization on RDF graphs is investigated in [1]. The object summarizes are identified for the graph representation of a relational database [3]. Graph summarization has also been studied on graph streams [4] and distributed computing setting [10]. In addition, a statistical method based aggregation for ontology graphical visualization is proposed [8]. Different from the above studies focusing on the general graph structure, we focus on optimizing k-summarization in hierarchical DAGs.

Graph analytic systems. Many graph analytic systems have been recently developed to handle different queries and scenarios in real applications [9, 2, 7, 7, 6]. AutoG [11] is an interactive system to alleviate the potentially painstaking task of graph query formulation. VizCS [7] presents an interactive graph system for online community search. MAGiQ [6] demonstrates a matrix algebra approach to answer RDF graph queries. All the above graph systems for various kinds of applications are different from our work that focuses on the HDAG data summarization and exploration.

4. DEMONSTRATION OVERVIEW

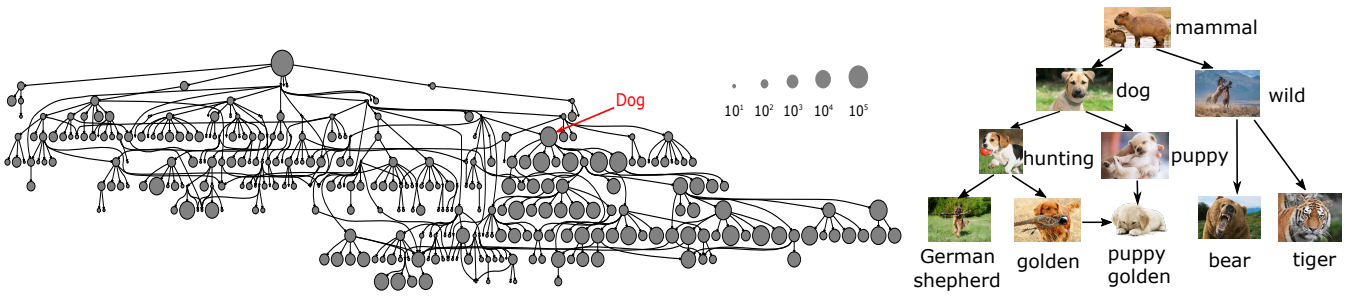
In this section, we introduce the demonstration of HDAG Explorer system. We implement the summarization algorithms of HDAG-Explorer in C++, use graph visualization tools (including D3.js, Inkscape, and Graphviz) and develop webpages to access server. HDAG-Explorer enables audiences to interact with our system and enjoy data exploration. Specifically, we demonstrate the system using two real-world applications and provide opportunities for users to interactively explore their own interested HDAGs.

Summarized topic recommendation in SIGMOD'19. We present an interesting application of summarized topic recommendation in SIGMOD'19 conference. The task is to recommend the top-k most attractive topics of SIGMOD'19. Each session has several papers under the same topic. Each paper has a cumulative number of downloads, which reflects its popularity. Each paper is represented by a secondary topic. Based on the Proceedings of SIGMOD'19,[2] we built a hierarchical DAG using 22 session topics and 104 secondary topics, where each secondary topic belongs to a session topic and is weighted by the number of downloads recorded in ACM Digital Library[3] (up to Dec 1, 2019) as shown in Figure 3(a). To evaluate the summarized quality, we conduct a usability test of topic recommendation that verifies users' preference. We asked 15 users, who published PVLDB/SIGMOD papers in recent three years, to recommend the top-k most attractive topics. We evaluate an accuracy rate of overlapping topics between users' choices and methods' selections. The survey results show that the summarization accuracy of HDAG-Explorer achieves the best performance on all different k, which is better than other competitive methods of FEQ [8], CAGG [8], and LASP [3]. This indicates an easy usability of HDAG-Explorer in the SIGMOD'19 attractive topic recommendation. Figure 3(b) shows our summarized results involving the attractive topics from two sessions of "Graphs" and

[1] https://en.wikipedia.org/wiki/Animation

[2] https://dblp.org/db/conf/sigmod/sigmod2019

[3] https://dl.acm.org/doi/proceedings/10.1145/3299869



(a) Mammal HDAG

(b) k-Summarized Snippet for $k = 10$

Figure 4: Graph visualization and summarized snippet on a mammal hierarchy in IMAGE.

"Optimization", and three secondary topics respectively on "Storage Engine", "Machine Learning Platform", and "Blockchain DB". Furthermore, besides SIGMOD'19, the contents of YouTube and Twitter are often classified/organized using a category hierarchy. The result of summarized topics should cover the diverse research areas and popular trends with high downloads. In addition, audiences are welcome to experience various summarization algorithms (e.g., HDAG-Explorer, FEQ, CAGG, and LASP) and different k for attractive topic recommendations.

Visual Data Exploration in ImageNet. The application of visual data exploration displays two functions of HDAG-Explorer: graph visualization and summarized snippet for data exploration. We use a subgraph of Image-net with 384 mammal catalogs called mammal HDAG. Each vertex represents an image catalog and its weight indicates the number of pictures associated with this catalog. Figure 4(a) shows the graph visualization of mammal HDAG. All the vertices are represented by circles. The vertex weight determines the circle size. The larger is the circle, the more important is the vertex. However, because of the large scale of the HDAG, it is unclear which are the important vertices in Figure 4(a). Even worse, it is difficult to summarize it and explore the important parts of HDAG. Figure 4(b) shows our solution of k-summarized snippet with $k = 10$. As we can see, it presents a clear summarization for the whole mammal HDAG. The "dog" catalog is the most important catalog as shown in the right area in Figure 4(a). Our results use "dog", "hunting", "puppy", "German shepherd", "golden", and "puppy golden" to for summarization. Moreover, for other areas, the "mammal", "wild", "bear", and "tiger" are selected as representatives to summarize them. The visualization of our summary results in Figure 4(b) reflects the diversity, small-scale, large coverage, and high correlation of our designed summary score function. In addition, the summarized results could offer a concise snippet for the such complex large-scale datasets in Figure 4(a), which makes users easily to further explore important components in the dataset.

Interactive Data Exploration on Your HDAGs. With the input module, audiences can upload other HDAG datasets and explore them instantly in our system. Except the default parameter setting of k, audiences can also easily drag the scroll bar to define different values of k. HDAG-Explorer then produces the corresponding k-summarization snippet. Since it is difficult to directly overview the complete structure of a dataset with more than hundreds of vertices, the k-summarization result displays a concise snippet. To ensure the complexity of displayed HDAG within human cognitive capacity, audiences can easily use a small number k to control the size of visualized snippet. Moreover, audiences can explore and interact with visualized data in the output panel. Specifically, audiences can further click the nodes to gain more hidden knowledge and understand the real-world background information behind them. In addition, the structural statistics report is also displayed for audiences to analyze the structural information of HDAGs in Figure 2. Last but not least, HDAG-Explorer has been implemented with several different summarization algorithms for competitiveness evaluations. Visualizing different summarization results for audiences is greatly helpful to understand the different approaches vividly and directly.

Acknowledgement. This paper is supported by NSFC 61702435, HK RGC GRF 12200917, 12201518, 12232716, HK RGC CRF C6030-18G, and Guangdong Basic and Applied Basic Research Foundation (Project No. 2019B1515130001). Xin Huang is the corresponding author.

5. REFERENCES

[1] Š. Čebirić, F. Goasdoué, and I. Manolescu. Query-oriented summarization of rdf graphs. *PVLDB*, 8(12):2012–2015, 2015.

[2] Y. Diao, P. Guzewicz, I. Manolescu, and M. Mazuran. Spade: A modular framework for analytical exploration of RDF graphs. *PVLDB*, 12(12):1926–1929, 2019.

[3] G. Fakas, Z. Cai, and N. Mamoulis. Diverse and proportional size-l object summaries for keyword search. In *SIGMOD*, pages 363–375, 2015.

[4] X. Gou, L. Zou, C. Zhao, and T. Yang. Fast and accurate graph stream summarization. In *ICDE*, pages 1118–1129, 2019.

[5] X. Huang, B. Choi, J. Xu, W. K. Cheung, Y. Zhang, and J. Liu. Ontology-based graph visualization for summarized view. In *CIKM*, pages 2115–2118, 2017.

[6] F. Jamour, I. Abdelaziz, and P. Kalnis. A demonstration of magiq: matrix algebra approach for solving rdf graph queries. *PVLDB*, 11(12):1978–1981, 2018.

[7] Y. Jiang, X. Huang, H. Cheng, and J. X. Yu. Vizcs: Online searching and visualizing communities in dynamic graphs. In *ICDE*, pages 1585–1588, 2018.

[8] X. Jing and J. J. Cimino. Graphical methods for reducing, visualizing and analyzing large data sets using hierarchical terminologies. In *AMIA*, volume 2011, page 635, 2011.

[9] Z. Li, X. Chen, X. Pan, P. Zou, Y. Li, and G. Yu. Shoal: large-scale hierarchical taxonomy via graph-based query coalition in e-commerce. *PVLDB*, 12(12):1858–1861, 2019.

[10] X. Liu, Y. Tian, Q. He, W.-C. Lee, and J. McPherson. Distributed graph summarization. In *CIKM*, pages 799–808, 2014.

[11] P. Yi, B. Choi, S. S. Bhowmick, and J. Xu. Autog: a visual query autocompletion framework for graph databases. *PVLDB*, 9(13):1505–1508, 2016.