VISION-KG: Topic-centric Visualization System for Summarizing Knowledge Graph

Jiaqi Wei
Peking University
Beijing, China
jiaqi97@pku.edu.cn

Shuo Han
Peking University
Beijing, China
hanshuo@pku.edu.cn

Lei Zou
Peking University
Beijing, China
zoulei@pku.edu.cn

ABSTRACT

Large scale knowledge graph (KG) has attracted wide attentions in both academia and industry recently. However, due to the complexity of SPARQL syntax and massive volume of real KG, it remains difficult for ordinary users to access KG. In this demo, we present VISION-KG, a topic-centric visualization system to help users navigate KG easily via *entity summarization* and *entity clustering*. Given a query entity v_0, VISION-KG summarizes the induced subgraph of v_0's neighbor nodes via our proposed facts ranking method that measures *importance*, *relatedness* and *diversity*. Moreover, to achieve conciseness, we split the *summarized graph* into several *topic-centric summarized subgraph* according to semantic and structural similarities among entities. We will demonstrate how VISION-KG provides a user-friendly visualization interface for navigating KG.

ACM Reference Format:
Jiaqi Wei, Shuo Han, and Lei Zou. 2020. VISION-KG: Topic-centric Visualization System for Summarizing Knowledge Graph. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3336191.3371863

1 INTRODUCTION

RDF data is enjoying an increasing popularity since the launching of knowledge graph (KG). As a de facto standard of KG, RDF represents knowledge as a collection of *triples (facts)*, denoted as ⟨subject, predicate, object⟩. However, the explosion in the volume of RDF data has brought many obstacles to data querying and exploring tasks. Two major challenging issues are:

- Although SPARQL is a standard protocol to access RDF data, it is impractical for ordinary users to write SPARQL due to its complicated syntax and the lack of schema knowledge.
- Browsing or visualizing RDF datasets suffers from poor readability because of the massive volume of data. For example, the latest English version of DBpedia [6] describes over 6 million entities with 1.3 billion RDF triples, averaging about 200 triples per entity. It is difficult for users to comprehend the excessive amount of data.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371863

To address the challenges above, we propose a topic-centric **VIS**ualizat**ION** system for **K**nowledge **G**raph (**VISION-KG**) to help users browse RDF repositories easily via *entity summarization* [2] and *entity clustering*. VISION-KG provides an interactive GUI, which makes the system more accessible to users. Firstly, users can input some keywords of interest, then relevant entities will be listed in our system. Once users want to inspect some entity to acquire more information, a *summarized graph* centered at the *query entity* will be demonstrated, which offers a concise overview of the entity. Moreover, entities from the *summarized graph* will be split into several clusters according to semantic and structural similarities among them. Entities in the same cluster are assumed to be relevant to one specific topic, so as to connect into a more concise *topic-centric summarized subgraph*. The following example illustrates the core functionality of VISION-KG.

Example 1.1. Figure 1 is a running example of our system. Suppose that one picks *Taylor_Swift* who is a famous American singer and actress. Our system firstly visualizes a summarized graph centered at *Taylor_Swift* by extracting a concise subset of the facts of *Taylor_Swift* within 2 hops, as shown in Figure 1-1. Diversified aspects of facts are demonstrated to the user.

Meanwhile, several topic-centric summarized subgraphs are provided, as shown in Subgraph I - IV, which can help users obtain desired information more intuitively and effectively. For example, if the user is interested in works of *Taylor_Swift*, she can focus on Subgraph I whose topic label is "Works". It is clear that Subgraph I consists of musical works, films and awards related to *Taylor_Swift*.

Related Work. Entity summarization has gained particular attention over the past years. Cheng et al. [2] introduced the problem of entity summarization and proposed RELIN which adopts the modified PageRank algorithm to extract both related and informative facts. Similarly, SUMMARUM [9] and LinkSUM [8] are also based on PageRank algorithm to generate ranking scores. Sydow et al. [7] illustrated the importance of diversity to entity summarization. FACES [3] partitions facts into diverse semantic groups using a conceptual clustering algorithm and then ranks them inside each group to generate diverse entity summary. DynES [5] considers the relevance between facts and the query context. REMES [4] focuses on summarizing a collection of entities and tries to make a connection between these entities.

However, most systems mentioned above focus on summarizing one-hop facts except for DiverSUM [7] and REMES. To provide an informative and diverse summary, multi-hop facts are essential. Continuing with Example 1.1, the user can obtain the answer from the multi-hop summarized graph intuitively if she wants to know who was born in the same place as *Taylor_Swift*. However, the

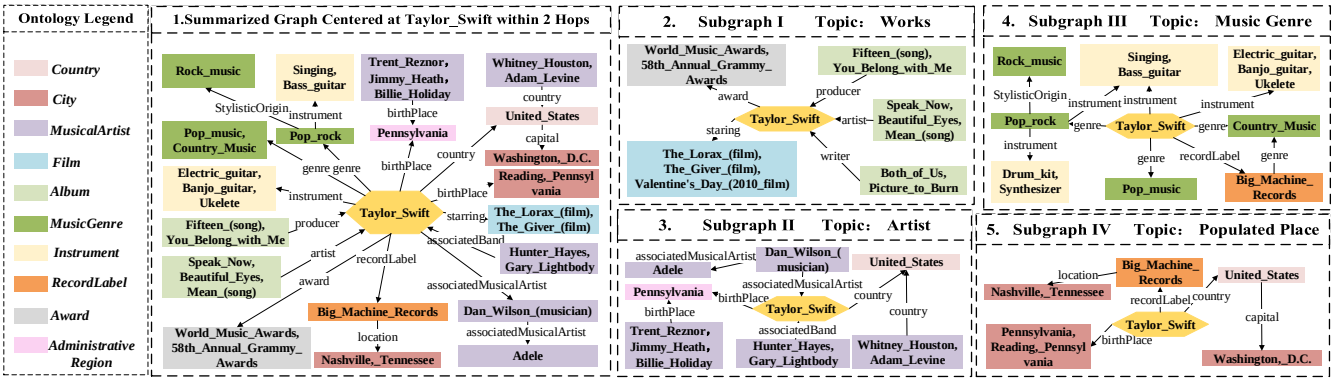

Figure 1: A running example of generating summarized graph and topic-centric summarized subgraphs. The background color of each rectangle specifies the category of each entity, which is in accordance with the Ontology Legend on the left panel. To prevent the canvas from overcrowding, entities that share common ingoing and outgoing edges are grouped into one rectangle.

main objective of REMES is related entity recommendation, which is different from ours. DiverSUM [7] is the closest system to the task we address in this paper, but it neglects semantic relatedness between the query entity and facts. Notice that in most cases, the subgraph may contain up to thousands of two-hop facts. However, only a few two-hop facts are relevant to the query entity. In order to filter out irrelevant facts, we need to consider not only importance and diversity but also relatedness between entities. For example, both *Whitney_Houston* (an American singer) and *Kobe_Bryant* (an American basketball player) are two-hop entities of *Taylor_Swift* because the three of them are Americans, and both *Singing* and *DXSP* (a radio station) are two-hop entities of *Taylor_Swift* because the three of them link with *Pop_rock*. For both of the examples above, it is obvious that the former entities (i.e., *Whitney_Houston* and *Singing*) are more expected to be selected in our summary because they are more relevant to the query entity.

Another innovation of our system is the topic-centric summarized subgraph that contains a group of similar (relevant to one specific topic) entities. However, an important question is: how to define a reasonable similarity measurement between entities? A straightforward solution is to cluster these entities utilizing the ontology information. Ontology is usually represented as a category hierarchy tree which specifies the schema of the underlying RDF data. Ideally, ontology provides adequate guidance for entity clustering. However, ontology is often incomplete or even missed in real RDF datasets. Furthermore, a topic-centric subgraph may contains several category of entities. Taking Subgraph III in Figure 1 as an example, most entities in Subgraph III are "music genres" and "instruments". The relevance among these categories is weak according to the category hierarchy tree. Actually, these entities are dense-connected and closely related. Recent research efforts on *knowledge graph embedding* gain a big success on representing KG in vector space, in which the inherent semantic and structural information of the KG is preserved. Distance between embeddings provides a quantitative measurement of similarity. Hence, we adopt embedding distance as the guidance of the clustering process.

Contributions. (i) We propose an approach to generate multi-hop entity summary considering three dimensions of ranking measurements: importance, relatedness, and diversity. And we propose an approximate algorithm using greedy strategy with a bounded approximation factor to tackle with the efficiency challenge. (ii) We introduce the notion of topic-centric summarized subgraph and propose a clustering algorithm to generate such summary. (iii) We present a visualization system with good readability and interactivity based on our proposed approach.

2 SYSTEM ARCHITECTURE

Figure 2 illustrates the architecture of VISION-KG. The system consists of four components: storage module, summarization module, clustering module, and GUI. The design and functionality of the former three components will be briefly reviewed in the following subsections and the details of user interaction mechanisms will be demonstrated in Section 3.

2.1 Storage Module

This module consists of two parts: *RDF data storage* and *embedding storage*. We use the latest English version of DBpedia as the underlying RDF dataset for demonstration. RDF data is stored in graph format by compressed adjacency-lists. Embedding vectors, which are stored in disk with indices, are used for extracting relatedness feature and diversity feature as ranking measurements during the summarization process, and calculating entity similarity to construct approximate k-NN graph during the clustering process.

Training KG Embedding Offline: *Structural embedding* of the entity is learnt from triples and *semantic embedding* is generated from the description texts with SSP [11]. SSP is a knowledge graph embedding method which jointly learns from the symbolic triples and textual descriptions by performing the embedding process in a semantic subspace. Choosing SSP is based on the following three reasons: (i) Distance between semantic embeddings generated from entity descriptions is a good measurement of relatedness between entities during summarization process. As shown in the example of the *Entity Description* in Figure 2, the more related two entities are, the more keywords they have in common in their descriptions. (ii) The core of SSP is that the semantically relevant entities are projected onto a consistent hyperplane approximately, which coincides with our objective of topic-centric clustering. (iii) The computation complexity of SSP is comparable to TransE which is the most efficient knowledge graph embedding method.

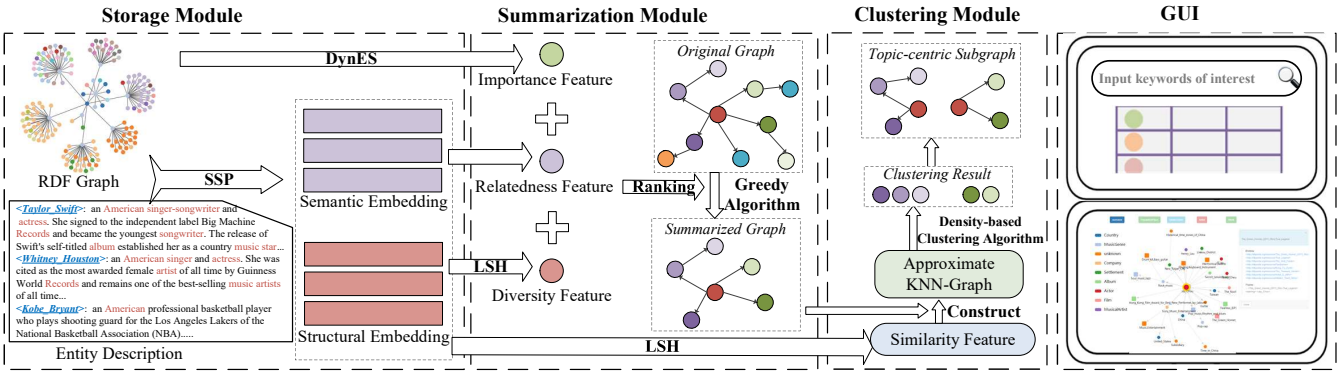

Figure 2: Overview of system architecture.

2.2 Summarization Module

In this component, we generate a summarized graph centered at query entity v_0 by extracting a small subset of relevant facts considering the following three features:

- **Importance:** The importance of fact f for the query entity, denoted by $I(f)$, reflects how informative the predicate-object pair is and how popular the object is. To compute the importance scores, we utilize the importance features defined in DynES [5]. DynES considers both frequency and specificity, which is similar to tf-idf.
- **Relatedness:** The relatedness between the query entity v_0 and the object entity (denoted as f_O) of fact f, indicated by $R(v_0, f_O)$, reflects the semantic relevance of f to v_0. The reciprocal of semantic embedding distance between query entity v_0 and entity f_O is applied to measure this feature.
- **Diversity:** The diversity feature can be quantized by pairwise embedding distances between selected facts. For facts f_1 and f_2, $D(f_1, f_2)$ denotes the structural embedding distance between object entities of f_1 and f_2.

By combining these features above, we define the objective score function as:

$$S(X) = \alpha \cdot \sum_{f \in X} I(f) + \beta \cdot \sum_{f \in X} R(v_0, f_O) + \gamma \cdot \sum_{f_1, f_2 \in X} D(f_1, f_2) \quad (1)$$

Specifically, given a set of v_0's facts within h hops $X_{v_0,h}$ and the size of summarized graph k, extract a subset X where $X \subseteq X_{v_0,h}$ and $|X| \leq k$, to generate a summarized graph by maximizing $S(X)$.

Notice that calculating exact distance between each pair of vectors leads to unaffordable computing cost for the online processing. To tackle with the efficiency challenge, we adopt LSH [1] techniques to estimate vector distances.

LSH Framework: Locality-sensitive hashing (LSH) is a lightweight indexing approach for approximate nearest neighbor search, and has shown significant performance in dealing with the curse of dimensionality. We briefly describe how to build a LSH family that maps embedding vectors onto a set of integers: Each vector $\boldsymbol{v}$ is projected to a real line $\boldsymbol{a}$ by computing the dot product $\boldsymbol{a} \cdot \boldsymbol{v}$, then we chop the line into equal-sized segments of appropriate size w and assign hash values to vectors based on which segment they project onto. Intuitively, if two vectors are close enough, they should collide (project onto the same segment) with high probability. Formally, the hash function is $h(\boldsymbol{v}) = \lfloor \frac{\boldsymbol{a} \cdot \boldsymbol{v} + b}{w} \rfloor$, where $\boldsymbol{a}$ is a d-dimensional vector, and b is a random offset in the range of $[0, w]$.

With a LSH family $\mathcal{H} = \{h_i, ..., h_n\}$, the problem of facts selection can be formulated as maximizing the following score function:

$$S(X) = \alpha \cdot \sum_{f \in X} I(f) + \beta \cdot \sum_{f \in X} R(v_0, f_O) + \gamma \cdot \sum_{i=1}^{n} |V_i(X)| \quad (2)$$

where $V_i(X)$ denotes the set of different hashing values of all the selected facts in X by the LSH function $h_i(\cdot)$, namely $V_i(X) = \bigcup_{f \in X} h_i(f_O)$ (f_O indicates the object entity of fact f). The larger cardinality of $V_i(X)$ means that hashing values of X under h_i diverge from each other, which indicates larger sum of pairwise vector distances.

We prove that the facts selection problem is NP complete. However, a greedy strategy can achieve $(1 - 1/e)$ approximation ratio because of submodularity. We greedily select the fact with the currently largest incremental on $S(X)$, until the cardinality of X is up to k. This selection process can be implemented in $O((k + n) \cdot |X_{v_0,h}|)$ time. To further improve $S(X)$, we iteratively adjust the candidate subset X by replacing from the set of unselected facts. Since the score function $S(X)$ is the linear combination of three features, we perform an extensive evaluation to analyze the performance of each individual feature and determine the distribution of these features. The detailed algorithm of fact selection and the evaluation result is given in the full research paper of this work.

2.3 Clustering Module

In this module, the summarized graph is split into several topic-centric summarized subgraph via entity clustering. We adopt embedding distance as the guidance of the clustering process. Existing clustering techniques are dedicated to offline analysis tasks, which are not capable of real-time requirement. To tackle with the efficiency challenge, we present a lightweight clustering algorithm based on approximate k-NN graph [10] and LSH techniques.

Constructing Approximate k-NN Graph with LSH: A k-nearest neighbor (k-NN) graph is a directed graph where each node is connected to its top-k nearest neighbor nodes. If we measure node similarity by Euclidean distance of embedding vectors, the time complexity of exact k-NN graph construction either grows exponentially wrt. the dimensionality, or grows super-linearly wrt. the number of nodes. Instead, we adopt the (K, L)-parameterized LSH framework to find approximate k-nearest neighbors. In brief, we maintain L hash tables. For each of them, we amplify the gap between similar vectors with high collision probability and the dissimilar vectors with low probability, by concatenating K hash values. The idea of constructing approximate k-NN graph is to iteratively

Figure 3: Screenshots of VISION-KG: A: Entity Search. B: Summarized Graph. C: Topic-centric Summarized Subgraph.

query each node in the L tables. From the property of LSH, we know that the nodes that are projected onto the same bucket with v_i have higher probabilities to be v_i's near neighbors, and the higher collision frequency in the L tables indicates the smaller distance between them. Thus we collect the nodes in the buckets addressed by the hash values of v_i, and estimate vector distances by collision frequencies to find approximate top-k nearest neighbors.

Clustering Algorithm: The quality of a cluster c_i is measured by $density(c_i) = \frac{\sigma(c_i)}{size(c_i) \cdot MIN(k, size(c_i)-1)}$,where $\sigma(c_i)$ denotes the weight sum of edges that both exist in the k-NN graph and connect nodes in c_i. Initially each entity forms a singleton cluster. Then the algorithm iteratively merges intermediate clusters in a bottom-up manner, until the number of clusters and the merging cost reach the thresholds.

Subgraph Generation: For each cluster, we generate a connected subgraph containing the query entity and these entities in the cluster by adding minimum edges and entities of the original summarized graph. Intuitively, we adopt the fine-grained ontology label that can cover most of entities in this cluster, as the topic label of the corresponding topic-centric summarized subgraph. The detailed algorithm description is given in our research paper.

3 DEMONSTRATION

In this section, We provide a scenario to elaborate how to use VISION-KG and how it helps to obtain desired information.

Operation Guideline: Consider the scenario that a user wants to know something about Albert Einstein. First, she starts from the entity search interface by inputing some keywords as shown in Figure 3-A, then relevant entities and their textual descriptions are listed in the table below. By clicking the corresponding circle button, the summarized graph centered at *Albert_Einstein* is shown in the center panel of the entity visualization page, as illustrated in Figure 3-B. Specifically, these circles represent singleton nodes, and each of these rectangles stands for a group of entities which share common ingoing and outgoing edges. The background color of each node specifies the category of the corresponding entity, which is in accordance with the ontology legend on the left panel. Furthermore, most standard graph interactions are supported, including zoom&pan, node drag&drop to fine-tune the layout, node hovering to show labels of adjacent edges and highlight adjacent nodes, and node click-selection to show the info box floating on the right panel. These labeled buttons in the top panel link to topic-centric summarized subgraphs. For example, if she clicks the button with label "Scientist", a subgraph which contains scientists related to *Albert_Einstein* will be shown, as illustrated in Figure 3-C.

Example Analysis: Continuing with the example of *Albert_Einstein*, the summarized graph generated by our system well satisfies those three features defined in Section 2.2. Most facts in the summarized graph are commonly known or unique to the query entity, which ensures *importance*. Different aspects of information concerning *Albert_Einstein* is provided, such that *diversity* is also guaranteed . Furthermore, most two-hop facts are related to "Scientist" or "Physics", which ensures *relatedness*. Although the summarized graph contains only 60 nodes, the canvas is overcrowded. In comparison, the topic-centric summarized subgraphs provide better readability because each of them contains concise information focused on one specific topic. It is worth mentioning that our clustering algorithm can generate fine-grained entity clusters. For example, scientists related to *Albert_Einstein* and family members of *Albert_Einstein* are grouped into two different subgraphs with label "Scientist" and label "Person" respectively.

A demonstration video is available at the following address: https://youtu.be/rvfbYMlvBWw.

Acknowledgments. This work is supported by The National Key Research and Development Program of China under grant 2018YFB 1003504 and NSFC under grant 61932001, 61961130390, 61622201 and 61532010. This work is also supported by Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*. ACM, 253–262.
[2] Cheng Gong, Thanh Tran, and Yuzhong Qu. 2011. RELIN: Relatedness and Informativeness-Based Centrality for Entity Summarization. *Lecture Notes in Computer Science* 7031 (2011), 114–129.
[3] Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit Sheth. 2015. FACES: Diversity-Aware Entity Summarization using Incremental Hierarchical Conceptual Clustering. In *Twenty-ninth Aaai Conference on Artificial Intelligence*.
[4] K Gunaratna, A. H. Yazdavar, K Thirunarayan, A Sheth, and G. Cheng. 2017. Relatedness-based Multi-Entity Summarization. In *International Joint Conference on Artificial Intelligence*.
[5] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2017. Dynamic factual summaries for entity cards. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 773–782.
[6] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.
[7] Marcin Sydow, Mariusz Pikuła, and Ralf Schenkel. 2013. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *Journal of Intelligent Information Systems* 41, 2 (2013), 109–149.
[8] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. 2016. LinkSUM: Using Link Analysis to Summarize Entity Data.. In *International Conference on Web Engineering*.
[9] Andreas Thalhammer and Achim Rettinger. 2014. Browsing DBpedia Entities with Summaries. In *Semantic Web: Eswc Satellite Events*.
[10] Jing Wang, Jingdong Wang, Gang Zeng, Zhuowen Tu, Rui Gan, and Shipeng Li. 2012. Scalable k-nn graph construction for visual descriptors. In *CVPR*. IEEE, 1106–1113.
[11] Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu. 2017. SSP: semantic space projection for knowledge graph embedding with text descriptions. In *Thirty-First AAAI Conference on Artificial Intelligence*.