A Survey on Extractive Knowledge Graph Summarization: Applications, Approaches, Evaluation, and Future Directions

Xiaxia Wang[2], **Gong Cheng**[1*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Department of Computer Science, University of Oxford, UK
xiaxia.wang@cs.ox.ac.uk, gcheng@nju.edu.cn

Abstract

With the continuous growth of large Knowledge Graphs (KGs), extractive KG summarization becomes a trending task. Aiming at distilling a compact subgraph with condensed information, it facilitates various downstream KG-based tasks. In this survey paper, we are among the first to provide a systematic overview of its applications and define a taxonomy for existing methods from its interdisciplinary studies. Future directions are also laid out based on our extensive and comparative review.

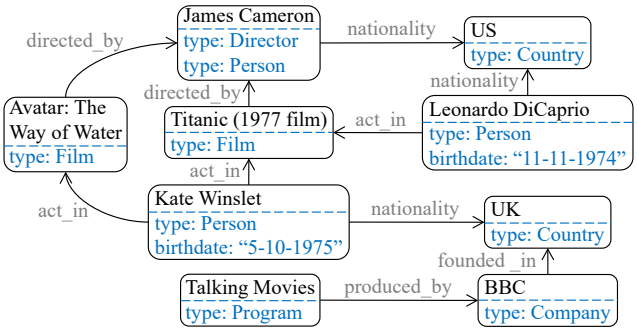

Figure 1: A small KG as our running example.

1 Introduction

Knowledge Graph (KG) has been a popular knowledge representation [Hogan *et al.*, 2022]. In this graph data model, as illustrated in Figure 1, nodes represent entities of interest, which are often annotated with types and attributes, and edges represent typed relations between entities; attributes and relations are collectively called properties. KGs have been widely employed in various AI systems and application fields [Peng *et al.*, 2023], such as question answering [Yang *et al.*, 2017] and machine translation [Zhao *et al.*, 2020]. With the emergence of large language models (LLMs), KGs have exhibited increased importance as they can also be used for training and augmenting LLMs [Pan *et al.*, 2023].

Motivation of KG Summarization. Given numerous KGs available on the Web, the effective discovery and selection of suitable KGs for reuse has become a challenge for humans since KGs are often large and cover a variety of topics. Such magnitude and diversity also hinder machines from efficient processing in AI systems. To tackle this problem, a straightforward idea is to generate a succinct description for a given KG to indicate its coverage or reflect its main content. Such a size-reduced representation containing concise, meaningful, and faithfully represented information from the original KG is referred to as a *summary* of the KG [Cebiric *et al.*, 2019].

Scope of the Survey. KG summarization is distinguished from summarizing conventional homogeneous graphs such as the Web graph and social networks [Liu *et al.*, 2018] by the different types of entities and relations in a KG. Accordingly, grouping-based methods for KG summarization merge entities having similar types and properties into super-nodes connected by super-edges representing relations between entities in the super-nodes. Such methods have been covered in previous surveys [Cebiric *et al.*, 2019; Scherp *et al.*, 2023]. In parallel, extractive methods select an optimal subgraph from a given KG as its *extractive summary*. This type of KG summary, a.k.a. KG snippet, faithfully exemplifies the content of the original KG, and can be effortlessly comprehended by humans and directly processed by machines in the same manner as the original KG. For these advantages, extractive KG summarization has attracted considerable research attention from interdisciplinary areas related to AI and big data, and has been used in KG profiling [Ellefi *et al.*, 2018], query optimization [Heling and Acosta, 2023], search [Chapman *et al.*, 2020], exploration [Lissandrini *et al.*, 2022], and many other KG-driven applications. However, to the best of our knowledge, *this is the first survey paper on extractive KG summarization*, providing a comprehensive overview of its applications, approaches, evaluation, and future research directions.

Outline. We broadly divide extractive KG summaries into two categories: *static summaries* are context-independent and always the same for a given KG; *dynamic summaries* are customized according to the user's information needs. They are generated for different applications (Section 3), by different methods (Sections 4 and 5), and are evaluated in different ways (Section 6). We present a taxonomy of existing methods in Figure 2. To demarcate this survey, we firstly distinguish extractive KG summarization from related tasks (Section 2).

*Corresponding author

2 Related Work

Our work differs from the following surveys on research problems that are related to extractive KG summarization.

Liu *et al.* [2018] reviewed a wide range of methods for **graph summarization**, but they focused mostly on homogeneous graphs rather than KGs where nodes and edges may be labeled with different types. For labeled graphs, their survey covered **grouping-based** but not extractive methods, which represent two distinct paradigms of summarization. Previous surveys on KG summarization [Cebiric *et al.*, 2019; Scherp *et al.*, 2023] also gave primary attention to grouping-based methods, based on the concept of quotients in particular. Since a grouping-based summary can be viewed as an implicit schema discovered from the KG, the task is also referred to as **schema discovery** [Kellou-Menouer *et al.*, 2022].

Entity summarization selects a subset of properties for each given entity in a KG [Liu *et al.*, 2021]. An entity summary can be viewed as a special type of extractive KG summary that is restricted to the neighborhood of a particular node. Such locality distinguishes entity summarization from the methods for extractive KG summarization covered in this survey which globally take the entire KG into consideration.

Ontology summarization extracts core concepts or a sub-ontology from a given ontology. Existing methods mainly represent an ontology as a graph to be summarized [Pouriyeh *et al.*, 2019]. Some of these graph representations resemble a KG but are at the schema level where nodes are classes instead of entities. Accordingly, the optimization objective of ontology summarization differs significantly from those of extractive KG summarization reviewed in this survey.

3 Applications of Extractive KG Summaries

Extractive KG summarization has proved to be useful for a variety of KG-based applications. Static and dynamic KG summaries are needed for supporting different tasks.

3.1 Applications of Static KG Summaries

Static summaries are context-independent and are extracted to capture the intrinsic characteristics of a KG, e.g., to reflect its main content, or to indicate its coverage. They have found application in KG profiling and query optimization.

KG Profiling

A KG profile is a formal representation of a set of features of a given KG, which is often used to aid KG discovery, recommendation, and comparison [Ellefi *et al.*, 2018]. An extractive KG summary that contains representative entities in the original KG is one such feature understood as a sample that accurately portrays the whole KG. It has been used in KG search engines to exemplify the content of a KG getting clicked on in the search results [Wang *et al.*, 2022], aiding the user in quickly filtering relevant KGs without expensively accessing the original, potentially large KGs.

KG Query Optimization

An extractive KG summary can be regarded as a view of a given KG and queried in the same manner [Fan *et al.*, 2014; Wang, 2017; Li *et al.*, 2016]. A common application of such views is federated KG query optimization [Montoya *et al.*, 2017] where multiple KGs are jointly queried, and the performance relies on the efficiency of the query plan, e.g., the estimation of join cardinalities for sub-queries. Extractive KG summaries have been used to compute accurate estimations of this kind using less time than the complete statistics derived from the original large KGs [Heling and Acosta, 2023].

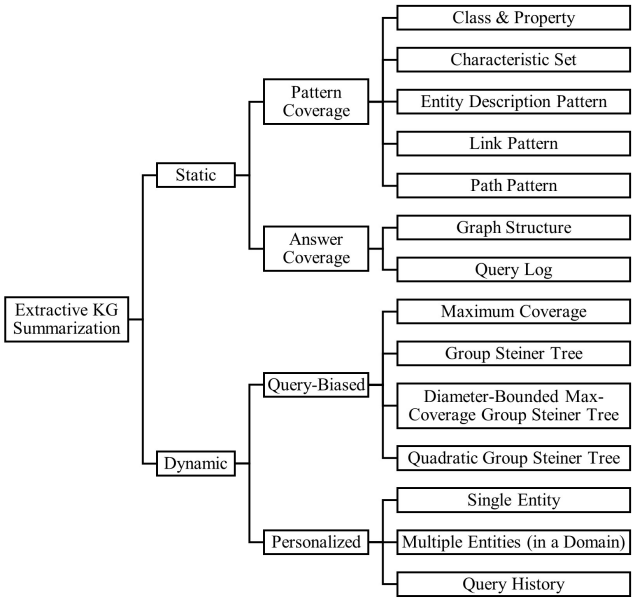

Figure 2: A taxonomy of methods for extractive KG summarization.

3.2 Applications of Dynamic KG Summaries

Dynamic summaries are customized and are extracted to satisfy each user's individual information needs, e.g., to show the relevance of a KG to a user's query, or to tailor the content of a KG to users' specific interests.

KG Search

KG search engines have become an important tool for finding open KGs for reuse [Chapman *et al.*, 2020]. Whereas early implementations rely on metadata descriptions to discover KGs and assist user judgment of their relevance, metadata has proved to be insufficient due to its quality and availability [Koesten *et al.*, 2017]. Complementary to metadata, various types of KG summaries have been extracted to enhance search: summaries that are centered on keyword matches to help judge the relevance of search results [Chen *et al.*, 2019; Wang *et al.*, 2022], and compact summaries that can be fed into a dense re-ranking model (which cannot accept a whole large KG) to improve search accuracy [Chen *et al.*, 2023].

KG Exploration

KG exploration refers to the process that gradually discovers and understands the content of a large and unfamiliar KG, often starting by means of a keyword query asking for matched nodes, edges, or subgraphs [Lissandrini *et al.*, 2022]. The matched relevant insights, as an extractive KG summary, can assist in a number of downstream tasks on this KG, such as the identification of its portions that can satisfy the current

information needs, the comprehension of its graph structure, and the formulation of formal (e.g., SPARQL) queries on it.

KG Reuse

KGs such as Wikidata [Vrandecic and Krötzsch, 2014] are huge and cover a wide variety of topics. For a specific application, it is often sufficient to only extract a relevant portion to be reused. For example, a personalized KG summary containing the subgraph most relevant to an individual user's interests has been extracted so that it can be stored and accessed on resource-constrained devices [Safavi *et al.*, 2019; Vassiliou *et al.*, 2023]. Minimal domain-specific KG summaries have been extracted to reduce the computation without compromising the accuracy of domain-specific applications [Lalithsena *et al.*, 2016; Lalithsena *et al.*, 2017]. Another interesting application is to enrich a news article with a relevant KG summary extracted from a background KG to help the reader comprehend the relations between entities mentioned in the news [Huang *et al.*, 2019; Li *et al.*, 2020].

4 Extraction of Static KG Summaries

Static summaries are extracted to reflect the coverage of the original KG. Depending on the application, existing methods cover different targets: data patterns, or query answers.

4.1 Pattern Coverage-Based Summarization

Entities in KGs are assigned different types, i.e., classes, and they are described using different properties. Extracting a summary that exemplifies the most representative data patterns for describing the entities in a KG is an important component of KG profiling [Ellefi *et al.*, 2018]. Existing methods in this category have been focused on extracting summaries that cover data patterns at different granularities, from independent instantiations of classes and properties to their joint instantiations associated with graph structure. Figure 3 outlines the expressivity increases among different data patterns.

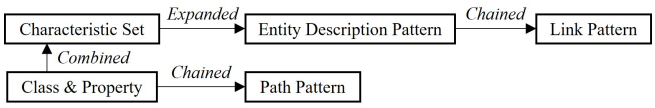

Figure 3: Expressivity increases among data patterns.

Class & Property

Class and property instantiations are the most elementary data patterns in a KG, such as the instances of the class `Person` and the property `nationality` in Figure 1. To exemplify all such instantiations during KG visualization, the 3-S approach [Sundara *et al.*, 2010] performs stratified sampling by uniformly sampling the instances of each class and each property independently. However, this method is only suitable for KGs that have a small schema.

Given a potentially large number of classes and properties used in a KG, the IlluSnip approach [Cheng *et al.*, 2017a] extracts a size-bounded connected subgraph that contains the most frequently instantiated classes and properties, as well as entities having the highest PageRank scores, to illustrate the main content of the KG. For example, the summary in Figure 4 includes `Person` and `act_in` which are the most frequently instantiated class and property in Figure 1, respectively. To fulfill this, a *Maximum-weight-and-coverage Connected Graph* problem (MwcCG) is formulated, and is solved by a greedy approximation algorithm for this new NP-hard optimization problem, where coverage and weights are jointly maximized to account for class/property instantiations and entity scores, respectively.

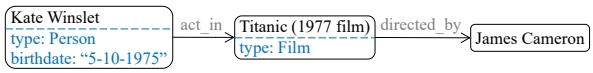

Figure 4: A summary extracted from Figure 1 containing the most frequently instantiated class `Person` and property `act_in`.

As a follow-up effort, Liu et al. [2019] design more efficient algorithms for MwcCG and adapt them to SPARQL endpoints for summarizing remotely accessed KGs.

Characteristic Set

On top of individual properties, the *characteristic set* [Heling and Acosta, 2020] of an entity is the set of all properties (including `type`) describing that entity in a KG, and each property can be associated with its average frequency of occurrence among all the entities having this characteristic set (i.e., average multiplicity). For example, the characteristic set of `James Cameron` in Figure 1 consists of {`nationality`, `type`}. All the characteristic sets instantiated in a KG, the number of entities having each characteristic set, and the average frequency of each property constitute a statistical profile of the KG which is required by a query optimizer for devising efficient query plans. When it is difficult to access or process the entire KG, Heling et al. [2020; 2023] use estimations for this profile derived from a KG summary containing the properties of a set of selected entities. Entities that have high outdegrees (i.e., rich descriptions) have a high chance of being selected.

Entity Description Pattern

In addition to the properties (i.e., outgoing edges) in a characteristic set, the *Entity Description Pattern* (EDP) [Wang *et al.*, 2021] of an entity distinguishes its types from other properties and, further, includes the properties that have this entity as a value (i.e., incoming edges). For example, Figure 5 shows E_1, the EDP of two entities `Kate Winslet` and `Leonardo DiCaprio` in Figure 1, and E_2, the EDP of `Titanic (1977 film)` and `Avatar: The Way of Water`. To extract a summary of a KG to exemplify all or the most frequent EDPs, the two-step PCSG approach [Wang *et al.*, 2021], in its first step, selects the minimum number of connected components that collectively cover those EDPs by formulating and solving a *Set Cover* problem. In the second step, it extracts an optimal subgraph from each selected connected component and merges these subgraphs. Specifically, the subgraph is a tree found by solving a minimum-size *Group Steiner Tree* problem [Ihler, 1991] where all the entities having the same EDP form a group. The extracted minimal tree is required to cover all the groups, i.e.,

contain at least one entity for each EDP; the types and properties of this entity are then extracted to exemplify its EDP.

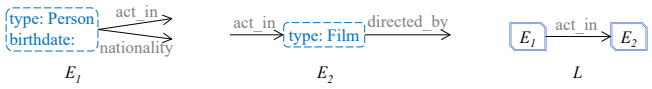

Figure 5: Two EDPs (E_1, E_2) and an LP (L) in Figure 1.

Link Pattern

Building on EDPs, *Link Pattern* (LP) [Wang *et al.*, 2021] takes a further step to characterize not only the neighborhood pattern of each individual entity, but also the patterns of relations between entities. Specifically, the LP of a relation between two entities is a triple consisting of the relation type and the EDPs of the two entities. For example, Figure 5 shows an LP shared by the three `act_in` relations in Figure 1. To extract a KG summary that also exemplifies all or the most frequent LPs, the aforementioned PCSG approach [Wang *et al.*, 2021] extends the formulation of the Group Steiner Tree problem by performing edge subdivision to convert each relation into a node; all the relations having the same LP form a group to be covered by the extracted minimal tree. For example, the summary in Figure 6 exemplifies the two EDPs and the LP in Figure 5. Note that the EDPs of the entities `James Cameron` and `UK` are not fully exemplified in this summary.

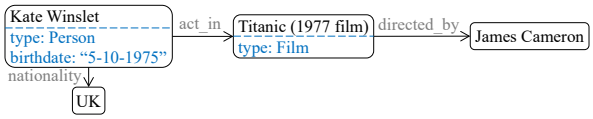

Figure 6: A summary extracted from Figure 1 exemplifying the two EDPs and the LP in Figure 5.

Path Pattern

Going beyond link patterns that focus on the direct connection patterns between adjacent entities, the more generalized *path pattern* [Mynarz *et al.*, 2016] addresses both direct and indirect connections between entities. It is an alternating sequence of classes and properties, derived from the paths in a KG where each entity is replaced by a type assigned to it. For example, Figure 7 shows the most frequent path pattern of length two in Figure 1. To present typical patterns for user comprehension, for each frequent path pattern in a KG, Mynarz et al. [2016] extract a set of diverse and representative instance paths. Diversification is achieved by choosing entities described using different properties, and representative entities are found by clustering entities having similar properties and then choosing from the medoids of the clusters.

Figure 7: A path pattern in Figure 1.

Remarks. An extractive summary aiming at covering data patterns at a low granularity such as class and property instantiations is able to concisely exemplify the content of a KG. However, higher-order structural patterns like EDPs provide more accurate characterization, accompanied with increased computational complexity and also redundancy, e.g., two EDPs may differ insignificantly by only one property. This trade-off is to be considered by the concrete application.

4.2 Answer Coverage-Based Summarization

Replacing a large KG with an extracted compact summary to be queried helps improve query performance [Fan *et al.*, 2014]. Existing methods in this category are targeted at maximizing a summary's coverage of query answers [Rietveld *et al.*, 2014] or its preservation of statistic features of data distribution in the original KG [Montoya *et al.*, 2017]. The latter is important to federated query optimization, for which some methods reviewed in Section 4.1 perform a biased sampling of entities according to their outdegrees [Heling and Acosta, 2020; Heling and Acosta, 2023], which is a common measure of node centrality in graphs. Below we will see other centrality measures and the utilization of query logs.

Graph Structure

Without relying on prior knowledge of queries, the SampLD approach [Rietveld *et al.*, 2014] attempts to purely employ graph topology to predict the relevance of edges for typical queries. Assuming that structurally central entities are likely to be included in common query answers, this approach extracts edges that are incident with the most central nodes in a KG. Three graph centrality measures are applied and compared: *PageRank*, *indegree*, and *outdegree*. For example, if indegree is used, the edge between `James Cameron` and `Titanic (1977 film)` in Figure 1 will be ranked high since these two nodes have the largest indegree in the KG.

Query Log

Besides graph centrality, it would be reasonable to assume that entities and properties that occur frequently in past queries or in the results of past queries are important and are likely to be included in the answers of future queries. For example, if the entity `BBC` in Figure 1 is frequently queried according to the query log, it will have a high priority to be included in the summary despite its relatively small indegree and outdegree in the KG. Guo et al. [2021] incorporate this idea and propose a hybrid approach that takes into account both graph structure and query log. They measure the importance of an edge by a combination of graph centrality (i.e., the indegree and outdegree of its endpoints) and query frequency (i.e., the frequency of co-occurrence of its endpoints in the query log). Node importance is measured in an analogous manner. Finally. top-ranked edges that are incident with top-ranked nodes are extracted to form a summary.

Remarks. Graph structure and query log are complementary to each other. Whereas observed query frequency represents an effective indicator for the likelihood of being queried in the future, when query log is not available, graph centrality could be used as a reasonable estimation for substitution.

5 Extraction of Dynamic KG Summaries

Dynamic summaries are extracted to tailor a KG to users' needs. Existing methods consider needs expressed in different manners: keyword queries, or personal interests.

5.1 Query-Biased Summarization

Snippet generation for KG search [Chen *et al.*, 2019] and query-based KG exploration [Lissandrini *et al.*, 2022] can both be formulated as extracting a query-biased subgraph from a given KG. Early methods in this category solve it as a node/edge ranking problem. Recently, various formulations in combinatorial optimization have been adopted, in particular based on Group Steiner Trees for reflecting connections among query keywords in the graph structure. Not limited to maximizing the coverage of query keywords, they consider answer compactness (i.e., small subgraph) and cohesiveness (i.e., semantically related entities). Figure 8 outlines the expressivity increases among different problem formulations.

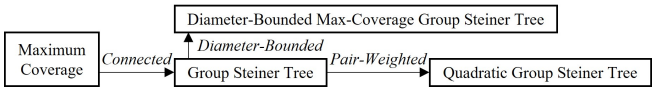

Figure 8: Expressivity increases among problem formulations for query-biased extractive KG summarization.

Early Heuristics

To acquaint the user with a KG in the context of keyword search, in an early work, Bai et al. [2008] extract two subgraphs: a *topic-oriented subgraph* for revealing the main topic of the KG, and a *query-oriented subgraph* for showing its relevance to the query. The topic-oriented subgraph is extracted from the neighborhood of a topic entity based on predefined priorities of its properties; the topic entity is one that has a large indegree and outdegree. The query-oriented subgraph contain edges matched with query keywords; priority goes first to the edges incident with the topic entity.

Maximum Coverage

Instead of separately considering data representativeness and query relevance in two subgraphs, the KSD approach [Wang *et al.*, 2019] jointly maximizes the coverage of all such elements of interest. Specifically, the elements of a KG that a summary is expected to cover include class and property instantiations, central entities, and keyword matches. To achieve it, a weighted *Maximum Coverage* problem is formulated, where each edge (or type/attribute assertion) is represented as a set that may cover a class instantiation, a property instantiation, one or two entities, and/or a number of query keywords. Each class and property to be covered is weighted by its frequency in the KG. Each entity to be covered is assigned a weight measuring its graph centrality based on indegree and outdegree. Solving this problem gives rise to a size-bounded optimum subgraph that maximizes the coverage of representative and query-biased data elements of the KG. As illustrated in Figure 9, the summary extracted by this approach is comparable with the summary extracted by IlluSnip shown in Figure 4. Given a keyword query *Titanic*

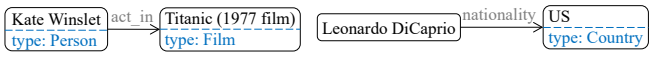

Figure 9: A summary extracted from Figure 1 containing the most frequently instantiated class `Person` and property `act_in`, and containing the query keywords *Titanic* and *US*.

US, whereas both summaries include the most frequently instantiated class `Person` and property `act_in` in Figure 1, the summary here further includes the entity `US` to match the query, although it is not required to be a connected subgraph.

Group Steiner Tree

Independent coverage of KG elements will easily produce a summary composed of multiple disconnected fragments. This weakens the structural cohesiveness of the summary, and misses the connections between keyword matches in the KG, which could be important to the user's judgment of query relevance. To overcome this limitation, a recent line of research formulates and solves a minimum-weight *Group Steiner Tree* (GST) problem [Ihler, 1991]. The KeyKG approach [Shi *et al.*, 2020] maps each query keyword to a group of entities, and the target is a minimum-weight tree that covers all the groups, i.e., contains at least one matched entity for each keyword. Edge weights can be defined by an off-the-shelf method depending on the application. The resulting tree concisely reflects a structural relationship among all the query keywords. For example, given a keyword query *Titanic US*, the summary in Figure 10 includes a path that connects the two query keywords. Considering the NP-hardness of this problem, KeyKG features a new approximation algorithm that employs an offline computed distance index to accelerate online extraction of shortest paths between groups to be merged into a tree.

Figure 10: A summary extracted from Figure 1 containing a path that connects the query keywords *Titanic* and *US*.

The aforementioned PCSG approach incorporates KeyKG as an efficient solver of the GST problem. Moreover, PCSG has been extended to QPCSG [Wang *et al.*, 2021], which covers EDPs, LPs, and also query keywords. All the entities and relations matched with a query keyword form a group to be covered. As illustrated in Figure 11, given a keyword query *Titanic US*, the summary extracted by this extended approach further includes the entity `US` to match the query, extending the summary in Figure 6 extracted by PCSG.

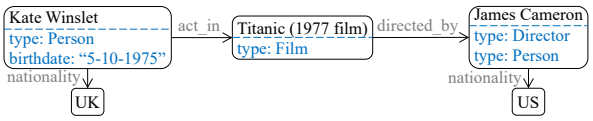

Figure 11: A summary extracted from Figure 1 exemplifying the two EDPs and the LP in Figure 5, and containing the query keywords *Titanic* and *US*.

Diameter-Bounded Max-Coverage Group Steiner Tree

The standard GST problem requires the extracted tree to cover all the groups, but this is not always achievable in practice. For example, given a KG that is a disconnected graph, a tree that connects all the keywords in a query may not exist. Even for a connected graph, a large tree may have to be extracted to cover keywords that are distant from each other in the graph structure, against the goal of extracting a compact summary. To achieve a trade-off between query coverage and structural compactness, Cheng et al. [2020] propose to allow relaxing a query by ignoring a minimum number of keywords to guarantee the compactness of the extracted summary.

As a generalization of this idea, Zhang et al. [2023] formulate a *Diameter-bounded max-Coverage Group Steiner Tree* (DCGST) problem. Compared with a standard GST, DCGST has a bounded diameter and is required to cover not necessarily all but the most possible groups. For example, given a keyword query *Titanic US talking* over Figure 1 and a diameter bound of two hops, the summary extracted here has to ignore the keyword *talking* but choose to be concisely focused on the connection between the other two keywords, as illustrated in Figure 10. By contrast, the summary extracted by the aforementioned KeyKG approach is a standard GST consisting of a long and unfocused path, as shown in Figure 12, although it covers all the three keywords. The DCGST problem is an emerging NP-hard optimization problem and is solved by PrunedCBA [Zhang *et al.*, 2023], an approximation algorithm performing pruned best-first search based on a novel hop-bounded index of precomputed distances.

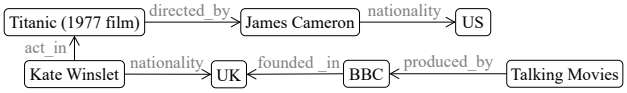

Figure 12: A summary extracted from Figure 1 containing a long path that connects the query keywords *Titanic*, *US*, and *talking*.

Quadratic Group Steiner Tree

Another extension of the GST formulation for query-biased KG summarization considers the semantic cohesiveness of the extracted tree [Cheng and Kharlamov, 2017]. Recall that the standard minimum-weight GST problem optimizes the total weight of the extracted nodes or edges, where weights reflect (inverse) salience. However, a set of salient nodes or edges may not always comprise a meaningful connection as a whole. It can be a semantically disjointed tree that connects a set of salient but disparate entities. To improve semantic cohesiveness, Shi et al. [2021a; 2021b] incorporate the minimization of semantic distances between extracted entities into the objective function. They formulate a more generalized *Quadratic Group Steiner Tree* (QGST) problem. Compared with a standard GST, QGST minimizes a linear combination of the sum of weights of its nodes and the sum of quadratic weights of its node pairs. Quadratic weight characterizes the semantic distance between a pair of entities, e.g., the angular distance between their embedding vectors. The extracted summary tends to include entities that are not only salient by themselves and matched with the query, but also semantically close to each other, hence forming a semantically cohesive whole. For example, given a keyword query *Avatar DiCaprio*, the top tree in Figure 13 contains salient entities like `US` but also disparate entities including a film, a director, a country, and an actor. By contrast, the entities in the bottom tree are a set of closely related films and actors, representing a more meaningful connection. To solve the QGST problem which is an NP-hard optimization problem, B^3F [Shi *et al.*, 2021b] is an exact algorithm that performs branch-and-bound best-first search. QO and EO [Shi *et al.*, 2021a] are two approximation algorithms with the idea of finding and merging small-weight paths from a root node to all the groups.

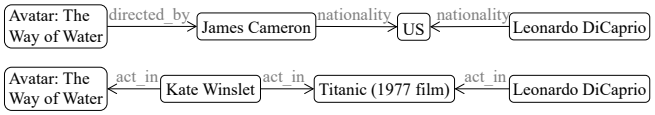

Figure 13: Two summaries extracted from Figure 1 containing the query keywords *Avatar* and *DiCaprio*.

Remarks. Compared with maximum coverage, GST-based formulations are expressive but hard to solve. Although KeyKG and PrunedCBA perform satisfyingly on large KGs with millions of nodes, they rely on a precomputed distance index which needs to be rebuilt when the KG evolves. For the QGST problem, no algorithm so far could respond to a query over a million-scale KG in real time. All these limitations should be taken into account when choosing the formulation.

5.2 Personalized Summarization

Besides keyword queries, users' interests can be explicitly expressed or implicitly discovered in various manners. Serving the user with a personalized summary extracted from a large KG provides cost-effectiveness in KG reuse [Safavi *et al.*, 2019; Vassiliou *et al.*, 2023]. Existing methods in this category tailor KGs to a user's needs in different forms: a single entity of interest, a class representing a domain or a set of entities specified by the user, or the user's own query history.

Single Entity

The iSummary approach [Vassiliou *et al.*, 2023] allows the user to specify an entity of interest in the KG as a *seed entity*, such as `James Cameron`, and it extracts a summary that compactly describes this entity as well as a set of its closely related entities. Specifically, each entity in the KG is assigned a weight. An entity with a higher frequency of co-occurrence with the seed entity in the query log has a higher weight. This frequency is believed to reflect the relatedness between entities. From the resulting node-weighted graph, a maximum-weight tree that connects the seed entity with its most related entities is extracted. This computational problem resembles the minimum-weight *Steiner Tree* problem and is NP-hard. It is solved by an approximation algorithm that extracts shortest paths to connect the seed entity with its most related entities.

Multiple Entities (in a Domain)

A user may also want to extract a domain-specific summary from a large KG. A domain of interest can be represented by a

set of *in-domain entities*, e.g., all the entities having a particular type such as `Film` in Figure 1. To fulfill this need, Lalithsena et al. [2016] traverse the KG starting from in-domain entities to extract domain-specific edges. Their primary focus is to determine the domain specificity score of each relation type, and they present two measures. The first measure utilizes the strength of association among the types of relations and intermediate entities derived from instance-level frequencies. The second and more effective measure relies on the domain specificity of intermediate relations and adopts Pointwise Mutual Information for measuring association.

In a follow-up work, Lalithsena et al. [2017] further employ the Wikipedia categories of each entity and restrict graph traversal to the entities in the top-ranked domain-specific categories. To determine the domain specificity score of a category, they collect evidences from the types of the entities in the category, its lexical label, and its structural abstractness characterized by its outdegree in the category hierarchy. Evidences are aggregated for ranking categories using Probabilistic Soft Logic, a statistical relational learning framework.

Instead of graph traversal, Jiang et al. [2012] firstly mine the frequent subgraph patterns in a KG, and extract their instance subgraphs that contain user-specified entities. These subgraphs are merged by intersection or union into a single graph, where each edge is weighted based on a combination of the frequencies of its constituent entities and relation type. Finally, a minimum-weight Steiner Tree that connects all the user-specified entities is extracted from this graph as a summary of the semantic associations among those entities.

Here we would like to distinguish the approaches in this category from the research on *relationship search* [Cheng, 2020], i.e., searching and ranking subgraphs that connect a given set of input entities [Cheng *et al.*, 2017b]. Their main difference is that the approaches here extract and summarize all the subgraphs relevant to the input entities, whereas relationship search is only focused on finding a few top-ranked subgraphs to be presented as search results.

Query History

When a user's personal query history is available, the GLIMPSE approach [Safavi *et al.*, 2019] extracts a personalized summary from a KG containing only a subgraph most relevant to the user's interest reflected in past queries. To this end, entities and relations are scored by a probabilistic framework estimated from their frequencies observed in the user's query history. The extraction of a size-bounded optimum subgraph is formulated as an optimization problem with a submodular objective function, hence featuring a greedy constant-factor approximation algorithm.

Remarks. Despite the diverse forms of expressing a user's needs, it is possible to convert them into each other to be handled in a consistent manner. For example, from a user's query history, a set of frequent entities can be mined, which, in turn, can be processed separately as single entities.

6 Evaluation of Extractive KG Summaries

Considering the magnitude of a KG, it is difficult—if not impossible—to manually extract a summary as the gold standard for evaluation. Instead, depending on the application, various *quality metrics* have been used to quantitatively assess the quality of a summary from different perspectives. For example, Wang et al. [2023] devise an evaluation framework for measuring a summary's coverage of the classes, properties, EDPs, and LPs in the original KG. They also publish a benchmark called BANDAR with thousands of real-world KGs to be used for evaluation. Cheng et al. [2017a] conduct a user study, inviting human users to rate and compare the quality of the summaries extracted by different approaches.

As to task-specific summaries, *extrinsic evaluation* aims to assess the quality of a summary indirectly by testing its performance in downstream tasks using the summary. For example, Rietveld et al. [2014] and Guo et al. [2021] measure a summary's coverage of query answers by comparing the results retrieved from the summary with the correct results retrieved from the original KG. Wang et al. [2021] conduct a user study to compare the usefulness of the summaries extracted by different approaches for assisting users in comprehending a large KG and then completing complex SPARQL queries over the KG. Useful summaries are expected to help complete SPARQL queries correctly and also quickly.

Besides, the *time* for extracting a summary is an important factor in evaluating the practicability of an algorithm, especially for real-time applications such as KG search engines.

7 Conclusion and Future Directions

We have systematically reviewed existing applications, approaches, and evaluation methods for extractive KG summarization. This interdisciplinary research topic spanning artificial intelligence, data mining, and information retrieval provides increasing research opportunities. We highlight the following research directions that are currently underexplored.

Neural Extraction. While existing approaches mainly exploit the symbolic features of KGs (e.g., ontological schema, graph structure), there is much room for exploring neural methods, given the rapid advances in deep learning models in recent years, particularly graph neural networks.

Supervised Extraction. Although unsupervised extraction has become mainstream to KG summarization, supervised methods deserve to be taken into consideration. The inadequacy of labeled data for training could be addressed by semi-supervised learning such as self-training and co-training.

Generative Extraction. Extractive summaries may also be obtained using a generative model. Motivated by the recent success in generative information extraction and retrieval, it would be an interesting investigation to apply pretrained generative models to KG summary extraction.

Comparative Extraction. When multiple KGs need to be summarized and compared, e.g., in the results page of a KG search engine, the extraction of their summaries should not be independent but can be performed jointly, attending to their common and distinctive features to facilitate user's selection.

Collaborative Extraction. Given the computational cost of processing a large KG, we anticipate a scenario where summaries are extracted not from scratch but building on stored summaries previously extracted for similar tasks, which can be inexpensively assembled and tailored to new needs.

Acknowledgments

This work was supported by the NSFC (62072224).

References

[Bai *et al.*, 2008] Xi Bai, Renaud Delbru, and Giovanni Tummarello. RDF snippets for Semantic Web search engines. In *Proc. OTM, Part II*, pages 1304–1318, 2008.

[Cebiric *et al.*, 2019] Sejla Cebiric, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. Summarizing semantic graphs: a survey. *VLDB J.*, 28(3):295–327, 2019.

[Chapman *et al.*, 2020] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. Dataset search: a survey. *VLDB J.*, 29(1):251–272, 2020.

[Chen *et al.*, 2019] Jinchi Chen, Xiaxia Wang, Gong Cheng, Evgeny Kharlamov, and Yuzhong Qu. Towards more usable dataset search: From query characterization to snippet generation. In *Proc. CIKM*, pages 2445–2448, 2019.

[Chen *et al.*, 2023] Qiaosheng Chen, Zixian Huang, Zhiyang Zhang, Weiqing Luo, Tengteng Lin, Qing Shi, and Gong Cheng. Dense re-ranking with weak supervision for RDF dataset search. In *Proc. ISWC, Part I*, pages 23–40, 2023.

[Cheng and Kharlamov, 2017] Gong Cheng and Evgeny Kharlamov. Towards a semantic keyword search over industrial knowledge graphs (extended abstract). In *Proc. IEEE BigData*, pages 1698–1700, 2017.

[Cheng *et al.*, 2017a] Gong Cheng, Cheng Jin, Wentao Ding, Danyun Xu, and Yuzhong Qu. Generating illustrative snippets for open data on the Web. In *Proc. WSDM*, pages 151–159, 2017.

[Cheng *et al.*, 2017b] Gong Cheng, Fei Shao, and Yuzhong Qu. An empirical evaluation of techniques for ranking semantic associations. *IEEE Trans. Knowl. Data Eng.*, 29(11):2388–2401, 2017.

[Cheng *et al.*, 2020] Gong Cheng, Shuxin Li, Ke Zhang, and Chengkai Li. Generating compact and relaxable answers to keyword queries over knowledge graphs. In *Proc. ISWC, Part I*, pages 110–127, 2020.

[Cheng, 2020] Gong Cheng. Relationship search over knowledge graphs. *SIGWEB Newsl.*, 2020(Summer):3:1–3:8, 2020.

[Ellefi *et al.*, 2018] Mohamed Ben Ellefi, Zohra Bellahsene, John G. Breslin, Elena Demidova, Stefan Dietze, Julian Szymanski, and Konstantin Todorov. RDF dataset profiling - a survey of features, methods, vocabularies and applications. *Semantic Web*, 9(5):677–705, 2018.

[Fan *et al.*, 2014] Wenfei Fan, Xin Wang, and Yinghui Wu. Answering graph pattern queries using views. In *Proc. ICDE*, pages 184–195, 2014.

[Guo and Wang, 2021] Jimao Guo and Yi Wang. Summarizing RDF graphs using node importance and query history. In *Proc. ICSS*, pages 51–58, 2021.

[Heling and Acosta, 2020] Lars Heling and Maribel Acosta. Estimating characteristic sets for RDF dataset profiles based on sampling. In *Proc. ESWC*, pages 157–175, 2020.

[Heling and Acosta, 2023] Lars Heling and Maribel Acosta. Characteristic sets profile features: Estimation and application to SPARQL query planning. *Semantic Web*, 14(3):491–526, 2023.

[Hogan *et al.*, 2022] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37, 2022.

[Huang *et al.*, 2019] Zixian Huang, Shuxin Li, Gong Cheng, Evgeny Kharlamov, and Yuzhong Qu. MiCRon: making sense of news via relationship subgraphs. In *Proc. CIKM*, pages 2901–2904, 2019.

[Ihler, 1991] Edmund Ihler. The complexity of approximating the class Steiner tree problem. In *Proc. WG*, pages 85–96, 1991.

[Jiang *et al.*, 2012] Xiaowei Jiang, Xiang Zhang, Wei Gui, Feifei Gao, Peng Wang, and Fengbo Zhou. Summarizing semantic associations based on focused association graph. In *Proc. ADMA*, pages 564–576, 2012.

[Kellou-Menouer *et al.*, 2022] Kenza Kellou-Menouer, Nikolaos Kardoulakis, Georgia Troullinou, Zoubida Kedad, Dimitris Plexousakis, and Haridimos Kondylakis. A survey on semantic schema discovery. *VLDB J.*, 31(4):675–710, 2022.

[Koesten *et al.*, 2017] Laura M. Koesten, Emilia Kacprzak, Jenifer Fay Alys Tennison, and Elena Simperl. The trials and tribulations of working with structured data: -a study on information seeking behaviour. In *Proc. CHI*, pages 1277–1289, 2017.

[Lalithsena *et al.*, 2016] Sarasi Lalithsena, Pavan Kapanipathi, and Amit P. Sheth. Harnessing relationships for domain-specific subgraph extraction: A recommendation use case. In *Proc. IEEE BigData*, pages 706–715, 2016.

[Lalithsena *et al.*, 2017] Sarasi Lalithsena, Sujan Perera, Pavan Kapanipathi, and Amit P. Sheth. Domain-specific hierarchical subgraph extraction: A recommendation use case. In *Proc. IEEE BigData*, pages 666–675, 2017.

[Li *et al.*, 2016] Jia Li, Yang Cao, and Xudong Liu. Approximating graph pattern queries using views. In *Proc. CIKM*, pages 449–458, 2016.

[Li *et al.*, 2020] Shuxin Li, Zixian Huang, Gong Cheng, Evgeny Kharlamov, and Kalpa Gunaratna. Enriching documents with compact, representative, relevant knowledge graphs. In *Proc. IJCAI*, pages 1748–1754, 2020.

[Lissandrini *et al.*, 2022] Matteo Lissandrini, Davide Mottin, Katja Hose, and Torben Bach Pedersen. Knowledge graph exploration systems: are we lost? In *Proc. CIDR*, 2022.

[Liu *et al.*, 2018] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM Comput. Surv.*, 51(3):62:1–62:34, 2018.

[Liu *et al.*, 2019] Daxin Liu, Gong Cheng, Qingxia Liu, and Yuzhong Qu. Fast and practical snippet generation for RDF datasets. *ACM Trans. Web*, 13(4):19:1–19:38, 2019.

[Liu *et al.*, 2021] Qingxia Liu, Gong Cheng, Kalpa Gunaratna, and Yuzhong Qu. Entity summarization: State of the art and future challenges. *J. Web Semant.*, 69:100647, 2021.

[Montoya *et al.*, 2017] Gabriela Montoya, Hala Skaf-Molli, and Katja Hose. The Odyssey approach for optimizing federated SPARQL queries. In *Proc. ISWC, Part I*, pages 471–489, 2017.

[Mynarz *et al.*, 2016] Jindrich Mynarz, Marek Dudás, Paolo Tomeo, and Vojtech Svátek. Generating examples of paths summarizing RDF datasets. In *Proc. Posters and Demos Track of SEMANTiCS and SuCCESS*, 2016.

[Pan *et al.*, 2023] Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. Large language models and knowledge graphs: Opportunities and challenges. *TGDK*, 1(1):2:1–2:38, 2023.

[Peng *et al.*, 2023] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. Knowledge graphs: Opportunities and challenges. *Artif. Intell. Rev.*, 56(11):13071–13102, 2023.

[Pouriyeh *et al.*, 2019] Seyedamin Pouriyeh, Mehdi Allahyari, Qingxia Liu, Gong Cheng, Hamid Reza Arabnia, Maurizio Atzori, Farid Ghareh Mohammadi, and Krys J. Kochut. Ontology summarization: Graph-based methods and beyond. *Int. J. Semantic Comput.*, 13(2):259–283, 2019.

[Rietveld *et al.*, 2014] Laurens Rietveld, Rinke Hoekstra, Stefan Schlobach, and Christophe Guéret. Structural properties as proxy for semantic relevance in RDF graph sampling. In *Proc. ISWC, Part II*, pages 81–96, 2014.

[Safavi *et al.*, 2019] Tara Safavi, Caleb Belth, Lukas Faber, Davide Mottin, Emmanuel Müller, and Danai Koutra. Personalized knowledge graph summarization: From the cloud to your pocket. In *Proc. ICDM*, pages 528–537, 2019.

[Scherp *et al.*, 2023] Ansgar Scherp, David Richerby, Till Blume, Michael Cochez, and Jannik Rau. Structural summarization of semantic graphs using quotients. *TGDK*, 1(1):12:1–12:25, 2023.

[Shi *et al.*, 2020] Yuxuan Shi, Gong Cheng, and Evgeny Kharlamov. Keyword search over knowledge graphs via static and dynamic hub labelings. In *Proc. WWW*, pages 235–245, 2020.

[Shi *et al.*, 2021a] Yuxuan Shi, Gong Cheng, Trung-Kien Tran, Evgeny Kharlamov, and Yulin Shen. Efficient computation of semantically cohesive subgraphs for keyword-based knowledge graph exploration. In *Proc. WWW*, pages 1410–1421, 2021.

[Shi *et al.*, 2021b] Yuxuan Shi, Gong Cheng, Trung-Kien Tran, Jie Tang, and Evgeny Kharlamov. Keyword-based knowledge graph exploration based on quadratic group Steiner trees. In *Proc. IJCAI*, pages 1555–1562, 2021.

[Sundara *et al.*, 2010] Seema Sundara, Medha Atre, Vladimir Kolovski, Souripriya Das, Zhe Wu, Eugene Inseok Chong, and Jagannathan Srinivasan. Visualizing large-scale RDF data using subsets, summaries, and sampling in oracle. In *Proc. ICDE*, pages 1048–1059, 2010.

[Vassiliou *et al.*, 2023] Giannis Vassiliou, Fanouris Alevizakis, Nikolaos Papadakis, and Haridimos Kondylakis. iSummary: workload-based, personalized summaries for knowledge graphs. In *Proc. ESWC*, pages 192–208, 2023.

[Vrandecic and Krötzsch, 2014] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

[Wang *et al.*, 2019] Xiaxia Wang, Gong Cheng, and Evgeny Kharlamov. Towards multi-facet snippets for dataset search. In *Proc. PROFILES & SEMEX*, pages 1–6, 2019.

[Wang *et al.*, 2021] Xiaxia Wang, Gong Cheng, Tengteng Lin, Jing Xu, Jeff Z. Pan, Evgeny Kharlamov, and Yuzhong Qu. PCSG: pattern-coverage snippet generation for RDF datasets. In *Proc. ISWC*, pages 3–20, 2021.

[Wang *et al.*, 2022] Xiaxia Wang, Tengteng Lin, Weiqing Luo, Gong Cheng, and Yuzhong Qu. CKGSE: A prototype search engine for Chinese knowledge graphs. *Data Intell.*, 4(1):41–65, 2022.

[Wang *et al.*, 2023] Xiaxia Wang, Gong Cheng, Jeff Z. Pan, Evgeny Kharlamov, and Yuzhong Qu. BANDAR: benchmarking snippet generation algorithms for (RDF) dataset search. *IEEE Trans. Knowl. Data Eng.*, 35(2):1227–1241, 2023.

[Wang, 2017] Xin Wang. Answering graph pattern matching using views: A revisit. In *Proc. DEXA, Part I*, pages 65–80, 2017.

[Yang *et al.*, 2017] Shuo Yang, Lei Zou, Zhongyuan Wang, Jun Yan, and Ji-Rong Wen. Efficiently answering technical questions - A knowledge graph approach. In *Proc. AAAI*, pages 3111–3118, 2017.

[Zhang *et al.*, 2023] Ke Zhang, Xiaoqing Wang, and Gong Cheng. Efficient approximation algorithms for the diameter-bounded max-coverage group Steiner tree problem. In *Proc. WWW*, pages 199–209, 2023.

[Zhao *et al.*, 2020] Yang Zhao, Jiajun Zhang, Yu Zhou, and Chengqing Zong. Knowledge graphs enhanced neural machine translation. In *Proc. IJCAI*, pages 4039–4045, 2020.