CKGSE: A Prototype Search Engine for Chinese Knowledge Graphs (Postprint)

Authors: Xiaxia Wang, Tengteng, Lin, Weiqing Luo, Gong, Cheng, Yuzhong, Qu, Gong, Cheng

Date: 2022-11-28T00:00:00+00:00

Abstract

Nowadays, with increasing open knowledge graphs (KGs) being published on the Web, users depend on open data portals and search engines to find KGs. However, existing systems provide search services and present results with only metadata while ignoring the contents of KGs, i.e., triples. It brings difficulty for users' comprehension and relevance judgement. To overcome the limitation of metadata, in this paper we propose a content-based search engine for open KGs named CKGSE. Our system provides keyword search, KG snippet generation, KG profiling and browsing, all based on KGs' detailed, informative contents rather than their brief, limited metadata. To evaluate its usability, we implement a prototype with Chinese KGs crawled from OpenKG.CN and report some preliminary results and findings.

Full Text

Preamble

CKGSE: A Prototype Search Engine for Chinese Knowledge Graphs

Xiaxia Wang, Tengteng Lin, Weiqing Luo, Gong Cheng† & Yuzhong Qu
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

Keywords: Knowledge graph; Search engine; Snippet generation; Dataset profiling; Browsing

Citation: Wang, X.X., et al.: CKGSE: A prototype search engine for Chinese knowledge graphs. Data Intelligence 4(1), 41-65 (2022). doi: 10.1162/dint_a_{00118}

Received: October 30, 2021; **Revised:** December 23, 2021; **Accepted:** January 19, 2022

ABSTRACT

Nowadays, with increasing open knowledge graphs (KGs) being published on the Web, users depend on open data portals and search engines to find KGs. However, existing systems provide search services and present results with only metadata while ignoring the contents of KGs, i.e., triples. This brings difficulty for users' comprehension and relevance judgement. To overcome the limitation of metadata, in this paper we propose a content-based search engine for open KGs named CKGSE. Our system provides keyword search, KG snippet generation, KG profiling and browsing, all based on KGs' detailed, informative contents rather than their brief, limited metadata. To evaluate its usability, we implement a prototype with Chinese KGs crawled from OpenKG.CN and report some preliminary results and findings.

1. INTRODUCTION

Reusing existing data, particularly knowledge graphs (KGs), eliminates duplicate effort and is therefore crucial for scientific research and application development. In recent years, substantial academic and industrial investment has been directed toward constructing reusable KGs, especially in specialized domains such as e-commerce, biomedicine, and education. Consequently, numerous KGs have been published on the Web as reusable resources [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. This trend has spurred the development of data sharing platforms, from early initiatives like Datahub (established in 2006) and the European Data Portal to hundreds of data portals worldwide [12]. Among the resources available on these portals—such as the 256 portals documented in [12]—KGs constitute a significant component. For instance, Data.Gov (https://www.data.gov/) had indexed over 10,000 KGs by November 2021, while the Linked Open Data Cloud (https://lod-cloud.net/) has indexed 1,301 KGs. These platforms provide a solid foundation for users to easily publish and access KG resources. Moreover, to help users efficiently locate KGs and assess their relevance, recent research has introduced various systems, ranging from general-purpose search engines like Google Dataset Search [13] to specialized systems such as LODAtlas [14] for KG-centric data. OpenKG.CN is a popular platform for Chinese open KGs that offers keyword-based search services.

Motivation. Although the aforementioned systems [13, 14] provide convenient search services, they rely exclusively on metadata—meta-level annotations provided by KG publishers that include authorship and license information. Generally, metadata annotations offer high-level descriptions but lack details about the actual KG content. This reliance on metadata introduces two key limitations in existing systems. **Limitation 1**: When a keyword query refers to KG content (i.e., triples), such as a specific entity, class, or property, these systems cannot effectively locate the target KG. In practice, such queries are common; according to one analysis [15], over 60% of KG search queries contain keywords referencing content. **Limitation 2**: For presenting search results, metadata cannot provide close-up views of the underlying KG content. Yet many user in-

teractions involved in relevance judgment depend on this content. In [16], users' actual data needs are categorized into ten types, with most—such as exploration and analysis—focusing primarily on data content. Furthermore, existing analyses of the KG search process [17] and search result presentation [18] demonstrate heavy reliance on KG content. Reference [17] divides the search process into four steps, where comprehension of the underlying KG content during the data handling step is critical to search effectiveness. Reference [18] identifies features for characterizing KGs, such as representative elements and instance-level statistics, that are based on content. In summary, while the utility of metadata is limited [15, 19], KG content should be incorporated into the search process.

Our Attempts. Given these two limitations, a content-based search engine for KGs is needed. However, as a novel approach in this direction, its practical viability remains uncertain. In this paper, as a preliminary effort to build and evaluate a content-based search system for open KGs, we present CKGSE (Chinese KG Search Engine). It comprises four components. To address Limitation 1, **KG Crawling and Storage** obtains KGs and their metadata via the CKAN API, then parses and stores the KGs locally. **Content-Based Keyword Search** parses keyword queries and retrieves and ranks relevant KGs using an inverted index that includes both metadata and content fields. To address Limitation 2, for each search result, **Content-Based Snippet Generation** extracts a sub-KG to demonstrate its query relevance. **Content Profiling and Browsing** provides detailed information about a KG, including a quality profile, statistical summaries (both abstractive and extractive), and a faceted browsing panel for exploring the original KG. To evaluate CKGSE' s practicality, we implemented a prototype (http://ws.nju.edu.cn/CKGSE) using Chinese KGs collected from OpenKG.CN. Our contributions are summarized as follows:

- We propose CKGSE, one of the first content-based search engines for open KGs;
- We present and discuss experimental results demonstrating the practicality of such a system.

Outline. The remainder of this paper is organized as follows. Related work is discussed in Section 2. Sections 3 and 4 introduce an overview of CKGSE and its detailed implementation, respectively. Experimental results are presented in Section 5. Section 6 concludes the paper and outlines future work.

This article extends our previous work [20] in six aspects: new system components, updated implementation, a new user study, and a more comprehensive research background.

- We revised our design and implementation of the inverted index in the content-based keyword search component (Section 4.2).
- We added query-relevant indexed fields to the content-based snippet generation component (Section 4.3).
- We added a quality profile with six quality metrics to the content profiling

component (Section 4.4).
• We added three visualized tag clouds to the statistical summary of the content profiling component (Section 4.4).
• We added a user study to verify CKGSE' s helpfulness by comparing it with OpenKG.CN (Section 5.3).
• We extended related work about KG summarization and snippet generation (Section 2.3).

We have accordingly updated our online system with new components and implementation.

2. RELATED WORK

We present related work for our system from three perspectives. First, Section 2.1 provides background on existing KG search systems and techniques. Second, since an important aspect of a KG search system is presenting each result KG to the user, Section 2.2 reviews methods for profiling KGs. Third, as CKGSE focuses on KG content, Section 2.3 discusses content-based summarization methods for illustrating and exemplifying KGs.

2.1 Open KG Portals and Search Engines

Various open data portals are available today [12]. They generally rely on metadata annotation under specific vocabularies such as W3C DCAT (https://www.w3.org/TR/vocab-dcat-2/) to collect and manage KG resources. For example, the European Data Portal [21] uses title and description values from each resource' s metadata for deduplication. Google Dataset Search [13] identifies data resources through metadata indexing without considering content, as its goal is to direct users to the original webpage for each resource. Some KG-centric systems also depend heavily on metadata, such as LODAtlas [14], which provides metadata-based faceted filters for KG selection.

Since metadata-based systems cannot provide close-up views of underlying KG content, they are vulnerable to unguaranteed metadata quality. To overcome this limitation, CKGSE incorporates KG content. First, to support keyword search over content, we include content elements in the inverted index for each KG. Second, to facilitate relevance judgment of search results, we extract a query-biased snippet for each KG. Third, to provide a close-up view of each KG, we employ various content-based profiles and exploration methods to ease comprehension.

2.2 KG Profiling

KG profiling aims to interpret a KG from various descriptive perspectives [17, 18, 22]. In [18], a taxonomy was proposed to categorize profiling features into seven types based on a survey of literature over two decades, most of which are metadata-related, such as Licensing and Provenance. The qualitative category

includes metrics for assessing resource usability [12, 23]. The statistical category encompasses data element counts and distributions, such as the number of instantiated classes or properties in a KG [24]. The general category contains methods for selecting representative elements from the KG, like structural summaries [25, 26, 27, 28] and pattern mining [29, 30].

Building on these fruitful research efforts, CKGSE incorporates several profiling techniques. In addition to metadata, CKGSE evaluates each KG using a set of qualitative metrics as a quality profile, provides element-level statistical summaries, mines frequent entity description patterns (EDPs) as an abstractive summary [31, 32], and illustrates content with an extractive summary [33, 34].

2.3 KG Summarization and Snippet Generation

KG summarization distills a small but significant portion from a large original KG to accomplish specific tasks, such as reducing storage costs or answering queries efficiently. Considerable research attention has been devoted to generating abstractive summaries for KGs [25]. These approaches focus on graph structures and aggregate frequent or common sub-structures as summaries. For example, entities can be aggregated based on EDP-based similarity [29, 30] or common multi-hop neighborhoods [26, 35]. This aggregation can also be hierarchical [28]. Reference [36] discusses the trade-off between summary size and restoration accuracy.

Complementing abstractive summaries, KG snippet generation methods extract a representative sub-graph from the original KG to exemplify its content. Depending on the application, KG snippets can be either query-relevant [31, 32, 37] or query-independent [33, 34].

To provide users with diversified views of KG content, CKGSE implements both abstractive KG summaries [31] to show representative patterns and extractive snippets [34] to illustrate underlying data content. Additionally, since snippets can be query-relevant, CKGSE also presents snippets [37] on the search results page to exemplify query relevance for each KG.

3. SYSTEM OVERVIEW

[Figure 1: see original paper] presents the user interaction flow and system architecture of CKGSE, which consists of four main components. When a user submits a query, **Content-Based Keyword Search** first parses the query, then retrieves relevant KGs and ranks them as a list. To exemplify the relevance of each KG in the list, **Content-Based Snippet Generation** presents not only query-relevant metadata information but also an extractive sub-KG as a snippet on the search results page. When a user selects a KG, **Content Profiling and Browsing** displays its detailed profiled information with content exploration support. In the backend, **KG Crawling and Storage** collects, parses, and stores all KGs along with their metadata.

KG Crawling and Storage. First, to collect KG resources in an offline process, CKGSE retrieves and stores all available metadata records from OpenKG.CN. Then, following the download links in these records, all accessible KG dump files are downloaded, parsed, and stored in a local database. Based on this, four indexes are built to support downstream tasks in other components; details are introduced in Section 4.1.

Content-based Keyword Search. In this component, any input keyword query is first parsed by segmenting it into (Chinese) words, with stop words removed. The parsed query is then searched against the content-based inverted index to obtain top-ranked KGs according to a relevance scoring function. The inverted index contains multiple fields from both metadata and content to support keyword matching on any of them, with each field having a boost factor for relevance scoring. Details are presented in Section 4.2.

Content-based Snippet Generation. For each KG in the ranked results list, a query-relevant snippet containing metadata and content information is presented to the user. As the first part, we display the indexed fields that match the query and highlight the matched keywords. To further exemplify the query relevance of the KG content through graph structure, in the second part an extractive content snippet is generated online. After loading all triples of the KG into memory, the generation method KSD [37] iteratively extracts a sub-KG considering both query relevance and content representativeness, supported by a label index. Finally, the output content snippet contains k top-ranked triples (where k is a pre-defined size limit), presented as a node-link diagram along with the metadata snippet on the search results page. An example of the content-based snippet is presented in [Figure 6: see original paper]. Details are presented in Section 4.3.

Content Profiling and Browsing. For each KG selected by the user, its profile is presented, including a quality profile and three content-based summaries at different levels. The KG quality profile [23] is a set of content-based quantitative metrics evaluating the intrinsic quality and usability of the KG, such as availability and understandability. The statistical summary includes counts, distributions, and cloud representations of KG elements, including classes, properties, and entities. The abstractive summary contains the most frequent entity description patterns (EDPs) [31, 32] instantiated in the KG. The extractive summary is an optimal sub-KG of k' triples generated by IlluSnip [33, 34] in terms of content representativeness, where k' is a pre-defined size limit. All quality metrics and summaries are pre-computed and indexed in an offline process. In addition to the profile, supported by an entity index, all entities in the KG can be explored in a faceted manner—by filtering entities using their classes and properties. By selecting any filtered entity, all triples describing it can be browsed. Details of the methods are presented in Section 4.4. Examples of content profiling and browsing are given in Section 5.

4. SYSTEM IMPLEMENTATION

We now introduce the detailed implementation of CKGSE by component.

4.1 KG Crawling and Storage

This component crawls all metadata and KGs from OpenKG.CN and stores them in a local database.

KG Crawling. Through the CKAN API, CKGSE first retrieves all available metadata records for resources from OpenKG.CN. Among the obtained records, 40 resources are identified as KGs based on formats such as RDF/XML, N-Triple, Turtle, and JSON-LD; the remaining resources are non-KG resources and are not our focus. Then, by accessing the download links in the metadata, all KG dump files are downloaded, parsed, and stored in a local MySQL database.

Metadata. Metadata for all KGs is stored as a table, where each row identifies a KG and each column represents a metadata field instantiated in the CKAN vocabulary, such as Title, Author, and License, most of which contain textual values. We note that incorrect and incomplete field values are relatively common, as they are freely submitted by KG publishers.

Triples. For each KG, all its dump files are parsed using Apache Jena 3.8.0. For some KGs that have multiple dump files, triple-level deduplication is performed before storing them in the database. The triples of each KG are stored in a table with three columns identifying the subject, predicate, and object. Additionally, each RDF term in the KG is labeled with a human-readable textual form, including the local name, the value of the rdfs:label property, and the textual form of literals. The term-label map is stored in the database and will be used in downstream components.

We report all storage costs of CKGSE in .

4.2 Content-based Keyword Search

This component parses the given keyword query, then retrieves and ranks relevant KGs using an inverted index.

Inverted Index. To support keyword matching on both KG metadata and content, an inverted index is created with eight fields, as presented in . Four fields are manually selected from existing metadata, as they are descriptive and likely to match query keywords. For KG content, which is not considered in existing systems, we divide all RDF terms in each KG into four categories following the RDF schema of classes and properties, and index all of them by their textual forms.

shows the metadata and content fields used in the inverted index. For all eight fields, we assign a boost factor between 0 and 1 when aggregating them into a relevance score. Currently, the boost factors are manually tuned but could be

adjusted in the future using more user query logs. We use Apache Lucene 7.5.0 to construct the index and report the construction time cost in .

Query Parsing. We implement the keyword search component with Apache Lucene 7.5.0. Since it only provides basic query analyzers that cannot effectively perform Chinese word segmentation, we incorporate the IK analyzer (https://code.google.com/archive/p/ik-analyzer/), an open-source tool for Chinese word segmentation.

KG Retrieving and Ranking. The keywords parsed from the query are then searched against the inverted index, with OR as the default Boolean operator between keywords. CKGSE adopts a multi-field query parser based on Lucene to retrieve relevant KGs across all eight fields using the BM25 scoring function, then combines all scores with the assigned boost factor for each field and normalizes the result as the overall relevance score. KGs are ranked by these overall relevance scores, and the top-ranked ones are returned.

4.3 Content-based Snippet Generation

Existing systems simply list metadata information to illustrate each returned KG, providing limited assistance for relevance judgment. In contrast, CKGSE provides a query-relevant snippet for each returned KG that includes both metadata and content parts to facilitate relevance judgment.

Query-relevant Fields Extraction. As the first part of the query-relevant snippet, indexed fields matching any keywords are presented in a flattened key-value format. Following conventional Web search engines, CKGSE highlights all matched parts in each field. An example is presented in [Figure 6: see original paper].

Label Index. To support query-relevant content snippet generation, a label index is constructed for each KG. This index records a mapping from each word to the IRIs in the KG whose textual form contains that word. It is used to measure the query relevance of each triple by counting the number of keywords contained in its textual form—that is, in the textual form of its subject, predicate, or object.

Content Loading. As a preprocessing step for content snippet generation, all triples of each KG in the results list are extracted from the database and loaded into memory. Based on the Label Index, the query relevance of each triple is computed during the loading process, along with other content representativeness measures such as the relative frequencies of the class or property instantiated in the triple.

Query-relevant Snippet Generation. CKGSE adopts KSD [37] to extract a snippet from the original KG content. By treating each triple as an element set containing query keywords, classes, properties, and entities, and assigning an importance weight to each element, KSD formulates snippet generation as a weighted maximum coverage problem. It aims to select at most k triples to

maximize the total weight of covered elements. We implement it using a greedy strategy: in each iteration, a triple containing the largest weight of uncovered elements is selected until reaching the pre-defined size limit k or until all elements are covered. Here we set k = 10. Details of KSD are introduced in [37].

4.4 Content Profiling

For each KG selected by the user, in addition to presenting metadata information as existing systems do (shown in [Figure 7: see original paper]), this component presents an offline-computed profile to illustrate its content, including a quality profile, a statistical summary, an abstractive summary, and an extractive summary.

Quality Profile. The quality profile aims to present a set of quantitative quality assessments as signals from both metadata and content to facilitate user judgment. Since quality metrics for KGs have been extensively studied [23], in CKGSE we select and implement six metrics that are relatively relevant to our search contexts and require no external resources. As shown in , three metrics relate to metadata: Availability measures the extent to which KG resources can be obtained; Licensing indicates whether the KG is under a specific license; Timeliness shows whether the KG has been recently updated. The other three metrics relate to content: Intra-KG interlinking represents the proportion of non-isolated entities in the KG; Inter-KG interlinking indicates how frequently entities in the KG are linked with entities in other KGs; Understandability reflects whether most entities in the KG have a human-readable textual form. An example of the quality profile is shown in [Figure 8: see original paper]. For each KG, all quality metrics are computed offline and stored in the database.

Statistical Summary Generation. The statistical summary consists of overall statistics for the selected KG and close-up views of different types of elements contained in the KG. Basic statistics include counts such as the number of triples and entities, presented to the user as a table (shown in [Figure 9: see original paper]). In addition, as summaries of KG elements, we implement distribution by relative frequencies for classes and properties as a pie chart (shown in [Figure 10: see original paper]). For top-ranked entities by PageRank score computed on the original KG, we present them as central content of the KG. Additionally, we visualize three tag clouds for classes, properties, and entities, where classes and properties are weighted by frequencies and entities are weighted by PageRank scores. An example entity tag cloud is shown in [Figure 11: see original paper].

Abstractive Summary Generation. Motivated by pattern mining techniques, CKGSE incorporates frequent entity description patterns (EDPs) [31, 32] into the KG profiling component. In a KG, each entity is described by a set of classes and properties—that is, schema-level elements. Each EDP captures a common description pattern shared among a set of entities. Therefore, frequent EDPs can be regarded as a pattern-level abstractive summary. Each consists of

a set of forward properties, backward properties, and classes that describe a set of entities. In the profile page, each EDP is presented as a node-link diagram, as shown in [Figure 12: see original paper].

Extractive Summary Generation. Complementing the abstractive schema-level summary represented by frequent EDPs, CKGSE also incorporates an extractive summary to directly exemplify KG content. To extract a connected sub-KG with maximum content representativeness, IlluSnip [33, 34] formulates triple selection as a combinatorial optimization problem. It defines content representativeness as coverage of the most frequently instantiated classes, properties, and the most important entities with the highest PageRank scores. For each KG, such a selected sub-KG can be viewed as an extractive summary. A greedy algorithm is applied in IlluSnip to generate a sub-KG containing at most k' triples. We set k' = 20. The result is presented as a node-link diagram on the profile page, as shown in [Figure 13: see original paper].

Summary Index. The statistical, abstractive, and extractive summaries for each KG are all computed offline and stored in a summary index.

4.5 Content Browsing

Beyond metadata and profile pages, CKGSE also enables users to interactively explore the selected KG by selecting and viewing entities.

Entity Index. For each KG, an entity index is built to support efficient entity filtering, containing two parts that map classes and properties to their entity instances. This index is also implemented using Lucene.

Faceted Entity Search. As shown in [Figure 14: see original paper], supported by the entity index, users are provided with a panel to select classes and properties instantiated in the KG. The selected classes and properties are then used as filters with AND as the Boolean operator to filter entities in the KG. Filtered entities are returned as a list, and each can be browsed by clicking.

Entity Browsing. For each filtered entity, CKGSE retrieves and presents all triples describing it as a node-link diagram, depicting the neighborhood of this entity in the original KG. By switching between entities in this diagram, users can explore any part of the KG according to their interests.

5. EXPERIMENTS

We implemented a CKGSE prototype on an Intel Xeon E7-4820 (2.0GHz) with 100GB memory for JVM. Our experiments primarily focused on evaluating CKGSE's practicality. We also conducted a case study comparing CKGSE with the search service provided by the current version of OpenKG.CN to demonstrate the usefulness of CKGSE' s unique features.

5.1.1 KG Crawling and Storage

shows statistics for the 40 KGs crawled from OpenKG.CN. Size and schema vary greatly among them. Some KGs are very large—the largest KG has a dump file size exceeding 28GB, parsed into more than 150 million triples. Most KGs have multiple dump files, with the largest number of dump files for a single KG reaching 46. Among 185 available dump file records, a total of 178 were successfully downloaded, parsed, and stored.

The time complexity of the parsing process is O(#Triples). As shown in , parsing all 40 KGs was affordably completed within one day. Among them, 10% (4/40) were very large and required over one hour to parse. We observed that the largest KG (http://www.openkg.cn/dataset/zhishi-me-dump) required 8 hours for parsing. Based on the parsed KGs, the other indexes were also built within acceptable time, with larger KGs generally requiring longer indexing time. Therefore, the overall time cost for parsing and indexing KGs is acceptable in practice.

shows the runtime of offline computation (in hours).

The space complexity of all indexes is O(#Triples). presents disk usage. In practice, the total size of the triple store and all indexes is smaller than the size of the original dump files, demonstrating CKGSE' s disk-use efficiency and practicality.

5.1.2 Content-based Keyword Search

As shown in , CKGSE spent 5.1 hours building an inverted index for all 40 KGs to support keyword search over both metadata and KG content. Note that we did not build them in parallel, which would have been much faster. According to , the inverted index only occupies 1.8GB, which is relatively small and affordable.

5.1.3 Content-based Snippet Generation

In addition to indexed fields, CKGSE also spent 2.5 hours constructing a label index to support query-relevant snippet generation, as shown in . Approximately half of this time (1.1 hours) was consumed by the largest KG. For the result index, which occupies 9.7GB (), about half is for the largest KG.

The time complexity of KSD is O(k $\cdot$ #Triples), where k is a pre-defined size limit. To evaluate the performance of online snippet generation by KSD, we created 10 keyword queries containing 1-5 keywords. We then retrieved the top-5 relevant KGs for each query and recorded KSD' s runtime for each of the 50 retrieved KGs. [Figure 2: see original paper] shows the complementary cumulative distribution of runtime across all these KGs. The median runtime is only 1 second, while for 12% (6/50) of KGs the runtime exceeded 10 seconds. This suggests that while KSD is fast enough for most KGs, further optimization is still needed, particularly for large KGs.

5.1.4 Content Profiling

As shown in , the content profiling component has two major time costs: for the quality profile and the summary index.

For quality profiles, each metadata-related metric (i.e., Availability, Licensing, and Timeliness) has time complexity O(1), while each content-related metric (i.e., intra-KG interlinking, inter-KG interlinking, and understandability) requires time complexity O(#Triples). CKGSE spent 1.6 hours computing values for six metrics across all KGs, which is relatively short.

[Figure 3: see original paper] shows the complementary cumulative distribution of quality scores across the 40 KGs. According to the three metadata quality metrics, most KGs have dump files available for download and parsing. For the licensing score, all KGs from OpenKG.CN have specific licenses recorded in metadata, thus each has a licensing score of 1 (for this reason, we do not show its cumulative distribution in the figure). The timeliness score measures how up-to-date a KG remains after its creation. However, over half of the KGs were never updated after submission, or their update time was missing from metadata, rendering the timeliness score undefined and treated as zero.

For content quality scores, the relatively high average intra-KG interlinking score suggests that in most KGs, entities are typically linked to others rather than isolated. Conversely, inter-KG interlinking scores are generally low for these KGs. For understandability, less than half of the KGs use rdfs:label to describe entities with human-readable tags, though we observed that some KGs use other properties with similar meaning, such as http://cndbpedia/ontology/实体名称 (entity name). Since such properties vary across KGs, we did not consider them.

The summary index contains element statistics, abstractive summary results, and extractive summary results for all 40 KGs. It uses only 165MB for storage () but required 34 hours for computation (). Of these 34 hours, CKGSE spent approximately 3 hours preparing statistical summaries, including 2 hours computing PageRank to identify central entities. Most of the remaining time was spent generating extractive summaries using IlluSnip. We used an anytime version of IlluSnip [34], allowing it to iteratively find better summaries within a maximum of 2 hours per KG. If needed, one could adjust to a smaller time limit to trade off between result quality and generation time. In our experiments, IlluSnip' s median runtime was 31 seconds. By comparison, generating abstractive summaries (i.e., EDPs) was much faster, even comparable to KSD, as shown in [Figure 2: see original paper].

5.1.5 Content Browsing

CKGSE spent 33 hours creating an entity index to support faceted entity search (), although the index was only 1.1GB upon completion (). We observed that significant room remains for improving the performance of our current imple-

mentation of this index, such as using better index structures and/or more efficient algorithms.

5.2 Case Study

We compared CKGSE' s performance with OpenKG.CN (assessed on October 28, 2021) through a case study.

5.2.1 Keyword Search As shown in [Figure 4: see original paper], given the query "哈利波特人物关系" ("relationships between characters in Harry Potter"), both OpenKG.CN and CKGSE successfully find the target KG, since both query keywords match the KG' s metadata.

However, for the query "格兰芬多人物关系" ("relationships between characters about Gryffindor"), where "格兰芬多" ("Gryffindor") refers to an entity not contained in the target KG' s metadata, only CKGSE finds this KG, as shown in [Figure 5: see original paper]. Thanks to content-based keyword search whose inverted index covers both metadata and KG content, CKGSE distinguishes itself from existing systems.

On CKGSE' s results page, each returned KG is presented with indexed fields including both metadata and content. In each field, query-relevant words are highlighted in red, as shown in [FIGURE:4(b)] and [FIGURE:5(b)].

5.2.2 Query-relevant Snippet Generation On search results pages, OpenKG.CN and other existing systems only display some metadata for each top-ranked KG. CKGSE goes further by providing a query-relevant snippet that includes both indexed fields and an extracted sub-KG, as shown in [Figure 6: see original paper].

The generation of this sub-KG is biased toward the keyword query—for example, containing the "格兰芬多"("Gryffindor") entity mentioned in the query. Therefore, it helps users quickly and accurately judge the relevance of the underlying KG to the query, even before browsing its full content, which could be time-consuming.

5.2.3 Profiling and Browsing When a KG is selected, in addition to the metadata information typically shown by OpenKG.CN and other existing systems ([Figure 7: see original paper]), CKGSE further presents a quality profile and content summaries for the KG.

Currently, each KG' s quality profile is presented as a table of metric values in CKGSE. As shown in [Figure 8: see original paper], each metric corresponds to an aspect relevant to search. This quality profile serves not only as quality signals for users but could also inform the search process. As a future direction, we will consider incorporating quality metrics into the KG ranking function.

The statistical summary consists of basic KG statistics such as triple count ([Figure 9: see original paper]) and element-level content distributions with

visualizations. [Figure 10: see original paper] shows the distribution of all properties instantiated in the KG, visualized as a pie chart. An entity tag cloud weighted by PageRank scores computed over the KG is visualized in [Figure 11: see original paper]. Such statistical summaries provide users with a brief overview of KG content.

The abstractive summary in [Figure 12: see original paper] describes the KG at the pattern level by presenting the most frequent EDPs in the KG, showing how entities are described—that is, by which combinations of classes and properties.

The extractive summary in [Figure 13: see original paper] presents an extracted sub-KG that differs from the snippet on the search results page. This sub-KG is query-independent but illustrates the most frequent classes and properties in the KG with a few concrete entities and triples. Compared to metadata and statistics, our content summaries provide a distinct close-up view of KG content, thus assisting users in comprehending the KG and judging its relevance before downloading it.

Finally, as shown in [Figure 14: see original paper], CKGSE allows users to interactively browse entities in the KG. Users can select classes and properties to filter entities. For each filtered entity, all its triples are visualized as a node-link diagram. With this simple yet effective browsing interface, many users do not need additional tools for KG browsing and can easily investigate KG content.

5.3 User Study

In addition to practicability analysis and case study, we conducted a user study to verify CKGSE' s usefulness for search result relevance judgment and KG content comprehension by comparing it with OpenKG.CN. We recruited 20 computer science students with necessary background knowledge and research experience in KGs via a mailing list.

5.3.1 Design and Process Following the general KG search process, each participant was first required to formulate a specific data need to find a target KG. The participant was then allowed to search using OpenKG.CN and CKGSE separately for the target KG. After viewing returned results and judging each KG' s relevance, the participant rated the usefulness of the two systems separately on a 1–5 scale, including how useful the system was for (1) search result relevance judgment and (2) KG content comprehension. We also asked participants to select the most useful system component for aspects (1) and (2).

5.3.2 Result Analysis User-rated usefulness results are summarized in . A paired two-sample t-test showed that CKGSE received significantly higher ratings than OpenKG.CN ($p < 0.01$) for both search result relevance judgment and KG content comprehension. Most participants (65%) gave higher ratings to CKGSE for helping them judge KG relevance, and all agreed that CKGSE

performed better than OpenKG.CN in helping them understand the main content of KGs.

shows user-rated usefulness for search result relevance judgment and KG content comprehension.

According to participants' selections of most useful components, 10 participants (50%) who rated OpenKG.CN 3 or higher for search result relevance judgment generally relied on title and description metadata to filter out irrelevant KGs. For CKGSE, all 20 participants (100%) rated over 3 for relevance judgment. Apart from title and description, 10 participants (50%) also selected the query-relevant snippet as especially useful for their judgment.

For helping users understand KG main content, OpenKG.CN received relatively low ratings with an average of 2.15, as it could not provide detailed KG elements or patterns to exemplify content. Compared to OpenKG.CN, CKGSE received much better usefulness ratings for comprehension with an average of 4.35. Participants selected the statistical summary (50%) and the content browsing component (25%) as most helpful for quickly learning about representative KG elements.

We also interviewed participants about their likes and dislikes of the two systems. Most participants (85%) preferred CKGSE for providing more detailed KG views, while some mentioned limitations, such as the need to speed up snippet generation and improve visualization methods for summaries.

6. CONCLUSION

In this paper, we present CKGSE, one of the first content-based search engines for open KGs. Complementing existing systems that only consider KG metadata, CKGSE incorporates content-level element fields such as classes and properties into the inverted index, enabling it to handle queries referencing KG content. To facilitate user relevance judgment, CKGSE provides a query-relevant snippet for each KG on the search results page. Beyond metadata information, CKGSE uses a quality profile, content-based summaries, and browsing capabilities to comprehensively provide users with a close-up view of the KG. We implemented a prototype using KGs crawled from OpenKG.CN. Our preliminary experimental results demonstrate the practicality and usability of this new paradigm for KG search, though system performance could be further optimized.

Our experiments also reveal several CKGSE limitations to address in the future. First, we will focus on improving efficiency for processing large KGs and enhance browsing and presentation of large KGs using entity summarization techniques [38, 39]. Additionally, based on user feedback, we will improve overall system performance and design better visualization methods for each component.

ACKNOWLEDGEMENTS
This work was supported by the National Science Foundation of China (No. 62072224).

REFERENCES

[1] Deng, C., et al.: GAKG: A multimodal geoscience academic knowledge graph. In: The 30th ACM International Conference on Information and Knowledge Management (CIKM 2021), pp. 4445–4454 (2021)
[2] Dsouza, A., et al.: Worldkg: A world-scale geographic knowledge graph. In: The 30th ACM International Conference on Information and Knowledge Management (CIKM 2021), pp. 4475–4484 (2021)
[3] Schindler, D., et al.: Somesci- A 5 star open data gold standard knowledge graph of software mentions in scientific articles. In: The 30th ACM International Conference on Information and Knowledge Management (CIKM 2021), pp. 4574–4583 (2021)
[4] Shen, Y., et al.: CKGG: A Chinese knowledge graph for high-school geography education and beyond. In: 2021 International Semantic Web Conference, pp. 429–445 (2021)
[5] Larmande, P., Todorov, K.: Agrold: A knowledge graph for the plant sciences. In: ISWC 2021: International Semantic Web Conference, pp. 496–510 (2021)
[6] Dimitrov, D., et al.: Tweetscov19 - A knowledge base of semantically annotated tweets about the COVID-19 pandemic. In: The 29th ACM International Conference on Information and Knowledge Management (CIKM2020), pp. 2991–2998 (2020)
[7] Walsh, B., Mohamed, S.K., Novácek, V.: Biokg: A knowledge graph for relational learning on biological data. In: The 29th ACM International Conference on Information and Knowledge Management (CIKM2020), pp. 3173–3180 (2020)
[8] Dessì, D., et al.: AI-KG: An automatically generated knowledge graph of artificial intelligence. In: 2020 International Semantic Web Conference (Part II), pp. 127–143 (2020)
[9] McCusker, J.P., et al.: Nanomine: A knowledge graph for nanocomposite materials science. In: 2020 International Semantic Web Conference (Part II), pp. 144–159 (2020)
[10] Michel, F., et al.: Covid-on-the-web: Knowledge graph and services to advance COVID-19 research. In: 2020 International Semantic Web Conference (Part II), pp. 294–310 (2020)
[11] Steenwinckel, B., et al.: Facilitating the analysis of COVID-19 literature through a knowledge graph. In: 2020 International Semantic Web Conference (Part II), pp. 344–357 (2020)
[12] Neumaier, S., Umbrich, J., Polleres, A.: Automated quality assessment of metadata across open data portals. ACM Journal of Data and Information Quality 8(1), Article No. 2 (2016)
[13] Brickley, D., Burgess, M., Noy, N.F.: Google dataset search: Building a search engine for datasets in an open web ecosystem. In: The World Wide Web Conference (WWW ' 19), pp. 1365–1375 (2019)
[14] Pietriga, E., et al.: Browsing linked data catalogs with LODAtlas. In: 2018 International Semantic Web Conference (Part II), pp. 137–153 (2018)

[15] Chen, J., et al.: Towards more usable dataset search: From query characterization to snippet generation. In: The 28th ACM International Conference on Information and Knowledge Management (CIKM2019), pp. 2445–2448 (2019)
[16] Degbelo, A.: Open data user needs: A preliminary synthesis. In: WWW ' 20: Companion Proceedings of the Web Conference 2020, pp. 834–839 (2020)
[17] Chapman, A., et al.: Dataset search: A survey. The VLDB Journal 29, 251–272 (2020)
[18] Ellefi, M.B., et al.: RDF dataset profiling—a survey of features, methods, vocabularies and applications. Semantic Web 9(5), 677–705 (2018)
[19] Wang, X., et al.: A framework for evaluating snippet generation for dataset search. In: 2019 International Semantic Web Conference (Part I), pp. 680–697 (2019)
[20] Wang, X., et al.: Content-based open knowledge graph search: A preliminary study with openkg.cn. In: China Conference on Knowledge Graph and Semantic Computing (CCKS 2021), pp. 104–115 (2021)
[21] Dutkowski, S., Schramm, A.: Duplicate evaluation - position paper by Fraunhofer Fokus. In: SDSVoc 2016, pp. 1–4 (2016)
[22] Koesten, L., et al.: Everything you always wanted to know about a dataset: Studies in data summarisation. International Journal of Human-Computer Studies 135, Article No. 102367 (2020)
[23] Zaveri, A., et al.: Quality assessment for linked data: A survey. Semantic Web 7(1), 63–93 (2016)
[24] Auer, S., et al.: LODStats –An extensible framework for high-performance dataset analytics. In: Knowledge Engineering and Knowledge Management (EKAW 2012), pp. 353–362 (2012)
[25] Cebiric, S., et al.: Summarizing semantic graphs: A survey. The VLDB Journal 28(3), 295–327 (2019)
[26] Song, Q., et al.: Mining summaries for knowledge graph search. IEEE Transactions on Knowledge and Data Engineering 30(10), 1887–1900 (2018)
[27] Khatchadourian, S., Consens, M.P.: ExpLOD: Summary-based exploration of interlinking and RDF usage in the linked open data cloud. In: Knowledge Engineering and Knowledge Management (EKAW 2010, Part II), pp. 272–287 (2010)
[28] Cheng, G., Jin, C., Qu, Y.: HIEDS: A generic and efficient approach to hierarchical dataset summarization. In: The International Joint Conference on Artificial Intelligence (IJCAI 2016), pp. 3705–3711 (2016)
[29] Zneika, M., et al.: RDF graph summarization based on approximate patterns. In: International Workshop on Information Search, Integration, and Personalization (ISIP 2015), pp. 69–87 (2015)
[30] Zneika, M., et al.: Summarizing linked data RDF graphs using approximate graph pattern mining. In: The 19th International Conference on Extending Database Technology (EDBT 2016), pp. 684–685 (2016)
[31] Wang, X., et al.: BANDAR: Benchmarking snippet generation algorithms for (RDF) dataset search. IEEE Transactions on Knowledge and Data Engineering (2021). Available at: https://ieeexplore.ieee.org/document/9477056. Accessed 20 January 2021

[32] Wang, X., et al.: PCSG: Pattern-coverage snippet generation for RDF datasets. In: 2021 International Semantic Web Conference, pp. 3–20 (2021)
[33] Cheng, G., et al.: Generating illustrative snippets for open data on the Web. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM 2017), pp. 151–159 (2017)
[34] Liu, D., et al.: Fast and practical snippet generation for RDF datasets. ACM Transactions on the Web13(4), Article No. 19 (2019)
[35] Tian, Y., Hankins, R.A., Patel, J.M.: Efficient aggregation for graph summarization. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pp. 567–580 (2008)
[36] Campinas, S., Delbru, R., Tummarello, G.: Efficiency and precision trade-offs in graph summary algorithms. In: Proceedings of the 17th International Database Engineering & Applications Symposium (IDEAS 2013), pp. 38–47 (2013)
[37] Wang, X., Cheng, G., Kharlamov, E.: Towards multi-facet snippets for dataset search. In: PROFLILES & SemEx 2019, pp. 1–6 (2019)
[38] Liu, Q., et al.: Entity summarization: State of the art and future challenges. Journal of Web Semantics 69, Article No. 100647 (2021)
[39] Liu, Q., et al.: Entity summarization with user feedback. In: European Semantic Web Conference (ESWC 2020), pp. 376–392 (2020)

AUTHOR BIOGRAPHY

Xiaxia Wang is an M.S. student in the Department of Computer Science and Technology at Nanjing University. She received her B.S. degree in Information and Computing Science from Nanjing University of Aeronautics and Astronautics in 2019. Her current research interests include data summarization and semantic search. She has published papers in journals such as IEEE Transactions on Knowledge and Data Engineering and conferences including the International Semantic Web Conference (ISWC) and ACM International Conference on Information and Knowledge Management (CIKM).

Tengteng Lin is an M.S. student in the Department of Computer Science and Technology at Nanjing University. She received her B.S. degree in Software Engineering from Shandong University in 2020. Her research interests include dataset retrieval and ranking.

Weiqing Luo is a B.S. student in the Department of Computer Science and Technology at Nanjing University. His research interests include dataset retrieval and ranking.

Gong Cheng is an Associate Professor in the Department of Computer Science and Technology at Nanjing University. He received his Ph.D. degree in Computer Software and Theory from Southeast University in 2010. His research interests include Semantic Web and knowledge graphs, particularly knowledge graph search and question answering. He has published more than 50 papers in major venues in these areas, including journals such as IEEE Transactions

on Knowledge and Data Engineering and conferences including the World Wide Web Conference (WWW) and International Semantic Web Conference (ISWC).

Yuzhong Qu is a Professor in the Department of Computer Science and Technology at Nanjing University. He received his Ph.D. degree in Computer Software from Nanjing University in 1995. His research interests include Semantic Web, question answering, and novel software technology for the Web. He has published more than 80 papers in major venues in these areas, including journals such as Journal of Web Semantics and conferences including the World Wide Web Conference (WWW) and International Semantic Web Conference (ISWC).

Note: Figure translations are in progress. See original paper for figures.