Incremental Entity Summarization with Formal Concept Analysis

Erhe Yang, Fei Hao, Yixuan Yang, Carmen De Maio *Member, IEEE*, Aziz Nasridinov, Geyong Min *Member, IEEE*, and Laurence T. Yang *Fellow, IEEE*

Abstract—Knowledge graph describes entities by numerous RDF data (subject-predicate-object triples), which has been widely applied in various fields, such as artificial intelligence, Semantic Web, entity summarization. With time elapses, the continuously increasing RDF descriptions of entity lead to information overload and further cause people confused. With this backdrop, automatic entity summarization has received much attention in recent years, aiming to select the most concise and most typical facts that depict an entity in brief from lengthy RDF data. As new descriptions of entity are continually coming, creating a compact summary of entity quickly from a lengthy knowledge graph is challenging. To address this problem, this paper firstly formulates the problem and proposes a novel approach of Incremental Entity Summarization by leveraging Formal Concept Analysis (FCA), called IES-FCA. Additionally, we not only prove the rationality of our suggested method mathematically, but also carry out extensive experiments using two real-world datasets. The experimental results demonstrate that the proposed method IES-FCA can save about 8.7% of time consumption for all entities than the non-incremental entity summarization approach KAFCA at best. As for the effectiveness, IES-FCA outperforms the state-of-the-art algorithms in terms of $F1-measure$, MAP, and $NDCG$.

Index Terms—Knowledge Graph, Entity Summarization, Formal Concept Analysis, Incremental Algorithm

1 INTRODUCTION

Knowledge Graph (KG), as one of the most important infrastructures of artificial intelligence, has received much attention in both academia [1]–[4] and industrial fields [5]–[8]. The mainstream large-scale knowledge graphs are all publicly available on the web, such as Wikidata [9], DBpedia [10], YAGO [11], [12], LinkMDB [13]. Entities in these knowledge graphs are described by the Resource Description Framework (RDF), which employs subject-predicate-object triples to describe all the resources and their relationships on the web. Nevertheless, people often suffer from information overload when searching through a considerable increment of RDF triples in the knowledge graph. For instance, the latest English version of DBpedia includes 1.7 billion RDF triples for 6.6 million entities, where each entity has 258 descriptions on average [14]. Thus, it is essential to provide a concise summary of the entity to end-users. In such a scenario, the technique of entity summarization has emerged and become a hot topic in recent years.

Entity summarization aims to provide concise information of the entity in the knowledge graph to depict the original lengthy entity. Most existing studies on entity summarization focus on one snapshot of entities in the knowledge graph while ignoring many constant descriptions of entities, including newly added descriptions. When the knowledge graph is complex, the efficiency of entity summarization can be low. In addition, the entities in the knowledge graph are constantly changing. Hence, recomputation of entity summarization every time can be time and computational resources consuming, especially when the knowledge graph is complex. To this end, we aim to improve the efficiency of entity summarization and make full use of computational resources using incremental entity summarization. To better understand the application of incremental entity summarization, Fig. 1 shows a motivating example.

Fig. 1. A motivating example.

Motivating Example. Fig. 1 shows the entity cards of the entities $Bill\quad Gates$ and $Mark\quad Zuckerberg$ searched by Google. The entities in entity cards are from Google KG and constructed with numerous RDF triples. The representative descriptions (i.e., entity summarization) of $Bill\quad Gates$ and

E. Yang, F. Hao and Y. Yang are with Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, China & School of Computer Science, Shaanxi Normal University, Xi'an, China. F. Hao is also with the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, U.K.
C. De Maio is with the Dept. of Information Eng., Electrical Eng. and Applied Mathematics, University of Salerno, Fisciano, Italy.
A. Nasridinov is with the Department of Computer Science, Chungbuk National University, Cheongju, Korea
G. Min the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, U.K.
L.T. Yang is with the Department of Computer Science, St. Francis Xavier University, Antigonish, Canada
**Corresponding author: Fei Hao; Email:feehao@gmail.com.*
*Manuscript received****, ****; revised ****, ****.*

$Mark\ \ Zuckerberg$ are selected from numerous descriptions in Google KG and displayed in the entity card panel. It is important to note that the descriptions of entities constantly change. For instance, the value of the net worth is updated yearly. To guarantee the summarization of entity is updated in time, it is necessary to improve the efficiency of entity summarization via incremental entity summarization.

Applications. The incremental entity summarization can be applied in various applications.

Application 1: Search Engine Optimization. As mentioned in the motivating example, the entity cards in search engine can provide a brief summary of the entity in KG. The incremental entity summarization can boost the efficiency of the entity cards acquisition, although the descriptions of entity are always massive and ever-changing.

Application 2: Question Answering Optimization. For the question answering based on the KG, the incremental entity summarization can be applied to reduce the size of KG. To be more concrete, the trivial triples of entity in the KG can be removed firstly by utilizing the incremental entity summarization, which can significantly improve the efficiency of question answering in the pruned KG.

Formal Concept Analysis (FCA) is a powerful data analysis method, which has been extensively applied in many ICT fields, such as software engineering [15], [16], data mining [17], [18], and information retrieval [19], to cite but a few. FCA performs well in analyzing the binary tabular data [20]. Considering that the predicates and objects in the RDF data for an entity can be converted into the form of binary tabular, it is reasonable to assume that FCA can be applied to entity summarization. For entity summarization using FCA, Kim et al. [14] proposed KAFCA, which can obtain the ranked RDF triples by the weights of extents of concepts in concept lattice. The experiment results demonstrate that KAFCA outperforms the state-of-the-art entity summarization methods.

Challenges. Due to the dynamic nature and massive scale of knowledge graphs, the efficiency of KAFCA is limited. To obtain a concise summarization of the entity, KAFCA considers the original RDF triples and the newly added RDF triples as a whole when building concept lattice. Considering that the construction of concept lattice in KAFCA is non-incremental, this method can be time-consuming, especially when the RDF entity descriptions are complex. Additionally, KAFCA considers giving the same scores to the concepts with the same cardinality of extents, which is unreasonable as the cardinality of the corresponding intents are also influential to the significance of concepts.

To tackle these challenges, we propose an incremental entity summarization approach to improve the efficiency of entity summarization with FCA. Furthermore, we improved the ranking algorithm by considering the $importance$, $redundancy$, and $uniqueness$ of triples for obtaining better summarization results. The main contributions of this paper are summarized as follows:

- **Formalization of Incremental Entity Summarization**: We pioneer the formalization of incremental entity summarization with FCA. Incremental entity summarization in this paper is based on FCA used to analyze the relationship between predicates and objects in RDF triples of the entity in the knowledge graph. Our main idea is to apply an incremental construction algorithm of concept lattice to entity summarization and rank the RDF triples by introducing the $importance$, $redundancy$, and $uniqueness$ of triples based on the hierarchy of concepts in concept lattice.
- **Incremental Entity Summarization Approach**: To address the low efficiency of KAFCA, this paper proposes IES-FCA, an original incremental entity summarization approach with FCA. The approach is applicable for the streaming data environment where the amount of data is constantly increasing and the order of data can not affect the summarization results. Firstly, original and newly added entity descriptions are constructed into formal contexts (K_1, K_2), and then these descriptions are built into concept lattices (C_1, C_2). Secondly, we take the intersection of extents of C_1 and C_2, based on which the final concept lattice can be built. Finally, we rank the RDF triples with the hierarchy of extents and intents in concept lattice and output the compact entity summary.
- **Improved Ranking Algorithm for Entity Summarization**: To address the shortage of KAFCA in ranking algorithm, our proposed approach IES-FCA modifies the scoring algorithm for the RDF triples. Concretely, we assign different scores for the concepts that has extents with the same cardinality while these scores in KAFCA are the same. In addition, the $importance$, $redundancy$, and $uniqueness$ of triples are considered in the ranking process, which guarantees the importance, compactness, and uniqueness of the summary results.
- **Evaluation**: We conduct extensive experiments to compare the proposed method with KAFCA and other state-of-the-art approaches on two real-world datasets. The experiment results demonstrate that our proposed method performs better than KAFCA. Specifically, the efficiency of entity summarization can be improved up to 8.7% for all entities. Further, for the entity whose RDF descriptions consist of the largest number of predicates, the summary efficiency can be improved up to 67%. Additionally, the effectiveness of IES-FCA has been proved compared with other state-of-the-art algorithms in terms of $F1-measure$, MAP (Mean Average Precision), and $NDCG$ (Normalized Discounted Cumulative Gain). The weighting tests and ablation study verified the rationality and effectiveness of the proposed ranking algorithm. Concretely, the results of $F1-measure$ improvement on ESBM (Entity Summarization Benchmark) v1.0 dataset range from 5.84% to 32.14% and the results of MAP improvement can reach to 17.87%. For the ESBM v1.2 dataset, the results of $F1-measure$ improvement and $NDCG$ improvement can be raised up to 4.68% and 2.41%, respectively.

The rest of this paper is organized as follows: Section 2 introduces the related work. Then, the problem formulation is presented in Section 3. Section 4 elaborates our novel approach. The experimental details are described and experimental results are discussed in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

Entity summarization provides concise information of the entity in the knowledge graph using various ranking algorithms. RELIN [21] ranks triples of the entity by adopting a variant of the random surfer model, which is based on non-uniform probability distributions and applies informativeness to the traditional relatedness-based centrality measure. In order to reduce the redundancy among the returned items and lower the risk of no item that people are interested in is returned, DIVERSUM [22] introduced the concept of diversity for the results of entity summarization. Gunaratna et al. [23] proposed a novel diversity-aware entity summarization approach, called FACES, which takes into account the dimensions of diversity, uniqueness, and popularity of descriptions for each entity. Their approach selects representative facts to form a concise and comprehensive summary using the clustering algorithm called Cobweb. FACES-E [4] is an extension of FACES that utilizes both object and data type properties to generate entity summarization. Xu et al. [24] proposed CD that considers the characteristic and diverse feature selection as a binary quadratic knapsack problem, in which they apply information theory into the feature characterizing. LinkSUM [25] is a generic relevance-centric summarization method that focuses more on objects rather than diversity of properties. Based on FCA, KAFCA [14] converts a knowledge graph into a formal concept employing the tokenized objects and predicates in RDF triples, and obtains the ranked RDF triples according to the weights of all predicate-object pairs. BAFREC [26] splits all facts of entities into categories and then rates each category using a specific metric, which balances the frequency and rarity metrics for obtaining summaries on the entity. Wei et al. proposed an LDA-based model MPSUM [27], which extends a probabilistic topic model by integrating the idea of predicate-uniqueness and object-importance for ranking triples. ES-LDA [28] is a probabilistic topic model that applies prior knowledge to statistical learning techniques for entity summarization, which selects top-k triples according to the probability distributions of triples. Wei et al. [29] presented a neural network model ESA and applied the supervised attention mechanism with BiLSTM to entity summarization task, which ranks facts by attention weights for the entity.

Most of the above-mentioned approaches of entity summarization are non-incremental, and thus the efficiency of entity summarization is low when the knowledge graph is complex. In addition, the entities in the knowledge graph change constantly and the corresponding entity summary should be created timely. Accordingly, it is necessary to enhance the efficiency of entity summarization. For this, the previously mentioned FACES [23] adopts an incremental approach using a modified incremental hierarchical conceptual clustering algorithm. FACES adapted an incremental hierarchical conceptual clustering algorithm named Cobweb for partitioning feature set, which can cluster items based on the probability of attribute-value pairs for the items. Incremental entity summarization can be regarded as one type of dynamic entity summarization with focus on the efficiency improvement rather than a comprehensive description of the entity from the perspective of time evolution. The literature [30] viewed dynamic entity summarization for entity cards as the query-dependent nature of the generated summaries and formulated two specific subtasks (i.e., fact ranking and summary generation) to address the problem. Tasmin et al. [31] envisioned an approach to create a summarization graph capturing the temporal evolution of entities across different versions of a knowledge graph. They converted different versions of a knowledge graph into RDF molecules and adopted FCA to these RDF molecules for generating the summary information.

3 Problem Formulation

This section first formally defines fundamental definitions about entity summarization and FCA, which has been depicted clearly in [28] and [32], respectively. Then, the problem of incremental entity summarization is formulated.

3.1 Entity Summarization

Entities in the knowledge graph are described by various RDF triples. Entity summarization simplifies the lengthy description of entity and provides a concise description.

Definition 1. [28] (**Entity Summarization**) Given an entity e and a positive integer k, a summarization of the entity e, denoted as $Sum(e,k)$, is the top-k subset of all predicates and corresponding objects that are most relevant to that entity.

3.2 Formal Concept Analysis

For the sake of simplicity, we only sketch the key notions of FCA. More preliminaries of FCA can be found in [20], [32]. To avoid confusion, notice that O and P represent the set of objects (denote objects in the formal context) and the set of predicates (denote attributes in the formal context) in RDF triples, respectively.

To better express the core of the work, we propose the definition of Tokenized Formal Context by modifying the basic definition of Formal Context [32] as follows:

Definition 2. (**Tokenized Formal Context**) A tokenized formal context is organized as a triple $K = (O, P, I)$, where $O = \{o_1, o_2, \cdots, o_n\}$ is the set of objects, $P = \{p_1, p_2, \cdots, p_m\}$ is the set of attributes, and I is composed of the direct relationship I' between O and P and underlying relationship I'' between tokenized objects set O' and P. Concretely, if o_i and p_i are object and predicate in a RDF triple respectively, we assume that there is a direct relationship: $(o_i, p_i) \in I'$. For two pairs of the objects and predicates (o_i, p_i) and (o_j, p_j), if o_i and o_j share the same terms by tokenizing the objects, we assume that there is a underlying relationship: $(o_i, p_j) \in I''$, $(o_j, p_i) \in I''$. Let $I = I' \bigcup I''$, $I \subseteq (O \bigcup O') \times P$, $(o_i, p_j) \in I$ denotes that object o_i has the relationship with p_j, and $(o_i, p_j) \notin I$ denotes that object o_i does not have the relationship with p_j, where $o_i \in O$, $p_j \in P$.

Here, “1” and “0” denote $(o_i, p_j) \in I$ and $(o_i, p_j) \notin I$, respectively.

$$\begin{cases} 1 & (o_i, p_j) \in I \\ 0 & (o_i, p_j) \notin I \end{cases}$$

For the sake of simplicity, we used terms Tokenized Formal Context and Formal Context interchangeably in the remainder of this paper. Based on the proposed Tokenized Formal Context, the following operators for building concepts are defined:

Definition 3. [32] For a formal context $K = (O, P, I)$, the operators $\uparrow$ and $\downarrow$ on $X \subseteq O$ and $B \subseteq P$ are respectively defined as:

$$X^{\uparrow} = \{p \in P| \quad \forall o \in X, (o, p) \in I\} \tag{1}$$

$$B^{\downarrow} = \{o \in O| \quad \forall p \in B, (o, p) \in I\} \tag{2}$$

$\forall o \in X$, let $\{o\}^{\uparrow}$=$o^{\uparrow}$, and $\forall p \in B$, let $\{p\}^{\downarrow} \in p^{\downarrow}$.

Definition 4. [32] (**Concept**) Given a formal context $K = (O, P, I)$, (X, B) is called a concept if (X, B) satisfies $X^{\uparrow} = B$ and $B^{\downarrow} = X$, where X and B are called the extent and intent of the concept, respectively.

Definition 5. [32] Let $C(K)$ denote the set of all formal concepts of the formal context $K = (O, P, I)$. If (X_1, B_1), $(X_2, B_2) \in C(K)$, then let

$$(X_1, B_1) \leq (X_2, B_2) \Leftrightarrow X_1 \subseteq X_2 (\Leftrightarrow B_1 \supseteq B_2) \tag{3}$$

then “ $\leq$ ” is a partial relation of $C(K)$.

Definition 6. [32] (**Concept Lattice**) A concept lattice $CL(K)$ = $(C(K), \leq)$ can be obtained by all formal concepts $C(K)$ of a formal context K with the partial order “ $\leq$ ”. Its graphical representation is a Hasse diagram. $EL(K)$ is the set of extents for all concepts in $CL(K)$.

3.3 Problem Description

In this section, we formulate the problem of incremental entity summarization addressed in this paper. Incremental entity summarization selects *top-k* descriptions of the entity in dynamic knowledge graph where new predicates or objects are frequently added. For the sake of simplicity, this paper only focuses on the increment of predicates for the entity. We also assume that there is no decrease of the RDF descriptions in the knowledge graph.

Input: A set of RDF triples R of the entity in the incremental knowledge graph, where R includes original and increased RDF triples.

Output: A set of ranked *top-k* RDF triples R_1.

Process: Firstly, we construct two formal contexts (K_1,K_2) for original and newly added RDF triples, respectively, and then obtain two concept lattices $CL(K_1)$ and $CL(K_2)$. After that, we make intersection T of the extents of $CL(K_1)$ and the extents of $CL(K_2)$, i.e., T = $EL(K_1) \cap EL(K_2)$. Based on obtained intersection, the final concept lattice can be built. Finally, we rank the RDF triples by the $importance$, $redundancy$, and $uniqueness$ of triples based on the hierarchy of extents and intents in the final concept lattice.

4 Proposed Approach

This section discusses: 4.1 the framework of incremental entity summarization; 4.2 how to construct the Tokenized Formal Context; 4.3 the details of our proposed approach; 4.4 a relevant proof on the correctness of our proposed approach; 4.5 the improved ranking algorithm for entity summarization; 4.6 the algorithm descriptions.

4.1 Framework of Incremental Entity Summarization

Recall from Section 1 that Kim et al. [14] presented KAFCA using FCA and proved that it achieves better entity

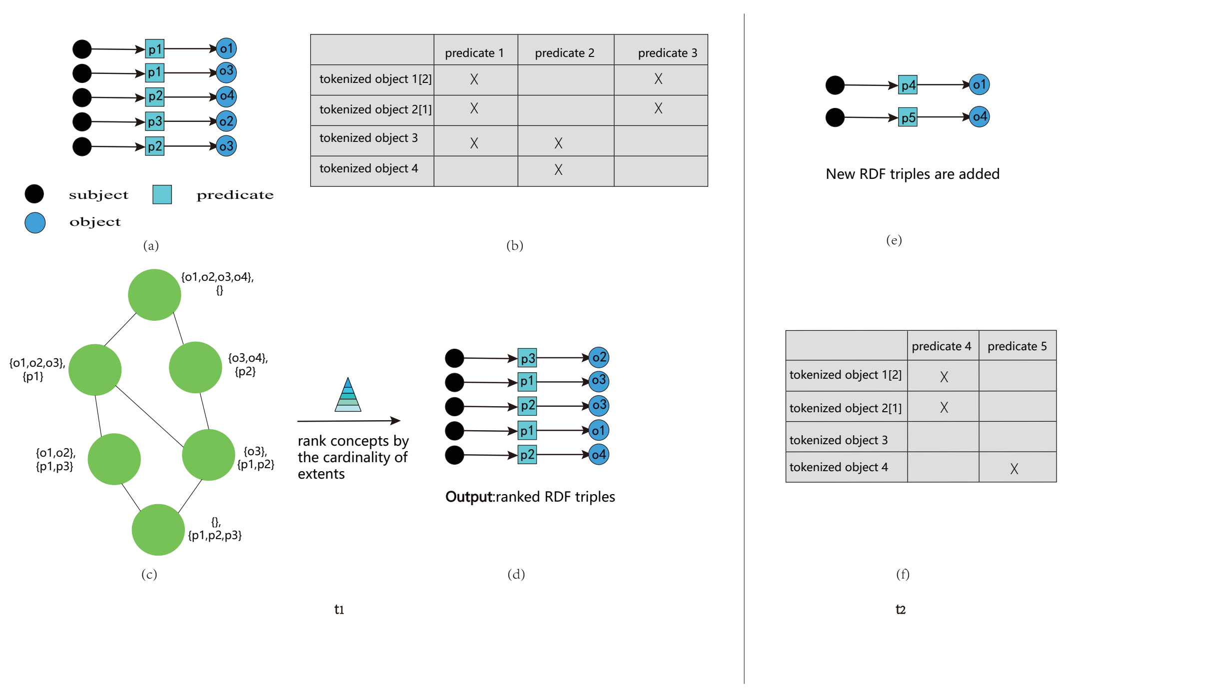


Fig. 2. The framework of incremental entity summarization.

summarization results than the state-of-the-art approaches. However, considering that KAFCA is non-incremental and the concept lattice can always be constructed in exponential time, the efficiency of entity summarization by KAFCA is limited, especially in the complex knowledge graph. Our proposed approach aims to reduce the time cost for generation of the entity summary by invoking an incremental algorithm for generating the concept lattice.

To better understand the problem, Fig. 2 depicts the framework of incremental entity summarization with F-CA. Here, o and p represent the object and predicate of the entity, respectively. We use the triples of actual entity to illustrate the Fig. 2. Concretely, p_1, p_2, p_3, p_4 and p_5 refer to $name$, $rdf-schema\#label$, $description$, $surname$ and $givenName$, respectively. o_1, o_1, o_3, and o_4 indicate "$Kippis, Andrew$"$@en$, "$Britishminister$", "$AndrewKippis$"$@en$, $Andrew$, respectively. As shown in Fig. 2 (a), first, the unordered RDF triples are input as initial data, and then they are constructed as a formal context using the binary relationships between the tokenized objects and predicates, as shown in Fig. 2 (b). Subsequently, a concept lattice is constructed based on the obtained formal context (Fig. 2 (c)). Finally, we select *top-k* RDF descriptions as an entity summarization by the proposed ranking algorithm that introduces the $importance$, $redundancy$, and $uniqueness$ of triples for entity summarization (Fig. 2 (d)). These mentioned procedures of entity summarization occurred at time t_1 are static, which only focuses on a snapshot of the entity.

However, the entity descriptions on the web are not static and change frequently. For instance, new RDF triples are added at time t_2. As concept lattices can grow exponentially large in the worst case [33], it is unnecessary to repeat the whole procedures for obtaining the entity summary. Thus, we presented a novel attribute-incremental algorithm for the construction of concept lattice to enhance the efficiency of entity summarization. The details of our proposed approach are described in the next subsection.

4.2 Tokenized Formal Context Construction

In this section, we illustrate how to tokenize the objects of triples and construct the tokenized formal context using the following triples of the actual entity "$3WAY\_FM$" in ESBM dataset [34]:
$(3WAY\_FM, subject, Category: Radio\_stations\_in\_Victoria)$ and $(3WAY\_FM, broadcastArea, Victoria\_(Australia))$.

The tokenized objects of triples can be obtained by splitting the objects into several single terms according to the segmentation principles including underline, camelcase, space, etc. For instance, the object $Category: Radio\_stations\_in\_Victoria$ can be tokenized as: $Category$, $Radio$, $stations$, in, and $Victoria$. According to *Definition 2*, the direct relationships between predicates and objects can be discovered in the formal context. Besides, if the objects of two triples share the same terms by tokenizing the objects, the underlying relationships between predicates and objects can also be discovered. For example, in Fig. 2 (b), we use the tokenized object 1[2] and tokenized object 2[1] to represent that the object 1 and object 2 share the same terms. More generally, for the predicate-object pairs $(subject, Category: Radio\_stations\_in\_Victoria)$ and $(broadcastArea, Victoria\_(Australia))$, the objects of which all contain the term of $Victoria$. Then, two potential relationships between the predicates and objects are added to construct the tokenized formal context: $(subject, Victoria\_(Australia))$, and $(broadcastArea, Category: Radio\_stations\_in\_Victoria)$. The direct and potential relationships between predicates and objects together form the tokenized formal context.

4.3 Incremental Entity Summarization with FCA

Inspired by our previous work [35], the proposed method can be described as follows:

Fig. 2 (b) and (f) are the formal context of original and newly added triples, respectively. The original formal context, the incremental formal context, and the final formal context are defined as: $K_1 = (O, P_1, I_1)$, $K_2 = (O, P_2, I_2)$, and $K = (O, P, I)$, respectively.

Firstly, we construct original formal context K_1 and newly added formal context K_2 according to the relationships between tokenized objects and predicates from RDF descriptions of the entity. Secondly, original concept lattice $C_1 = CL(K_1)$ and newly added concept lattice $C_2 = CL(K_2)$ are built using the obtained formal contexts. Thirdly, we take intersection T of $EL(K_1)$ and $EL(K_2)$. Afterwards, we obtain the intent i for each extent $e \in T$ according to $i \leftarrow e^{\uparrow}$, where the final concepts can be obtained. Finally, we obtain the ranked RDF triples using a modified algorithm that employs the $importance$, $redundancy$, and $uniqueness$ of triples based on [14]. More specifically, we grade and rank the RDF triples using the $importance$ of extents in concepts. The intuition of this approach is that the fewer objects an extent contains, the more important the objects are. Furthermore, the $redundancy$ is introduced to reduce the ranking score of the triples with the same object, while the $uniqueness$ of predicates is used to select the unique triples.

Example 1. Fig. 2 (c) is the initial concept lattice of K_1, whose concepts are: $(\emptyset, \{p_1, p_2, p_3\})$, $(\{o_1, o_2\}, \{p_1, p_3\})$, $(\{o_3\}, \{p_1, p_2\})$, $(\{o_1, o_2, o_3\}, \{p_1\})$, $(\{o_3, o_4\}, \{p_2\})$, $(\{o_1, o_2, o_3, o_4\}, \emptyset)$. Fig. 3 (a) is the concept lattice of the newly added formal context K_2, whose concepts are: $(\emptyset, \{p_4, p_5\})$, $(\{o_1, o_2\}, \{p_4\})$, $(\{o_4\}, \{p_5\})$, $(\{o_1, o_2, o_3, o_4\}, \emptyset)$. Then, we can obtain the extent set T by making intersection of T_1 and T_2, where $T_1 = EL(K_1), T_2 = EL(K_2)$. The extent set T are: $\{\{o_1, o_2, o_3, o_4\}$, $\{o_3, o_4\}$, $\{o_1, o_2, o_3\}$, $\{o_1, o_2\}$, $\{o_3\}$, $\{o_4\}$, $\emptyset\}$. Then, the corresponding intent i of each extent e in T is obtained by $i \leftarrow e^{\uparrow}$. Finally, we obtain the following concepts: $(\emptyset, \{p_1, p_2, p_3, p_4, p_5\})$, $(\{o_4\}, \{p_2, p_5\})$, $(\{o_3\}, \{p_1, p_2\})$, $(\{o_1, o_2\}, \{p_1, p_3, p_4\})$, $(\{o_3, o_4\}, \{p_2\})$, $(\{o_1, o_2, o_3\}, \{p_1\})$, $(\{o_1, o_2, o_3, o_4\}, \emptyset)$.

Fig. 3 (b) shows the actual concept lattice of the final formal context K, which is consistent with the obtained concepts by our method. Based on the obtained concept lattice, entity summarization can be generated.

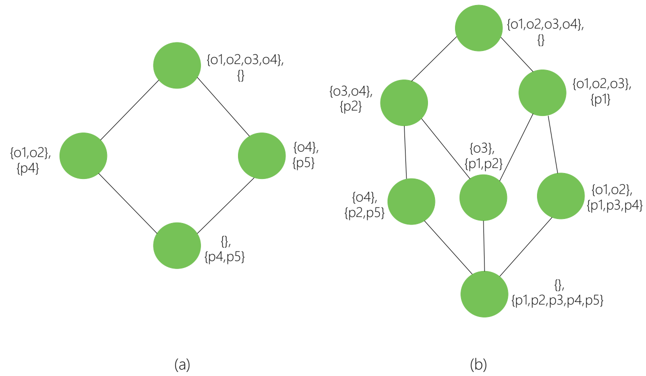


Fig. 3. Concept lattice of K_2 and K.

4.4 Correctness of the Proposed Approach

Considering that our proposed approach applies an incremental algorithm to entity summarization, it is necessary to prove the correctness of the method.

Theorem 1. Given three formal contexts $K_1 = \{O, P_1, I_1\}$, $K_2 = \{O, P_2, I_2\}$, and $K = (O, P_1 \cup P_2, I_1 \cup I_2)$, the relationship among the set of the extents of K_1, K_2, and K satisfies the following equation:

$$EL(K) = \{X_1 \cap X_2 | X_1 \in EL(K_1), X_2 \in EL(K_2)\} \quad (4)$$

where $EL(K)$ is the set of extents for all concepts in concept lattice $CL(K)$, and X_1 and X_2 are a set of extents in $EL(K_1)$ and $EL(K_2)$, respectively.

Proof:

1) For the original and newly added formal context K_1, K_2, the sets of extents $EL(K_1)$ and $EL(K_2)$, the sets of attributes P_1 and P_2, $\exists X_1 \in EL(K_1)$, $X_2 \in EL(K_2)$, $B_1 \subseteq P_1$, $B_2 \subseteq P_2$, assume that concept $(X_1, B_1) \in$ concept lattice $CL(K_1)$, concept $(X_2, B_2) \in$ concept lattice $CL(K_2)$. According to *Definition 3*, we have that $X_1 \cap X_2 = B_1^{\downarrow} \cap B_2^{\downarrow} = (B_1 \cup B_2)^{\downarrow}$. Due to $B_1 \cup B_2 \subseteq P_1 \cup P_2$, we have $((X_1 \cap X_2), (B_1 \cap B_2)^{\downarrow\uparrow}) = ((B_1 \cap B_2)^{\downarrow}, (B_1 \cap B_2)^{\downarrow\uparrow}) =$ concept lattice $CL(K)$, hence, $X_1 \cap X_2 \subseteq$ the set of extents $EL(K)$.
   Moreover, for the formal context K, the set of extents $EL(K)$, the sets of attributes P_1 and P_2, $\exists X \in EL(K)$, $B \subseteq P_1 \cup P_2$, assume that $(X, B) \in$ concept lattice $CL(K)$. According to *Definition 3*, we have that $X = B^{\downarrow} = (B \cap (P_1 \cup P_2))^{\downarrow} = ((B \cap P_1) \cup (B \cap P_2))^{\downarrow} = (B \cap P_1)^{\downarrow} \cap (B \cap P_2)^{\downarrow}$. Due to $B \cap P_1 \subseteq P_1$ and $B \cap P_2 \subset P_2$, we have $(B \cap P_1)^{\downarrow} \in$ the set of extents $EL(K_1)$ and $(B \cap P_2)^{\downarrow} \in$ the set of extents $EL(K_2)$, respectively. Therefore, $EL(K) = \{X_1 \cap X_2 | X_1 \in EL(K_1), X_2 \in EL(K_2)\}$.
2) Typically, for $P_2 = \{m\}$, $K_2 = \{O, m, I_2\}$, $\exists X \in EL(O, P_1, I)$, we have that the set of extents $EL(O, P_1 \cup \{m\}, I) = EL(O, P_1, I) \cup EL(O, \{m\}, I_2) = EL(O, P_1, I) \cup \{X \cap m^{\downarrow}\}$. According to 1), we have the set of extents $EL(O, \{m\}, I_2) = \{m^{\downarrow}, \emptyset^{\downarrow}\} = \{m^{\downarrow}, O\}$.

According to ***Theorem 1***, we have that the set of extents of the formal context K equals to the intersection of the set of extents of formal contexts K_1 and K_2.

4.5 Improved Ranking Algorithm for Entity Summarization

This section describes the modification of ranking algorithm that introduces the $importance$, $redundancy$, and $uniqueness$ of triples for entity summarization based on [14]. In [14], the authors rank the RDF triples according to the cardinality of extents for the concepts in concept lattice, the intuition of which is that the concept is more important when the cardinality of extent of concept is smaller. However, the cardinality of intents is also an important factor that can not be ignored. Thus, we improved the ranking algorithm by considering the cardinality of extents and intents simultaneously. Additionally, in order to reduce the $redundancy$ of RDF triples and quantize the $importance$ and $uniqueness$ of each triple, the following ranking indicators are defined:

$$uniqueness(s, p, o) = \frac{len(entity)}{number(p)} \quad (5)$$

where $len(entity)$ denotes the number of RDF triples of the entity, and $number(p)$ is the number of predicate p in all triples. From Equation (5), we can observe that the rarer the predicate of the triple in all triples is, the more unique the triple is, which means that the triple can be more representative of the entity. For all the RDF triples, by calculating the uniqueness of each triple, more triples containing unique properties can be assigned with higher scores and be selected. Then, the score of each triple $ranking(s, p, o)$ can be defined accordingly:

$$\begin{aligned} ranking(s, p, o) = len(entity) - hierarchy \\ -redundancy + uniqueness \end{aligned} \quad (6)$$

where $hierarchy$ and $redundancy$ are related to the hierarchy of concepts in concept lattice. When we re-rank all the concepts according to the ascending order of the cardinality of extents, the $importance$ of extents in the obtained concepts decreases as the cardinality of extents increases. Consequently, the $hierarchy$ can be utilized to obtain more important triples, because the concepts with fewer objects are located at higher layers and can be assigned with higher scores. In addition, due to the same object in RDF various triples, the selected triples should avoid triples with the same object occurrence. Thus, we use $redundancy$ to lessen the ranking score when the triples with the same object have been selected.

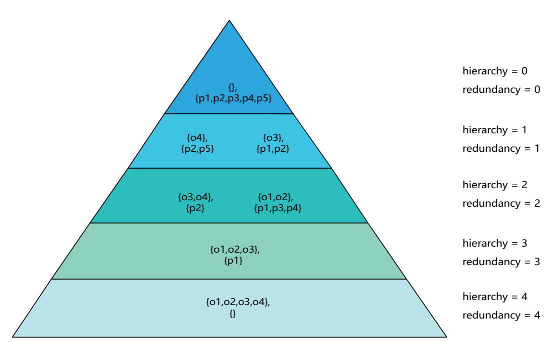


Fig. 4. The ranking process for the concept lattice of K.

Example 2. In Fig. 3 (b), the obtained concepts are: $(\{o_3, o_4\}, \{p_2\})$, $(\{o_1, o_2, o_3\}, \{p_1\})$, $(\{o_1, o_2\}, \{p_1, p_3, p_4\})$, $(\{o_3\}, \{p_1, p_2\})$, $(\{o_4\}, \{p_2, p_5\})$. Fig. 4 illustrates the ranking process for the obtained concepts. Firstly, we re-ranked the concept lattice based on the cardinality of extents for the concepts. Typically, the concepts with the same cardinality of extents are at the same layer and the concepts with less cardinality of extents are at higher layer. For the original 5 triples in Fig. 2: (s, p_1, o_1), (s, p_1, o_3), (s, p_2, o_4), (s, p_3, o_2), (s, p_2, o_3), and the newly added 2 triples: (s, p_4, o_1), (s, p_5, o_4), we can obtain $len(entity) = 7$. According to the Equation (5), the values of $uniqueness$ for all triples are calculated as follows:

$$uniqueness(s, p_2, o_4) = 3, uniqueness(s, p_5, o_4) = 7$$
$$uniqueness(s, p_2, o_3) = 3, uniqueness(s, p_1, o_1) = 3$$
$$uniqueness(s, p_3, o_2) = 7, uniqueness(s, p_4, o_1) = 7$$
$$uniqueness(s, p_1, o_3) = 3$$

Concretely, because the number of predicates p_2 and p_5 in all triples is 2 and 1, respectively, $uniqueness(s, p_2, o_4) = 3$ and $uniqueness(s, p_5, o_4) = 7$ by the Equation (5). When assigning the scores to triples, we traverse all concepts and calculate the scores of triples $ranking(s, p, o)$ according to the hierarchy of the re-ranked concepts. More specifically, we traverse the concepts in different layers as the cardinality of extents of concepts (or the layer of concepts) increases. For the concepts at the same layer, the cardinality of intents of the concept is bigger, and the concept is calculated first. For example, $(\{o_4\}, \{p_2, p_5\})$ and $(\{o_3\}, \{p_1, p_2\})$ are both at the second layer and the concepts are calculated first compared to the concepts in other layers. Due to $(\{o_4\}, \{p_2, p_5\})$ and $(\{o_3\}, \{p_1, p_2\})$ have the same number of extent and intent, they are given the same score. Here, the score for a triple (s, p, o) is determined by the concept that first appeared. For instance, $(\{o_4\}, \{p_2, p_5\})$ and $(\{o_3, o_4\}, \{p_2\})$ are located at the second and third layer, respectively. Then, the score of the triple (s, p_2, o_4) that contains o_4 is calculated by the $(\{o_4\}, \{p_2, p_5\})$ rather than $(\{o_3, o_4\}, \{p_2\})$, although the latter also contains o_4. In terms of the $redundancy$, it is added into the Equation (6) only when the score of triple that contains the same object is calculated again. For example, when calculating the concept $(\{o_4\}, \{p_2, p_5\})$ that refers to the following two triples: (s, p_2, o_4) and (s, p_5, o_4), the redundancy is added into the Equation (6) when calculating the ranking score of the (s, p_5, o_4) as (s, p_2, o_4) contains the same object o_4. Therefore, the traversal sequence of the concepts and the corresponding scores of the triples can be obtained as follows:

$$ranking(s, p_1, o_3) = 7 - 1 + 3 = 9$$
$$ranking(s, p_2, o_3) = 7 - 1 - 1 + 3 = 8$$
$$ranking(s, p_2, o_4) = 7 - 1 + 3 = 9$$
$$ranking(s, p_5, o_4) = 7 - 1 - 1 + 7 = 12$$
$$ranking(s, p_1, o_1) = 7 - 2 + 3 = 8$$
$$ranking(s, p_4, o_1) = 7 - 2 - 1 + 7 = 11$$
$$ranking(s, p_3, o_2) = 7 - 2 + 7 = 12$$

Finally, the RDF triples can be sorted in descending order by the ranking scores.

Compared with KAFCA, our improved ranking algorithm can perform better on distinguishing the $importance$ of these concepts with the same cardinality of extents. In addition, the $uniqueness$ and $redundancy$ of triples are also considered into the ranking process, which can ensure that the most representative triples are selected and the performance of entity summarization is improved.

4.6 Algorithms

Algorithm 1 Incremental Entity Summarization Algorithm

Input:
A set of RDF triples for the entity, R
The parameter of output RDF triples, k
Output:
A set of ranked *top-k* RDF triples R_1

```
1: Initialize K_1 = ∅, K_2 = ∅
2: begin
3:     Get tokenized objects O, original predicates P_1, incremental
       predicates P_2 by segmentation operation from R
4: end
5: begin
6:     K_1 = (O, P_1, I_1)
7:     K_2 = (O, P_2, I_2)
8:     C ← IncrementalConcept(K_1,K_2)
9: end
10: Obtain R_1 by invoking Ranking Algorithm
```

Based on *Theorem 1*, we propose an incremental entity summarization algorithm listed as ***Algorithm 1***. Firstly, a set of RDF triples for the entity, R, and the parameter of output RDF triples, k (given by users), are given in input. Then Line 1 initializes original formal context K_1 and newly added formal context K_2. The purpose of Lines 2-4 is to obtain the tokenized objects O, original predicates P_1, incremental predicates P_2 from initial data R. After that, original formal context K_1 and incremental formal context K_2 can be assigned with binary relation value ("0" or "1") according to the relationships between the obtained objects and predicates (Lines 6-7). At Line 8, by invoking the algorithm ***IncrementalConcept(**K_1**,**K_2**)***, the final concept lattice can be built. Finally, we rank RDF triples of the entity via ***Ranking Algorithm*** at Line 10.

Algorithm 2 Non-incremental Entity Summarization Algorithm

Input:
A set of RDF triples for the entity, R
The parameter of output RDF triples, k
Output:
A set of the ranked *top-k* RDF triples R_1

```
Initialize K = ∅, C = ∅
begin
    Get tokenized objects O, predicates P by segmentation opera-
    tion from R
end
begin
    K = (O, P, I)
    C ← BasicConcept(K)
end
Obtain R_1 by invoking Ranking Algorithm
```

For comparison, ***Algorithm 2*** details the algorithm of non-incremental entity summarization [14]. The differences between this algorithm and ***Algorithm 1*** lie at Lines 2-4 and Lines 5-8. On the one hand, ***Algorithm 2*** considers the initial input of RDF triples for the entity as a whole, thus the entire tokenized objects O and predicates P can be acquired (Lines 2-4). On the other hand, Lines 5-8 in ***Algorithm 2*** obtain the final concepts by ***BasicConcept(K)***. The ranked RDF triples R_1 and R_2 are output as shown at Line 9.

Algorithm 3 IncrementalConcept(K_1,K_2)

Input:
The formal contexts K_1, K_2
Output:
A set of concepts C
1: Initialize $C=\emptyset$, $C_1=\emptyset$, $C_2=\emptyset$, $T=\emptyset$, $T_1=\emptyset$, $T_2=\emptyset$
2: **begin**
3: $C_1 \leftarrow$ ***BasicConcept(K_1)***
4: $C_2 \leftarrow$ ***BasicConcept(K_2)***
5: **end**
6: **for** each concept $(X,B) \in C_1$
7: $T_1 \leftarrow X \cup T_1$
8: **end**
9: **for** each concept $(X,B) \in C_2$
10: $T_2 \leftarrow X \cup T_2$
11: **end**
12: $T \leftarrow T_1 \cap T_2$
13: **for** each extent $e \in T$
14: $i \leftarrow e^{\uparrow}$
15: $C \leftarrow (e,i) \cup C$
16: **end**

Algorithm 4 BasicConcept(K)

Input:
A formal context K
Output:
A set of concepts C
1: Initialize $T=\emptyset$, $P=\emptyset$, $C=\emptyset$
2: **begin**
3: $T \leftarrow$ Add the set that contains all objects in K
4: $P \leftarrow$ Add all attributes in K
5: **end**
6: **for** each attribute $a \in P$
7: **for** each extent $e \in T$
8: $T \leftarrow e \cap a^{\downarrow}$
9: **end**
10: **end**
11: **for** each extent $e \in T$
12: $i \leftarrow e^{\uparrow}$
13: $C \leftarrow (e,i) \cup C$
14: **end**
15: **Return** C

As for algorithm ***IncrementalConcept(K_1,K_2)***, Line 1 initializes concept sets (C, C_1, C_2), extent sets (T, T_1, T_2). After that, Lines 2-5 assign with values to C_1 and C_2 through ***BasicConcept(K_1)*** and ***BasicConcept(K_2)***, respectively. Based on the obtained C_1 and C_2, the extent sets T_1 and T_2 can be obtained by two loop operations (Lines 6-11), respectively. Followed by taking the intersection of T_1 and T_2 (Line 12), we utilize the obtained intersection T to construct the final concept lattice (Lines 13-16).

BasicConcept(K) is a non-incremental construction algorithm of concept lattice. Firstly, Line 1 initializes the extent set T, attribute set P, concept set C. Then Lines 2-5 are the assignment operations for T and P. Finally, we can obtain the all extent set T (Lines 6-10) and concepts set C (Lines 11-15) according to *Definition 4*.

Algorithm 5 is the modified algorithm of entity summarization based on FCA, which considers the $importance$, $redundancy$, and $uniqueness$ of triples in ranking the RDF triples of the entity compared to [14]. Line 1 initializes the final score $final\_score$ of each triple, other variables. Line 3 ranks the concepts C according to the cardinality of extents and intents in C, where the concepts C are firstly ranked by the cardinality of extents, and then ranked according to the cardinality of intents when the cardinalities of extents are the same. Line 4 obtain the subject, predicate, and object from R. Then, we calculate the $final\_score$ (Lines 6-33) considering the $importance$, $redundancy$, and $uniqueness$ of triples.

Algorithm 5 Ranking Algorithm

Input:
A set of concepts C
A set of RDF triples for the entity, R
The parameter of output RDF triples, k
Output:
A set of the ranked *top-k* RDF triples R_1
1: Initialize $final\_score$, $hierarchy$, $redundancy$, $uniqueness = 0$, $i = 1$, $object\_list = \emptyset$
2: **begin**
3: $C_1 \leftarrow$ Rank concepts according to the cardinality of extents and intents in C
4: s, p, $o \leftarrow$ Obtain the $subject$, $predicate$, $object$ from R
5: **end**
6: **for** each concept $(X,B) \in C_1$
7: **for** each extent $e \in X$
8: $number\_p = count(p)$
9: $uniqueness = \frac{length(R)}{number\_p}$
10: **if** $entity \in$ *'dbpedia'*
11: **if** $extent \in object\_list$
12: $final\_score[s,p,extent] =$
13: $length(R) - hierarchy - redundancy$
14: $object\_list \leftarrow object\_list \cup extent$
15: continue
16: **end if**
17: $final\_score[s,p,extent] = length(R) -$
18: $hierarchy + uniqueness$
19: $object\_list \leftarrow object\_list \cup extent$
20: **else if** $entity \in$ *'lmdb'*
21: **if** $extent \in object\_list$
22: $final\_score[s,p,extent] =$
23: $length(R) - redundancy$
24: $object\_list \leftarrow object\_list \cup extent$
25: continue
26: **end if**
27: $final\_score[s,p,extent] = length(R) +$
28: $uniqueness$
29: $object\_list \leftarrow object\_list \cup extent$
30: **end if**
31: **end**
32: $hierarchy += 1, redundancy += 1$
33: **end**
34: **begin**
35: $final\_score \leftarrow$ Rank $final\_score$ in descending order according to its value
36: **end**
37: **for** each s, p, $o \in final\_score$
38: **if** $i <= k$
39: $R_1 \leftarrow R_1 \cup (s,p,o)$
40: **end if**
41: i++
42: **end**

More specifically, the $importance$ of triples is calculated according to the hierarchy of concepts in C_1. In other words, if an extent in concepts has fewer objects, the objects are more important and the corresponding scores for these objects are higher. Due to the existence of the same objects in various triples that should avoid being selected as the summarization of the entity, the $redundancy$ is introduced to lessen the scores of triples that the triples with the same objects have been in existence. By utilizing the $uniqueness$, the more unique and representative triples can be selected, because the predicates of triples usually represent one respect of the entity and the rarity of the predicates can be selected as the uniqueness of the entity. Intuitively, the more rare the predicates are, the more representative the triples that contain the predicates are.

Concretely, Lines 8-9 calculate the number of predicate p in all triples and the corresponding $uniqueness$ of p. Then, the scores of the triples from the DBpedia dataset and LinkedMDB dataset are obtained at Lines 10-19 and Lines 20-33, respectively. For avoiding redundancy of the summarization, Lines 11-16 and Lines 20-26 lessen the scores of the triples with the same objects. Lines 17-19 calculate scores of the triples on the DBpedia dataset by considering the $importance$ and $uniqueness$, while Lines 27-30 calculate scores of the triples on the LinkedMDB dataset by considering the $uniqueness$. The reason why we omit the $importance$ from the LinkedMDB dataset is that the objects of the triples are in the form of a specific number rather than meaningful token. This prevents hierarchy of concepts from distinguishing the $importance$ of concepts and triples. Line 32 assigns incremental values to $hierarchy$ and $redundancy$ with traversing the concepts in C_1. After that, Lines 34-36 rank the $final\_score$ in descending order according to its value. Finally, the remaining procedures (Lines 37-42) output the ranked top-k RDF triples.

5 Experiments

In this section, we first introduce the datasets and implementation detail of our experiments, and then depict the evaluation criteria. Afterwards, we present the comparison approaches and discuss the experimental results. All experiments are implemented with Inter(R) Core (TM)i5-8250U CPU@1.60GHz 1.80GHz 16GB-RAM PC under Windows10 system.

5.1 Datasets and Implementation

The real-world dataset ESBM [1] we employed in experiments is available in [34], which contains two benchmark datasets for evaluating entity summarization. ESBM is currently the largest available benchmark dataset that can be found in the real-world. ESBM v1.0 and v1.2 consist of 140 entities and 175 entities selected from DBpedia[2] and LinkedMDB[3], respectively. For each entity, ESBM provides its original descriptions, with the addition of 6 top-5 and 6 top-10 ground-truth summaries created by crowdsourcing. Concretely, ESBM v1.0 is a total of 100 DBpedia entities whose types consist of Agents, Events, Locations, Species, and Works, and 40 entities of LinkedMDB related to Films and Persons. On the basis of v1.0, ESBM v1.2 adds another 5 entities for each type of entity. We conducted the following three comparison experiments on ESBM v1.0 in terms of the efficiency, with the addition of a performance comparison experiment on ESBM v1.0 and v1.2 compared to other state-of-the-art algorithms:

- **Experiment I:** First, we obtained the files of formal context using the Entity Summarization Benchmark datasets v1.0 and v1.2 [34]. After that, we converted the obtained files to adjacent matrices that are formal contexts of entities, as initial data in our experiments. Afterwards, we split the formal context into two categories, original formal context (K_0) and incremental formal context (K_1, K_2, K_3, K_4, K_5, K_6). For example, K_2 means that the formal context has two incremental attributes. For these entities, we compared our proposed method with KAFCA in terms of runtime.
- **Experiment II:** Second, we selected the entity@115 (refers to the entity with ID "115") that contains the largest number of predicates from all 140 entities and divided these predicates into two parts, original predicates and incremental predicates. In this experiment, we aim to explore how the various partitions of predicates influence the efficiency of entity summarization.
- **Experiment III:** Third, we conducted experiments on diverse predicate increment inc (inc=1, 2, 3) but with the same number of objects to find out the variation trend of the efficiency influenced by the predicate increment.
- **Experiment IV:** Fourth, we compared IES-FCA to KAFCA and other algorithms with regard to $F1-measure$, MAP and $NDCG$ performance measurements on both ESBM v1.0 and ESBM v1.2. Due to the attribute increment does not affect the final results of entity summarization, we set the attribute increment $inc = 3$ in the experiments for Table 3 to 6. Additionally, to study the influence of the $uniqueness$ factor of the ranking algorithm, the results of the weighting tests are also provided. Concretely, we assign weight α to $len(entity) - hierarchy - redundancy$ and $(1-\alpha)$ to $uniqueness$, respectively.
- **Experiment V:** Finally, to validate the rationality and effectiveness of each factor in Equation (6), we conduct the ablation study that only reserves one factor from $importance$, $redundancy$, and $uniqueness$. The ablation study contains three different variants of IES-FCA, including IES-FCA$_i$, IES-FCA$_r$, and IES-FCA$_u$ that denote the $importance$, $redundancy$, and $uniqueness$ factors only considered in Equation (6), respectively.

Fig. 5 (a), 5 (b), and 6 depict the result of Experiment I, II and III, respectively. TABLE 1 and 2 show the improvement of efficiency in Experiment II and statistics of entities in Experiment III, respectively. TABLE 3, 4 present the results of $F1-measure$ and MAP, and TABLE 5, 6 show the results of $F1-measure$ and $NDCG$ for IES-FCA and other algorithms, respectively. TABLE 7 presents the ablation test results of $F1-measure$, MAP and $NDCG$ on ESBM v1.0 and ESBM v1.2. Before discussing the experimental results, we first introduce the evaluation criteria and comparison approaches for our experiments.

1. https://w3id.org/esbm
2. http://dbpedia.org/
3. http://www.linkedmdb.org/

5.2 Evaluation Criteria and Protocol

In this section, we will introduce the evaluation criteria that is adopted in [34], [36]. We utilize the following three indicators: $F1-measure$ (so-called F1-score), MAP (Mean Average Precision), and $NDCG$ (Normalized Discounted Cumulative Gain). $F1-measure$ calculates the harmonic average of the P (Precision) and R (Recall). MAP denotes the mean of AP (Average Precision) for all entities, of which AP is the average precision of the obtained summaries for each entity. $NDCG$ has been widely applied in the field of information retrieval, which can assess the quality of the obtained summaries.

$$P = \frac{|S_m \bigcap S_h|}{|S_m|}, R = \frac{|S_m \bigcap S_h|}{|S_h|}, F1 = \frac{2 \cdot P \cdot R}{P+R} \tag{7}$$

where S_m and S_h are summaries by a certain entity summarization approach and ground-truth summaries created by crowdsourcing, respectively.

$$AP = \frac{\sum\limits_{i=1, S_m[i-1] \in S_h}^{M} P(S_h, S_m(i-1))}{H} \tag{8}$$

where M, H, $S_m[i-1]$, $S_m(i-1)$ represents the size of S_m, the size of S_h, the $i-1th$ element of S_m and the subset of S_m that contains the elements from $0th$ to $i-1th$, respectively. Accordingly, the MAP can be obtained as follows:

$$MAP = \frac{\sum\limits_{i=1}^{G} AP}{G} \tag{9}$$

Here, G denotes the number of the ground-truth summaries for each entity by various human experts.

Let S_{gt} and $Desc(e)$ represent a ground-truth summary and an entity description, respectively. For a triple $t \in Desc(e)$, the relevant function rel is defined as follows:

$$rel(t) = \begin{cases} 1 & if \quad t \in S_{gt} \\ 0 & if \quad t \notin S_{gt} \end{cases} \tag{10}$$

where $rel(t) = 1$ means that it is relevant for the triple t when $t \in Desc(e)$ and $t \in S_{gt}$.

The $NDCG$ of the ranking at position $i (1 \leq i \leq I)$ can be defined as follows:

$$NDCG@i = \frac{DCG@i}{IDCG@i} \tag{11}$$

$$DCG@i = \sum_{j=1}^{i} \frac{rel(r_{j-1})}{log(j+1)}, IDCG@i = \sum_{j=1}^{i} \frac{1}{log(j+1)} \tag{12}$$

where I is with the setting parameters of 5 and 10 in the experiments.

Note that, we first calculate the mean value of $F1-measure$, MAP and $NDCG$ for 6 ground-truth summaries by comparing the summarization result with each ground-truth summary. Then, we further obtain the average scores of the mean value of the three indicators (i.e., $F1-measure$, MAP and $NDCG$) for all entities, respectively.

5.3 Comparison Approaches

Considering that KAFCA is one of the most relevant approaches to our work and performs better than other approaches, this paper aims to improve the efficiency as well as the effectiveness of entity summarization compared with KAFCA. Note that FACES [23] is also an incremental approach that leverages Cobweb for partitioning feature set, while IES-FCA employs an incremental algorithm concept lattice construction for the FCA-based entity summarization approach. Nevertheless, this paper focuses more on the efficiency improvement compared to KAFCA and thus, FACES is excluded from the efficiency comparison experiment. Accordingly, we use the following comparison approaches:

- **Non-incremental Entity Summarization:** The compared entity summarization approach [14] is non-incremental. This method employs initial and newly added RDF triples R as input, and then formal context K is obtained by the relationship between tokenized objects and predicates of R, which are regarded as objects and attributes in formal context, respectively. After concept lattice is built by ***BasicConcept(*** K ***)*** algorithm, the ranked RDF triples are output according to ***Ranking Algorithm***.
- **Incremental Entity Summarization:** The proposed incremental method in this paper is based on the compared entity summarization method, with the addition of the ***IncrementalConcept(*** K_1,K_2 ***)*** algorithm. The algorithm is an incremental construction algorithm of concept lattice, the central idea of which is to take the intersection of the extents of C_1 and the extents of C_2 and then obtain the final concept lattice by the intersection. Finally, we output the ranked RDF triples using ***Ranking Algorithm***.

TABLE 1
The Improvement of Efficiency in Experiment II.

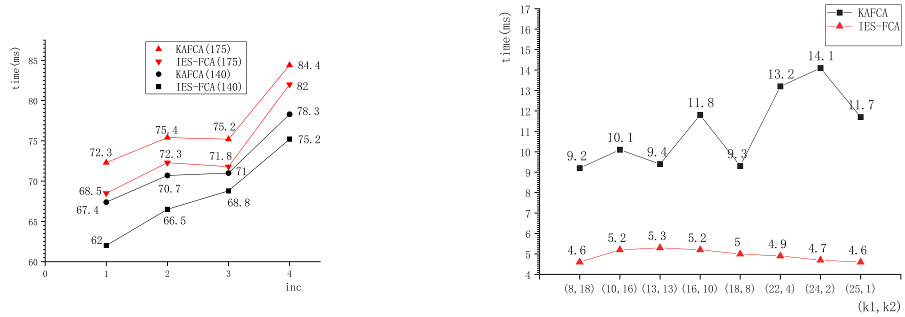
The partitions of predicates	The Improvement of Efficiency
(8,18)	50%
(10,16)	49%
(13,13)	44%
(16,10)	56%
(18,8)	46%
(22,4)	63%
(24,2)	67%
(25,1)	61%

TABLE 2
The statistics of entities in experiment III.

Entity Number	The Num of Predicates	The Num of Concepts
Entity@4	11	14
Entity@5	15	22
Entity@27	18	18
Entity@105	20	16
Entity@134	9	11

5.4 Experimental Results

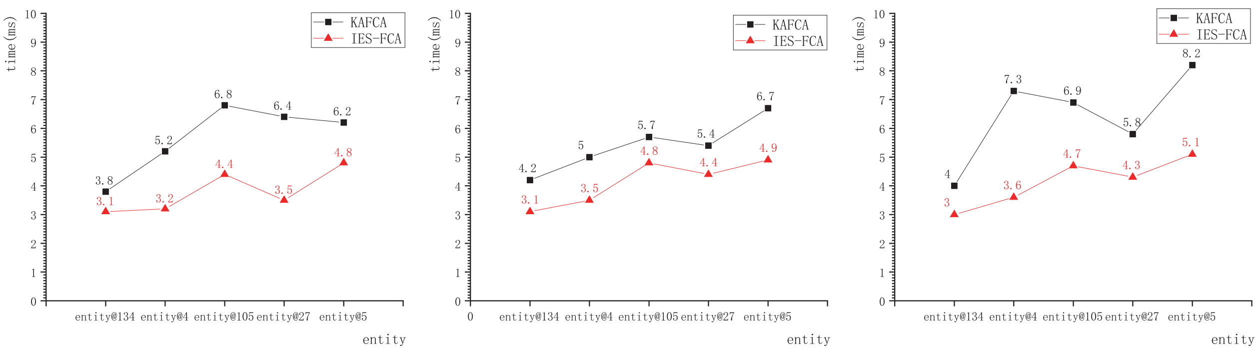
For the consistency of inputs, we added the runtime of concept lattice construction for original formal context into the comparison approaches when we calculated the runtime.



(a) The efficiency of our method compared with baseline method for 140 and 175 entities.

(b) The efficiency of our method compared with baseline method for the entity that contains the largest amount of predicates.

Fig. 5. The results of Experiment I and Experiment II.



(a) $inc = 1$.

(b) $inc = 2$.

(c) $inc = 3$.

Fig. 6. The efficiency of our method compared with baseline method for the entity that contains the same number of objects and different predicate increment.

The pre-processing time is not considered in the experimental results. Furthermore, we ran the comparison approaches 10 times for each result.

As shown in Fig. 5 (a), the result declares that our method has better performance on the evaluation of runtime than the compared method. The black and red curve represent the runtime changes using 140 entities and 175 entities, respectively. Specifically, for the case of $inc = 1$, the efficiency of entity summarization can be increased up to 8.7% and 5.5 % than KAFCA for all 140 entities and 175 entities, respectively.

Fig. 5 (b) signifies that our incremental approach can reduce the time consumption dynamically for the entity@115 that contains the largest number of predicates. It is clear that the difference of efficiency between KAFCA and our method is distinct when the number of predicates is large. Particularly, the data of efficiency improvement is listed in TABLE 1. Note that the efficiency of entity summarization can be raised up to 67%.

The results of Experiment III are reported in Fig. 6, where all the entities have 40 objects, but with diverse number of predicates. The number of predicates and the concepts of the entities are detailed in TABLE 2. Looking at a single diagram in Fig. 6, we can observe that the runtime increases with the number of predicates as concepts increase. Interestingly, the summary efficiency of entity@105 is lower than entity@27, although entity@27 has more concepts. The reason is that entity@105 has more predicates, which indicates that both the number of predicates and concepts affect the efficiency of entity summarization. Lastly, we can conclude that IES-FCA performs better than KAFCA when different number of attributes is added.

TABLE 3
F1-measure of the selected entity summarizers on ESBM v1.0.

Model	DBpedia		LinkedMDB		ALL	
	$k = 5$	$k = 10$	$k = 5$	$k = 10$	$k = 5$	$k = 10$
RELIN [21]	0.250	0.468	0.210	0.260	0.239	0.409
DIVERSUM [22]	0.260	0.522	0.222	0.365	0.249	0.477
FACES [23]	0.272	0.439	0.160	0.259	0.240	0.388
FACES-E [4]	0.285	0.527	**0.252**	0.348	0.276	0.476
LinkSUM [25]	0.290	0.498	0.117	0.255	0.240	0.428
CD [24]	0.299	**0.531**	0.215	0.326	0.267	0.467
KAFCA [14]	**0.332**	**0.531**	0.249	**0.399**	**0.308**	**0.493**
IES-FCA	**0.374 (▲ 12.65%)**	**0.562 (▲ 5.84%)**	**0.333 (▲ 32.14%)**	**0.436 (▲ 9.27%)**	**0.363 (▲ 17.86%)**	**0.526 (▲ 6.69%)**
IES-FCA($\alpha = 0.2$)	0.374	**0.564 (▲ 0.02%)**	0.333	**0.438 (▲ 0.02%)**	0.363	**0.528 (▲ 0.02%)**

TABLE 3 and 4 show the $F1 - measure$ and MAP results of entity summarization on ESBM v1.0 for the comparison approaches, which declares that the superiority of IES-FCA by comparing with the state-of-the-art approaches. Concretely, compared to other representative approaches, the results of $F1 - measure$ improvement range from 5.84% to 32.14% and the results of MAP improvement can reach to 17.87%. For different α of the weighting tests of the $uniqueness$ factor, the best experimental results can be reached when $\alpha = 0.2$. Compared with the proposed IES-FCA, the majority of results about $F1 - measure$ and

TABLE 4
MAP of the selected entity summarizers on ESBM v1.0.

Model	DBpedia		LinkedMDB		ALL	
	$k=5$	$k=10$	$k=5$	$k=10$	$k=5$	$k=10$
LinkSUM [25]	0.246	0.386	0.120	0.254	0.210	0.348
FACES [23]	0.247	0.386	0.140	0.261	0.216	0.351
DIVERSUM [22]	0.316	0.511	0.269	0.388	0.302	0.476
RELIN [21]	0.348	0.532	0.243	0.337	0.318	0.476
FACES-E [4]	0.354	0.529	0.258	0.361	0.326	0.481
CD [24]	-	-	-	-	-	-
KAFCA [14]	**0.402**	**0.597**	**0.319**	**0.428**	**0.378**	**0.549**
IES-FCA	**0.447 (▲ 11.19%)**	**0.634 (▲ 6.20%)**	**0.376 (▲ 17.87%)**	**0.457 (▲ 6.78%)**	**0.427 (▲ 12.96%)**	**0.584 (▲ 6.38%)**
IES-FCA($\alpha=0.2$)	0.447	**0.635 (▲ 0.01%)**	**0.377 (▲ 0.01%)**	**0.459 (▲ 0.02%)**	0.427	**0.585 (▲ 0.01%)**

MAP are improved when considering the weight of the $uniqueness$ factor into Equation (6).

TABLE 5
F1-measure of the selected entity summarizers on ESBM v1.2.

Model	DBpedia		LinkedMDB		ALL	
	$k=5$	$k=10$	$k=5$	$k=10$	$k=5$	$k=10$
RELIN [21]	0.242	0.455	0.203	0.258	0.231	0.399
DIVERSUM [22]	0.249	0.507	0.207	0.358	0.237	0.464
FACES [23]	0.270	0.428	0.169	0.263	0.241	0.381
FACES-E [4]	0.280	0.488	0.313	0.393	0.289	0.461
CD [24]	0.283	0.513	0.217	0.331	0.264	0.461
LinkSUM [25]	0.287	0.486	0.140	0.279	0.245	0.427
BAFREC [26]	**0.335**	0.503	**0.360**	0.402	**0.342**	0.474
MPSUM [27]	0.314	0.512	0.272	**0.423**	0.302	0.486
ESA [29]	0.310	**0.525**	**0.320**	0.403	0.312	**0.491**
KAFCA [14]	0.314	0.509	0.244	0.397	0.294	0.477
IES-FCA	**0.357 (▲ 6.58%)**	**0.546 (▲ 4.00%)**	**0.319**	**0.434 (▲ 2.60%)**	**0.346 (▲ 1.17%)**	**0.514 (▲ 4.68%)**
IES-FCA($\alpha=0.2$)	0.357	**0.547 (▲ 0.001%)**	0.319	**0.435 (▲ 0.001%)**	0.346	**0.515 (▲ 0.001%)**

TABLE 6
NDCG of the selected entity summarizers on ESBM v1.2.

Model	DBpedia		LinkedMDB		ALL	
	$k=5$	$k=10$	$k=5$	$k=10$	$k=5$	$k=10$
RELIN [21]	0.699	0.795	0.586	0.690	0.666	0.765
DIVERSUM [22]	0.646	0.757	0.589	0.714	0.630	0.745
FACES [23]	0.523	0.711	0.390	0.565	0.485	0.669
FACES-E [4]	0.735	0.836	0.674	0.765	0.718	0.816
CD [24]	-	-	-	-	-	-
LinkSUM [25]	0.505	0.699	0.371	0.574	0.467	0.663
BAFREC [26]	**0.752**	0.832	**0.773**	**0.827**	**0.758**	**0.830**
MPSUM [27]	0.745	0.831	0.694	**0.787**	0.730	0.819
ESA [29]	0.743	0.847	0.694	0.779	0.729	0.827
KAFCA [14]	0.737	**0.851**	0.640	0.754	0.709	0.823
IES-FCA	**0.783 (▲ 4.12%)**	**0.875 (▲ 2.82%)**	**0.703**	**0.786**	**0.760 (▲ 0.26%)**	**0.850 (▲ 2.41%)**
IES-FCA($\alpha=0.2$)	0.782	0.875	0.703	**0.787 (▲ 0.001%)**	0.760	0.850

TABLE 7
The results of ablation tests on ESBM v1.0 and ESBM v1.2.

Model	DataSet	Metrics	DBpedia		LinkedMDB		ALL	
			$k=5$	$k=10$	$k=5$	$k=10$	$k=5$	$k=10$
IES-FCA$_i$	v1.0	F1	0.335	0.530	0.242	0.406	0.308	0.494
		MAP	0.405	0.590	0.348	0.438	0.388	0.546
	v1.2	F1	0.317	0.510	0.235	0.399	0.294	0.478
		NDCG	0.741	0.841	0.676	0.763	0.722	0.819
IES-FCA$_r$	v1.0	F1	0.169	0.563	0.133	0.282	0.158	0.324
		MAP	0.259	0.638	0.230	0.335	0.245	0.393
	v1.2	F1	0.171	0.342	0.135	0.282	0.161	0.325
		NDCG	0.611	0.722	0.550	0.684	0.594	0.711
IES-FCA$_u$	v1.0	F1	0.335	0.563	0.333	0.436	0.334	0.526
		MAP	0.399	0.595	0.376	0.457	0.392	0.556
	v1.2	F1	0.326	0.545	0.319	0.434	0.324	0.513
		NDCG	0.736	0.839	0.703	0.786	0.726	0.824

TABLE 5 and 6 present the $F1-measure$ and $NDCG$ results on ESBM v1.2 for the comparison approaches. Another three latest approaches [26], [27], [29] are added into the comparison. Note that our proposed approach shows the superiority over other approaches in the majority of settings. Typically, compared with BAFREC, the $F1-measure$ and $NDCG$ improvement can be raised up to 6.58% and 4.12% on the DBpedia dataset with the setting of $k=5$, respectively. On the LinkedMDB dataset, the difference between IES-FCA and ESA is negligible with the setting of $k=5$ on $F1-measure$. In several settings, although IES-FCA is inferior to BAFREC and MPSUM on the LinkedMDB dataset, IES-FCA performs better than those approaches in most settings. Moreover, IES-FCA performs better on the DBpedia dataset than the LinkedMDB dataset. The reason for this phenomenon is that the objects of RDF triples on the LinkedMDB dataset are in the form of a specific number, while the objects in DBpedia dataset are composed of several meaningful words. Namely, IES-FCA can distinguish the relatedness among the objects of the RDF triples better on the DBpedia dataset than that on the LinkedMDB dataset. Similar with the results of weighting tests on ESBM v1.0, the results of IES-FCA ($\alpha=0.2$) on ESBM v1.2 are the best when $\alpha=0.2$ and better than IES-FCA in most settings.

TABLE 7 shows the results of ablation tests in terms of $F1-measure$, MAP, and $NDCG$ on both ESBM v1.0 and ESBM v1.2. Clearly, it is concluded that the experimental results that only consider $uniqueness$ factor are better than the results that only consider $redundancy$ or $importance$ factor in Equation (6). Besides, the $redundancy$ factor has slight impact on the results of entity summarization, due to many triples of the entity have no objects in common. For instance, when the $uniqueness$ factor is considered only, the results of $F1-measure$ and MAP on ESBM v1.0 reach to 0.526 and 0.556 respectively, which is higher than the results with the consideration of $redundancy$ or $importance$ factor. If the $redundancy$ factor is considered only, the $F1-measure$ value (0.325) and $NDCG$ value (0.711) on ESBM v1.2 are lower than the results that only one of other two factors is taken into account.

Although, the effectiveness of entity summarization on ESBM v1.2 in several settings shows unsatisfactory results, overall, IES-FCA performs better entity summarization results than KAFCA and other approaches in most settings. Note that, for all entities on ESBM v1.0 and ESBM v1.2, IES-FCA shows the superiority over other approaches on the $F1-measure$, MAP and $NDCG$. The weighting tests illustrate that assigning higher weights to $uniqueness$ factor can facilitate the performance of entity summarization but other factors are equally indispensable. The ablation study verified the rationality and effectiveness of each factor in Equation (6). The $uniqueness$ factor has bigger influence on the results of entity summarization than $redundancy$ and $importance$ factors. In terms of the efficiency of entity summarization, IES-FCA outperforms KAFCA on ESBM v1.0 and ESBM v1.2.

6 Conclusions

This paper presents an efficient Incremental Entity Summarization approach by utilizing FCA, named IES-FCA. Through FCA, the underlying relationships between predicates and objects in RDF descriptions of entity can be discovered, which has been proved to be promising in entity summarization. Specifically, we have firstly formulated the problem of incremental entity summarization and applied an incremental algorithm of concept lattice construction to entity summarization with FCA. Moreover, we have verified the correctness of our proposed method mathematically. In terms of efficiency, the experimental results indicate that our approach performs better than KAFCA, a state-of-the-art method for entity summarization. Under the best conditions, the efficiency of incremental entity summarization can be increased up to 8.7% than KAFCA for all entities. Further, for the RDF descriptions of the entity that has the largest number of predicates, the efficiency improvement of entity summarization is up to 67%, compared to KAFCA. Also, IES-FCA can achieve better summarization results than KAFCA and other state-of-the-art approaches in terms of $F1-measure$, MAP and $NDCG$. As for the future work, we are going to study further more complex situations of incremental entity summarization, such as the objects increment, predicates and objects increment simultaneously. In addition, to improve the performance on entity summarization, we plan to investigate more fine-grained ranking algorithms via considering the hierarchy of FCA and various types of entities. Also, it would be interesting to summarize and re-rank triples by automatically deciding k and further optimize the results of entity summarization. Concretely, the k can be trained by using deep reinforcement learning with the comprehensive consideration of the $importance$, $redundancy$, and $uniqueness$ on triples.

Acknowledgment

This work was funded in part by the National Natural Science Foundation of China (Grant No. 61702317), the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 840922, and the Fundamental Research Funds for the Central Universities (GK202103080). This work reflects only the authors' view and the EU Commission is not responsible for any use that may be made of the information it contains.

References

[1] S. Zwicklbauer, C. Seifert, and M. Granitzer, "Robust and collective entity disambiguation through semantic embeddings," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 425–434.

[2] A. Pappas, G. Troullinou, G. Roussakis, H. Kondylakis, and D. Plexousakis, "Exploring importance measures for summarizing rdf/s kbs," in *European Semantic Web Conference*. Springer, 2017, pp. 387–403.

[3] D. Diefenbach and A. Thalhammer, "Pagerank and generic entity summarization for rdf knowledge bases," in *European Semantic Web Conference*. Springer, 2018, pp. 145–160.

[4] K. Gunaratna, K. Thirunarayan, A. Sheth, and G. Cheng, "Gleaning types for literals in rdf triples with application to entity summarization," in *European Semantic Web Conference*. Springer, 2016, pp. 85–100.

[5] X. Fu, X. Ren, O. J. Mengshoel, and X. Wu, "Stochastic optimization for market return prediction using financial knowledge graph," in *2018 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 2018, pp. 25–32.

[6] T. Yu, J. Li, Q. Yu, Y. Tian, X. Shun, L. Xu, L. Zhu, and H. Gao, "Knowledge graph for tcm health preservation: design, construction, and applications," *Artificial Intelligence in Medicine*, vol. 77, pp. 48–52, 2017.

[7] D. Song, F. Schilder, S. Hertz, G. Saltini, C. Smiley, P. Nivarthi, O. Hazai, D. Landau, M. Zaharkin, T. Zielund *et al.*, "Building and querying an enterprise knowledge graph," *IEEE Transactions on Services Computing*, 2017.

[8] C. Lu, P. Laublet, and M. Stankovic, "Travel attractions recommendation with knowledge graphs," in *European Knowledge Acquisition Workshop*. Springer, 2016, pp. 416–431.

[9] S. Allison-Cassin, A. Armstrong, P. Ayers, T. Cramer, M. Custer, M. Lemus-Rojas, S. McCallum, M. Proffitt, M. Puente, J. Ruttenberg *et al.*, "Arl white paper on wikidata: Opportunities and recommendations," 2019.

[10] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia-a crystallization point for the web of data," *Journal of web semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[11] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, "Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames," in *International semantic web conference*. Springer, 2016, pp. 177–185.

[12] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A large ontology from wikipedia and wordnet," *Journal of Web Semantics*, vol. 6, no. 3, pp. 203–217, 2008.

[13] M. P. Consens, "Managing linked data on the web: The linkedmdb showcase," in *2008 Latin American Web Conference*. IEEE, 2008, pp. 1–2.

[14] E. Kim and K.-S. Choi, "Entity summarization based on formal concept analysis," in *1st International Workshop on Entity REtrieval, EYRE*, 2018.

[15] S. Roongsangjan, T. Sunetnanta, and P. Mongkolwat, "Using fca implication to determine the compliance of model practice implementation for software process," in *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, 2017, pp. 64–70.

[16] A. Shatnawi, A.-D. Seriai, and H. Sahraoui, "Recovering software product line architecture of a family of object-oriented product variants," *Journal of Systems and Software*, vol. 131, pp. 325–346, 2017.

[17] A. Buzmakov, E. Egho, N. Jay, S. O. Kuznetsov, A. Napoli, and C. Raïssi, "On mining complex sequential data by means of fca and pattern structures," *International Journal of General Systems*, vol. 45, no. 2, pp. 135–159, 2016.

[18] J. Poelmans, P. Elzinga, S. Viaene, and G. Dedene, "Formal concept analysis in knowledge discovery: a survey," in *International conference on conceptual structures*. Springer, 2010, pp. 139–153.

[19] F. Dau, J. Ducrou, and P. Eklund, "Concept similarity and related categories in searchsleuth," in *International Conference on Conceptual Structures*. Springer, 2008, pp. 255–268.

[20] B. Ganter and R. Wille, "Formal concept analysis–mathematical foundations berlin," 1999.

[21] G. Cheng, T. Tran, and Y. Qu, "Relin: relatedness and informativeness-based centrality for entity summarization," in *International Semantic Web Conference*. Springer, 2011, pp. 114–129.

[22] M. Sydow, M. Pikuła, and R. Schenkel, "The notion of diversity in graphical entity summarisation on semantic knowledge graphs," *Journal of Intelligent Information Systems*, vol. 41, no. 2, pp. 109–149, 2013.

[23] K. Gunaratna, K. Thirunarayan, and A. P. Sheth, "Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering," 2015.

[24] D. Xu, L. Zheng, and Y. Qu, "Cd at ensec 2016: Generating characteristic and diverse entity summaries." in *SumPre@ ESWC*, 2016.

[25] A. Thalhammer, N. Lasierra, and A. Rettinger, "Linksum: using link analysis to summarize entity data," in *International Conference on Web Engineering*. Springer, 2016, pp. 244–261.

[26] H. Kroll, D. Nagel, and W.-T. Balke, "Bafrec: balancing frequency and rarity for entity characterization in linked open data," in *1st International Workshop on Entity REtrieval, EYRE*, 2018.

[27] D. Wei, S. Gao, Y. Liu, Z. Liu, and L. Hang, "Mpsum: entity summarization with predicate-based matching," *arXiv preprint arXiv:2005.11992*, 2020.

[28] S. Pouriyeh, M. Allahyari, K. Kochut, G. Cheng, and H. R. Arabnia, "Es-lda: entity summarization using knowledge-based topic modeling," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017, pp. 316–325.

[29] D. Wei and Y. Liu, "Esa: entity summarization with attention," *arXiv preprint arXiv:1905.10625*, 2019.

[30] F. Hasibi, K. Balog, and S. E. Bratsberg, "Dynamic factual summaries for entity cards," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 773–782.

[31] M. Tasnim, D. Collarana, D. Graux, F. Orlandi, and M.-E. Vidal, "Summarizing entity temporal evolution in knowledge graphs," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 961–965.

[32] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang, "k-clique community detection in social networks based on formal concept analysis," *IEEE Systems Journal*, vol. 11, no. 1, pp. 250–259, 2015.

[33] L. Zou, Z. Zhang, and J. Long, "A fast incremental algorithm for constructing concept lattices," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4474–4481, 2015.

[34] Q. Liu, G. Cheng, K. Gunaratna, and Y. Qu, "Esbm: an entity summarization benchmark," in *European Semantic Web Conference*. Springer, 2020, pp. 548–564.

[35] Y. Yang, F. Hao, B. Pang, K. Qin, Z. Pei, and B. Li, "A quick algorithm on generating concept lattice for attribute-incremental streaming data," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2019, pp. 2811–2816.

[36] Q. Liu, Y. Chen, G. Cheng, E. Kharlamov, J. Li, and Y. Qu, "Entity summarization with user feedback," in *European Semantic Web Conference*. Springer, 2020, pp. 376–392.

Erhe Yang received the B.Sc. degree in Oil-Gas Storage and Transportation Engineering from Shenyang University of Chemical Technology, China, in 2016. He is currently working toward the M.Sc. degree at the School of Computer Science, Shaanxi Normal University, China. His current research interests include social computing, formal concept analysis, and knowledge graph.

Fei Hao received the B.Sc. degree in Information and Computing Science and the M.Sc. degree in Computer Software and Theory from Xihua University, China, in 2005 and 2008, respectively, and the Ph.D. degree in Computer Science and Engineering from Soonchunhyang University, South Korea, in 2016. Since 2016, he has been with Shaanxi Normal University, Xi'an, China, where he is an Associate Professor. He is currently taking a Marie Sklodowska-Curie Individual Fellowship at the University of Exeter, Exeter, United Kingdom. His research interests include social computing, ubiquitous computing, big data analysis and processing and mobile cloud computing.

Yixuan Yang received the B.Sc. degree in Software Engineering from Taiyuan University of Technology, China, in 2017. She received the M.Sc degree in Software Engineering from Shaanxi Normal University, China, in 2020. Her research interests include social computing and formal concept analysis.

Carmen De Maio graduated and received the Ph.D. degree in Computer Sciences, both from the University of Salerno, Italy, in 2007 and 2011, respectively. From 2007 until now, she collaborates to several research initiatives mainly focused on Knowledge Extraction and Management from structured and unstructured data defining intelligent systems based on the combination of techniques from Soft Computing, Semantic Web, areas in which she has many publications. Specifically, she has been deeply involved in several research projects and she has published extensively about Fuzzy Decision Making, Ontology Elicitation, Situation and Context Awareness, Semantic Information Retrieval. Recently, she is working in the field of Social Media Analytics and Semantic Web to define intelligent features such as microblog summarization and context aware information retrieval. In 2014, she is an assistant professor at Department of Computer Science at University of Salerno.

Aziz Nasridinov received the B.Sc. degree from the Tashkent University of Information Technologies in 2006 and the M.Sc. and Ph.D. degrees from Dongguk University in 2009 and 2012, respectively. He is currently an Associate Professor of computer science with Chungbuk National University. His research interests include databases, big data, data mining, and distributed processing system applications.

Geyong Min received the BSc degree in computer science from the Huazhong University of Science and Technology, China, in 1995, and the PhD degree in computing science from the University of Glasgow, United Kingdom, in 2003. He is a Professor of High Performance Computing and Networking in the Department of Mathematics and Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. His research interests include next generation internet, wireless communications, multimedia systems, information security, ubiquitous computing, modelling, and performance engineering.

Laurence T. Yang received the B.E. degree in Computer Science and Technology from Tsinghua University, China and the PhD degree in Computer Science from University of Victoria, Canada. He is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology, China, and in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data. His research has been supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation.