Knowledge-Based Systems 294 (2024) 111671

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Using dynamic knowledge graphs to detect emerging communities of knowledge

Joao T. Aparicio [a,b,*], Elisabete Arsenio [b], Francisco Santos [a], Rui Henriques [a]

[a] INESC-ID and Instituto Superior Tecnico, Universidade de Lisboa, Lisbon, Portugal
[b] Department of Transport, LNEC, Lisbon, Portugal

ARTICLE INFO

Keywords:
Knowledge graphs
Dynamic
Community finding
Network science
Text2kg
KDD

ABSTRACT

Knowledge graphs represent relationships between entities. These graphs can take dynamic forms to trace changes along time through text models and further used by reasoning systems with the intent to answer queries. In this research we explore their applicability for extracting temporal patterns of knowledge in the form of communities. To this end, we propose a method for generating knowledge relationships over unconnected components of a knowledge graph, allowing for a targeted exploration of emerging contents in corpora. This analysis is applied to the corpora of the Conference on Knowledge Discovery and Data Mining (KDD) publications over the last decade. We find the key knowledge communities over time and rank the underlying concepts. Results show that the publication efforts increasingly focus on graph research and the creation of relationships instead of new concepts. The acquired results confirm the validity of the proposed knowledge discovery methodology for community-centered analysis of emerging changes in dynamic knowledge graphs.

1. Introduction

The creation and completion of knowledge graphs (KG) has been increasingly important in recent years [1–3] as search and knowledge engines require these structures to perform reasoning tasks. Although the existing literature has made significant progress on the generation and optimization of static knowledge graphs for reasoning tasks [4,5], the capacity to tap into the dynamic character and longitudinal evolution of knowledge within specific domains is not yet mature [6].

The application of network science methodologies may prove to be pertinent in the examination of the evolution of knowledge structures [7]. This can be achieved by the quantification of structural changes within knowledge networks. This critical void deprives a variety of stakeholders–from academia to industry and government–of the means to comprehend how expertise and knowledge evolve over time, a deficiency that could have repercussions for innovation and strategic decision-making. To address this gap, this work focuses on the following question: How can network science be used to trace the evolution of knowledge within a dynamic knowledge graph? And, can this analysis be used to generate high impact knowledge candidates? Grounded on techniques from network science and community detection, our study introduces a new analytical framework that shifts the paradigm from static reasoning to dynamic topological and temporal analysis. We do not limit ourselves to theoretical contributions; rather, we apply this framework to papers from the KDD Conference in a real-world case study. In doing so, we not only open a new frontier for knowledge graph research, but also provide actionable insights that are crucial for the success of organizations across multiple industries.

This research is focused on capturing not only trends in knowledge graphs, but on the evolution of knowledge in a particular field. In this context, a tool to these ends can help relate abstract ideas [8] based on network science principles, being of notable relevance. The knowledge evolution is based on formal knowledge advancement and emphasizes group knowledge creation through knowledge graphs. Consequently, it is essential to consider methods for tracing changes in ideas when attempting to capture the development of collective knowledge. In network science, concepts are represented as clusters of words due to the fact that communities share and enhance concepts through the use of language in their discourse in order to advance collective knowledge [9]. The research in graph generation and evolution is quite recent and growing at a fast pace [10].

The behavior of entities and how they contribute to the formation of facts over time can be better understood using dynamic knowledge graphs that capture temporal dependencies between facts in addition to

* Corresponding author at: INESC-ID and Instituto Superior Tecnico, Universidade de Lisboa, Lisbon, Portugal.
E-mail address: joao.aparicio@tecnico.ulisboa.pt (J.T. Aparicio).

https://doi.org/10.1016/j.knosys.2024.111671
Received 22 March 2023; Received in revised form 20 January 2024; Accepted 18 March 2024
Available online 29 March 2024

the relational (structural) dependencies [11]. Two major distinct definitions of temporal knowledge graphs can be found in the literature. First, Temporal Knowledge Graphs (TKGs) [11,13], are structures that tie each piece of information or occurrence to a definite timestamp, reflecting the knowledge state at that exact point. They are technically defined as G={(e_1, e_2, r, t), where,}, where every triple contains a subject entity, a relation, an object entity, and a specific time point (t). This architecture records the chronological progression of interrelations.

On the other hand, Discrete Time Dynamic Graphs (DTDGs) [12] present information in separate, unchanging segments, each symbolizing the conditions at a certain period. A DTDG consists of multiple static graphs G = {G1,G2,…,GT}, with each Gt = {(e_1,r,e_2): e_1,e_2 ∈ E,r ∈ R} containing nodes and connections applicable for the period . This approach facilitates interval-based scrutiny, providing a detailed but segmented perspective on the development of knowledge. Here, E is the set of all entities and R is the set of all relationships. Each triple in Gt is valid at time t. For example, if G1 contains the triple (“Paris”, “is capital of”, “France”), this triple is valid at the time represented by G1. This definition captures the dynamic nature of knowledge graphs by arranging them as a sequence of static snapshots, each representing the state of the world at a specific time.

Our study particularly implements the DTDG definition, recognizing its strategic advantage in temporal analysis. In this sense, we aim to use network community finding and network metrics over time to understand the evolution of knowledge in a specific context. This means that the focus of this analysis deviates from the classic reasoning focused approach. Instead, it focuses on looking at evolving topological features and network science techniques to grasp characteristics of meta-knowledge. Meta-knowledge refers to higher-level information that describes the structural and semantic properties of the graph, rather than the individual data points (nodes and edges) themselves. This includes the identification of communities within the graph, which are clusters of nodes that are more densely interconnected with each other than with the rest of the graph. Meta-knowledge also captures centrality measures that identify the most influential or well-connected nodes within these communities or the graph at large. These higher-order features provide a macroscopic view of the knowledge graph, offering insights into its overall architecture and the roles or importance of specific entities within it. a technical contribution of this idea is the analysis of evolving topological features of knowledge as well as a baseline for generative knowledge graph-based impact recommender system (Fig. 3). In the field of recommender systems, the pursuit of enhancing connections within Knowledge Graphs (KGs) is pivotal for driving innovation and interpretability. Traditional methods, heavily reliant on singular connected components, limit the breadth and depth of recommendations. Our approach, delineated in this paper, introduces an algorithm—the Knowledge Connector Candidate (KC2)—that transcends these constraints by targeting the most influential connections across weakly linked subgraphs. By leveraging the power of logarithmic normalization of centrality, our method not only identifies but also prioritizes the integration of previously isolated knowledge components.

This meta-knowledge discovery is important because it can help people in business, science, medicine, academia, government, and other fields to extract information from datasets [14]. To this end, the paper’s main contributions are two fold. First, it proposes a dynamic knowledge graph longitudinal analysis based on network science and community detection principles. Allowing us to understand the evolution and size of different Knowledge domains, distribution of degrees, and potential connections among knowledge communities. Secondly, the instantiation of this analysis on a particular context and corresponding implications. This knowledge understanding is vital for any organization’s success [15]. In particular, we apply this analysis to the papers from KDD (Conference on Knowledge Discovery and Data Mining) since the focus on this case study allows us to acquire meta-knowledge on research about knowledge discovery. Hence by analyzing such relevant proceedings, that devote by several decades on the topic of knowledge understanding and discovery, results unveil future impactful scientific connections.

The paper is structured in five sections, starring by providing an overview of the context and aims of the paper, in section one. In section two, it presents the literature review. The third section conveys the methodological approach used to achieve the objectives of the study. The fourth section presents the empirical part of the study, comprehending the results and discussion. Finlay, the fifth section presents the main conclusions, contributions, implications of the present study.

2. Background and related work

2.1. Knowledge graph construction

The construction of knowledge graphs has been suggested as a promising area of study for the improvement of search and question-answering engines [16–18]. A paper by Khapra et al. [19] examines the challenge of automatically generating question-answer pairs from a given knowledge network. Using existing biology literature, Lamurias et al. [20] aimed to construct a knowledge network for tolerogenic cell therapy automatically. Under this idea of knowledge graph construction, Yochum et al. [21] are developing a knowledge-based recommendation algorithm for the travel industry. Recent research by Krinkin et al. [22] discusses the problems now faced by cable TV operators’ telecommunications network monitoring systems and the models used to resolve these problems. In that proposal for user monitoring, a knowledge graph is generated based on a set of ontological representations and is used to integrate knowledge from different sources and models. Knowledge graphs, which provide context regarding genuine and informal interactions between items in the real world, are important if semantically rich data formats are to be used in intelligent real-world situations. This need is based on the importance of knowledge graphs. Rezaei et al. [23] provide a strategy for constructing knowledge graphs that meet this objective in the research. Crowdsourcing techniques like task delegation and reverse captcha generation have been used to build a knowledge graph in the realm of educational systems [24]. The most recent generation of knowledge graphs lacks a clear representation of the information offered in research papers. In light of this, Dessi et al. [4] offer an architecture for extracting entities and relationships from research papers and integrating them in a large-scale knowledge network using Natural Language Processing and Machine Learning methodologies. Du et al. contributed to the literature on constructing knowledge graphs by building a method that indexes organized and unstructured content on a single knowledge graph [25]. Venkatesh developed an original cloud-based method for establishing a knowledge graph. This was done in order to produce a unified body of knowledge [26] by relating data with both structured and unstructured formats. Kittenberger’s [27] method for including and managing ambiguity insertion in knowledge graphs is another important and possibly game-changing piece of work. When it comes to creating knowledge graphs, KnowGL is a recent and high-preforming algorithm to generate triplets from phrases [28]. Unlike the previously mentioned works, the research presented in this paper accounts for the temporal (chronological) component of the existing literature through generating dynamic knowledge graphs. Zhang et al. [16], Li et al. [17], and Veena et al. [18] build knowledge graphs for search and Q&A, but neglect temporal aspects, which our research incorporates. As in Khapra et al. [19] focus on QA pair generation within knowledge networks, omitting the dynamic capabilities we introduce. However, Lamurias et al. [20] craft domain-specific biomedical knowledge graphs but lack a generalized approach, which our work provides. As well, Yochum et al. [21] specialize in travel recommendations but do not adapt to other contexts, an adaptability our research aims for. Krinkin et al. [22] create knowledge graphs for cable TV network monitoring, lacking broader applicability, whereas our method is more general. Rezaei et al. [23] prioritize semantic richness but are not dynamically adaptive, a gap our study fills. Dessi et al. [4] extract research paper entities but overlook temporal research trends, which we capture. Other studies, Du et al. [25] and Venkatesh [26] integrate structured and unstructured data but are not dynamically evolving, a feature we incorporate. Kittenberger [27] manages knowledge graph ambiguity but excludes time-sensitive data, which our approach includes.

2.2. Dynamic knowledge graphs

Studies on the generation of dynamic knowledge graphs or dynamic knowledge graph creation (KGC) are scarce, yet rapidly emerging. To this purpose, Trivedi et al. [11] propose Know-Evolve, a deep evolutionary knowledge network that acquires and adapts non-linearly changing entity representations. Using knowledge-to-text comprehension and a low-variance gradient estimator for discrete latent variable models, Das et al. [29] build upon this research. To tackle the aforementioned problem, Wu et al. [30] propose a context-aware Dynamic Knowledge Graph Embedding (DKGE) approach that enables the embedding of learning in a web-based application, this kind of application uses the dynamic features of the graph to perform link prediction and question answering. Using a dialogue corpus from a well-known television series, Tuan et al. [31] provide a method for constructing knowledge graphs using conversational comprehension tasks (DyKgChat). To contribute to the current literature, Xie et al. [32] present a new deep recurrent neural network-based dynamic KGC model that depends on structural and textual features equally (DKGC-JSTD) allowing for a topological dependence on the completion of the knowledge graph. Most current KGE methods do not take into account how the structure of dynamic knowledge graphs has evolved over time (DKGs). To solve this issue, Tang et al. [33] suggest a solution called Timespan-aware Dynamic knowledge Graph Embedding Evolution (TDG2E), which accounts for the evolving nature of DKGs across time. Tay et al. [34] propose Parallel Universe TransE (puTransE), which consists of an adaptation of a translational model to make more precise predictions about connections within evolving networks of information. The current work will extend the concept of dynamic knowledge graphs by using KnowGL, the best method for generating static knowledge graphs at the present time [28]. This extension is justified since the knowledge on the papers is assumed in equal intervals (yearly) and the descriptive analysis would be impacted by the link prediction over the aggregate graph. In that sense, we use only the knowledge described within the papers.

Among the works dealing with dynamic knowledge graphs, Trivedi et al. [11] make an interesting proposal with Know-Evolve but fall short of providing a robust methodology for the actual construction of these dynamic graphs. Das et al. [29] and Wu et al. [30] touch upon some of the same challenges but do not integrate historical data in a meaningful way. Tay et al. [34] propose puTransE but limit its scope to making predictions within evolving networks, without considering a method for creating these dynamic networks. While Tang et al. [33] do account for the temporal aspects of dynamic knowledge graphs, their methodology is not generalizable for all types of graphs, which is a gap our work aims to fill.

2.3. Pattern finding in dynamic knowledge graphs

Another relevant challenge in this area is finding patterns in dynamic knowledge graphs. Patterns in dynamic knowledge graphs are significant because they can be used to describe the local characteristics of dynamic networks and forecast future behavior [35]. Some of these patterns can be seen as sub-graphs contained inside a larger knowledge graph. This is done on a massive search space of seed candidates, which means a lot of time is wasted looking for candidates that are not likely to be efficient when attempting to find isomorphic subgraphs covering a given area [36]. Subgraph indexed sequential subdivision is a technique suggested to address these difficulties by accelerating the process of continuously matching subgraphs on dynamic knowledge graphs [37]. This contribution builds upon several others. Borgwardt et al. [38] consider how pattern mining on static graphs may be applied to time series of graphs. Blin et al. [39] offered a precise technique for querying graph patterns based on dynamic programming and color-coding.

To classify both static and dynamic networks, Gehweiler et al. [40] provide a distribution heuristic that relies only on limited, locally observable information. To find similar concepts in ontologies, Benik et al. [41] proposed a technique for annotation (taxonomy) graph analysis based on a combination of tree distance metrics and the inspection of a dense subgraph. Agarwal et al. [42] provide methods for identifying time-sensitive occurrences in microblog posts (expressed as knowledge graphs). This allows for instant notification of such events depending on the density of individual subclusters. To better understand dynamic networks and, in particular, the temporal behavior of vertices using betweenness centrality through time, Fairbanks et al. [43] suggest a technique for mixing sparse parallel graph algorithms with dense parallel linear algebra algorithms. While not the focus of this section, it's worth mentioning that graph summarization techniques [44] can also play a role in pattern mining. These methods aim to simplify the graph while preserving its essential structural and semantic properties, thus potentially aiding in the efficient discovery of meaningful patterns. Shah et al. [44] demonstrate how to do so by lowering the encoding cost inside a data compression paradigm. They then apply this method to the construction of TimeCrunch, a scalable and parameter-free tool for finding dynamic graph summaries. To bridge the knowledge gap between human tasks and current methodologies, Gao et al. [45] suggested an end-to-end video classification system based on a structured knowledge graph replicating dynamic knowledge accumulation in movies over time. Using an iterative recognition of atomic changes and information gain, Kapoor et al. [46] provide an interestingness metric on elements of the KGs. The idea of minimal description length inspired this research (MDL).

Complementary, summarization tasks on KGs are based on finding groups of nodes that are highly cohesive. In this sense, we may extend the idea of community-finding in graphs to knowledge graphs as well. Zakrzewska's work [47] extend the concept of community-finding in a dynamic graph scenario. The process of locating communities in dynamic graphs may be divided into the following steps: 1) application of a greedy static technique that optimizes modularity. 2) keeping a record of merging history.

In the realm of pattern discovery within knowledge graphs, existing works like those by Qin [35], Nguyen [36], and Borgwardt et al. [38] focus primarily on static graphs or the computational aspects of dynamic graphs but overlook the significance of the structural and semantic changes over time. Gao et al. [45] venture into the territory of time-sensitive knowledge graphs but their work is based on LSTM (Long-Short Term Memory) and Attention mechanisms just like many of the reasoning systems that are seen. Kapoor et al. [46] introduce an interestingness metric, but their methodology is focused on quantifying informativeness of a pattern for summarizing, instead is not suitable for generating potential important relationships that impact the structure of knowledge. Even though, we have seen algorithms [47] for finding communities in dynamic graphs, we have not seen this applied in dynamic knowledge graphs.

2.4. DKG applications

In terms of applications of such research to text reasoning tasks, Choudhury et al. [48] offer an end-to-end framework for building bespoke knowledge graph-driven analytics. Lang et al. [49] present a semantic knowledge reasoning graph model based on the multidimensional axiomatic fuzzy sets (AFS). Another proposal built a Narrative Analytics-Assisted System (NAAS) that uses a knowledge graph [50]. In order to produce the knowledge graph (KG) and construct reasoning routes for reading comprehension tasks through unsupervised learning.

We also have seen other application in text mining, namely search query finding [51], chatbot development, data integration, and semantic search [52]. We also have seen the applications of knowledge graphs on text mining in the construction of interpretable features that are of potential use for the task of text classification [53]. Lastly we see KGs being used for entity linking and entity retrieval of different types of texts [54, 55] as well as creating embedding representations [56,57].

In addition, we have seen DKGs being used in the geoscience field to perform knowledge discovery of encyclopedic discipline knowledge [58]. KGs are also being utilized in today's trends in integrating collected, analyzed, and managed pavement knowledge assets, which is fundamental to the problem solving process in pavement engineering [59]. We have also seen that it can be used for reasoning over geoscience metadata repositories [60]. This has also been seen as essential for making geographic data accessible for the Semantic Web and machine learning [61] as well as efficient storage and retrieval systems for archival data [62]. This idea is also extended to manage the relationships between geographic entities and derive other relationships [63].

Applications in other domains are also seen. Zhao et al. [64] developed knowledge graphs to aid software developers and project managers in making sense of software repositories. Tolerogenic cell therapy is one area where we see instances of knowledge networks being automatically constructed from studies published in biomedical literature [20]. Through the use of crowdsourcing and reverse captchas, Weng et al. developed a knowledge graph in the area of educational systems [24]. In the field of computer vision we see Kalanat et al. [65] use a scene graph, a graph representation of an image, to capture visual components and features.

2.5. Related studies in knowledge graphs

Advancements in knowledge graph refinement techniques have proliferated their use across various domains. These techniques range from data pruning to gap-filling methods [66], automated knowledge base generation from electronic medical data [67], and employing commonsense knowledge for natural language understanding [68]. There are also efforts focused on precise recommendations using Knowledge-aware Graph Neural Networks (KGNNLS) [69], knowledge graph completion through pre-trained language models [70], and textual inference frameworks for answering intuitive queries [71].

Recent advancements in graph stream summarization have led to significant improvements in real-time graph analytics. The work by Jia et al. [84] introduces a method for persistent graph stream summarization (PGSS), which allows for efficient querying of graph streams over arbitrary historical time ranges. The proposed approach, grounded on hierarchically organized hashmaps, shows substantial accuracy and efficiency over traditional methods. The proposed PGSS-BDH and PGSS-MDC sketches can significantly enhance query performance, making it a promising solution for large-scale dynamic graph analysis. In the domain of image clustering, the study by Gao et al. [85] presents Gomic, a technique that leverages self-supervised learning within heterogeneous graphs. This approach facilitates multi-view image clustering and exhibits the potential to outperform existing methods in terms of both clustering accuracy and computational efficiency. The co-learning aspect of the method can provide new insights into the community detection within knowledge graphs. These studies highlight the dynamic nature of graph and image data processing, paving the way for more sophisticated analytical tools in knowledge graph evolution and community detection.

Recent review works have systematized the fundamental concept of knowledge reasoning [72], provided summaries and future directions [73], and even explored taxonomies in the field [73]. The work by Ojo et al. [74] focuses on finding hidden relationships between datasets through a knowledge trees. Despite the progress, existing works have limitations when it comes to considering the dynamic and temporal aspects of knowledge graphs. For instance, [66] lacks focus on dynamic changes over time, while [67] is limited to medical data. Similarly, [68, 69] do not address the adaptability of models to evolving data. This gap represents a significant shortcoming, particularly in the context of our research that aims to study the temporal aspects of network science in knowledge graphs. Our work seeks to extend the existing body of knowledge by focusing on the temporal dynamics, thereby filling a crucial gap in the literature.

3. Methods

3.1. Understanding the gap

Taking into account the previously referenced literature, we believe it is relevant to obtain an overview of the research that was carried out in this context. Using the Scopus database and the following query: ALL ("knowledge graph" AND "generate") and filtered only articles that are in their final form in proceedings and journals, eliminating drafts or articles under review, we are able to extract the articles and conference papers pertaining to the aforementioned fields. Using co-occurrence of keywords on the resulting 1961 documents, we are able to build a keyword network that shows us the sub-areas and related areas within this field. To better understand this network, we performed community finding. Fig. 1 shows the edge density representation of this network where each color represents a community of keywords and the higher density color represents higher density of co-occurrence. Quite clearly, we can see that 'semantics' is a central term to this graph, even though it was not on the search query. Interestingly it is also in a high betweenness node (a bridge node) among different communities. Among the tree generated communities we have: 1) (on the top) the knowledge graph and semantics group, 2) (on the left) the machine learning and computational linguistics group, and 3) (on the right) the context of application of the previous techniques (where we can see with less emphasis keywords like: 'humans', 'controlled study' and 'covid-19'). This visualization helps us understand the gap in literature as it reveals a lack of studies in both community finding and dynamic knowledge graphs.

3.2. Building the knowledge graph

Using the Web of Science database, we extracted the SIGKDD conference publications between 2013 and 2021, including the year, title, authors and abstract. After accessing the core literature, we proceeded to design the experiment. Using KnowGL [28] on the extracted abstracts, we retrieved the triplets with entities and relations. The KnowGL parser provides RDF compliant knowledge through the usage of a domain-agnostic Knowledge Exploitation Patterns abstract the induced KG [75]. The resulting triplet structure has the following structure " [(subject mention # subject label # subject type) |relation label |(object mention # object label # object type)] " [28], allowing for the definition of annotations on the subjects and objects. An instance of this can be found in the following example: "Learning is not necessarily machine learning" is turned into [(machine learning#Machine learning#academic discipline) | different from | (Learning#Learning#biological process)]. This imposes the assignment of a type based on context, yet this may be redefined later on based on other edges or via the analysis of inconsistencies on the knowledge graph. Yet, that is not the focus of this study. Using the triplets, we built the knowledge graph by representing the entities as nodes and adding edges between them with the corresponding directed relation. To do so, we use the subject and object labels to minimize the redundancy of nodes that have the same meaning, instead of subject mention. This means that for instance in the following text: "We propose an ML algorithm." And "Machine learning is used for classification.", generates the following 2 triplets:

[(ML algorithm#Machine learning#academic discipline) |instance of |(algorithm#Algorithm#work)];

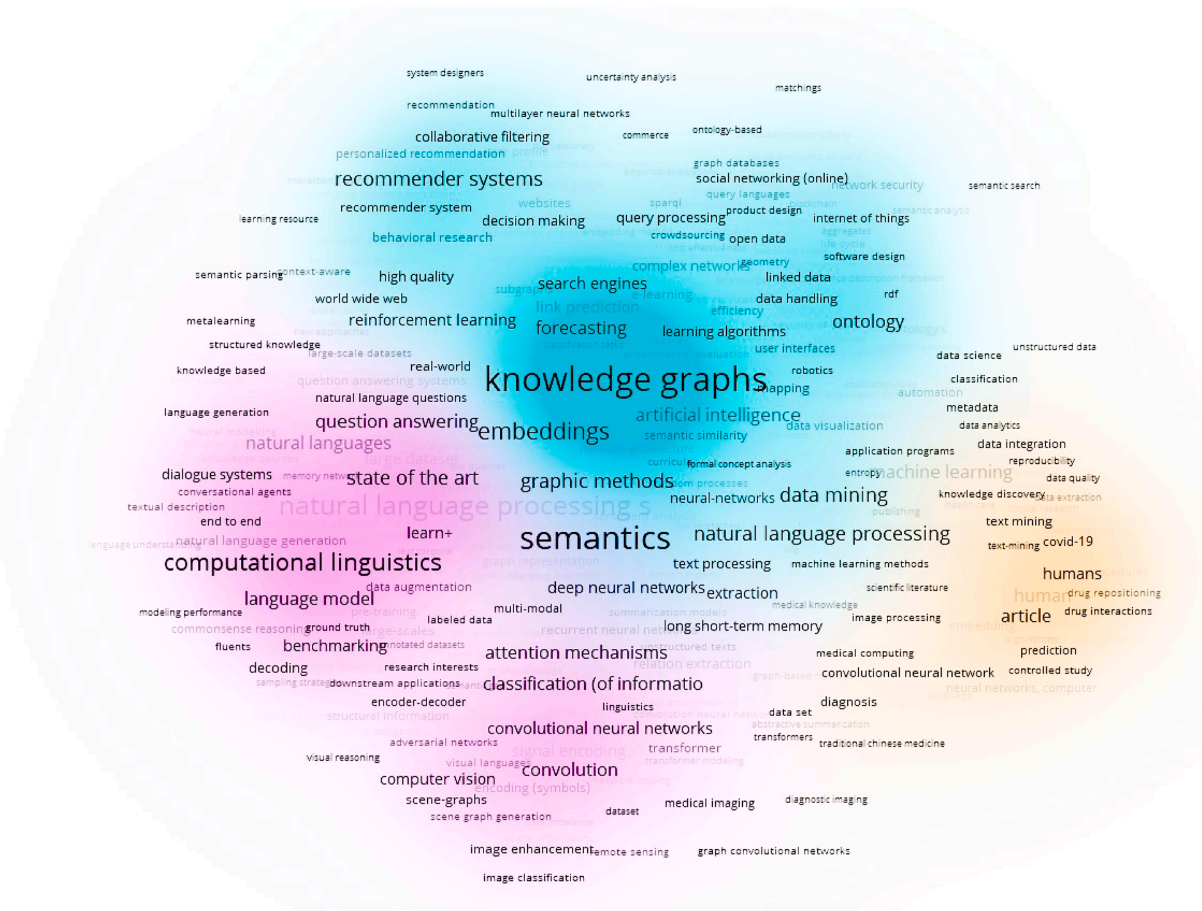

Fig. 1. Knowledge graph density representation shows 3 main clusters regarding research themes: blue, pink and orange.

[(classification#Statistical classification#machine learning) |part of|(Machine learning#Machine learning#academic discipline)]

This results in the 'Machine learning' node will be a single node with degree of 2 instead of a 2 nodes with a degree of one. Additionally, since this is based on a sequence-to-sequence model similar to BART [76] it is able to handle some typos and still generate subjects and objects on the correct label.

In the construction of our knowledge graph, we interpret the various mentions as aliases (skos:altLabel) of the entities, thereby enhancing the graph's semantic richness and ensuring consistency across diverse nomenclatures within individual documents. For example, a single node might aggregate multiple subject mentions such as [Machine Learning, ml, ML, learning], reflecting the exact references to a single concept (skos:exactMatch) based on structures like "Machine learning (ML)". When associating a node with a subject or object mention, such as 'learning', we first consult the entity's alias (skos:altLabel) from that document. This procedure allows us to determine whether the mention aligns with existing records, promoting ontological coherence. If a congruence is identified, we elect to circumvent the originally designated node from type and instead merge the entities within the node in the designated document. This methodology not only preserves semantic integrity but also fortifies the interconnectedness and comprehensive representation of concepts within the knowledge graph. Effectively classifying learning in the machine learning node, instead of a separate node. Yet, a limitation of the current approach is if the type is incorrectly assigned throughout an entire abstract and then joined into the full knowledge graph. This may cause the new knowledge to be added to the wrong node. This may be further studied in the future using better alternatives to label assignment than KnowGL or knowledge graph refinement techniques that are context sensitive. On the other hand, this method is generalizable to knowledge from many disciplines and is able to generate new entities and relation labels that have not been previously seen based on sentence structure.

Since the resulting structure can have several relations between the same pair of nodes, this is a MultiDiGraph (or a multilayer directed graph). The MultiDigraphs are generated by separating each set of abstracts by year, so we have one MultiDiGraph per year.

3.3. Analyzing the knowledge graph

The analysis of the resulting MuliDiGraphs is done by acquiring informative statistics and understanding how they evolve along time. A central concept is the degree $\langle k \rangle$ of a node, which is the number of neighbours of that node. Another important concept is betweeness centrality,

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t \mid v)}{\sigma(s,t)},$$

measuring how important a node is in connecting communities of nodes. V is the set of nodes in the network, $\sigma(s,t)$ is the number of shortest paths in a network and $\sigma(s,t \mid v)$ is the fraction that flow through the node v. A high betweenness centrality means that node is an important bridge between communities, and literally, it means the percentage of shortest paths in the network flow through that node, as defined in [77]. Closeness centrality measures how close a node is to every other, i. e.

$$C_c(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)},$$

the reciprocal of the average shortest path distance from any other node in the network to the node, u. Where, n is the number of nodes in the network and $d(v,u)$ is the distance between the nodes v and u.

In the discipline of network science, community finding is a promising clustering technique. A community is a set of densely interconnected nodes relative to their number of connections to external nodes. Thus, it is possible to divide a connected component into two or more communities [78]. We also used community finding based on modularity optimization of groups of nodes. This algorithm maximizes intra-cluster modularity,

$$Q = \sum_{c=1}^{n}\left[\frac{L_c}{m} - \gamma\left(\frac{k_c}{2m}\right)^2\right],$$

where, L_c are the links within a community, c is the number of communities, m is the total number of edges, is the resolution parameter, k_c are the degrees of the nodes in a community individually summed and then multiplied. The optimization process is the one defined by Clauset et al. [79]. This operation was done using the undirected collapsed version of the graph, since we can consider that if we know the relationship between two entities in one direction we can also consider that we know the inverse relationship.

To understand the knowledge communities over time, we measure how their size and main topic evolves along time in yearly graph, by looking at the communities of the knowledge of each year.

3.4. Extending the knowledge graph

Going beyond the giant weakly connected component over time, we aggregated the knowledge of the subgraphs. The following artifact then allows us to contribute to challenge of interpretable recommendations based on KGs, going beyond current KG based recommender systems [80]. The most common approaches include collaborative filtering, content-based filtering, or hybrid methods that often work within the context of a single (usually strongly) connected component. They are generally optimized for objectives like accuracy, diversity, or serendipity of recommendations within the scope of available connections in the graph. This proposal is a foundation for generative knowledge graph-based recommender systems (Fig. 2). We propose an algorithm for generating edges with the highest impact on knowledge graph connection of previously unconnected components, Knowledge Connector Candidate (KC2). The edge candidate generation algorithm is described below in Algorithm 1.

The main idea is to iteratively generate edges between weakly connected components and measure the impact of said edge by the C_C of the two connected nodes. Let, X and Y be the two previously unconnected components, I(x, y)

$$I(x,y) = log_2(C_C(x)) + log_2(C_C(y)),$$

where $x \in X$ and $y \in Y$ are the two candidate nodes for connection. The $\log_2$ are added to normalize the results an bring upward pairs of important nodes in smaller communities within a rank. This takes on the assumption that a higher impact comes from connecting two important terms from two sizable and previously unconnected components, through the most well-connected node. We propose this feature for selecting and preforming logarithmic normalization of centrality, that is a higher-order network property that KG recommender systems do not

Algorithm 1
Assign high impact candidate connections.

```
function EdgeCandidateImpact(G)
  wcc ← weaklyConnectedComponents(G) ▷ dict id: node list
  cc ← clossnessCentrality(G) ▷ dict node: closeness centrality
  maxCC ←
  for k,v ∈ wcc do
    maxCC[k] ← max(v,key ← lambda(x: cc[x]))
  end for
  for X ∈ range(len(wcc)) do
    for Y ∈ range(len(wcc)) do
      if X ≠ Y then
  x ← maxCC(X) ▷ max value of each set
        y ← maxCC(Y)
        if (l,k)/∈ dictCandidates.keys() then
          dictCandidates[x,y] ← I(x,y)
        end if
      end if
    end for
  end for
return dictCandidates
  end function
```

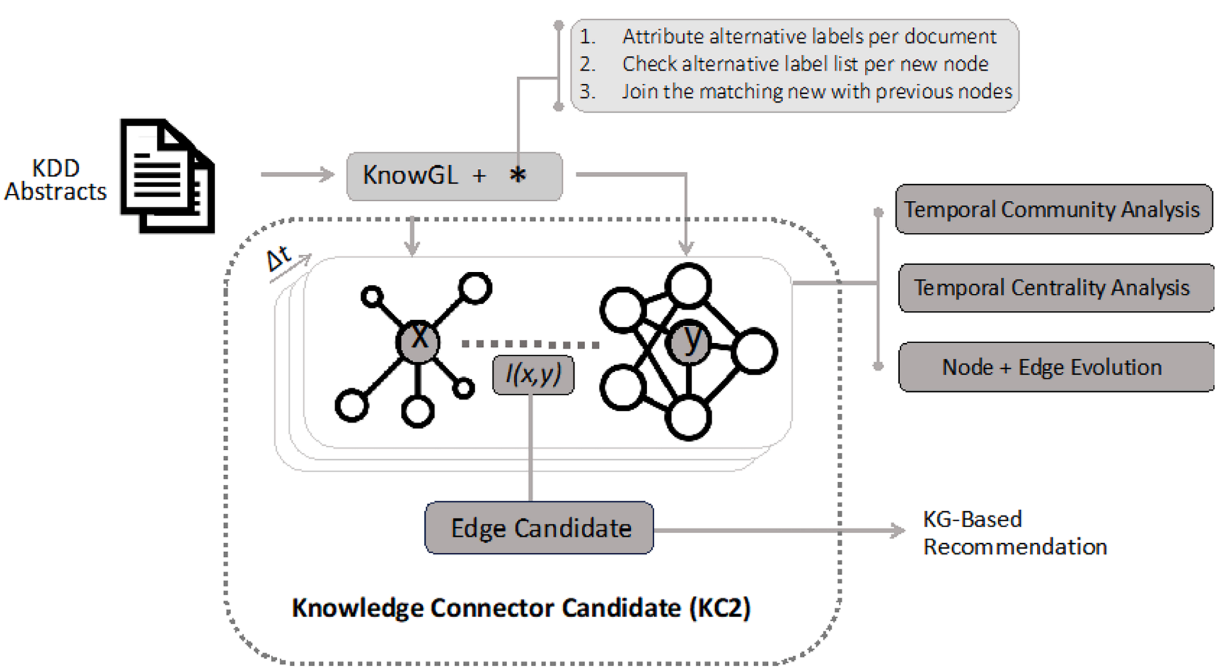

Fig. 2. Methodological approach.

consider [81]. The emphasis on closeness centrality as a measure of node importance for connecting different components is unique. KG recommender systems often use other forms of scoring, like personalized PageRank or matrix factorization, to rank items for individual users [81].

These can be then used for ideation of future projects by conference authors, as well as ideas for creating novel tracks within the conference. This study can be replicated in both scientific journals as well as in conference proceedings. For example, it can identify emerging areas of interest or flag declining trends in existing topics, thus providing actionable insights for the academic community. By adopting this technical approach, we aim to bridge the gap between community finding and knowledge to provide a more cohesive solution that provides pairs that maximize connectivity. This will allow us to derive meaningful narratives and potential knowledge triplets that can stimulate future work in various domains.

3.5. System design implications

The usage of communities within knowledge graph, that we can denote as communities of knowledge, has several implications in terms of system design. This is relevant for Community-Based Indexing and similarly for Selective Caching as well as Query Routing and Community Monitoring.

Proposition 1. *Indexing nodes belonging to the same community together reduces the average number of disk seeks for query execution.*

Proof. Let n be the total number of nodes and m be the number of communities. For a random query Q, let $p(Q,C_i)$ be the probability that Q is contained within community C_i.

The expected number of disk seeks E_s without community-based indexing is proportional to n:

$$E_s \propto n.$$

With community-based indexing, the expected number of disk seeks E'_s is:

$$E'_s \propto \sum_{i=1}^{m} p(Q,C_i) \cdot |C_i|.$$

Since $|C_i| \ll n$ and $p(Q,C_i)$ is often substantially higher within a community, $E'_s < E_s$.

Proposition 2. *Selective caching of frequent query results within a community reduces the computational load.*

Proof. Let $F(Q,C_i)$ be the frequency of query Q within community C_i, and $T(Q)$ and $T'(Q)$ be the computational time for Q without and with caching, respectively.

We have $T'(Q) < T(Q)$ due to caching. Therefore, the computational load with caching, $F(Q,C_i) \cdot T'(Q)$, is strictly less than without caching, $F(Q,C_i) \cdot T(Q)$.

Proposition 3. *Using community information optimizes query routing, thereby reducing the execution time of paths.*

Proof. Let $P(Q)$ be the set of all possible paths for query Q. Without community information, the query processor selects $p \in P(Q)$ uniformly at random.

With community information, the query processor can prioritize paths $p' \subset P(Q)$ that are within a community. The expected time E_p for path selection becomes:

$$E'_p < E_p.$$

Since community-based paths are more likely to be cached or indexed, leading to faster execution.

Proposition 4. *Real-time tracking of community structures ensures the system adapts to evolving data.*

Proof. By continuously monitoring the modularity Q or other community metrics, the system can dynamically recompute communities and update indices and caches, ensuring E'_s remains optimal.

4. Results and discussion

In this section we explore the network patterns, namely the evolution of:

- publication volume;
- centralities of top terms;
- network community characteristics;
- top terms and their communities;
- entity and relationship creation over time.

Additionally, we also analyse the aggregate graph with the knowledge from all years and analyse the importance distributions and apply the proposed method for the generation of knowledge candidates and links.

4.1. Publication volume

To give a context to this analysis, Fig. 3 shows the number of publications over the years. The number of publications in 2021 are already more than double the number of 2013, with a steady growth.

Using the defined methodology, we then create the knowledge graphs. Fig. 4 shows a representation of an instance of the generated graphs. This figure shows us a slice of the full dynamic knowledge graph. We can see several nodes that have high degree, yet most of them have a low degree.

4.2. Centralities of top terms

Using their topological information, we plot the evolution of degree of the top ten nodes (Fig. 5) with the highest average. This shows us that the entity 'algorithm' is clearly the most important regarding the number of elements that are semantically connected to it. The remaining nodes have an upward trend of degree, meaning that each year there are more articles that relate that concept with another. Nevertheless, this effect can be caused by the growing number of articles. The exception to this rule is the term 'network', over the last year.

Using the betweenness centrality indicator, we can see who is connecting the majority of the semantic paths in the network (Fig. 6). We see a steady downward trend of the entity 'algorithm'. The stronger contender for the most important bridging term is 'machine learning'. This can be a sub-product of a hype trend cycle [82].

In terms of closeness centrality (Fig. 7), 'machine learning' has now overturned 'algorithm'. This means that it is closer to the remaining terms within the focus of the conference. Again, indicating a shift in focus of the venue.

The degree distribution of the knowledge graphs of each year helps us better understand the extent of the connections and how they evolve. In this case we see that the distribution of degrees over time is only changed by the number of nodes. However, the distribution itself is not particularly dynamic in the sense that the proportion of each type of degree group is the same over time (as seen in Fig. 8).

The results of the distribution of knowledge generated in this venue resembles a scale-free network degree distribution [83]. In the Probability density functions, if (loglikelihood ratio between the two distributions) is positive and (the statistical significance of that ratio) is less than 0.05, this suggests that the power-law model is a better fit to the

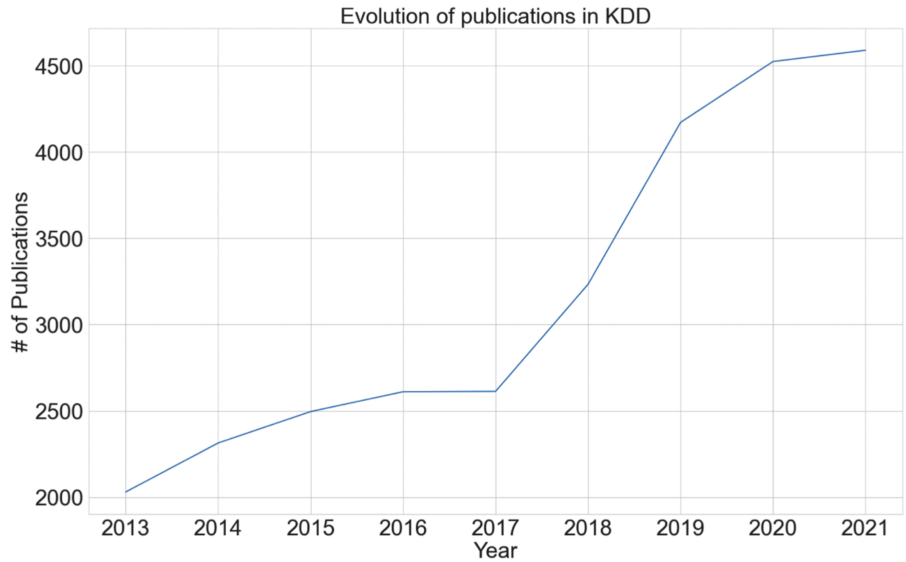

Fig. 3. Evolution of KDD publications over the years.

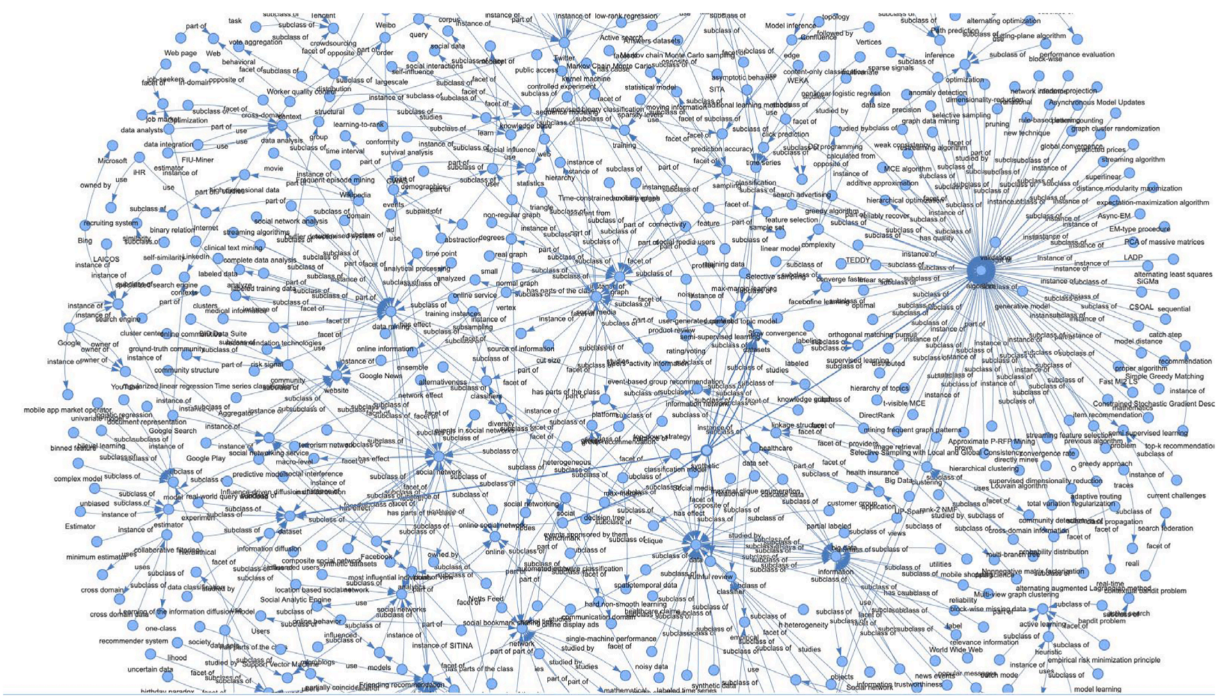

Fig. 4. Network representation for the 2013 KG.

data than the exponential model. In the CCDP we can see what proportion of nodes has a degree higher than the degree corresponding to these percentiles, showing very close values to a power law. This may corroborate that the knowledge generated seems to follow a preferential attachment of topics, which is to be expected since this venue is focused on specific fields. And these fields act as centers for community structures. Interestingly this might mean that semantic knowledge follows a fractal geometry, hence why analogies may work from some sub fields to others, given the self-similar, fractal nature of scale-free networks means that small sub-networks can resemble the overall structure of the network. This could lead to localized insights that, surprisingly, apply to the entire knowledge graph.

4.3. Network community characteristics

To understand the community structure, we look at the knowledge communities. Fig. 9 shows a network plot for the communities generated for a particular year. This shows us that each community has a small set of very important nodes, i.e. 1 or 2 most representative nodes per community.

Then, by applying the intra-cluster modularity optimization community finding algorithm by Clauset et al. [79] over the years, we see the distribution of community size leaning towards bigger communities (see Fig. 10). This is interesting since we are considering the contributions per year without considering the aggregated knowledge from previous years. This shift means that the knowledge generated is more interconnected by itself.

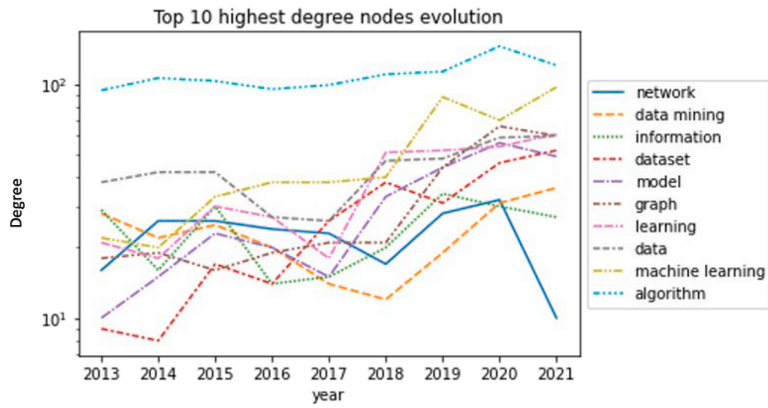

Fig. 5. Evolution of degree in top 10 highest degree nodes on a log scale.

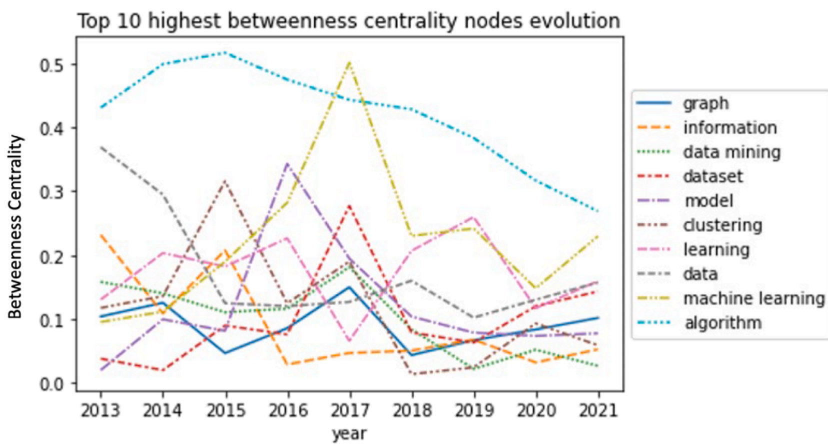

Fig. 6. Evolution of betweenness centrality in top 10 highest betweenness centrality nodes.

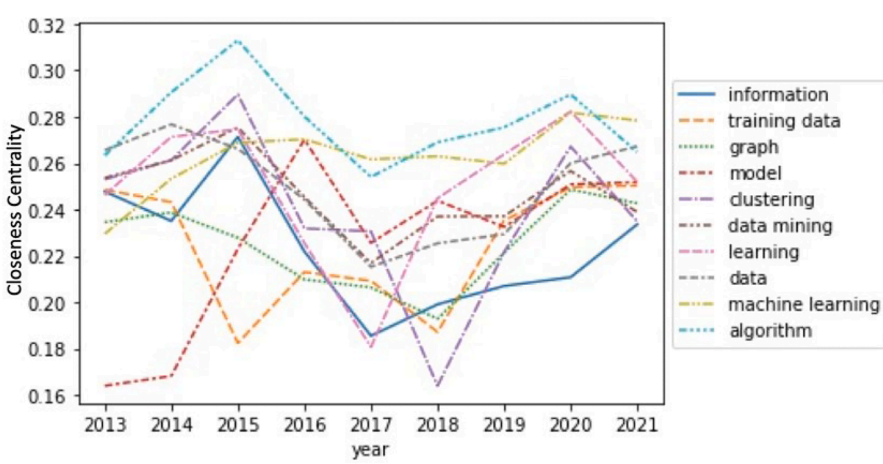

Fig. 7. Evolution of top 10 nodes in closeness centrality over time.

4.4. Top terms and their communities

To better understand what these communities mean, we look at the most important entity per community over the years through the closeness centrality (see Fig. 11). This representation looks at communities with at least 20 entities to minimize entropy on plotting and analyses yearly contributions independently. Using the parallel coordinates, we can see that the community containing 'algorithm' has always been the most important until 2019, when 'learning' sets the biggest community. This is also an important transition year since it was the last time that 'machine learning' term was the most important of its cluster. We also see that since 2020, the 'graph' community has been the biggest. We observe an increasing importance on the 'learning' and 'e-commerce' as well. Using the generated visualizations we are able to filter the community flows based on a specific community. Fig. 11 shows the filter of the community containing 'graph' in 2021. Here we can trace the origins of the terms for this large community.

Using the generated community structures we then measure how

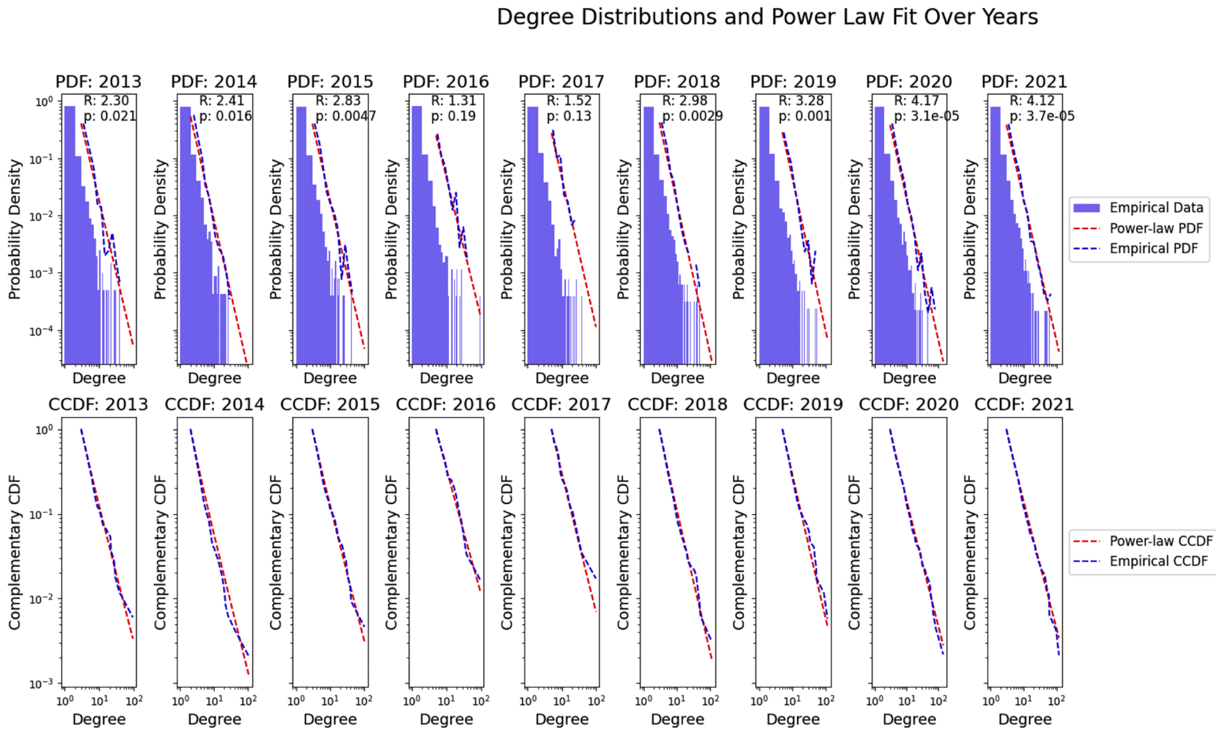

Fig. 8. Degree distribution, Probability density functions and Complementary Cumulative Distribution Function over the years.

modular these are (Fig. 12). The curve is approximately constant given the range of modularity. This means that the cohesion of the bigger communities remains stable. So, do we have a shift in growth? Are we witnessing a higher connection more in connecting existing entities than creating new ones, in the SIGKDD conference?

Fig. 13 answers the above questions. We see a clear trend on generating more relations each year than entities. So, the focus of SIGKDD at this point is to bridge the semantic gaps by connecting existing ideas with new relations. Nevertheless, we still see an increasing number for both over the years.

Now that we understand the evolution of the knowledge communities over time, we may look at the aggregated knowledge graph, comprising the publications over eight years (2013–2021) of the SIGKDD Conferences used in the analysis. This network has 20,228 entities and 21,297 edges. From which we may generate more edges based on the KEG approach of the methodology. We see in Fig. 14 that we have a similar distribution to individual year graphs. The importance of nodes in Fig. 15, based on degree centrality, shows us that there is a low number of nodes with heightened importance. This tells us that there is a very clear set of central topics within the conference.

In Fig. 16 with the aggregated KG, we can see quite clearly 5 communities: the Algorithm community (blue) which is quite dispersed and well connected with the remaining communities within the giant component, the machine learning (ML) community (purple) that contains a set of learning algorithms, interestingly the optimization community is much smaller and separated from ML, the data and information community (green) that contains concepts regarding the data used in different studies, and lastly the knowledge community which is parallel to the machine learning community and still well connected to the data community. Yet, looking at the full network, there are still many unconnected nodes in the full aggregated graph.

4.5. Entity and relationship creation over time

So, there are many impactful relations that may improve the connectivity in the published research knowledge. To this end we apply the Algorithm 1 to generate high impact candidate connections based on the defined importance function (4). Below, we show the top importance pairs generated sorted by descendent order of impact. The tuple that generates the highest impact in network connectivity is '('algorithm', 'Commonwealth of Nations')', effectively increasing the richness of the entire network.

[('algorithm', 'Commonwealth of Nations'),
('algorithm', 'outlier'),
('Transformer', 'real estate'),
('Transformer', 'outlier'),
('algorithm', 'urban area'),
('Transformer', 'constraint'),
('outlier', 'urban area'),

The algorithm identifies a potentially valuable link between computational methods (algorithms) and the Commonwealth of Nations. This might imply a need or opportunity for algorithmic approaches to challenges or data related to the Commonwealth. A link between 'algorithm' and 'outlier' suggests that algorithms might play an important role in outlier detection, which is often true in data science. The Transformer model may have applications in the real estate industry, perhaps for property valuation, recommendation systems, or natural language queries about properties. Another pair, 'Transformer' and 'outlier', could imply that Transformer models have applicability in identifying outliers, perhaps in sequential data or time series. Algorithms may also have significant applications in urban planning, crowd control, or smart city initiatives, as indicated by the pair 'algorithm' and 'urban area'. The pair 'Transformer' and 'constraint' suggests that Transformer models might be relevant when dealing with constraint-based problems, perhaps in optimization scenarios. Finally, 'outlier' and 'urban area' indicate that outlier detection could be an important aspect of various urban phenomena, from traffic patterns to utility usage.

The theoretical implications of this paper can be broken down into two categories. First, the paper proposes a dynamic knowledge graph longitudinal analysis based on network science principles, and the principles of community detection. Second, the paper focuses on the

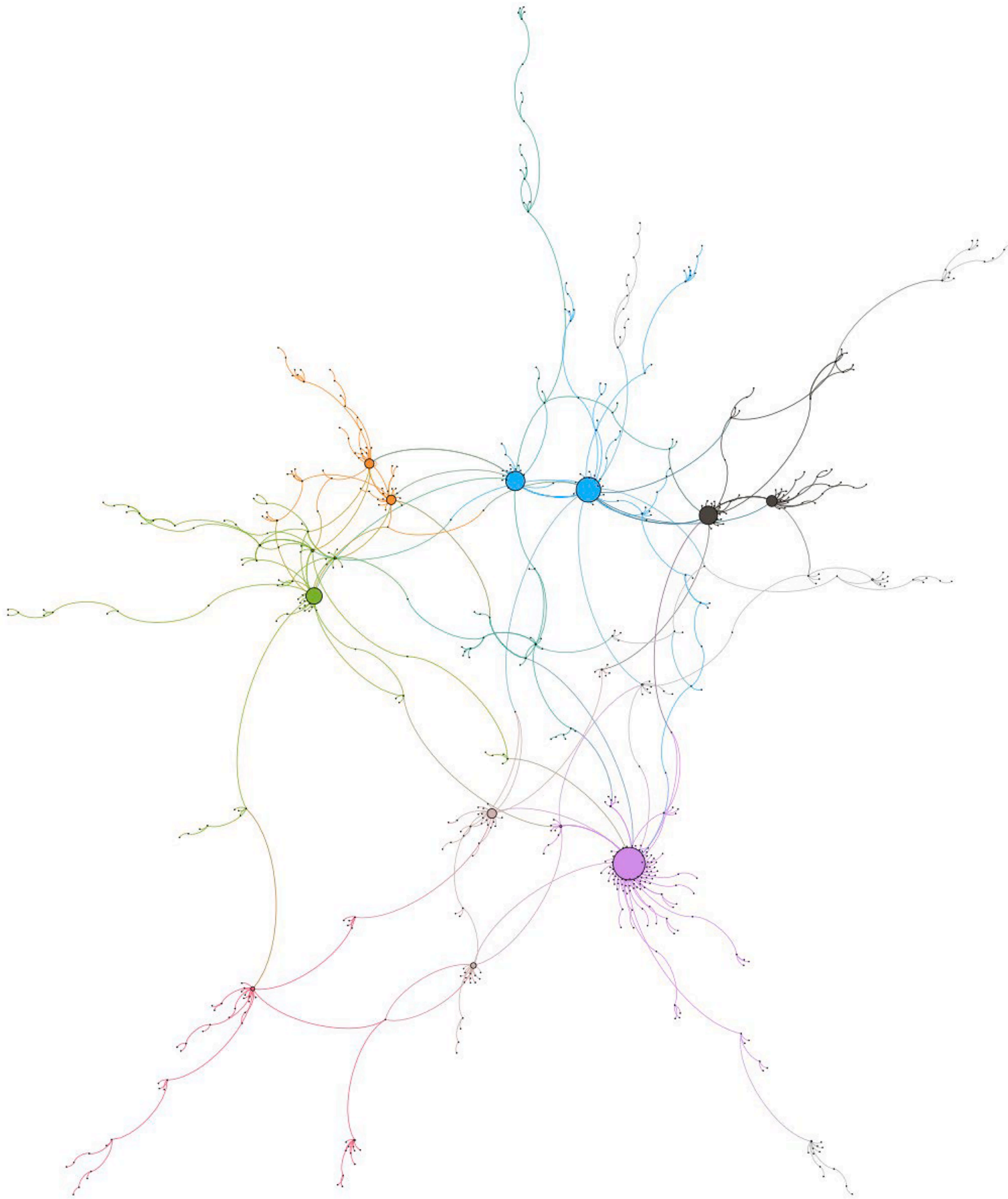

Fig. 9. Communities generated for the 2013 Knowledge graph.

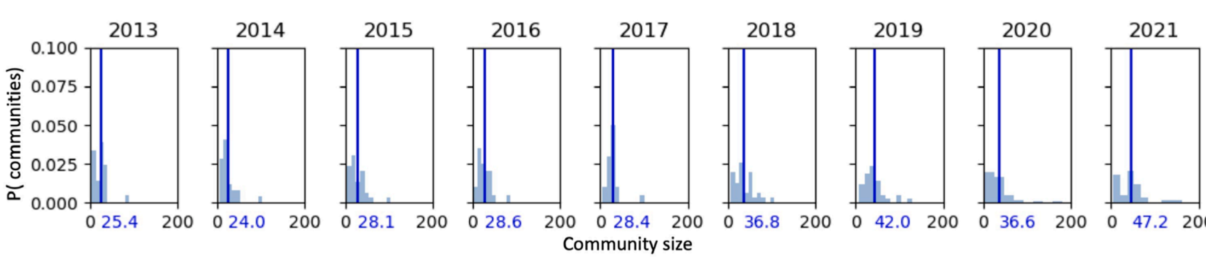

Fig. 10. Community size distribution over the years.

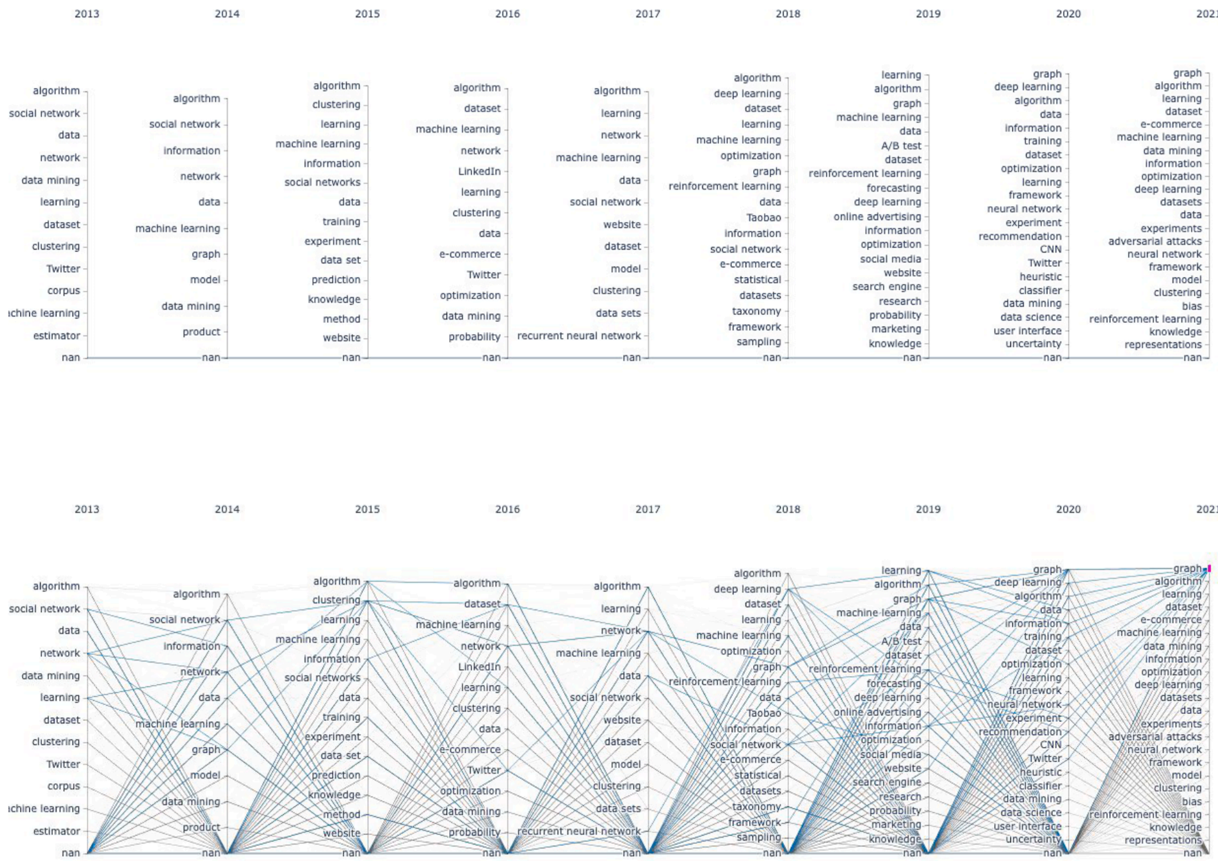

Fig. 11. Highest closeness centrality concept per community over the years, with graph filter below.

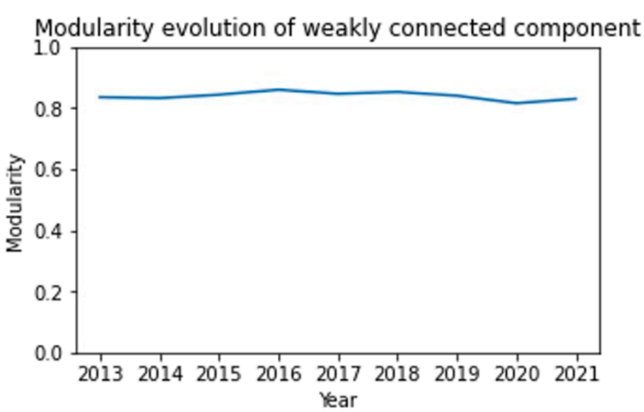

Fig. 12. Modularity of concepts remains the same.

application of this analysis on a specific context and the corresponding implications of doing so. Hence, this study implies that future research on dynamic meta-knowledge network representation can use the proposed methodology to better understand trends and evolution of knowledge in different fields. As practical implications of this study, the KDD conference program committees may identify future connected research topics and tracks based on unconnected knowledge. Additionally, we see empirical evidence for more a higher focus of knowledge integration instead of the construct creation.

5. Conclusion and future research

Research results show that the KDD conference and contributing communities has been evolving over the last nine years. The KDD conference is increasingly focused on a limited set of domains. Emerging publications in KDD appear to favor the creation of relations instead of entities.

Cohesion of the generated communities over time is constant and the average community size grew over time. Focus of the papers evolves with an increasing importance of ‘graph’, ‘learning’ and ‘e-commerce’ communities is observed. The ‘algorithm’ community was historically the most important. However, in the recent years, the ’graph’’ community is positioned at the top for the first time.

In this work, a review of state-of-the-art principles to generate and analyze dynamic knowledge graphs is undertaken. A further discussion of complementary application domains is undertaken, motivating the viability of the targeted analytics towards the KDD corpus in a temporal domain. A brief review of the application domains and the techniques form finding patterns in such knowledge representation was also done. To instantiate the concepts explored, the corpora from the KDD papers was used to generate a dynamic knowledge graph over the period of 2013–2021. The analysis of communities was done along with the centrality analysis over time. A method for generating high impact relations was proposed and instantiated. In the future, the analysis presented can shed a better understanding on different domains of applications. New methods regarding generation of candidate knowledge triplets can be the subject of further development. The study of the relationships between the connected components, communities and the research community itself, may be of interest.

CRediT authorship contribution statement

Joao T. Aparicio: Writing – original draft, Software, Methodology, Conceptualization. **Elisabete Arsenio:** Writing – review & editing, Supervision. **Francisco Santos:** Supervision. **Rui Henriques:** Writing – review & editing, Supervision.

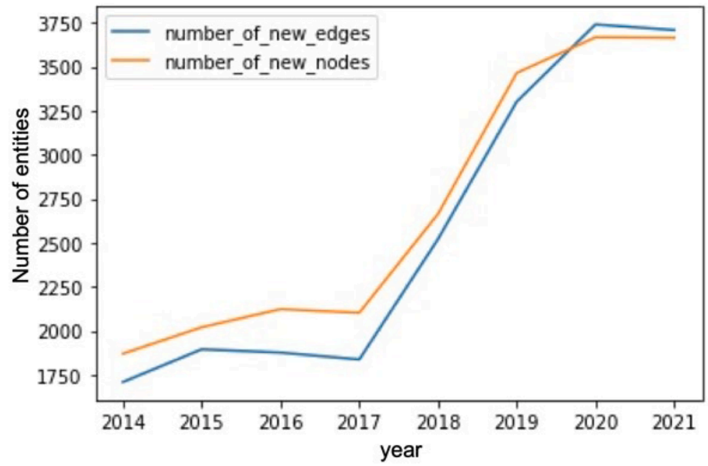

Fig. 13. New entities and relations by year.

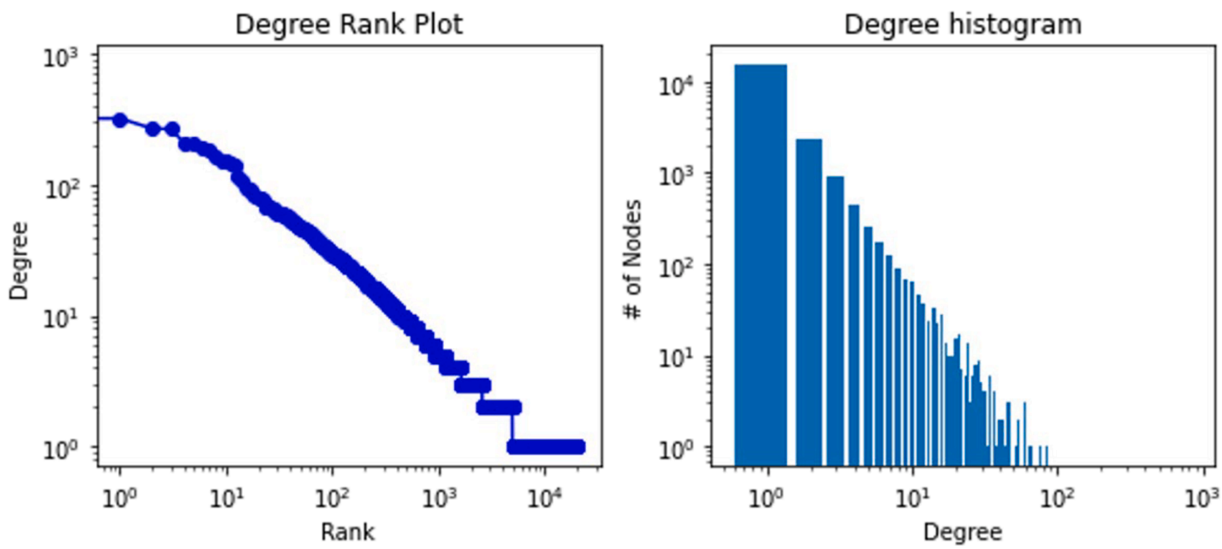

Fig. 14. Degree distribution over the aggregated knowledge graph.

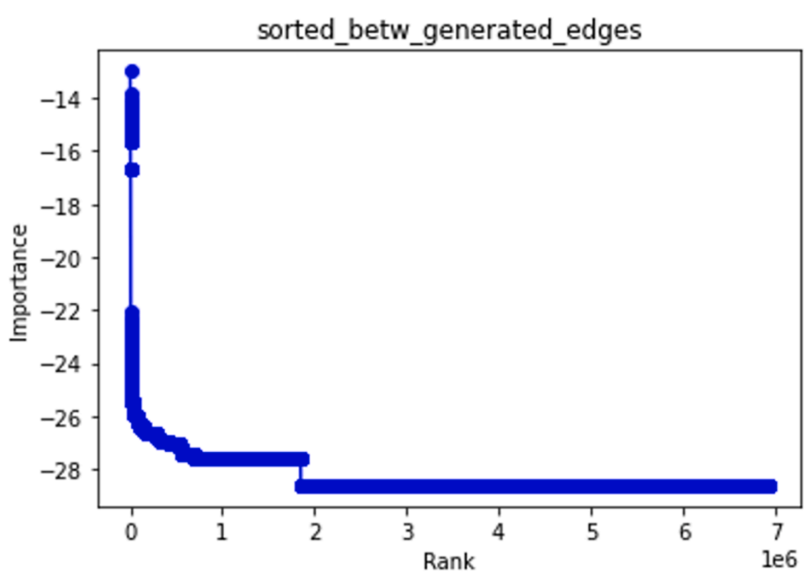

Fig. 15. Rank of Importance of generated entity pairs.

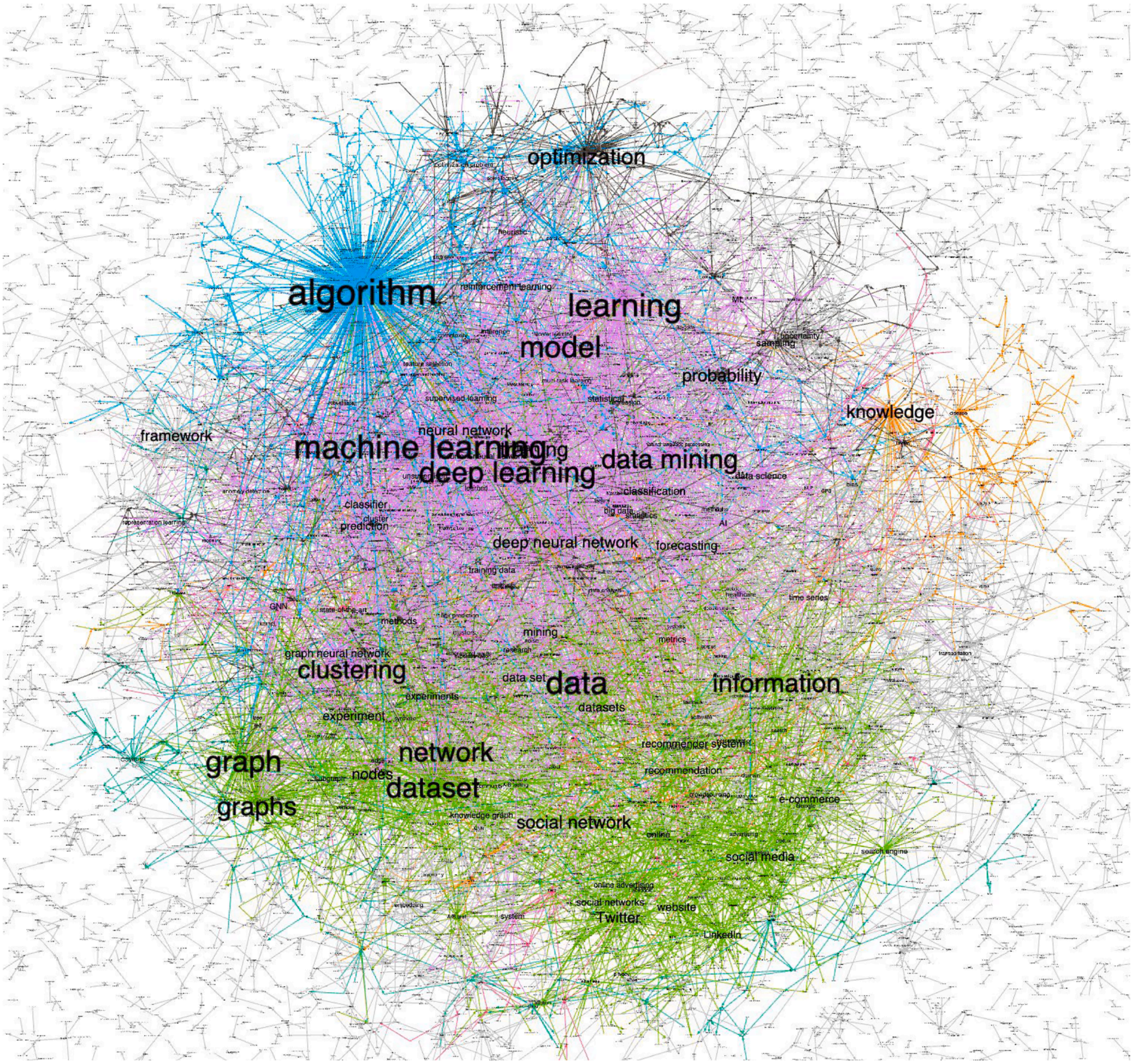

Fig. 16. Representation of Full knowledge graph with colors representing generated clusters and high degree nodes.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Joao Tiago Aparicio reports financial support was provided by Foundation for Science and Technology.

Data availability

Only open data was used.

Acknowledgments

This research was funded by Fundaço para a Ciência e Tecnologia (FCT) under PhD Grant number UI/BD/153587/2022 to JTA, ADMIRAL (HORIZON-CL5-2022-D6-02-01 under agreement 101104163) and ATE (02/C05-i01.02/2022 under agreement PC644914747-00000023).

References

[1] M. Pellegrino, Knowledge graphs within everyone's means, in: Proceedings of the Interactive Experiences and Doctoral Consortium, 2021, pp. 36–40.

[2] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, Why we need knowledge graphs: applications. Knowledge Graphs, Springer, 2020, pp. 95–112.

[3] X. Zou, A survey on application of knowledge graph, J. Phys. Conf. Ser. 1487 (2020) 012016. Vol.

[4] D. Dessì, F. Osborne, D.R. Recupero, D. Buscaldi, E. Motta, Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain, Future Gener. Comput. Syst. 116 (2021) 253–264.

[5] H. Ye, N. Zhang, H. Chen, H. Chen, Generative knowledge graph construction: a review, arXiv preprint arXiv:2210.12714 (2022).

[6] A. Veremyev, A. Semenov, E.L. Pasiliao, V. Boginski, Graph-based exploration and clustering analysis of semantic spaces, Appl. Netw. Sci. 4 (2019) 1–26.

[7] C.S. Siew, Applications of network science to education research: quantifying knowledge and the development of expertise through network analysis, Educ. Sci. 10 (4) (2020) 101.

[8] L. Sheetz, V. Dunham, J. Cooper, Professional development for network science as a multi-disciplinary curriculum tool, in: Proceedings of the IEEE Integrated STEM Education Conference, IEEE, 2015, pp. 178–182.

[9] A. Ohsaki, J. Oshima, R. Oshima, Knowledge-building analytics based on network science, in: Proceedings of the NetSciLA21 Workshop, 2021.

[10] G. Buchgeher, D. Gabauer, J. Martinez-Gil, L. Ehrlinger, Knowledge graphs in manufacturing and production: a systematic literature review, IEEE Access 9 (2021) 55537–55554.

[11] R. Trivedi, H. Dai, Y. Wang, L. Song, Know-evolve: deep temporal reasoning for dynamic knowledge graphs, in: Proceedings of the International Conference on Machine Learning, PMLR, 2017, pp. 3462–3471.
[12] S.M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, P. Poupart, Relational representation learning for dynamic (knowledge) graphs: a survey, arXiv preprint arXiv:1905.11485 12 (2019).
[13] A. García-Durán, S. Dumančić, M. Niepert, Learning sequence encoders for temporal knowledge graph completion, arXiv preprint arXiv:1809.03202 (2018).
[14] M.J. Pazzani, Knowledge discovery from data? IEEE Intell. Syst. Their Appl. 15 (2) (2000) 10–12.
[15] J. Numata, K. Hane, B. Lei, Y. Iwashita, Knowledge discovery and sharing in an information system, in: Proceedings of the Innovation in Technology Management. The Key to Global Leadership. PICMET'97, IEEE, 1997, pp. 713–716.
[16] X. Zhang, L. Shou, J. Pei, M. Gong, L. Wen, D. Jiang, A graph representation of semi-structured data for web question answering, arXiv preprint arXiv:2010.06801 (2020).
[17] Y. Li, J. Cao, Y. Wang, Implementation of intelligent question answering system based on basketball knowledge graph, in: Proceedings of the IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), IEEE, 2019, pp. 2601–2604. Vol. 1.
[18] G. Veena, S. Athulya, S. Shaji, D. Gupta, A graph-based relation extraction method for question answering system, in: Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 944–949.
[19] S.R. Indurthi, D. Raghu, M.M. Khapra, S. Joshi, Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017, pp. 376–385.
[20] A. Lamurias, J.D. Ferreira, L.A. Clarke, F.M. Couto, Generating a tolerogenic cell therapy knowledge graph from literature, Front. Immunol. 8 (2017) 1656.
[21] P. Yochum, L. Chang, T. Gu, M. Zhu, W. Zhang, Tourist attraction recommendation based on knowledge graph, in: Proceedings of the International Conference on Intelligent Information Processing, Springer, 2018, pp. 80–85.
[22] K. Krinkin, A. Vodyaho, I. Kulikov, N. Zhukova, Models of telecommunications network monitoring based on knowledge graphs, in: Proceedings of the 9th Mediterranean Conference on Embedded Computing (MECO), IEEE, 2020, pp. 1–7.
[23] N. Rezaei, M.Z. Reformat, R.R. Yager, Image-based world-perceiving knowledge graph (wpkg) with imprecision, in: Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2020, pp. 415–428.
[24] J. Weng, Y. Gao, J. Qiu, G. Ding, H. Zheng, Construction and application of teaching system based on crowdsourcing knowledge graph, in: Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Springer, 2019, pp. 25–37.
[25] H. Du, Q. Huang, C. Zhang, D. Zhao, Knowledge-enhanced iterative instruction generation and reasoning for knowledge base question answering, in: Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Springer, 2022, pp. 431–444.
[26] P.R. Venkatesh, et al., Knowledge graphs using cloud services, Turk. J. Comput. Math. Educ. (TURCOMAT) 12 (10) (2021) 3979–3991.
[27] P. Kittenberger, Generating knowledge graphs with specified ambiguities, Ph.D. Thesis, Wien (2021).
[28] G. Rossiello, M.F.M. Chowdhury, N. Mihindukulasooriya, O. Cornec, A. Gliozzo, Knowgl: knowledge generation and linking from text, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023.
[29] R. Das, T. Munkhdalai, X. Yuan, A. Trischler, A. McCallum, Building dynamic knowledge graphs from text using machine reading comprehension, arXiv preprint arXiv:1810.05682 (2018).
[30] T. Wu, A. Khan, M. Yong, G. Qi, M. Wang, Effiiently embedding dynamic knowledge graphs, Knowl. Based Syst. 250 (2022) 109124.
[31] Y.L. Tuan, Y.N. Chen, H.Y. Lee, Dykgchat: benchmarking dialogue generation grounding on dynamic knowledge graphs, arXiv preprint arXiv:1910.00610 (2019).
[32] W. Xie, S. Wang, Y. Wei, Y. Zhao, X. Fu, Dynamic knowledge graph completion with jointly structural and textual dependency, in: Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, Springer, 2020, pp. 432–448.
[33] X. Tang, R. Yuan, Q. Li, T. Wang, H. Yang, Y. Cai, H. Song, Timespanaware dynamic knowledge graph embedding by incorporating temporal evolution, IEEE Access 8 (2020) 6849–6860.
[34] Y. Tay, A.T. Luu, S.C. Hui, Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs, in: Proceedings of the Thirtyfirst AAAI Conference on Artificial Intelligence, 2017.
[35] G. Qin, J. Yang, L. Gao, J. Li, Evolution pattern discovery in dynamic networks, in: Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 2011, pp. 1–6.
[36] L.B. Nguyen, I. Zelinka, V. Snasel, L.T. Nguyen, B. Vo, Subgraph Mining in a Large Graph: A Review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2022, p. e1454.
[37] Y. Sun, G. Li, M. Guan, B. Ning, Subgraph-indexed sequential subdivision for continuous subgraph matching on dynamic knowledge graph, Complexity 2020 (2020) 1–18.
[38] K.M. Borgwardt, H.P. Kriegel, P. Wackersreuther, Pattern mining in frequent dynamic subgraphs, in: Proceedings of the Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 818–822.
[39] G. Blin, F. Sikora, S. Vialette, Querying protein-protein interaction networks, in: Proceedings of the International Symposium on Bioinformatics Research and Applications, Springer, 2009, pp. 52–62.
[40] J. Gehweiler, H. Meyerhenke, A distributed diffusive heuristic for clustering a virtual p2p supercomputer, in: Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), IEEE, 2010, pp. 1–8.
[41] J. Benik, C. Chang, L. Raschid, M.E. Vidal, G. Palma, A. Thor, Finding cross genome patterns in annotation graphs, in: Proceedings of the International Conference on Data Integration in the Life Sciences, Springer, 2012, pp. 21–36.
[42] M.K. Agarwal, K. Ramamritham, M. Bhide, Real time discovery of dense clusters in highly dynamic graphs: identifying real world events in highly dynamic environments, arXiv preprint arXiv:1207.0138 (2012).
[43] J.P. Fairbanks, R. Kannan, H. Park, D.A. Bader, Behavioral clusters in dynamic graphs, Parallel Comput. 47 (2015) 38–50.
[44] N. Shah, D. Koutra, T. Zou, B. Gallagher, C. Faloutsos, Timecrunch: interpretable dynamic graph summarization, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1055–1064.
[45] J. Gao, T. Zhang, C. Xu, Watch, think and attend: end-to-end video classification via dynamic knowledge evolution modeling, in: Proceedings of the 26th ACM International Conference on Multimedia, 2018, pp. 690–699.
[46] S. Kapoor, D.K. Saxena, M. van Leeuwen, Online summarization of dynamic graphs using subjective interestingness for sequential data, Data Min. Knowl. Discov. 35 (1) (2021) 88–126.
[47] A. Zakrzewska, D.A. Bader, Fast incremental community detection on dynamic graphs, in: Proceedings of the International Conference on Parallel Processing and Applied Mathematics, Springer, 2015, pp. 207–217.
[48] S. Choudhury, K. Agarwal, S. Purohit, B. Zhang, M. Pirrung, W. Smith, M. Thomas, Nous: construction and querying of dynamic knowledge graphs, in: Proceedings of the IEEE 33rd International Conference on Data Engineering (ICDE), IEEE, 2017, pp. 1563–1565.
[49] Q. Lang, X. Liu, W. Jia, AFS graph: multidimensional axiomatic fuzzy set knowledge graph for open-domain question answering, IEEE Trans. Neural Netw. Learn. Syst. 34 (2022) 10904–10918.
[50] N. AfzaliSeresht, Y. Miao, Q. Liu, A. Teshome, W. Ye, Investigating cyber alerts with graph-based analytics and narrative visualization, in: Proceedings of the 24th International Conference Information Visualisation (IV), IEEE, 2020, pp. 521–529.
[51] J. Dörpinghaus, M. Jacobs, Knowledge detection and discovery using semantic graph embeddings on large knowledge graphs generated on text mining results, in: Proceedings of the 15th Conference on Computer Science and Information Systems (FedCSIS), 2020, pp. 169–178.
[52] M. Masoud, B. Pereira, J.P. McCrae, P. Buitelaar, Automatic construction of knowledge graphs from text and structured data: a preliminary literature review, in: International Conference on Language, Data, and Knowledge, 2021.
[53] N. Petrelková, B. krlj, N. Lavra, Knowledge graph aware text classification, 2020.
[54] L. Dietz, A. Kotov, E. Meij, Utilizing knowledge graphs for text-centric information retrieval, in: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018.
[55] R. Feldman, J. Sanger, The text mining handbook - advanced approaches in analyzing unstructured data, 2006.
[56] L.S. Martín, J.H. Boockmann, A. Henrich, Fast pathfinding in knowledge graphs using word embeddings, in: Proceedings of the Deutsche Jahrestagung für Künstliche Intelligenz, 2020.
[57] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: a survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (2017) 2724–2743.
[58] C. Zhou, H. Wang, C. Wang, Z. Hou, Z. Zheng, S. Shen, Q. Cheng, Z. Feng, X. Wang, H. Lv, et al., Geoscience knowledge graph in the big data era, Sci. China Earth Sci. 64 (7) (2021) 1105–1114.
[59] Z. Yang, Y. Bi, L. Wang, D. Cao, R. Li, Q. Li, Development and application of a field knowledge graph and search engine for pavement engineering, Sci. Rep. 12 (1) (2022) 1–12.
[60] M. Cheatham, A.A. Krisnadhi, R. Amini, P. Hitzler, K. Janowicz, A. Shepherd, T. W. Narock, M.B. Jones, P. Ji, The geolink knowledge graph, Big Earth Data 2 (2018) 131–143.
[61] E. Demidova, A. Dsouza, S. Gottschalk, N. Tempelmeier, R. Yu, Creating knowledge graphs for geographic data on the web, ACM SIGWEB Newsl. (2022) 1–8.
[62] C. Guo, T. Xu, L. Liu, Construction of knowledge graph based on geographic ontology, in: Proceedings of the IOP Conference Series: Earth and Environmental Science 252, 2019.
[63] C. Deng, Y. Jia, H. Xu, C. Zhang, J. Tang, L. Fu, W. Zhang, H. Zhang, X. Wang, C. Zhou, Gakg: a multimodal geoscience academic knowledge graph, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021.
[64] Y. Zhao, H. Wang, L. Ma, Y. Liu, L. Li, J. Grundy, Knowledge graphing git repositories: a preliminary study, in: Proceedings of the IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, 2019, pp. 599–603.
[65] N. Kalanat, A. Kovashka, Symbolic image detection using scene and knowledge graphs, arXiv preprint arXiv:2206.04863 (2022).
[66] H. Paulheim, Knowledge graph refinement: a survey of approaches and evaluation methods, Semant. Web 8 (3) (2017) 489–508.
[67] M. Rotmensch, Y. Halpern, A. Tlimat, S. Horng, D. Sontag, Learning a health knowledge graph from electronic medical records, Sci. Rep. 7 (1) (2017) 1–11.

[68] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, X. Zhu, Commonsense knowledge aware conversation generation with graph attention, in: Proceedings of the IJCAI, 2018, pp. 4623–4629.

[69] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph and text jointly embedding, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1591–1601.

[70] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, arXiv preprint arXiv:1909.03193 (2019).

[71] B.Y. Lin, X. Chen, J. Chen, X. Ren, Kagnet: knowledge-aware graph networks for commonsense reasoning, arXiv preprint arXiv:1909.02151 (2019).

[72] X. Chen, S. Jia, Y. Xiang, A review: knowledge reasoning over knowledge graph, Expert Syst. Appl. 141 (2020) 112948.

[73] S. Ji, S. Pan, E. Cambria, P. Marttinen, S.Y. Philip, A survey on knowledge graphs: representation, acquisition, and applications, IEEE Trans. Neural Netw. Learn. Syst. 33 (2) (2021) 494–514.

[74] A. Ojo, O. Sennaike, Constructing knowledge graphs from data catalogues, in: Proceedings of the International Conference on Distributed Computing and Internet Technology, Springer, 2020, pp. 94–107.

[75] N. Mihindukulasooriya, M. Sava, G. Rossiello, M.F.M. Chowdhury, I. Yachbes, A. Gidh, J. Duckwitz, K. Nisar, M. Santos, A. Gliozzo, Knowledge graph induction enabling recommending and trend analysis: a corporate research community use case, in: Proceedings of the International Semantic Web Conference, Springer, 2022, pp. 827–844.

[76] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-To-Sequence Pretraining For Natural Language generation, translation, and Comprehension, arXiv preprint arXiv: 1910.13461 (2019).

[77] J.T. Aparicio, E. Arsenio, F.C. Santos, R. Henriques, Lines: multimodal transportation resilience analysis, Sustainability 14 (13) (2022) 7891.

[78] M. Liu, C. Chato, A.F. Poon, From components to communities: bringing network science to clustering for molecular epidemiology, Virus Evol. 9 (1) (2023) vead026.

[79] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.

[80] J. Chicaiza, P. Valdiviezo-Diaz, A comprehensive survey of knowledge graph-based recommender systems: technologies, development, and contributions, Information 12 (6) (2021) 232.

[81] C.C. Aggarwal, et al., Recommender Systems, Springer, 2016. Vol. 1.

[82] D.E. O'Leary, Gartner's hype cycle and information system research issues, Int. J. Account. Inf. Syst. 9 (4) (2008) 240–252.

[83] L. Li, D. Alderson, J.C. Doyle, W. Willinger, Towards a theory of scalefree graphs: definition, properties, and implications, Internet Math. 2 (4) (2005) 431–523.

[84] Y. Jia, et al., Persistent Graph Stream Summarization for Real-Time Graph Analytics, World Wide Web, 2023, pp. 1–21.

[85] U. Fang, et al., Gomic: Multi-View Image Clustering via Self-Supervised Contrastive Heterogeneous Graph Co-Learning, 26.4, World Wide Web, 2023, pp. 1667–1683.