Journal of Informetrics 15 (2021) 101109

Contents lists available at ScienceDirect

Journal of Informetrics

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

Regular article

SciKGraph: A knowledge graph approach to structure a scientific field

Mauro Dalle Lucca Tosi[a,*], Julio Cesar dos Reis[b]

[a] Faculty of Science, Technology and Communication, University of Luxembourg, Belval, Luxembourg
[b] Institute of Computing, University of Campinas, Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 25 February 2020
Received in revised form
24 September 2020
Accepted 2 November 2020

Keywords:
Knowledge graphs
Knowledge representation
Overlap clustering
Semantic annotation
Document classification

ABSTRACT

Understanding the structure of a scientific domain and extracting specific information from it is laborious. The high amount of manual effort required to this end indicates that the way knowledge has been structured and visualized until the present day should be improved in software tools. Nowadays, scientific domains are organized based on citation networks or bag-of-words techniques, disregarding the intrinsic semantics of concepts presented in literature documents. We propose a novel approach to structure scientific fields, which uses semantic analysis from natural language texts to construct knowledge graphs. Then, our approach clusters knowledge graphs in their main topics and automatically extracts information such as the most relevant concepts in topics and overlapping concepts between topics. We evaluate the proposed model in two datasets from distinct areas. The results achieve up to 84% of accuracy in the task of document classification without using annotated data to segment topics from a set of input documents. Our solution identifies coherent keyphrases and key concepts considering the dataset used. The SciKGraph framework contributes by structuring knowledge that might aid researchers in the study of their areas, reducing the effort and amount of time devoted to groundwork.

1. Introduction

The amount of publications a researcher must absorb has been increasing over the past years (Bornmann & Mutz, 2015). This issue is a regular part of academic life, requiring constant updating with the most recent articles and discoveries in the researcher's knowledge area. Consequently, the coming era of big scholarly data makes it hard for researchers to identify interesting documents to read (Xia, Liu, Lee, & Cao, 2016). Considering that research work is arduous even for experts, it is quite hard for newcomers to accomplish it in a reasonable time. Therefore, computational mechanisms are essential for assisting scientists in finding what they seek, rather than indiscriminately searching for information. That is why research has been conducted to construct and improve recommendation systems for scientific articles (Gupta & Varma, 2017; Xia et al., 2016).

[*] Corresponding author.
E-mail addresses: mauro.dalleluccatosi@uni.lu (M.D.L. Tosi), jreis@ic.unicamp.br (J.C. dos Reis).

https://doi.org/10.1016/j.joi.2020.101109
1751-1577/© 2020 Elsevier Ltd. All rights reserved.

Nowadays, state-of-the-art methods usually use classification techniques to segment academic documents in pre-defined areas (Yau, Porter, Newman, & Suominen, 2014). In these approaches, the most cited articles from relevant areas are recommended to researchers. However, this research line has two main flaws. First, it neglects the fact that users may not have the required prerequisites to understand the recommendations. Second, it either segments the area based on manually predefined areas, imparting a bias to the results, or considers only the meta-data of the analyzed documents, disregarding their content.

In this sense, the content consumed by a novice researcher is far from ideal, leading him/her to waste time studying topics unrelated to his/her primary goal. This problem is caused by the difficulty in understanding how the scientific field the researcher is studying is organized, which is an indicator that the way knowledge is structured and visualized should be enhanced.

Improving the organization of scientific knowledge is not a trivial task. This occurs not only because scientific knowledge is continually evolving, as new documents are published, but because of the comprehensive scope of the problem, which involves a lot of data. Therefore, approaches based on semantic analysis in natural language texts are a challenge, as they are more complex, having to deal with the interpretation of concepts in different contexts (Cambria & White, 2014). To exemplify this, when analyzing a phrase with the word “apple” in it, this word could represent distinct concepts, such as a fruit, a brand, or New York City (The Big Apple), depending on the context in which it is. That is why most research addressing this problem proposes minor changes to current solutions, maintaining their classification techniques to segment academic articles.

This article proposes SciKGraph, a framework to structure the knowledge of a scientific field considering the semantics of the concepts extracted from textual documents of that field. Our solution aims to identify segments of a field of knowledge in its sub-areas, presenting a short textual description from them. We study the behavior of the framework with different datasets and its application to distinct knowledge areas.

In our framework, rather than using only meta-data and citation information, we construct a knowledge graph (KG), a knowledge-based system that integrates information and applies a reasoner to it to generate new knowledge (Ehrlinger & Wöß, 2016); processed from a set of textual documents to represent concepts belonging to the studied scientific field. Our proposal takes as input a collection of academic documents, identifies their concepts, and constructs a knowledge graph based on their co-occurrence in the documents. Then, our framework identifies clusters of concepts representing the sub-areas of the studied scientific field. Finally, the proposed framework extracts from both the field and its sub-areas key concepts (relevant concepts composed of one or more words) and keyphrases (major phrases composed of one or more concepts), outputting to the user the organized knowledge graph.

We evaluate our implemented proposal based on two datasets. First, we use the WOS-5736 dataset, which is composed of a collection of academic articles with their areas and sub-areas annotated. In this context, our proposal identifies in which of the automatically identified topics each article belongs, and we analyze the accuracy in the document classification task. Second, we construct a dataset of Artificial Intelligence (AI) documents by gathering 1018 articles from the *Artificial Intelligence* area. We use it to qualitatively evaluate topic segmentation and knowledge extraction, based on the key concepts and keyphrases identified. Furthermore, we compare the knowledge graphs constructed based on the AI and WOS datasets, analyzing their similarity to evaluate if they exhibit the same structure, indicating that our solution is sufficiently generic and applicable to distinct scientific fields.

The results reveal up to 84% of accuracy by identifying to which academic area a set of articles belongs, without relying on annotated data, which expresses the novelty of our proposal. The results show that the identified key concepts and keyphrases are suited to clarify what each AI topic represents, and their overlapping structure shows the topic’s correlations. The analyses of the two knowledge graphs indicate similar structures and trends, which corroborates our assumption that our proposed framework can be used to represent distinct scientific fields.

Our solution is suited to provide users with information regarding the structure in sub-areas (topics) generated from a set of textual documents as input, considering connections and intersections between identified topics and concepts. Users can consult the topics and analyze the extracted concepts from the scientific field.

This article is organized as follows: Section 2 discusses background work. Section 3 introduces the proposed framework to structure and analyze a scientific field. Section 4 describes the experimental evaluation, including implementation aspects, datasets and reports on the achieved results. We discuss the obtained findings in Section 5. Section 6 exhibits the final considerations of this article.

2. Background

Nowadays, most academic textual knowledge is structured based on three approaches: (1) classification algorithms, which are used to infer in which pre-defined area a piece of text belongs (Kowsari et al., 2019); (2) citation networks, applying clustering methods to citation graphs to identify their main topics (Silva, Amancio, Bardosova, Costa, & Oliveira Jr, 2016); or (3) manually, requesting researchers to assign academic articles to predefined categories, as the ones defined in Rous (2012).

Kowsari et al. (2019) state that most text classification algorithms are divided into four modules: feature extraction, which identifies the main features of a piece of text; dimensionality reduction, which is optional and used to optimize the algorithm;

learning model, responsible for determining the machine learning model used to classify the texts; and evaluation, which determines the metric used to appraise the algorithm.

Concerning the feature extraction module, Kowsari et al. (2019) describe several common techniques, such as Term Frequency-Inverse Document Frequency (TF-IDF) (Salton & Buckley, 1988), Word2Vec (Goldberg & Levy, 2014), and Global Vectors for Word Representation (GloVe) (Pennington, Socher, & Manning, 2014). One could identify similar features based on their distance using Word2Vec or GloVe. Such techniques do not disambiguate their elements, but based on their proximity, one could infer their similarity. However, they require training to identify the main features of the text.

Considering this, Tosi and dos Reis (2019) proposed C-Rank, a keyphrase extraction technique that, in order to mitigate these issues, uses Babelfy (Moro, Raganato, & Navigli, 2014) to extract disambiguated concepts from single academic articles. Keyphrases are expressions composed of single or multiple words that are usually used to represent the content of a document, highlighting its main topics. Babelfy is a graph-based approach to disambiguate and link entities and concepts between texts and BabelNet (Navigli & Ponzetto, 2012), a knowledge graph constructed based on WordNet (Miller, 1995), DBPedia (Auer et al., 2007), and other sources. Moreover, C-Rank builds a co-occurrence graph based on the disambiguated concepts and ranks them according to their centrality in the graph, which is further used to identify the keyphrases from a scientific document.

Kowsari et al. (2019) indicate distinct learning models, such as Rocchio classification (Rocchio, 1971), Naïve Bayes Classifier (Maron, 1961), k-nearest neighbor (Altman, 1992), and support vector machine (SVM) (Cortes & Vapnik, 1995). According to the authors, those and other cited models either are computationally expensive, do not solve nonlinear problems, are not robust, or require supervised training. In general, using supervised training to perform textual classification is not an issue. Still, when segmenting a scientific field based on its data, it is disadvantageous to bias this process using manually annotated examples, unless one is trying to classify texts based on preexisting categories.

On the other hand, citation networks are used to classify documents without requiring supervised training to segment the network areas. Jung and Segev (2014) performed this segmentation, and besides automatically extracting the main communities of a scientific field, their work inferred future changes in those communities. The segmentation of a scientific area based on the citation network approach depends on the construction of a citation graph. In this structure, each academic article of the studied area is a node, and every citation between those articles is an edge. Then, it is possible to cluster the graph, identifying its main sub-areas.

The clustering process identifies groups of related articles belonging to the same area. It is fundamental to understand the network topology and the problem at hand to select the clustering algorithm most suited to the target application. Although there are different types of clustering techniques, two main differences between them are keys: (1) whether they accept overlapping clusters; (2) whether they are hierarchical. Methods that accept overlapping clusters determine that an element can belong to multiple clusters simultaneously. Hierarchical clustering algorithms define different levels of clusters in which the user can navigate. It is similar to the structure of a tree, in which the root represents the broader cluster and their children the smaller sub-clusters inside it.

Also, to determine the best type of clustering for each application, it is necessary to choose among several algorithms, which is usually done based on an evaluation metric. One of the most well-known metrics to evaluate clustered graphs is modularity, proposed by Newman (2006) and Newman and Girvan (2004). It is a metric that ranges from -1 to 1 and compares the number of edges that link elements inside the same clusters with edges that connect elements from different ones. The closer the result is to 1, the better the esteemed segmentation of the network and, therefore, the organization of the clusters. Furthermore, there are several variants of the modularity metric, for example, those compared by Chen and Szymanski (2015), studying the best metrics to evaluate overlapping clusters.

The definition of the adequate clustering algorithm plays a key role in the citation network approach to automatically identify areas of a scientific field. However, regardless of the chosen algorithm, this approach neglects a fundamental variable involved in the problem, the content of the documents analyzed. In this regard, Silva et al. (2016) proposed to perform text analysis to identify keywords of their areas in addition to the clustering of citation networks. Their work aimed to construct the taxonomy of the studied scientific field. Although it improved the contextualization of the identified areas, their proposal does not use the extracted keywords to assist in the segmentation of these areas, which are determined based on metadata information only. Besides, it would be relevant to investigate the relationship among these keywords, which was not done.

Nowadays, diverse software tools are used to assist researchers in understanding how a scientific field is organized. Examples include Sci2 (Lewis & Alpi, 2017), HistCite (Garfield, 2009), CitNetExplorer (Van Eck & Waltman, 2014), CiteSpace II (Chen, 2006), Metaknowledge (McLevey & McIlroy-Young, 2017), ScientoPy (Ruiz-Rosero, Ramirez-Gonzalez, & Viveros-Delgado, 2019), Bibliometrix (Aria & Cuccurullo, 2017), and VOSviewer (Van Eck & Waltman, 2011). These software tools base their analyses on metadata of scientific articles, generating bibliometric and scientometric results. Part of them use only the articles' authors, titles, references, and citations to construct networks from which users can better understand and segment scientific fields. Some of these proposals (CiteSpace II, Metaknowledge, ScientoPy, Bibliometrix, and VOSviewer) also explore abstracts and/or keywords to improve their approaches. CiteSpace II, similarly to Silva et al. (2016), extracts keywords from the title and abstract of articles to enhance the visualization of their networks. Metaknowledge constructs a co-occurrence graph using authors' keywords. This tool then clusters the graph to segment the scientific field's main areas. ScientoPy uses the most frequent keywords of articles as the sub-areas of the studied scientific field. Bibliometrix

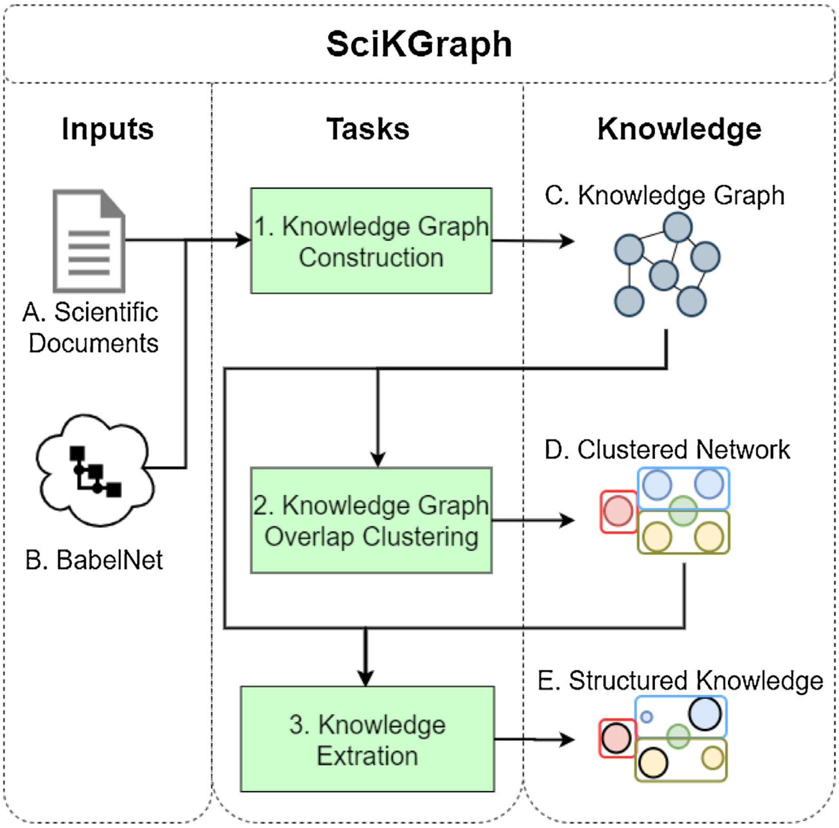

Fig. 1. SciKGraph pipeline to structure a scientific field as a knowledge graph.

uses the words of the articles' abstracts to construct a co-occurrence graph; it then clusters the graph to find the sub-areas of the studied field. VOSviewer clusters co-occurrence graphs to identify related terms (sub-areas) of scientific fields; it performs this task using not only the co-occurrence of words present in titles and abstracts, but also words from entire texts.

In our literature analysis, we found that the citation network and document classification methods could be improved to better represent and segment a scientific area when considering segmentation without predefined known groups and processing of textual data from documents.

In this work, we originally propose a framework to construct knowledge graphs to represent and segment scientific fields. The knowledge graphs are constructed based on features extracted as in document classification tasks, which are modeled as graphs, as in C-Rank (Tosi & dos Reis, 2019). Then, they are clustered similarly to the citation network, VOSviewer, and Bibliometrix approaches, identifying the key areas of the scientific field and their concepts. Moreover, our approach not only considers the textual data from the documents but also addresses their semantics based on links connecting their concepts to existing background knowledge. This may improve the understanding of the analyzed area by researchers and the analysis of the knowledge graphs constructed.

3. SciKGraph: Structuring science as a knowledge graph

This section describes SciKGraph, a framework to structure and analyze a scientific field as a knowledge graph. Fig. 1 illustrates our proposal organized into three key tasks. The "Knowledge Graph Construction" task (cf. Section 3.1), based on the C-Rank co-occurrence graph (Tosi & dos Reis, 2019), constructs a knowledge graph receiving as input BabelNet knowledge and the collection of documents that represents the studied scientific field. The "Knowledge Graph Overlap Clustering" task (*cf.* Section 3.2) clusters the previously constructed graph, identifying the main topics of the studied scientific field. The "Knowledge Extraction" task (*cf.* Section 3.3) extracts relevant information regarding the studied scientific field and its main topics based on the knowledge graph structure and its clusters.

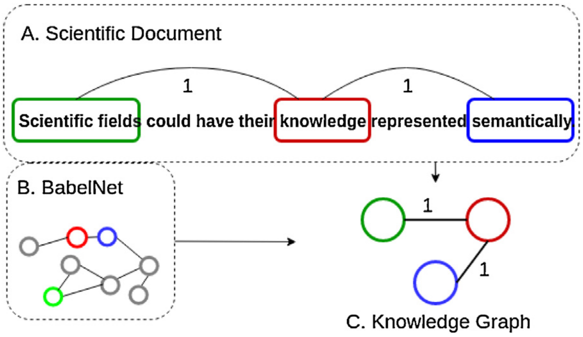
A. Scientific Document
1
1
Scientific fields could have their knowledge represented semantically
B. BabelNet
1
1
C. Knowledge Graph

Fig. 2. Representation of the knowledge graph construction task.

3.1. Knowledge graph construction

Knowledge Graph Construction is the task that outputs a scientific field knowledge graph based on a collection of documents used to describe it (*cf.* Fig. 2). First, it parses the input documents into simple texts without images, equations, or citations. Then, it sends those texts to Babelfy HTTP API, which identifies their concepts, disambiguates them, links them with their correspondents in BabelNet, and returns their babel synsets, which are the identification codes used by BabelNet to represent each concept.

The knowledge graph is constructed using as vertices all concepts identified by Babelfy in the collection of documents; edges are undirected and refer to direct co-occurrence of concepts in the text, weighted by the number of times the co-occurrence appears in the collection. This approach is based on the co-occurrence graph constructed in C-Rank (Tosi & dos Reis, 2019), but it structures a collection of documents rather than single articles.

Fig. 2 represents the construction of an illustrative knowledge graph that received as input the sentence “Scientific fields could have their knowledge represented semantically”. This sentence presents 3 correspondent concepts identified in BabelNet, which are the concepts that we use as vertices of the Knowledge Graph. In the example, the concepts labeled as “Scientific fields” and “knowledge” co-occurred in the sentence. Therefore, they are linked by an edge, which also occurred with the concepts labeled “knowledge” and “semantically”. Moreover, both edges represent co-occurrences that appeared a single time, resulting in a weight of “1”. If this co-occurrence appears again in the collection of documents used to construct the knowledge graph, this weight is updated by the number of times the co-occurrence appears.

3.2. Knowledge graph overlap clustering

Following the construction of the knowledge graph, the identification of its sub-areas is the next relevant step. The framework does this by using a clustering technique, which divides the knowledge graph into groups of vertices that are more connected among themselves than with vertices outside their group. We propose this structure to segment knowledge, considering that concepts belonging to the same area are more connected among themselves than with others. As areas may be interdisciplinary or have concepts in common, their representation with crisp clusters would not be appropriate, as such an approach does not allow concepts to belong to multiple areas simultaneously. In our work, the accurate representation of sub-areas of a scientific field is performed by using overlapping clustering algorithms.

As the Knowledge Graph itself is highly connected, it has to be preprocessed before the clustering process. Preprocessing consists of 3 steps.

1. Removal from the knowledge graph of the edges that are weighted below a certain threshold, named here $threshold_{edges}$; this decreases the network connectivity without interfering with the most important connections of the graph.
2. Removal of the nodes with higher centrality in the network, numbered by $threshold_{centrality}$. This step decreases the possibility of creating clusters centered on general concepts that are relevant for the scientific field as a whole, but not for any of the identified clusters.
3. The final step removes small disjointed sub-graphs that were created after the two previous processes.

During preprocessing, three metrics are explored: (1) centrality; (2) $threshold_{edges}$; and (3) the $thereshold_{centrality}$. Centrality is the metric used to rank the concepts based on their relevance to the network. It is used to identify the most generic concepts, which would negatively impact the clustering process, and the most relevant ones, used in the Knowledge Extraction task. In C-Rank, Tosi and dos Reis (2019) identified that degree centrality is the best metric to rank the concepts of their co-occurrence graph. Therefore, as this framework structures a scientific field based on the C-Rank graph, it uses the same centrality measure, degree centrality. It is a simple metric that considers that the higher the number of vertices connected to vertex v, the higher the centrality of v in the graph.

Regarding the thresholds, both of them are domain-dependent and vary according to the textual collection size. $threshold_{edges}$ is directly related to the amount of information and the clustering quality of the network. Increasing this value increases the quality of the clustering but reduces the amount of information in the knowledge graph. This value can change from one dataset to another and must be defined based on a manual analysis performed by the user, who has to find a number that results in a KG with a balance between amount of information and clustering quality.

$threshold_{centrality}$ is related to the granularity of the clusters, which means that the higher its value, the more small clusters are defined. Basically, the concepts inside the threshold are considered too general to represent important information for the network, which disturbs the clustering process. Therefore, they are excluded. So the user must analyze and determine the number of concepts with higher centrality in the knowledge graph that are too general and could belong to any sub-area of the scientific field. This threshold, like $threshold_{edges}$, is also found based on the studied dataset.

We recommend a semi-automated approach to achieve a fine-tuning of these parameters, in which the user is not obliged to remove nodes and edges within the thresholds. In this sense, the nodes and edges to be removed would be previously displayed to the user as a suggestion list, which can be modified based on his or her criteria. The results of varying these threshold values are presented and discussed in Section 5.

After preprocessing, the clustering process segments the knowledge graph into clusters, representing topics of the studied scientific field. This procedure must identify to which cluster each concept of the graph belongs. In our study context, some concepts fundamentally belong to different topics simultaneously. For this purpose, the clustering process must accept overlapping to represent the problem correctly. As the idea behind our approach is to elucidate a scientific field structure for researchers, we assume that the end-user has no prior knowledge of the studied domain and, therefore, cannot determine the optimal number of clusters to organize the knowledge graph.

Our investigation analyzed several algorithms to perform the clustering of the knowledge graph. Among them, only OClustR (Pérez-Suárez, Martínez-Trinidad, Carrasco-Ochoa, & Medina-Pagola, 2013) and SLPA (Xie, Szymanski, & Liu, 2011) allowed overlapped clustering without requiring the number of clusters to be input by the user. Both of them have low computational complexity, which reinforces their usage in this problem. The SLPA algorithm is not deterministic and excludes some vertices during the clustering process, so we discarded it. Consequently, we adopted the OClustR algorithm during the clustering process. It is a graph-based clustering technique that allows overlapping and identifies the optimal number of clusters automatically.

We found that as OClustR identifies the optimal number of clusters automatically, it may segment the knowledge graph into too many topics to be directly analyzed. To mitigate this issue, we suggest applying agglomerative techniques to reduce the number of clusters identified, merging them until the desired number of groups is achieved. Although this step goes against the idea of not requiring users to input the ideal number of clusters, we highlight that this step is optional. The researcher takes his/her decisions based on the previously clustered knowledge graph, which may make the task easier than choosing the number of clusters a priori.

In this work, we propose three agglomerative techniques to reduce the number of clusters obtained, selecting only the n clusters desired. In the following sections, we present the agglomerative techniques. Section 4 describes the methodology applied to evaluate and compare the effectiveness of these algorithms in the studied datasets.

3.2.1. Simple-threshold

This is a baseline that selects the n clusters with more elements and discards the others. This approach is simple and reduces the number of concepts clustered, resulting in information loss.

3.2.2. Top-modularity

Algorithm 1 describes the Top-modularity technique, which compares the modularity that would be obtained by merging two clusters or leaving them apart. The technique determines how this merging process impacts the network modularity, always aiming at its maximization. The technique fixes a sub-set of clusters that are compared with the others to reduce the computational complexity of the solution, instead of comparing all clusters among themselves.

First, Algorithm 1 receives as input a set of clusters *Clusters* and the final length n desired for this set. In line 1, it sorts *Clusters* in descending order based on the length of its elements. Then, in lines 2 and 3, it segments this set into two groups, C_{top} containing the n clusters with more elements, which are compared with all others; and C_{small} that consists of the other clusters. Next, between lines 4 and 18, the algorithm iterates over the clusters $j \in C_{small}$ until it is empty. During the iteration, between lines 7 and 15, the algorithm calculates the cluster $i \in C_{top}$ that produces the higher modularity when merged with j. Based on this, it considers i the best cluster to merge with j and, in lines 16 and 17, the algorithm merges those two clusters so $i = i \cup j$ and deletes j from C_{small}. Lastly, it returns C_{top}, a set of n clusters containing all elements from the original *Clusters* set.

Algorithm 1. Top-modularity agglomerative algorithm

```
    input  : Clusters: a set of clusters.
             n: number of desired clusters.
    output: $C_{top}$: a set of n clusters.
 1  $Clusters = descending\_sort(Clusters)$;
 2  $C_{top} = Clusters[: n]$;
 3  $C_{small} = Clusters[n :]$;
 4  for $j$ in $C_{small}$ do
 5      $best\_modularity = MAX\_FLOAT$;
 6      $best\_cluster = -1$;
 7      for $i$ in $C_{top}$ do
 8          $mod\_i = calc\_Modularity(i)$;
 9          $mod\_j = calc\_Modularity(j)$;
10          $iUj = i + j$;
11          $mod\_iUj = calc\_Modularity(iUj)$;
12          if $mod\_i + mod\_j - mod\_iUj < best\_modularity$ then
13              $best\_modularity = mod_i + mod\_j - mod\_iUj$;
14              $best\_cluster = i$;
15      end
16      $i = merge\_clusters(i, j)$;
17      delete $j$;
18  end
19  return $C_{top}$;
```

3.2.3. Best-modularity

Algorithm 2 describes the Best-modularity technique, which is similar to Algorithm 1. In its operation, in line 1, it sorts *Clusters* in descending order based on the length of its elements. Then, unlike Algorithm 1, between lines 2 and 18, it iterates over clusters $j \in$ *Clusters* until it reaches the desired number of *n* clusters, checking if this occurred in line 3. If so, in line 4, it returns *Clusters*, a set of *n* clusters containing all elements from the original inputted set. Otherwise it continues and, between lines 7 and 16, calculates the best cluster $i \in$ *Clusters*, $i \neq j$ to merge with *j*. Then, in lines 17 and 18, after determining the best cluster *i*, the algorithm merges it with *j* so $i = i \cup j$ and deletes *j* from *Clusters*. This algorithm compares all clusters among themselves. Therefore, it always merges the clusters that produce the best modularity at the cost of higher computational complexity.

Algorithm 2. Best-modularity agglomerative algorithm

```
    input  : Clusters: a set of clusters.
             n: number of desired clusters.
    output: Clusters: a set of n clusters.
 1  $Clusters = descending\_sort(Clusters)$;
 2  for $j$ in $Clusters$ do
 3      if $length(Clusters) \leq n$ then
 4          return $Clusters$;
 5      $best\_modularity = MAX\_FLOAT$;
 6      $best\_cluster = -1$;
 7      for $i$ in $Clusters$ do
 8          if $i \neq j$ then
 9              $mod\_i = calc\_Modularity(i)$;
10              $mod\_j = calc\_Modularity(j)$;
11              $iUj = i + j$;
12              $mod\_iUj = calc\_Modularity(iUj)$;
13              if $mod\_i + mod\_j - mod\_iUj < best\_modularity$ then
14                  $best\_modularity = mod_i + mod\_j - mod\_iUj$;
15                  $best\_cluster = i$;
16      end
17      $i = merge\_clusters(i, j)$;
18      delete $j$;
19  end
```

3.3. Knowledge extraction

Knowledge extraction is the last task performed in our proposal. It takes as input the previously constructed structures, organizes them, and outputs knowledge ready to be analyzed by the user. This task produces three results: (1) the segmented topics of a scientific field; (2) their main concepts sorted by relevance; (3) their keyphrases.

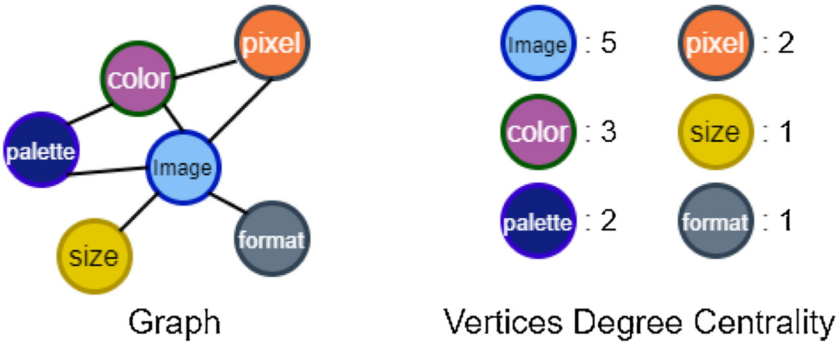

Fig. 3. Example of degree centrality usage to calculate relevance of concepts.

The segmented topics are represented by the clusters obtained in the knowledge graph clustering task (Section 3.2). Their structure might illuminate the topology of a scientific field for scientists; their relation may indicate the amount of interaction between topics; their overlapping areas specify the correlations between them.

The relevance of the concepts inside each topic is fundamental to the user analysis as it is complicated to extract meaning from hundreds of concepts together. Sorting concepts based on their relevance enables users to identify those that better represent each topic and, therefore, can be the basis for future analysis. The relevance of a concept is directly connected with its centrality in the network. Degree centrality is explored to calculate the relevance of each concept. Boudin (2013) showed the effectiveness of this metric compared to similar ones and Tosi and dos Reis (2019) corroborated it, determining that degree centrality was the best metric to identify relevant concepts in the structure they proposed.

Fig. 3 presents an example by illustrating a graph with 6 vertices connected among themselves. The vertex labeled “Image” is connected to another 5 vertices, making its degree centrality equal to 5. It is the vertex with higher degree centrality, which means that it is considered the most relevant vertex of the graph. Therefore, if one would have to describe the graph of Fig. 3 with one of its vertices, the most relevant one, “Image”, would be appropriate. Furthermore, in the proposed framework, instead of using the total number of vertices connected to a concept as its degree centrality, we use its normalized value, which is equal to the division of the number of vertices connected to a concept by the total number of concepts in the knowledge graph, generating a value in the range $[0, 1]$.

Still, because of the complexity of the task, we cannot effectively extract the main concepts of the network based on degree centrality only. Considering this, to enhance our approach, we use three out of four heuristics presented by Tosi and dos Reis (2019). The heuristics are: (1) discard concepts labeled with words that are not nouns, verbs, or adjectives; (2) cut lower-ranked concepts; and (3) favor the centrality of concepts labeled with multiple words, powering them by the inverse of their length. Although these heuristics are designed to assist in the keyphrase extraction task, the idea behind them suits our framework. Therefore, we use them to reduce noisy concepts that can impair the identification of the main concepts of the knowledge graph.

Lastly, keyphrases are expressions used to represent the content of textual structures. Unlike the most relevant concepts, keyphrases can be formed by multiple concepts. Usually they are used to highlight the main topics of a document. Nevertheless, it is not common to identify keyphrases from whole scientific fields or their topics, as their extraction is not easy without a background dataset or supervised training.

In our solution, keyphrase extraction is performed in the knowledge graph and its clusters using an adaptation of the C-Rank algorithm (Tosi & dos Reis, 2019), which does not require external data to extract keyphrases. Considering that we structure a scientific field based on the C-Rank graph and use it to extract keyphrases, we can expect and assume similar quality from the keyphrases identified. C-Rank takes as input a document, uses it to construct a co-occurrence graph, ranks the graph based on the degree centrality of its nodes, applies 4 heuristics, merges concepts to form keyphrases, and outputs a ranked list of keyphrases.

In our work, as keyphrases are extracted from the knowledge graph rather than single documents, three adaptations are necessary in C-Rank for this investigation. First, as the knowledge graph was previously constructed and ranked based on the C-Rank co-occurrence graph, graph construction and ranking can now be skipped. Second, unlike C-Rank, SciKGraph uses undirected edges; thus, instead of distinguishing between incoming and outgoing edges, SciKGraph considers them the same. Third, the heuristic concerning the exclusion of candidate keyphrases appearing at the beginning of the document cannot be applied. This heuristic considers that keyphrases are usually introduced at the beginning of a document, and as the framework extracts keyphrases from whole areas, this factor is irrelevant to its context.

Consequently, to extract keyphrases from the knowledge graph, one must apply the other C-Rank heuristics, which are the same ones used to identify the main concepts of the knowledge graph (previously described). Then, one must merge keyphrases formed by multiple concepts.

Finally, the knowledge graph is re-ranked, considering the concepts formed by multiple concepts, and the list of most relevant keyphrases is outputted to the user. The extraction of keyphrases from sub-areas of the knowledge graph requires the creation of a sub-graph with its vertices and edges and the application of the explained procedure.

4. Experimental evaluation results

This section describes the methodology applied to evaluate SciKGraph and the obtained results in several analyses. First, we explain how we implemented the developed solution and present the datasets used in the evaluation (Section 4.1). Section 4.2 describes the procedure performed to evaluate SciKGraph, by providing details of the purpose and organization of the conducted analyses.

4.1. Implementation and datasets

SciKGraph was developed in Jupyter Notebook (Kluyver et al., 2016) to facilitate its reproducibility. This allows displaying code along with textual elements and figures, enhancing interactivity between user and content. Our Jupyter Notebook uses Python 3 as the programming language and is available online.[1]

The concept linking process between our textual documents and BabelNet was implemented by using *pybabelfy*[2] (a library that links Babelfy HTTP API with Python) with minor changes to work with Python 3. We constructed our knowledge graph with *networkx* (Hagberg, Swart, & S Chult, 2008), a Python package for the study of complex networks. In addition, we implemented OClustR and the proposed agglomerative methods (*cf*. Section 3.2), both of them available online.[1,3]

The evaluation of SciKGraph for structuring a scientific field relied on two datasets. We used the WOS-5736 (Kowsari et al., 2017) and the AI datasets. Below we explain the reason for choosing them and present their characteristics.

4.1.1. WOS-5736

The WOS-5736 (WOS) dataset was constructed by Kowsari et al. (2017) using *Web of Science* data and meta-data of published papers to validate document classification methods. It is composed of 5736 annotated academic abstracts with 3 categories and 11 subcategories. The domain from those categories are very different among themselves as they represent "Psychology", "Biochemistry", and "Electrical Engineering" areas, with 3, 4, and 4 sub-areas, respectively.

Our "Knowledge Graph Overlap Clustering" identifies topics of a scientific field despite not being originally developed to classify documents based on pre-existing categories. The WOS dataset was used to determine if there is a relation between the topics identified by SciKGraph and those predefined categories (expected answer). This dataset has annotated data and enables us to compare our method with document classification algorithms.

4.1.2. AI

Unlike WOS, we constructed the AI dataset. It is composed of 1,018 academic articles, published between 1962 and 2019, crawled from the IEEE Xplorer website[4] using "Artificial Intelligence" as the search term, sorting the results based on their number of citations.

The documents were obtained in PDF, which we converted to XLST using GROBID (Grobid), a machine learning library to parse PDF. Then, we removed their citations, mathematical formulas, and images and converted the documents to simple texts. In this process, 33 articles could not be correctly parsed and were discarded, resulting in the 1018 articles composing the dataset.

We chose to construct this dataset because we did not find one composed of full academic articles available to use, which we wanted to adopt to compare its results with those of a dataset constructed using only the abstracts from articles, disregarding the rest of their content. Moreover, as the idea of our framework is to structure a scientific field, the usage of a dataset automatically constructed exemplifies how the model deals using real noisy data.

4.2. General procedure and organization of analyses

Table 1 presents the quantitative and qualitative analyses conducted with the respective datasets (performed in the WOS and AI datasets).

For the development of our analyses, we firstly defined the AI and WOS datasets as representations of the scientific fields we would like to represent. Then, we constructed a knowledge graph for each one using their data as input. Afterwards, we clustered both knowledge graphs, identifying their topics.

We observed that the high amount of clusters could negatively impact the visualization of a scientific field. On the basis of this motivation, we studied the problem of minimizing the number of clusters (cf. Sections 4.4 and 4.5). Based on the

[1] https://github.com/maurodlt/SciKGraph.

[2] https://github.com/aghie/pybabelfy.

[3] https://github.com/maurodlt/OClust-R.

[4] https://ieeexplore.ieee.org/.

Table 1
Proposed analyses and used datasets. This table indicates the list of evaluations conducted; for each one it describes the datasets used (WOS, AI, or both datasets (WOS – AI)).

	Accuracy	Modularity	Structure	Content
Knowledge graph construction and visualization			WOS | AI	WOS | AI
Accuracy comparison of the agglomerative techniques	WOS			
Modularity comparison of the agglomerative techniques		WOS | AI		
Top-modularity accuracy and modularity correlation	WOS	WOS		
Knowledge graph size and modularity correlation		AI	AI	
Knowledge graphs clusters relations		WOS | AI	WOS | AI	
Key concepts and clusters keyphrases				AI

obtained results, we determined that the *Top-modularity* technique presented the best trade-off between results produced and computational cost. We applied it to obtain 20 clusters in the AI and 15 clusters in the WOS knowledge graphs (cf. Section 4.3) to assist the visualization of our analyses.

In order to better study and visualize the knowledge graph structure, we used crisp clusters in our analyses, defined by maintaining overlapping vertices only in the biggest clusters to which they belong. Therefore, if a concept belongs to two clusters composed of 200 and 150 concepts each, it will be excluded from the one composed of 150 concepts and maintained in the bigger one. In addition, we experimentally studied the topology of the knowledge graphs (cf. Sections 4.6 and 4.7) and extracted knowledge from them, such as their key concepts, keyphrases, and relations among their clusters (cf. Sections 4.8 and 4.9). Below we further explain the conducted analyses.

4.2.1. *Knowledge graph construction and visualization:*

This analysis (cf. Section 4.3) provides the results of the construction of the AI and WOS knowledge graphs using SciKGraph. It allows the user to visualize the structure obtained by applying the proposed framework to whole-documents and abstracts-only datasets.

4.2.2. *Accuracy comparison of the agglomerative techniques*

The accuracy of the document classification problem quantifies to what extent the identified topics are related to pre-existing areas. We compare the accuracy obtained in classifying documents using the constructed knowledge graph, varying the agglomerative techniques to reduce the number of clusters (cf. Section 4.4). This analysis was performed using the WOS dataset and calculates the accuracy of classifying documents in their respective areas and sub-areas. It could not be performed in the AI dataset because it requires the use of an annotated dataset, which is not the case here.

4.2.3. *Modularity comparison of the agglomerative techniques*

This analysis (cf. Section 4.5) compares the clusters' modularity variation after applying the suggested agglomerative techniques. It is performed using both the AI and WOS knowledge graphs. Moreover, it investigates modularity variation by using crisp and overlapping clusters.

4.2.4. *Top-modularity accuracy and modularity correlation*

This analysis (cf. Section 4.6) verifies whether there is a direct correlation between the modularity of a set of clusters and their accuracy when classifying documents. It is performed using the WOS knowledge graph and the Top-modularity technique to reduce the number of clusters identified. It could not be performed in the AI dataset because it requires the use of an annotated dataset, which is not the case here.

4.2.5. *Knowledge graph size and modularity correlation*

This investigates if there is a direct correlation between the size variation of the proposed knowledge graph and its modularity (cf. Section 4.7). The size of the knowledge graph varies by modifying the $threshold_{edges}$ parameter, which allows updating of the number of nodes and edges of the knowledge graph, excluding the most irrelevant ones. We show its results in the AI knowledge graph but we observed the same trends in the WOS graph.

4.2.6. *Knowledge graphs cluster relations*

This analysis (cf. Section 4.8) exemplifies how one may analyze the relations among the topics of the studied scientific field using the SciKGraph framework. It is performed by using the overlapping clusters identified through the Top-modularity technique in the WOS and AI knowledge graphs.

4.2.7. *Key concepts and cluster keyphrases*

This analysis (cf. Section 4.9) shows the key concepts extracted from the AI knowledge graph. Furthermore, it identifies and presents the keyphrases from its clusters, defined by using the Top-Modularity technique. It was not performed in the WOS knowledge graph because SciKGraph is designed to extract key concepts and keyphrases from representations of a single scientific area. The WOS dataset is composed of multiple areas of science. Hence, the key concepts and keyphrases

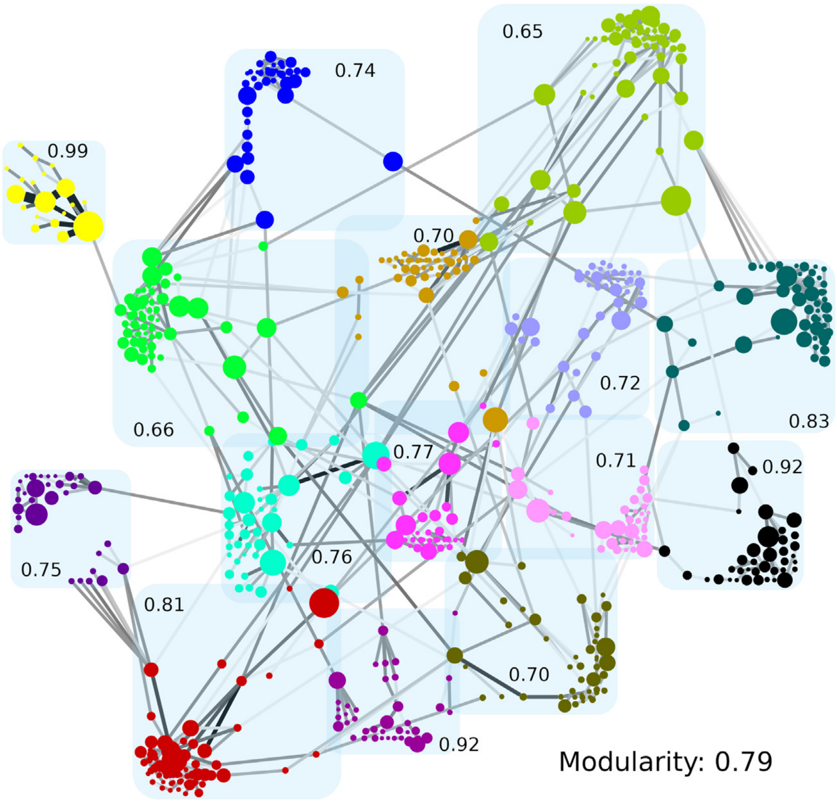

Fig. 4. WOS knowledge graph, its modularity, and the percentage of inner edges of each cluster.

extracted from the WOS dataset would not properly represent examples of what someone could expect using SciKGraph, nor be meaningful to analyze.

4.3. Knowledge graph construction

We constructed the WOS-5736 knowledge graph and identified 37,591 different concepts and 390,296 connections between them. Prior to clustering, the preprocessing step reduced them to 667 concepts and 665 connections. The clustering procedure identified 210 different clusters. Since this volume of clusters is hard to analyze, we assessed different agglomerative techniques to reduce it (cf. Sections 4.4 and 4.5), reaching 15 clusters with 1.087 of overlapping rate using the *Top-Modularity* algorithm.

In the construction of the AI knowledge graph, we identified 40,373 different concepts and 834,678 connections between them. The preprocessing step reduced them to 2495 concepts and 6602 edges, which were clustered in 436 distinct topics. Different agglomerative techniques were analyzed to merge these clusters and favor their interpretation, reaching 20 clusters with 1.188 of overlapping rate using the *Top-Modularity* algorithm.

We present the whole knowledge graph to enable a broader visualization of the studied scientific field. The nodes represent concepts and are sized based on the number of times they appeared in the textual collection, the edges represent the co-occurrence of the concepts and their width is based on their weight, and the clusters are crisp. This visualization was constructed using Cytoscape (Shannon et al., 2003) software and CoSE (Dogrusoz, Giral, Cetintas, Civril, & Demir, 2009) algorithm. We also exhibit the Q_{ov}^{L} modularity metric, as suggested by Chen and Szymanski (Chen & Szymanski, 2015), and the percentage of edges that each cluster has linking its own elements. This enabled us to observe the effectiveness of the clustering process and how the concepts in the network interact with each other.

4.3.1. Results for the WOS dataset

Fig. 4 illustrates the knowledge graph created from the WOS dataset with nodes weighted based on their degree centrality and colored based on a crisp clustering.

Fig. 4 presents the whole WOS knowledge graph, from which one can attest the effectiveness of the clustering, observing how well-segmented and clustered the concepts are, with 0.795 of modularity. The yellow cluster, for example, has only a single edge linking it with another cluster. Therefore, 99% of its edges link concepts that also belong to the yellow cluster.

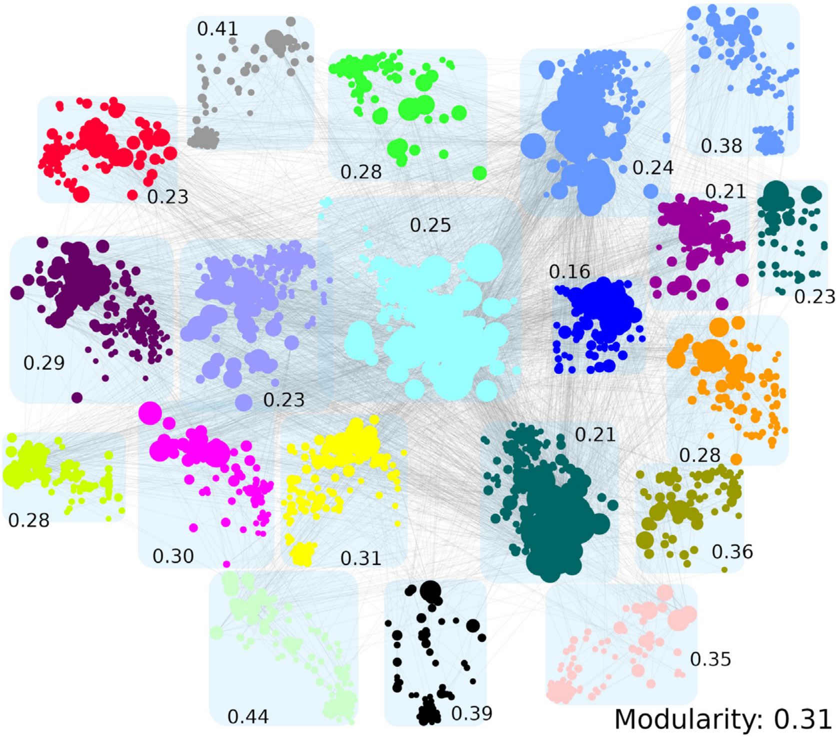

Fig. 5. Artificial Intelligence knowledge graph, its modularity, and the percentage of inner edges of each cluster.

This visualization allows the analysis of the connections among concepts, which is impaired in this example because the image size does not render the concepts' labels large enough to be readable.

4.3.2. Results for the AI dataset

Fig. 5 presents the knowledge graph constructed from the AI dataset with nodes weighted based on their degree centrality and colored based on a crisp clustering.

We observe in Fig. 5 that, unlike Fig. 4, the segmentation of the clusters is not so well defined, with many connections among clusters, which impacts their percentage of inner edges. Fig. 5 illustrates different topics from the same scientific field, segmenting the AI knowledge graph into 20 groups with almost 10 times more connections than the WOS knowledge graph. This explains the edges connecting different clusters.

This analysis enables a broader visualization of the studied areas. The plotted WOS and AI knowledge graphs, along with their modularity, later discussed in Section 4.5, show the effectiveness of the clustering process and how the concepts of the network interact with each other. We note that some concepts in both representations could have been clustered differently, such as the purple nodes on the left of Fig. 4, which are segmented into two different groups. We constructed this analysis using crisp clustering because it is impracticable to represent a network with 15 overlapping clusters in a single image. This issue occurs because each cluster represents a dimension of the network, and, ideally, this network should be represented using a 15-dimension plot rather than an image that has only 2 dimensions. Consequently, despite not reducing the dimensionality of the problem, we use the crisp representation to diminish the overlap between those dimensions, increasing their disconnection and improving their visualization by using a simple image.

4.4. Accuracy comparison of the agglomerative techniques

Using the WOS dataset allows us to compare the topics identified by our proposed solution with areas and sub-areas previously annotated in the dataset. Even though the aim of our proposal is not to classify documents based on pre-existing categories, we explored this analysis as a criterion to assess to what extent the topics are coherently segmented. As this analysis requires an annotated dataset and the AI dataset is not annotated, we used only the WOS dataset for accuracy measurement.

The goal in this analysis is to identify in which predefined area and sub-area a determined document belongs. To this end, we take as input the WOS clustered knowledge graph and the WOS dataset, which has documents annotated with their correct areas and sub-areas. Then, we divide the dataset into two sub-sets, *train* and *test*. The *train* set contains the first 80% of the WOS documents, and the *test* set the complementary 20%. Next, we use the training set to *train* which annotated areas

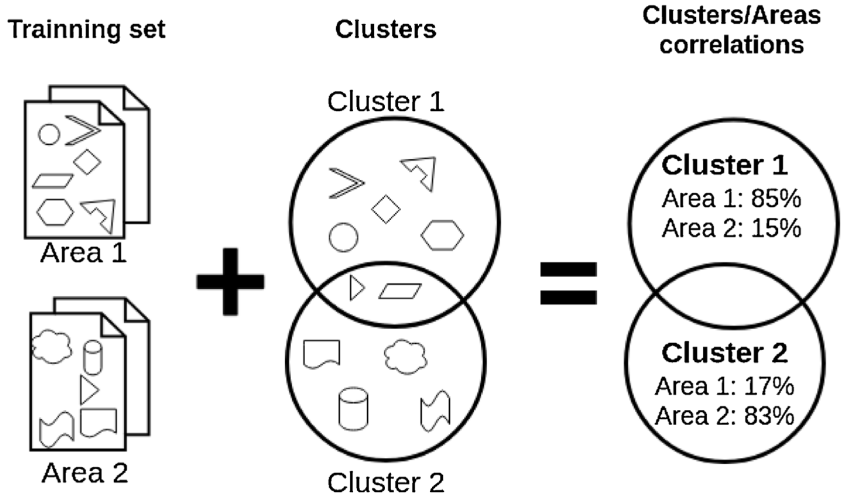

Fig. 6. Example of clusters/areas correlation percentage calculation.

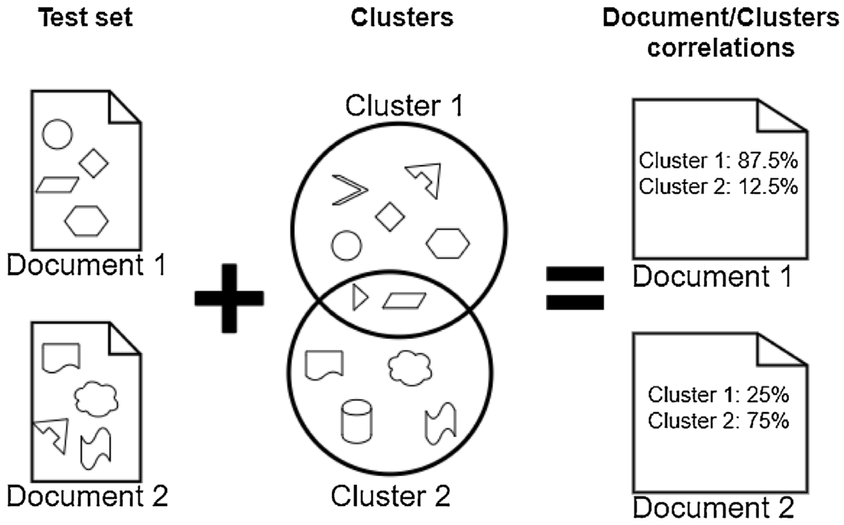

Fig. 7. Example of document/clusters correlation percentage calculation.

and sub-areas each of the knowledge graph clusters represents. Afterward, we determine which clusters better describe the content of *test* documents to be classified. Finally, we infer the area and sub-area of documents, which are the ones that the cluster which better describes the analyzed document represents.

For instance, Fig. 6 exemplifies the identification of which annotated area each cluster represents. It receives the training set and the clusters, identifies the concepts they have in common, and outputs the clusters/areas correlation percentage. In this example, Cluster 1 shares 6 of 7 concepts with the documents from Area 1, resulting in a correlation of 85%.

Fig. 7 shows how to compute in which cluster each document belongs. First, it identifies the document/clusters correlation percentage. In our example, all concepts from Document 1 are shared with Cluster 1, but one of them is also shared with Cluster 2. Therefore, we assume that the shared concept belongs equally to both clusters, and, consequently, half of its belonging coefficient is directed to Cluster 1 and the another half to Cluster 2. Thus, in this example, Document 1 has 87.5% of correlation with Cluster 1.

Furthermore, after identifying the clusters/areas and document/clusters correlations, we process the probability of the document belonging to each area. For example, in Fig. 8, Document 1 has 0.875 of correspondence with Cluster 1, which has 0.85 of correspondence with Area 1; Document 1 also has 0.125 of correspondence with Cluster 2, which has 0.17 of correspondence with Area 1; the probability of Document 1 belonging to Area 1 is $P(Document_1 \in Area_1) = 0.875 * 0.85 + 0.125 * 0.17 = 0.765$. Hence, as Area 1 is the area to which Document 1 has the highest probability of belonging, we indicate that it is the main area of Document 1.

Consequently, considering that the documents from the *test* set are annotated with the areas and sub-areas to which they belong, we evaluate the accuracy of the results of our framework, calculating the percentage of document areas and

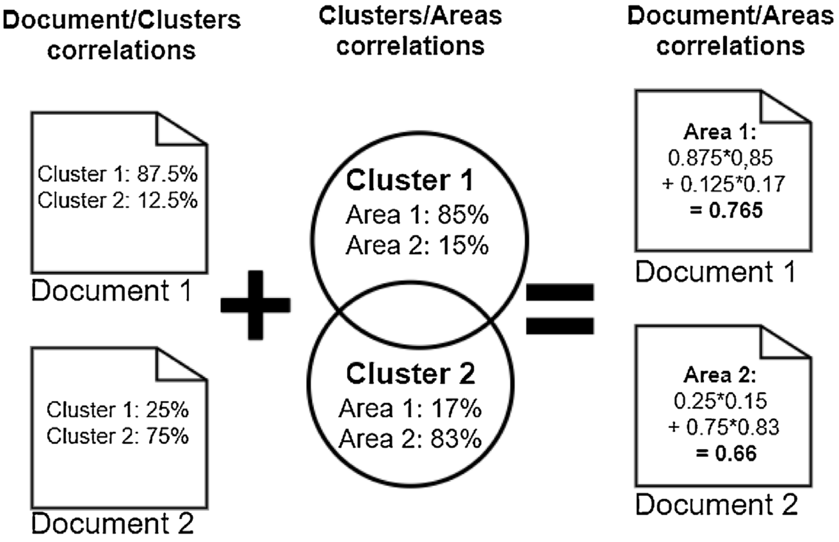

Fig. 8. Example of the area of a document.

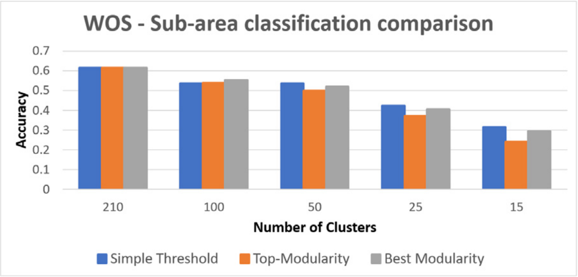

Fig. 9. Accuracy comparison of agglomerative methods in classifying WOS documents in pre-annotated sub-areas.

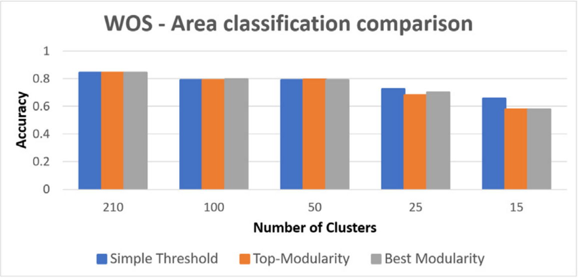

Fig. 10. Accuracy comparison of agglomerative methods in classifying WOS documents in pre-annotated areas.

sub-areas that it correctly determined. Accordingly, we can use this metric to compare the accuracy obtained by varying the proposed agglomerative techniques and the number of final clusters that they identify.

Fig. 9 presents the accuracy obtained in classifying documents in their respective topics.

From the results presented in Fig. 9, we observe that the reduction of the number of clusters negatively impacted the document classification accuracy. All three proposed agglomerative techniques followed the same trend and obtained similar results. Even though the accuracy of Top-Modularity decreased faster than the other methods, it obtained similar results in segmenting higher amounts of clusters.

Fig. 10 shows the accuracy obtained in classifying documents in their respective areas.

Fig. 10 exhibits similar results to Fig. 9. It illustrates that the reduction of the number of clusters impacts the accuracy obtained and that the proposed agglomerative methods obtain similar results. In contrast, it achieves higher accuracy because it is simpler to identify the right area among three options than the right sub-area among 11.

Our results reveal that the use of the agglomerative technique negatively impacted accuracy in the document classification task. However, this was expected as, theoretically, the OClustR algorithm had already found the optimal number of clusters

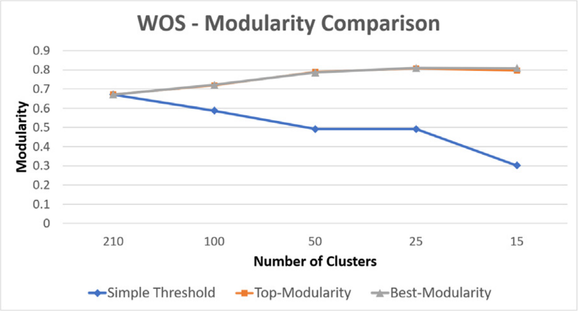

Fig. 11. Modularity comparison of agglomerative methods merging WOS crisp clusters.

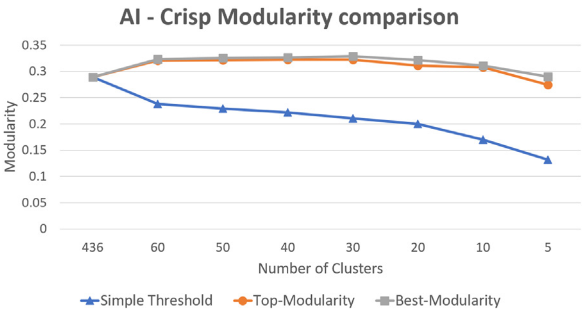

Fig. 12. Modularity comparison of agglomerative methods merging AI crisp clusters.

of the dataset. Thus, the process of merging clusters would only degrade this segmentation. The *Simple Threshold* technique obtained better results, followed by *Best-Modularity* and *Top-Modularity*. This occurs because *Simple Threshold* excludes concepts instead of re-classifying them in other clusters, reducing noise and the complexity of the problem. *Best-Modularity* always chooses the best clusters to merge, so it was expected that it would perform better than *Top-Modularity*. We note that the difference between the accuracy of these two metrics is low compared to the computational complexity difference between them.

4.5. Modularity comparison of the agglomerative techniques

This analysis aims to determine how segmented the obtained clusters are and, therefore, how disjointed the identified topics are. For this purpose, we use the modularity metric, as suggested by Chen and Szymanski (2015). This metric was calculated not only to determine whether the knowledge graph was clustered correctly, but also to compare how well segmented the obtained clusters were according to the agglomerative methods proposed.

Furthermore, we investigated how the overlapping clustering impacts the modularity obtained, performing this analysis in both overlapping and crisp clusters.

4.5.1. Results for the WOS dataset

Fig. 11 illustrates the modularity variance using the different agglomerative techniques in the WOS dataset. We observe that the Top-Modularity and Best-Modularity methods, when applied to the WOS crisp clusters, obtain almost identical modularities, increasing their results when we reduce the number of output clusters. On the other hand, the Simple Threshold method loses modularity in this process, achieving the worst results.

4.5.2. Results for the AI dataset

Fig. 12 presents the modularity variance using the different agglomerative techniques in the AI crisp clusters, whereas Fig. 13 describes the same variance, but using the overlapping clusters.

We note that Figs. 12 and 13 present the same trends as those observed in the WOS dataset. However, we highlight that by comparing the results from these two analyses, we observe a difference between the modularity calculated with crisp and overlapping clusters, the latter presenting the worst modularity.

We found that the overlapping clustering decreased cluster modularity by an average of 14%. This showed that unlike the accuracy analysis, *Simple Threshold* obtained the worst modularity; the other two techniques improved the modularity metric by reducing the number of clusters until reaching 20 clusters. The analysis performed using the WOS knowledge graph identified the same trends using crisp (cf. Fig. 11) and overlapping clusters.

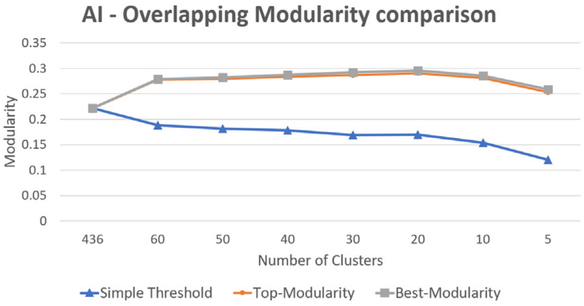

Fig. 13. Modularity comparison of agglomerative methods merging AI overlapping clusters.

Table 2
Pearson's correlation table comparing modularity and accuracy values obtained by varying the number of clusters in the WOS dataset using the Simple Threshold (ST), Top-Modularity (TM), and Best-Modularity (BM) algorithms. Statistically significant values highlighted with †.

		Modularity			Accuracy		
		ST	TM	BM	ST	TM	BM
Modularity	ST	1†					
	TM	−0.897†	1†				
	BM	−0.915	0.999†	1†			
Accuracy	ST	0.989†	−0.840	−0.861	1†		
	TM	0.979†	−0.807	−0.830	0.993†	1†	
	BM	0.979	−0.791	−0.816	0.991†	0.996†	1†

From 20 clusters, modularity decreased using the *Top-Modularity* and *Best-Modularity* techniques (cf. Figs. 12 and 13), achieving almost identical results. We interpret this decrease as an over-reduction of the number of clusters of the AI knowledge graph. Thus, we are basically reducing the number of clusters of the scientific field to less than the actual number of sub-areas of that field, which impacts the modularity obtained. Based on this, we recommend users to observe the modularity of their knowledge graph and choose the number of clusters that best improve this metric. In our analyses, we divided the AI knowledge graph representation into 20 clusters.

Furthermore, the AI clusters achieved lower modularity compared to the WOS clusters, but this was already expected as the AI dataset is composed of articles of a single area, and WOS is composed of three distinct areas, which are easier to be segmented. Newman and Girvan (Newman & Girvan, 2004) state that real-world community structures typically present modularity values between 0.3 and 0.7. Thus, the fact that the *Top-Modularity* metric and the crisp clusters stayed most of the time above 0.3 indicates that the clusters were well defined enough to reach the modularity value of other real-world known community structures. This is an indication that the clustering occurred successfully.

4.6. *Top-modularity accuracy and modularity correlation*

This analysis studies whether there is any direct correlation between the accuracy and modularity metrics investigated. As accuracy was calculated only for the WOS dataset, we did not use the AI dataset in this analysis.

Table 2 presents Pearson's correlation coefficients measuring the correlation between the modularity and accuracy values obtained by varying the number of clusters using the Simple Threshold (ST), Top-Modularity (TM), and Best-Modularity (BM) algorithms. Moreover, we highlighted with "†" the statistically significant coefficients, the ones with *p-value* < 0.05.

This analysis shows statistically significant correlations of the accuracy values obtained by applying the distinct agglomerative algorithms. Comparing modularity and accuracy values, the only statistically significant data observed were the positive correlations between the Simple Threshold modularities and the Simple Threshold and Top-Modularity accuracies. However, analyzing the correlations between the modularity of the Top-Modularity and Best-Modularity algorithms and their respective accuracies, not only do they lack statistical significance, but, unlike the Simple Threshold algorithm, present a negative correlation. Therefore, this analysis indicates that there is no clear correlation between the modularity and accuracy values obtained that is independent from the agglomerative algorithm used.

Unlike what was expected, we found that there is no clear correlation between cluster modularity and accuracy in the document classification task. However, this does not mean that modularity should not be further analyzed in this context, since it still quantifies cluster segmentation.

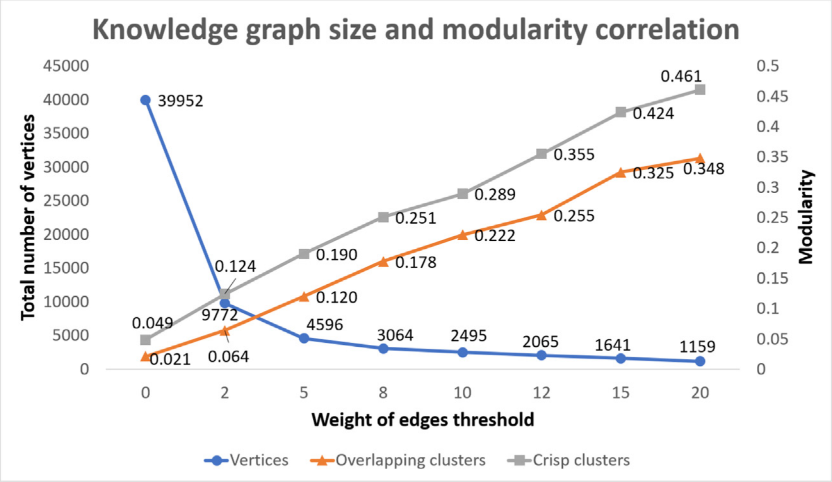

Fig. 14. Analysis of correlation between Knowledge Graph size and modularity by varying the $threshold_{edges}$ parameter.

4.7. Knowledge graph size and modularity correlation

This analysis searches for correlations between the size of the knowledge graph and the modularity of its clusters. This result can help researchers define the $threshdold_{edges}$ value. If a correlation between these values is observed, one can modify $threshdold_{edges}$ to change the size of the knowledge graph and improve its segmentation.

Fig. 14 presents the results for the AI dataset. The x-axis refers to the $threshold_{edges}$ value parameter, whereas the y-axis presents the number of vertices and modularity. The results show an inverse correlation between the size of the AI knowledge graph and its modularity. By increasing the $threshold_{edges}$ value, the number of vertices in the network exponentially decreases. On the other hand, knowledge graph modularity linearly increases for both overlapping and crisp clusters. Therefore, this experiment shows that one can increase the $threshold_{edges}$ value to improve the segmentation of the knowledge graph.

The analysis of the *Simple Threshold* metric indicates that the amount of nodes and edges of a graph is related to its modularity and accuracy in classifying documents. The results shown in Fig. 14 corroborate this assumption, illustrating that when the *edges threshold* value increases, the total number of concepts of the knowledge graph exponentially decreases and modularity linearly increases. Considering this, we recommend users choose the minimum $threshold_{edges}$ value that produces knowledge graphs with at least 0.30 of modularity. Actually, we used this experiment to set $threshold_{edges} = 10$, so the modularity obtained in the AI knowledge graph would stay close to 0.30, excluding the minimum number of concepts as possible.

4.8. Knowledge Graphs cluster relations

After determining clustering correctness, we calculated the connections between the clusters. This analysis can be used by researchers to understand the relations among areas of a scientific field. It is represented as a graph, in which a vertex represents each cluster, and all the connections between two clusters are merged in a single edge. This visualization is constructed using Cytoscape (Shannon et al., 2003), which enables us to graphically observe not only to what extent the clusters are highly connected to themselves compared to other connections, but also which clusters interact more with one another.

4.8.1. Results for the WOS dataset

Fig. 15 presents the relations between the WOS clusters. We observe the topic's relations and the high modularity of the WOS knowledge graph, showing how clusters have more connections within themselves than with other clusters. Take cluster 12 as an example; among its relations, its self-connection is stronger than the others, which are in general weak, except for the one that links it with cluster 7, illustrating their strong relation compared to the other cluster 12 connections.

4.8.2. Results for the AI dataset

Fig. 16 illustrates the relations between the AI clusters. We observe the same characteristics obtained with the WOS dataset (Fig. 15). Take as an example Cluster 16; its self-connection is strong, and, compared to all other connections in the graph, its connections with other clusters are weak. This means that Cluster 16, compared to most of the other clusters, represents a more independent topic.

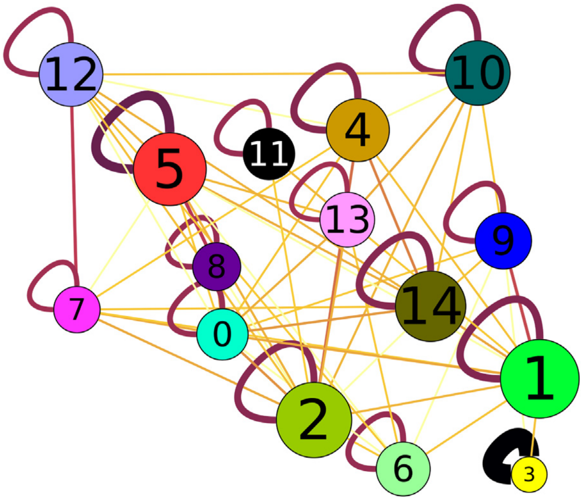

Fig. 15. Graph of WOS cluster relations. The vertices' sizes are proportional to the clusters' amount of concepts and their relevance. The width and color of the edges are proportional to the number and strength of the connections between clusters – stronger connections are represented by thicker and darker edges.

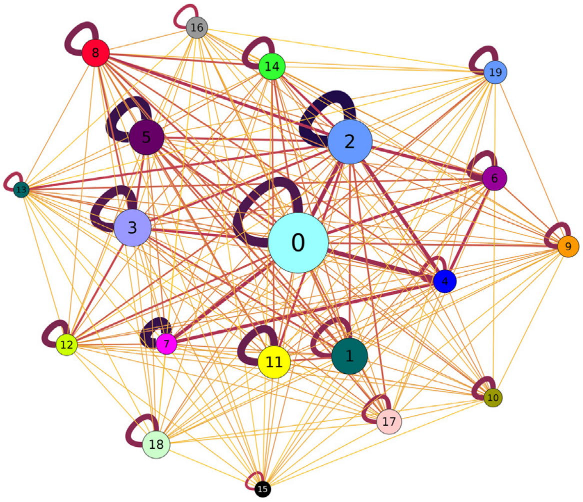

Fig. 16. Graph of AI cluster relations. The vertices' sizes are proportional to the clusters' amount of concepts and their relevance. The width and color of the edges are proportional to the number and strength of the connections between clusters – stronger connections are represented by thicker and darker edges.

The representation used to plot the knowledge graphs allows us to confirm the high modularity of the clusters, with stronger self-connections than the other ones. This might assist researchers in identifying the topics that interact more among themselves. For example, in Fig. 15, topic *4*, which interacts more with *13* and *14* than it interacts with the others.

4.9. Key concepts and cluster keyphrases

In order to analyze the main concepts of the knowledge graph, we extracted them based on their degree centrality and the C-Rank heuristics. To quantitatively assess the quality of the extracted keyphrases we would need a scientific field with manually annotated keyphrases, with which we could compare our results. However, this dataset is not available and is impracticable to construct in a short-term period. As we constructed the knowledge graph using a similar structure to the one used in C-Rank, we considered that the same metric can be used to identify the key concepts of the graph. Similarly, we presume that as the SciKGraph structure was based on the C-Rank algorithm, it can be used to extract the keyphrases from the knowledge graph clusters. In this sense, we qualitatively analyzed each cluster as a separate graph and applied C-Rank to extract its keyphrases.

Table 3
AI Knowledge Graph key concepts sorted by degree centrality and C-Rank (Tosi & dos Reis, 2019) heuristics.

1. Neural network	4. Feature space	7. Computing science
2. Learning algorithms	5. Training set	8. Pattern recognition
3. Computer vision	6. Evolutionary process	9. Provide

Table 4
AI cluster keyphrases ranked using C-Rank. Bold values highlight examples of clusters that can be used to summarize the topics that they represent.

Clusters	Keyphrases		
0	1. provide	2. data table	3. paper
	4. information	5. mode	6. task
1	1. region graph	2. face database	3. regions
	4. size steps	5. space input	6. texture color
2	1. performs best	2. performs task	3. input pattern
	4. performs	5. generated	6. steps
3	1. node layer	2. node	3. weight connection
	4. node input	5. node rules	6. layer input
4	1. percentage rate	2. see table	3. compare table
	4. compare	5. size step	6. compare methods
5	**1. changes illumination**	**2. computing science**	**3. illumination variation**
	4. changes	**5. scale change**	**6. camera view**
6	1. classifiers ensemble	2. achieve best	3. achieve
	4. achieve high	5. recognition task	6. SVM classifier
7	**1. computer vision**	**2. learning algorithms technique**	**3. learning algorithms**
	4. work	**5. computer vision learning algorithms**	**6. computer vision task**
8	1. positive sample	2. vectors input	3. training set
	4. vectors	5. matrices weight	6. vectors represent
9	1. term linguistic	2. term	3. single element
	4. define term	5. single best	6. best
10	1. visual information	2. age estimation	3. range
	4. setting estimation	5. depth information	5. setting
11	1. distance steps	2. position orientations	3. based classifiers
	4. steps size	5. setting size	6. face detector
12	**1. neural network**	**2. evolutionary process**	**3. neural network NN**
	4. control input	**5. SVM classifier**	**6. based**
13	1. increased decreases	2. improved	3. improved recognition
	4. reduce	5. recognition rate	6. technique based
14	**1. frames sequence**	**2. steps size**	**3. sequence videos**
	4. weight rules	**5. frames**	**6. weight inputs**
15	1. attributes	2. input space	3. attributes selection
	4. based classifiers	5. best position	6. strategy based
16	1. input patterns	2. patch size	3. face representation
	4. patch	5. local forms	6. face patterns
17	1. generated rules	2. initial estimates	3. motion
	4. distributions	5. represent	6. setting
18	1. operator mutation	2. operators	3. important application
	4. human operator	5. operators crossover	6. mutation chance
19	1. IEEE members	2. main	3. lines represent
	4. IEEE	5. main idea	6. important

Table 3 presents the key concepts – composed of one or more words - of the AI knowledge graph, identified based on their degree centrality and the C-Rank heuristics. Table 4 presents the keyphrases - composed of one or more concepts - of each of the AI clusters, identified using the C-Rank algorithm.

We observe in Tables 3 and 4 that the key concepts and keyphrases identified correspond to relevant concepts in the Artificial Intelligence area. For example, the concept “Neural Network” in Table 3 is the one with the highest centrality, indicating the popularity of neural networks in artificial intelligence approaches. Moreover, the keyphrases identified in Table 4, despite suffering from some overlapping, are able to describe the topics that their clusters represent. Take as an example Cluster 14, which has all keyphrases related to the video processing area such as “frames sequence”, “sequence videos”, and “frames”.

In Table 3, great part of the extracted terms express fundamental concepts of the Artificial Intelligence area, such as *Neural Network*, *Computer Vision*, and *Pattern Recognition*. Concepts such as *Computing Science* and *provide* can be considered too generic to be key concepts of the “Artificial Intelligence” area. This occurred because we rank our concepts based on their degree centrality and generic concepts tend to co-occur with many concepts, and, consequently, have a high ranking in our key concepts extraction. That is why we have the $threshold_{centrality}$ metric to mitigate this issue. In our experiments we set $threshold_{centrality} = 50$ because we considered that most concepts from the 51st were relevant to the scientific field.

However, this metric proved not to be optimal, as many generic concepts appeared after that one; that is why we previously suggested an interactive model, so that the user can fine-tune these key concepts, excluding the most generic ones.

The extracted keyphrases of the AI topics applying the C-Rank algorithm in each cluster can be used to summarize the topics that they represent and assist the user in understanding the results. For example, Cluster 5 represents the image processing sub-area containing the keyphrases: "*camera view*", "*illumination variation*", and "*scale change*"; Cluster 7 represents the computer vision sub-area with the keyphrases: "*computer vision*", "*computer vision learning algorithms*", and "*computer vision task*"; Cluster 12 represents machine learning techniques based on the keyphrases: "*neural network*", "*evolutionary process*", and "*SVM classifier*"; and Cluster 14 represents video processing based on the keyphrases: "*sequence videos*", "*frames sequence*", and "*frames keyphrases*". On the other hand, the keyphrases of Cluster 0 and Cluster 19 indicate that they are formed by generic keyphrases such as "*provide*", "*information*", "*paper*", "*IEEE*", and "*lines represent*". We observed that the clusters formed by generic keyphrases are usually the ones with higher and lower modularity. In the first case, their concepts represent common characteristics and jargons in academic articles, such as "*main ideas*", "*lines represent*", and their publisher. In the second case, the clusters contain generic concepts that should have been eliminated in the preprocessing step, which can be accomplished using an interactive approach. Therefore, despite the generic clusters, these results show that this clustering was able to divide the network into different topics, which can be represented by automatically identified keyphrases.

5. Discussion

The task of understanding relationships among concepts in an area from the reading of scientific articles remains a very challenging issue. This work contributed to the design, implementation, and evaluation of the SciKGraph framework to represent and structure scientific fields based on textual documents as input. The outcome of this research can assist researchers in understanding how concepts in scientific fields are organized and correlated.

For evaluation purposes, the use of our framework enabled the construction of knowledge graphs from two distinct datasets. The conducted experimental analyses were suited to study different aspects of the generated knowledge graphs. We found that our solution is suited to obtain the clustering of knowledge graphs representing topics in the content of input documents.

We used SciKGraph to represent two datasets containing academic documents from different areas and observed the same trends in all experiments performed. This implies that the proposed structure can be used to represent different areas expecting similar results. Considering that SciKGraph depends on Babelfy to identify its concepts, we cannot ensure that it can be applicable to every scientific field. Still, we strongly believe that BabelNet should cover most of scientific knowledge, as it was constructed based on multiple data sources and is made up of about 16 million entries (Babelnet). Based on this, we suggest using SciKGraph in other academic areas to enhance their understanding by researchers. This was achieved because the proposed structure is constructed based on concepts of the academic articles, rather than only their metadata information. As our key finding, SciKGraph uses the relations among concepts to automatically identify clusters in the knowledge graph by dividing an area into its sub-areas, a process not biased by manually-defined categories.

The clustering of the proposed structure is performed using the OClustR algorithm, which automatically identifies the optimal number of clusters of the graph. In order to enhance the visualization of results, one may have to reduce the number of clusters identified. To this end, this work proposed and tested three simple agglomerative methods. *Simple Threshold* obtains the best accuracy in our experiments, but it deletes relevant information and achieves the worst modularity; *Top-Modularity* and *Best-Modularity* produced similar results, with the latter achieving better accuracy in one of the experiments, but being more computationally costly. Therefore, because of the good trade-off between results produced and computational cost, we recommend using the *Top-Modularity* agglomerative method.

The results using the *Top-Modularity* agglomerative method achieved over 0.30 of modularity concerning the clustered knowledge graphs, indicating high segmentation. This was impacted by the $threshold_{edges}$ metric, which can be increased to improve knowledge graph modularity. We found that the higher this threshold, the more information is lost during clustering preprocessing. On this basis, we recommend increasing this metric as little as possible to preserve adequate modularity of the obtained clusters.

In the framework, we found that it is possible to describe the topics (identified clusters) according to keyphrases defined based on degree centrality algorithms and heuristics executed over the clusters. However, because of the amount of noise data, we recommend executing it interactively.

In order to further explore and take advantage of the framework for literature analysis, further studies need to conceive interactive software tools to enable users to filter key concepts and keyphrases. Additional investigations on visualization methods and tools can enhance users' experience in understanding the processed documents and identified topics. Moreover, it would be beneficial to further examine the usage of other background semantic networks along with BabelNet to ensure the coverage of the proposed approach in specific domains. Also, it would be valuable to study how semantic similarity and word embedding techniques can be combined with our framework to reduce noise concepts after the preprocessing step. Still, our current solution is accurate enough to validate the structure and to give researchers a broader view of a scientific area.

6. Conclusion

The amount of scientific information produced has hugely increased throughout the years. This impacts the lives of researchers, who have to keep abreast of discoveries and relevant papers related to their areas of expertise. They have to invest a lot of time in groundwork, but during this process, a significant portion of this time is wasted on topics that are unrelated to their primary goals. This occurs because of the difficulties in finding the desired material to consult. In this article, we proposed to represent scientific fields as knowledge graphs. Our approach explored the semantics of natural language texts from input documents rather than metadata and citation information to structure knowledge. This approach considered all concepts from the studied field, clustering them into topics. The results of our experimental analyses achieved up to 84% of accuracy in identifying the areas of input documents. This finding shows that despite not being constructed to segment the knowledge graph into predefined areas, it could successfully segment the areas into topics without using annotated data in the process. The topics and keyphrases extracted from the graph are coherent with the dataset area, which indicates that they were correctly identified. To obtain optimal results, we recommend using our framework interactively, by taking the threshold values as suggestions to assist users. They can manually fine-tune the recommended thresholds and keyphrases to better represent the identified topics. The results revealed that the way our framework generates knowledge graphs can be used to represent other areas from distinct domains. In future studies, we will investigate interactive techniques to provide a simplified and representative visualization of the knowledge graph.

Authors' contribution

Mauro Dalle Lucca Tosi: conceived and designed the analysis, collected the data, contributed data or analysis tools, performed the analysis, wrote the paper. Julio Cesar dos Reis: conceived and designed the analysis, wrote the paper, supervisor.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgments

This work was financially supported by the São Paulo Research Foundation (FAPESP) (grants #2017/02325-5 and #2013/08293-7) and the Coordenação de Aperfeicoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.joi.2020.101109.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175–185.

Aria, M., & Cuccurullo, C. (2017). bibliometrix: An r-tool for comprehensive science mapping analysis. *Journal of Informetrics*, *11*(4), 959–975.

Babelnet | the largest multilingual encyclopedic and semantic network. https://babelnet.org/about (Accessed 07 May 2020).

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z., et al. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*. pp. 722–735. Springer.

Bornmann, L., & Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, *66*(11), 2215–2222.

Boudin, F. (2013). A comparison of centrality measures for graph-based keyphrase extraction. *Proceedings of the sixth international joint conference on natural language processing*, 834–838.

Cambria, E., & White, B. (2014). Jumping nlp curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, *9*(2), 48–57.

Chen, M., & Szymanski, B. K. (2015). Fuzzy overlapping community quality metrics. *Social Network Analysis and Mining*, *5*(1), 40.

Chen, C. (2006). Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information SCIENCE and Technology*, *57*(3), 359–377.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Dogrusoz, U., Giral, E., Cetintas, A., Civril, A., & Demir, E. (2009). A layout algorithm for undirected compound graphs. *Information Sciences*, *179*(7), 980–994.

Ehrlinger, L., & Wöß, W. (2016). *Towards a definition of knowledge graphs*. pp. 48. SEMANTiCS (Posters, Demos, SuCCESS)

Garfield, E. (2009). From the science of science to scientometrics visualizing the history of science with histcite software. *Journal of Informetrics*, *3*(3), 173–179.

Goldberg, Y., & Levy, O. (2014). *word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method*. arXiv:14023722

Grobid. https://github.com/kermitt2/grobid, 2008–2019. 1:dir:6a298c1b2008913d62e01e5bc967510500f80710.

Gupta, S., & Varma, V. (2017). Scientific article recommendation by using distributed representations of text and graph. In *Proceedings of the 26th international conference on world wide web companion*. pp. 1267–1268. International World Wide Web Conferences Steering Committee.

Hagberg, A., Swart, P., & S Chult, D. (2008). *Exploring network structure, dynamics, and function using networkx*. Los Alamos, NM (United States): Los Alamos National Lab.(LANL). Technical report.

Jung, S., & Segev, A. (2014). Analyzing future communities in growing citation networks. *Knowledge-Based Systems*, *69*, 34–44.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., et al. (2016). Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*. pp. 87–90. IOS Press.

Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017). Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. pp. 364–371. IEEE.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Lewis, D. M., & Alpi, K. M. (2017). Bibliometric network analysis and visualization for serials librarians: An introduction to sci2. *Serials Review*, *43*(3-4), 239–245.

Maron, M. E. (1961). Automatic indexing: An experimental inquiry. *Journal of the ACM (JACM)*, *8*(3), 404–417.

McLevey, J., & McIlroy-Young, R. (2017). Introducing metaknowledge: Software for computational research in information science, network analysis, and science of science. *Journal of Informetrics*, *11*(1), 176–197.

Miller, G. A. (1995). Wordnet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244.

Navigli, R., & Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences United States of America*, *103*(23), 8577–8582.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), 026113.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Pérez-Suárez, A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., & Medina-Pagola, J. (2013). Oclustr: A new graph-based algorithm for overlapping clustering. *Neurocomputing*, *121*, 234–247.

Rocchio, J. (1971). *Relevance feedback in information retrieval. The Smart retrieval system-experiments in automatic document processing*. pp. 313–323.

Rous, B. (2012). Major update to ACM'S computing classification system. *Communications of the ACM*, *55*(11), 12–12.

Ruiz-Rosero, J., Ramirez-Gonzalez, G., & Viveros-Delgado, J. (2019). Software survey: Scientopy, a scientometric tool for topics trend analysis in scientific publications. *Scientometrics*, *121*(2), 1165–1188.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., et al. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, *13*(11), 2498–2504.

Silva, F. N., Amancio, D. R., Bardosova, M., Costa, L. F., & Oliveira, O. N., Jr. (2016). Using network science and text analytics to produce surveys in a scientific topic. *Journal of Informetrics*, *10*(2), 487–502.

Tosi, M. D. L., & dos Reis, J. C. (2019). C-rank: A concept linking approach to unsupervised keyphrase extraction. In *Research conference on metadata and semantics research*. pp. 236–247. Springer.

Van Eck, N. J., & Waltman, L. (2011). *Text mining and visualization using vosviewer*. arXiv:11092058

Van Eck, N. J., & Waltman, L. (2014). Citnetexplorer: A new software tool for analyzing and visualizing citation networks. *Journal of Informetrics*, *8*(4), 802–823.

Xia, F., Liu, H., Lee, I., & Cao, L. (2016). Scientific article recommendation: Exploiting common author relations and historical preferences. *IEEE Transactions on Big Data*, *2*(2), 101–112.

Xie, J., Szymanski, B. K., & Liu, X. (2011). Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 IEEE 11th international conference on data mining workshops*. pp. 344–349. IEEE.

Yau, C. K., Porter, A., Newman, N., & Suominen, A. (2014). Clustering scientific documents with topic modeling. *Scientometrics*, *100*(3), 767–786.