Universidade Estadual de Campinas
Instituto de Computação

Mauro Dalle Lucca Tosi

Constructing Knowledge Graphs from Textual Documents for Scientific Literature Analysis

Construindo Grafos de Conhecimento utilizando Documentos Textuais para Análise de Literatura Científica

CAMPINAS
2020

Mauro Dalle Lucca Tosi

Constructing Knowledge Graphs from Textual Documents for Scientific Literature Analysis

Construindo Grafos de Conhecimento utilizando Documentos Textuais para Análise de Literatura Científica

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Julio Cesar dos Reis

Este exemplar corresponde à versão final da Dissertação defendida por Mauro Dalle Lucca Tosi e orientada pelo Prof. Dr. Julio Cesar dos Reis.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

T639c

Tosi, Mauro Dalle Lucca, 1995-
Constructing knowledge graphs from textual documents for scientific literature analysis / Mauro Dalle Lucca Tosi. – Campinas, SP : [s.n.], 2019.

Orientador: Julio Cesar dos Reis.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Grafo (Sistema de computador). 2. Redes complexas. 3. Centralidade (Teoria dos grafos). 4. Computação semântica. 5. Conhecimento científico. I. Reis, Julio Cesar dos, 1987-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Construindo grafos de conhecimento utilizando documentos textuais para análise de literatura científica
Palavras-chave em inglês:
Graph (Computer system)
Complex networks
Centrality (Graph Theory)
Semantic computing
Scientific knowledge
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Julio Cesar dos Reis [Orientador]
Ricardo da Silva Torres
Guilherme Alberto Wachs Lopes
Data de defesa: 09-03-2019
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-0218-2413
- Currículo Lattes do autor: http://lattes.cnpq.br/4812437948087902

Universidade Estadual de Campinas
Instituto de Computação

Mauro Dalle Lucca Tosi

Constructing Knowledge Graphs from Textual Documents for Scientific Literature Analysis

Construindo Grafos de Conhecimento utilizando Documentos Textuais para Análise de Literatura Científica

Banca Examinadora:

- Prof. Dr. Julio Cesar dos Reis
  Instituto de Computação, Universidade Estadual de Campinas (UNICAMP)
- Prof. Dr. Ricardo da Silva Torres
  Norwegian University of Science and Technology (NTNU)
- Prof. Dr. Guilherme Alberto Wachs Lopes
  Fundação Educacional Inaciana Padre Sabóia de Medeiros (FEI)

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 09 de março de 2020

[...] a mind needs books as a sword needs a whetstone, if it is to keep its edge — Tyrion Lannister
(A Game of Thrones, George R. R. Martin)

Acknowledgements

First, I would like to thanks Renata, my wife, for all the support, love, and companionship that she promoted to me, which motivated me to conclude this work.

Then, I thanks Julio, my supervisor, for his patience, comprehension, and competency that enabled me to overcome this challenge.

Finally, I acknowledge the entities that financially supported this work, which made this thesis possible.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

This work was financially supported in partial by the São Paulo Research Foundation (FAPESP) (grants #2017/02325-5 and #2013/08293-7)[1].

[1]The opinions expressed in here are not necessarily shared by the financial support agency.

Resumo

O número de publicações científicas que pesquisadores tem que ler vem aumento nos últimos anos. Consequentemente, dentre várias opções, é difícil para eles identificarem documentos relevantes relacionados aos seus estudos. Ademais, para entender como um campo científico é organizado, e para estudar o seu estado da arte, pesquisadores geralmente se baseiam em artigos de revisão de uma área. Estes artigos podem estar indisponíveis ou desatualizados dependendo do tema estudado. Usualmente, pesquisadores têm que realizar esta árdua tarefa de pesquisa fundamental manualmente.

Pesquisas recentes vêm desenvolvendo mecanismos para auxiliar outros pesquisadores a entender como campos científicos são estruturados. Entretanto, estes mecanismos são focados exclusivamente em recomendar artigos relevantes para os pesquisadores ou os auxiliar em entender como um ramo da ciência é organizado ao nível de publicação. Desta forma, estes métodos limitam o entendimento sobre o ramo estudado, não permitindo que interessados estudem os conceitos e relações abstratas que compõe um ramo da ciência e as suas subáreas.

Esta dissertação de mestrado propõe um framework para estruturar, analisar, e rastrear a evolução de um campo científico no nível dos seus conceitos. Ela primeiramente estrutura o campo científico como um grafo-de-conhecimento utilizando os seus conceitos como vértices. A seguir, ela automaticamente identifica as principais subáreas do campo estudado, extrai as suas frases-chave, e estuda as suas relações. Nosso framework representa o campo científico em diferentes períodos do tempo. Esta dissertação compara estas representações, e identifica como as subáreas do campo estudado evoluiram no decorrer dos anos.

Avaliamos cada etapa do nosso framework representando e analisando dados científicos provenientes de diferentes áreas de conhecimento em casos de uso. Nossas descobertas indicam o sucesso em detectar resultados similares em diferentes casos de uso, indicando que nossa abordagem é aplicável à diferentes domínios da ciência. Esta pesquisa também contribui com uma aplicação com interface web para auxiliar pesquisadores a utilizarem nosso framework de forma gráfica. Ao utilizar nossa aplicação, pesquisadores podem ter uma análise geral de como um campo científico é estruturado e como ele evolui.

Abstract

The amount of publications a researcher must absorb has been increasing over the last years. Consequently, among so many options, it is hard for them to identify interesting documents to read related to their studies. Researchers usually search for review articles to understand how a scientific field is organized and to study its state of the art. This option can be unavailable or outdated depending on the studied area. Usually, they have to do such laborious task of background research manually.

Recent researches have developed mechanisms to assist researchers in understanding the structure of scientific fields. However, those mechanisms focus on recommending relevant articles to researchers or supporting them in understanding how a scientific field is organized considering documents that belong to it. These methods limit the field understanding, not allowing researchers to study the underlying concepts and relations that compose a scientific field and its sub-areas.

This Ms.c. thesis proposes a framework to structure, analyze, and track the evolution of a scientific field at a concept level. Given a set of textual documents as research papers, it first structures a scientific field as a knowledge graph using its detected concepts as vertices. Then, it automatically identifies the field's main sub-areas, extracts their keyphrases, and studies their relations. Our framework enables to represent the scientific field in distinct time-periods. It allows to compare its representations and identify how the field's areas changed over time.

We evaluate each step of our framework representing and analyzing scientific data from distinct fields of knowledge in case studies. Our findings indicate the success in detecting the sub-areas based on the generated graph from natural language documents. We observe similar outcomes in the different case studies, indicating that our approach is applicable to distinct domains. This research also contributes with a web-based software tool that allows researchers to use the proposed framework graphically. By using our application, researchers can have an overview analysis of how a scientific field is structured and how it evolved.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Context and Motivation

Nowadays, the scientific literature is mostly organized based on the metadata of published articles. Conventional factors used to segment and structure literature are articles' authors, citations, publication years, and journals or conferences, in which they were published. Moreover, manually defined data, usually annotated by the authors, is also used. For example, the areas and sub-areas to which an academic article belongs and keywords.

When searching for content on a specific topic, researchers usually use those supra-cited-metrics to identify relevant articles for their studies. If they have background knowledge in the studied area, they can follow specific authors, journals, and conferences which they consider relevant. On the other hand, those that are new to an area tend to perform broader searchers focused on the keywords.

Other than just studying the current state of the art of a specific topic, it is very relevant that researchers may analyze its evolution over the years. This type of investigation allows researchers to understand trends in a scientific field and predict how it may advance in the future [24]. Researchers usually publish those findings in article reviews. Therefore, to understand the evolution of an area and its current state of the art, a researcher would usually searches for updated article reviews or, if they are not available, do their own background research on the investigated area.

Over the last years, the number of documents published has been rapidly increasing. This phenomenon triggered researches in the Big Scholarly Data area, which studies the fast growth of scholarly data [63]. This growth makes it more laborious and time-consuming for researchers to identify content related to their studies among all newly published documents.

In this context, literature has presented mechanisms to assist researchers in understanding and finding academic content related to their studies, mitigating their amount of work and time invested in these tasks. Those mechanisms usually focus on recommending academic articles to researchers [20][62], structuring and analyzing an academic area [55], or tracking its evolution [24].

Meanwhile, researchers have been investigating novel approaches to structure and represent knowledge. Based on the hypothesis that humans organize their knowledge using abstract concepts relationships [56][15][11], computational approaches that structure

knowledge based on these abstract concepts have been developed such as Ontologies [16], Formal Concept Analysis [17], and Knowledge Graphs [14], which we refer here as concept-level representations. Those representations allow the comprehension of concepts and their relations, letting users analyze not just how an area of knowledge interacts with other areas, but understand and study individual concepts.

At the current state, mostly mechanisms used to assist researchers in understanding and represent scientific knowledge are based on manually defined data or metadata information extracted from academic articles that belong to the studied scientific field. In these approaches, which we refer here as document-level approaches, researchers cannot apply them on large-scale and study their concepts because manually defined data is not scalable, and metadata-based mechanisms structure scientific knowledge considering only the data about the documents published, not their content. For the best of our knowledge, only Semantic Scholar [53] and Meta [34] are using concept-level approaches to recommend articles to researchers. Semantic Scholar identifies the concept searched by the user, recommends academic documents related to the searched concept, outputs its Wikipedia description, and shows related topics that may interest the user. Meta, on the other hand, based on concepts inserted by the user, generates a feed of recommended articles, which it updates considering new publications related to the inputted concepts. However, both approaches focus mostly on the recommendation of articles. Therefore, they do not further explain to the researcher how scientific areas are structured and how their concepts are correlated.

Taking this into account, we understand that further investigations focused on new approaches to structure and extract information from scientific data must be developed. Different from the current ones, it is necessary to develop scalable mechanisms that consider the fully content of documents to represent a scientific field. In addition, instead of structuring science based on published documents, we assume that we should structure it based on the concepts that compose scientific areas. Therefore, to better understand an area of knowledge, researchers could investigate how its concepts are organized and correlated among themselves.

1.2 Research Problem and Challenges

To the best of our knowledge, it is still an open research challenge how to assist researchers in extracting and understanding knowledge at a concept level from a scientific field based on textual information from scientific papers. This M.Sc. thesis addresses the representation and analysis of a scientific field at a concept level as a research problem. We address this challenge by studying the following four sub-problems.

1. **Automatic representation of a scientific field at a concept level**

   Scientific literature is organized based on published articles. Therefore, it is necessary a collection of academic documents that belong to the scientific field as input to structure a scientific field. Since a high number of articles are essential to encompass all facets of a scientific field, it is necessary to deal with how extracting a high number of concepts from the documents without huge human efforts and

pre-annotated datasets. We highlight that the representation of a scientific field at a concept level is not straightforward because of the difficulties in identifying the most relevant concepts and their relations among all input academic articles.

2. **Automatic identification of the main sub-areas of a scientific field using a concept-level representation**

   The automatic identification of topics in a scientific field is usually based on metadata from academic articles that belong to the analyzed area. Usually, it does not consider the concepts that compose an academic area, limiting the user's understanding of the scientific field topology. Approaches based on the semantics of textual documents could mitigate this issue, identifying the sub-areas of the scientific field and to which of them each represented concept belongs. Though, they are more complex, having to manage abstract concepts in different contexts [8]. This occurs because these approaches would have to deal with issues as structure scientific knowledge based on tens of thousands of concepts that compose a scientific field; identify the correlations among those concepts considering their application in different contexts; and distinguish when the same concepts belong to multiple sub-areas at the same time.

   Furthermore, as sub-areas may be related and have concepts in common among themselves, it is necessary to acknowledge that they can overlap each other. In order to best identify the sub-areas of a scientific field, the method used for this task must consider that concepts can belong to multiple sub-areas simultaneously. This fact hinders the task, as the chances of overlap increase the possibilities of the identification of the sub-areas, reducing the method correctness. In this sense, the automatic identification of sub-areas of a scientific field at a concept level is not trivial. An adequate method to solve this problem must manage concepts in different contexts and their overlapping in distinct sub-areas.

3. **Automatic extraction of keyphrases from scientific fields and its sub-areas**

   The identification of keyphrases - relevant phrases composed of one or more concepts - is usually performed in single documents. Most studies and datasets focused on keyphrases, related to scientific data, deal with identifying them from academic articles. To the best of our knowledge, its automatic unsupervised identification from broader perspectives has not been studied yet. This problem consists of identifying the most relevant keyphrases from a scientific field, considering as input a collection of documents used to represent this field. Keyphrases are phrases that describe the content that they are representing, being composed of one or more words. In this context, they would assist users in understanding what a sub-area of a scientific field represents.

   The key challenging part of this problem is its evaluation. There are no available datasets with annotated keyphrases of scientific fields to use as an evaluation reference. This lack of datasets occurs because of the difficulties to identify keyphrases in an extensive compendium of documents. Moreover, this identification, usually

performed manually, would demand experts in the scientific field, which hinders the process.

4. **Software tool support for tracking the evolution of a scientific field at a concept level**

   The tracking of a scientific field evolution is the process of identifying the changes that occurred in a field and its sub-areas over time. Researchers can use this information to understand how a scientific field reached its current state of the art and to study how it will behave in the future, based on prior observations. At a concept level, researchers could not just identify changes in the structure of a scientific field, but also in the concepts and relations that compose it. For instance, this tracking would enable the identification of new concepts introduced to a sub-area, or that lost relevance in it.

   The problem of tracking the evolution at a concept level is directed connected with the problems of representing and segmenting a scientific field in its sub-areas. This correlation occurs because it is necessary to compare representations of a scientific field in distinct time-periods to track its evolution. The same issue occurs to identify changes appearing in specific sub-areas. In the second case, before comparing the sub-area representations, it is necessary to determine when two groups of concepts from distinct time-periods represent the same sub-area. This problem is not straightforward, considering that both the structure and the concepts that shape a group can be modified, and they can still represent the same sub-area. Thus, when comparing two groups, it is not trivial to compute if some parts of the structure changed, were replaced, suffered from noise data, or the groups are not correlated.

1.3 Research Goal and Approach

In this MS.c. thesis, our goal is to develop a framework to structure, represent, and analyze a scientific field over-time at a concept level. Our solution is based on a collection of academic documents as input. In addition, we aim to develop a software tool to facilitate its usage and better assist researchers from distinct areas in using the proposed framework. More specifically, we define the following four key objectives:

1. To structure and represent scientific textual data at a concept level.
2. To automatically extract keyphrases from the previously mentioned structure without supervision.
3. To automatically identify sub-areas of a scientific field at a concept level.
4. To track the evolution of a scientific field and its sub-areas.

To accomplish our goal, we structure our supra cited objectives as illustrated in Figure 1.1. Considering the challenges in evaluating if we properly structured a scientific field and extracted its keyphrases, our first two objectives focus on proposing and appraising

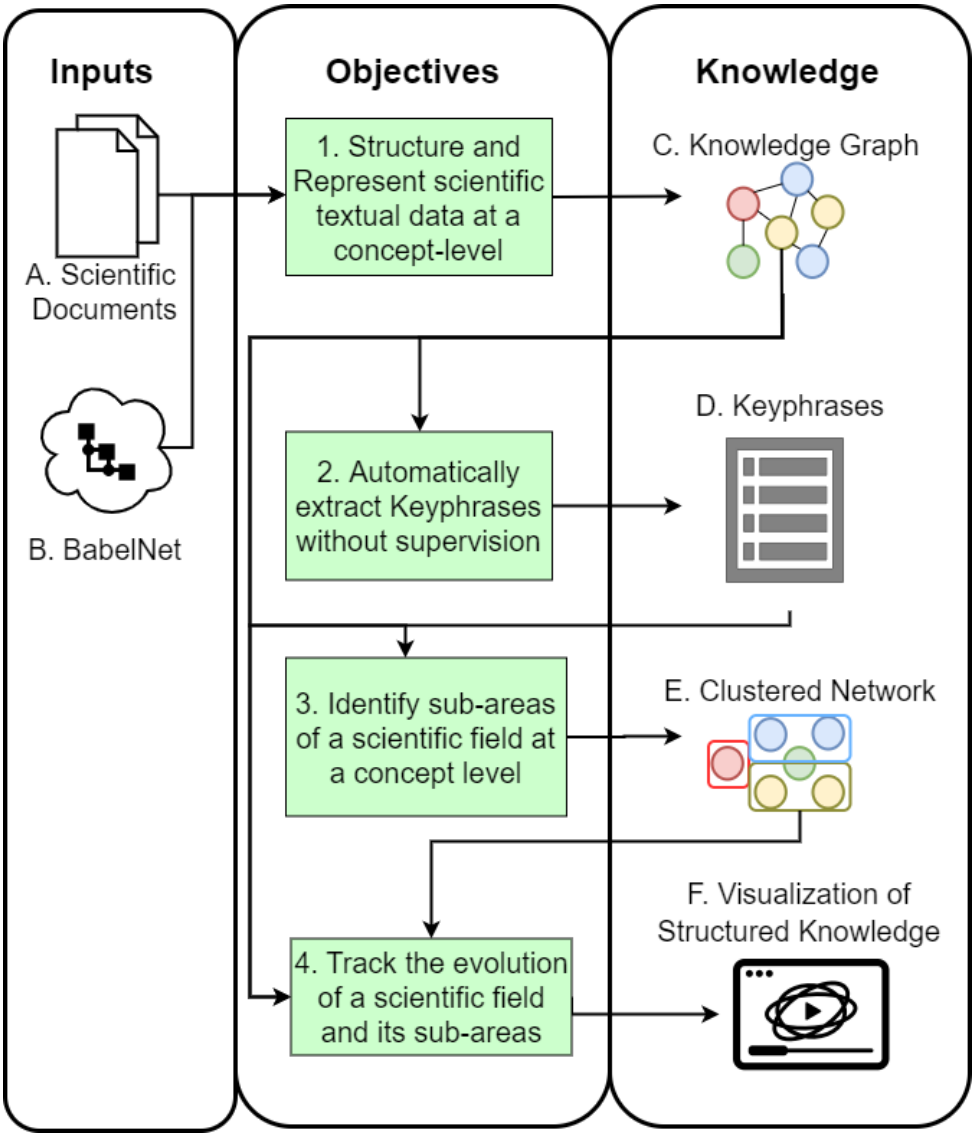

Figure 1.1: Methodology to fulfill key objectives to develop a framework to structure, represent, and analyze a scientific field over-time at a concept level.

approaches that deal with shorter scientific textual data. Therefore, instead of analyzing whole scientific fields, we first structure the text of single scientific articles, or their abstract, using our proposed knowledge graph (cf. Section 2.3) - a knowledge base system that integrates information from distinct sources and applies a reasoner to identify new information [14] - which has article concepts as vertices and their correlation in the text as edges. Then, we developed a keyphrase extraction technique to identify the most relevant keyphrases from it. This way, we evaluate and compare the results of our keyphrase extraction with other similar methods that extract keyphrases from academic documents, obtaining state-of-the-art results in this task (cf. Section 2.4.3).

Accordingly, we consider that as we could identify appropriate keyphrases from our representation, it was structured correctly. In this investigation, we assume that when using the same methodology to structure a single document or a compendium of documents, we can expect similar results in its representation and its keyphrase extraction.

Therefore, we propose to use the previously knowledge graph structure to represent a whole scientific field, using as input a collection of articles that belongs to the analyzed field (cf. Section 3.3.1).

Considering that we have a scientific field structured at a concept level, our objective is to identify its sub-areas. In our approach, we achieve this identification by investigating the clustering of knowledge graphs. Our proposal applies a clusterization algorithm that identifies overlapping groups (sub-graphs) that contain concepts (vertices) that are more connected among themselves than with concepts (vertices) from other groups (sub-graphs). Here, we follow the idea that words belonging to the same sub-area are usually used together in a text, and, therefore, tend to be clustered together inside co-occurrence graphs (inspired by [60]). However, instead of analyzing words, we hypothesize that concepts from the same sub-area are more connected in our knowledge graph; hence, they will be clustered together in the clusterization process (cf. Section 3.3.2).

The clusterization process completes our third objective. In this context, we constructed a knowledge graph representation of a dataset used for document classification, which has articles annotated with their respective areas. Then, we clustered this graph and detected in which of the identified areas each annotated article belonged. We evaluated the accuracy of this document classification approach. In particular, we investigate that our clusterization method, even not being developed to classify documents in predefined areas, could obtain satisfactory results in such task. This result demonstrates the coherency of our approach to represent a scientific field and identify its sub-areas (cf. Section 3.4).

Furthermore, we fulfill our fourth objective by proposing a model to track the evolution of a scientific field (cf. Section 4.3). It identifies the changes that occurred in the scientific field sub-areas comparing their representations on different time-periods. To identify those changes, we first need to determine if two groups of concepts - identified in the previously discussed clusterization process - represent the same sub-area. We determined a metric that calculates the similarity between groups of concepts to compute if they are correspondent. Then, after determining groups that represent the same sub-area, our approach compares them and returns the concepts and relations that changed between both representations.

At last, in this Ms.c. thesis, we constructed an application software tool to assist researchers in understanding, analyzing and visualizing a scientific field that they are studying (cf. Section 4.4). Our software graphically allows researchers to accomplishing the following:

- Insert the collection of documents used to represent their scientific field;
- Structure the scientific field as a knowledge graph;
- Identify its main sub-areas (clusters of concepts);
- Analyze their structure and keyphrases detected;
- Study and visualize the scientific field evolution.

1.4 Contributions

We describe the main contributions produced during the development of this Ms.c. thesis as follows:

1. C-rank (cf. Chapter 2), an automatic unsupervised keyphrase extraction technique. It extracts keyphrases from academic texts without demanding external inputs from users other than the texts from which it identifies the keyphrases. Considering the case that a researcher wants to extract the keyphrases from an academic text without inserting other data, C-Rank achieves state-of-the-art results, considering the tests performed in the SemEval 2010 dataset [27].

2. SciKGraph (cf. Chapter 3), a framework that structures and analyzes scientific fields at a concept level. Receiving as input a collection of academic articles used to represent a scientific field, SciKGraph represents this field as a knowledge graph with the use of background knowledge; the framework identifies its sub-areas, their relations and concepts. The solution enables to analyze the generated structure by identifying the concepts that belong to multiple sub-areas, how they are correlated, quantify the segmentation of its sub-areas, and extracts their keyphrases - relevant phrases composed of one or more concepts - and key-concepts - relevant concepts composed of one or more words.

3. A method to track the evolution of a scientific field at a concept level (cf. Section 4.3). Our method automatically identifies how sub-areas of a scientific field changed over time. Researchers can use this information to obtain knowledge related to the evolution of sub-areas and their concepts. Examples of those findings include: 1) determining if two sub-areas merged into a single one; 2) identifying concepts that are hubs among multiple sub-areas; 3) observing when a concept was introduced to a sub-area; 4) and identifying which concepts are central to a sub-area and do not vary over time.

4. An online interactive software tool that encompasses the SciKGraph framework and the method to track the evolution of a scientific field (cf. Chapter 4.4). Our tool provides a web interface and can be used by researchers that do not have programming skills to structure and analyze a scientific field in a specific period overtime. This tool uses Cytoscape [54] - a software for the visualization of complex networks - to visually represent the structures in which the user is working. Our proposed visualization solution is interactive and enables users to graphically manipulate the vertices and edges of the graph, zooming into specific concepts, and opening its correspondents in Babelnet [42], a multilingual semantic network.

5. Another technological contribution, we make available all libraries, algorithms, and case studies produced during the development of the whole investigation. They are available online[1] and enable researchers to automate our proposed methods and techniques into their softwares.

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

1.5 Organization

This MS.c. thesis is organized as a collection of three articles published or under review for publication. Each one of these articles corresponds to a chapter of this thesis, as described below.

Chapter 2 corresponds to the article "*C-Rank: A Concept Linking Approach to Unsupervised Keyphrase Extraction*" [57], which was published and presented in the 13th International Conference on Metadata and Semantics Research (MTSR 2019). It introduces C-Rank, an unsupervised technique to extract keyphrases from academic articles and its intermediate structure, used to represent scientific textual data at a concept level.

Chapter 3 corresponds to the article "*SciKGraph: A Knowledge Graph Approach to Structure a Scientific Field*" [58], which was submitted to an international journal and is under review. It presents SciKGraph, a framework that uses knowledge graphs to structure and analyze scientific fields at a concept level.

Chapter 4 corresponds to the article "*Understanding the evolution of a scientific field by clustering and visualizing knowledge graphs*" [59], which was submitted to an international journal and is under review. This work proposes a method to track the evolution of the sub-areas of a scientific field by comparing their representations using SciKGraph in distinct time-periods. This also presents our developed software tool to help end-users in this task.

Chapter 5 presents a in-depth discussion focused on each of the research objectives of this MS.c. thesis. Then, we list the limitations observed in our framework. Additionally, we present a synthesis of our findings and recommendations.

At last, Chapter 6 concludes this MS.c. thesis. It elaborates on how our research can be extended into future investigations. Then, it highlights the overall contributions of this thesis. Furthermore, it shows how we disseminated our research and its findings. Finally, it presents our final considerations.

Chapter 2

C-Rank: A Concept Linking Approach to Unsupervised Keyphrase Extraction

2.1 Introduction

Keyphrases are expressions intended to represent the content of a document and highlight its main topics. They may be single or multi-termed and may be provided by the author, which is uncommon in most of the non-scientific texts. Keyphrases are used by potential readers to decide whether or not the topics approached in the document are relevant to them. Furthermore, they may be used to recommend articles to readers, analyze research trends over time, among other NLP tasks [3]. However, the automatic keyphrase extraction is a challenging task as it varies from domains, suffers from the lack of context, and its result keyphrases may be formed by multiple words [3]. Therefore, despite the improvements achieved in the last years, it still is an active research topic that deserves further studies.

The keyphrase extraction task can be performed based on different approaches. Hasan and Ng [22] segmented the keyphrase extraction task in Supervised, that demands an annotated training set; and Unsupervised, that does not depend on annotated data, which is the line followed in this article.

The unsupervised methods can be developed to extract the keyphrases of a document based on different inputs other than the document text itself. The background data varies according to the method and can consider web-pages, specific-domain documents and general scientific texts [28]. Although the best results have been achieved by most of the methods using background data, it may demand information, training time or both, that the user does not necessarily possess. Therefore, approaches that do not require training nor other data to be inputted by the user should be investigated.

A predefined-domain-independent knowledge resource could improve the extraction results without requesting further data nor training from users. Babelnet[1] [42] is a wide-coverage multilingual semantic network automatically constructed that has about 16 million entries, which are called synsets. Each synset represents a given concept or a named entity and contains all its synonyms and translations in different languages. Despite the

[1] `https://babelnet.org/`

amount of relevant information contained in Babelnet, its usage would be limited without the Babelfy [39], which is a graph-based approach to simultaneously perform Entity Linking (EL) and Word Sense Disambiguation (WSD) on Babelnet.

In this article, we propose C-Rank as a novel approach to automatic perform unsupervised keyphrase extractions from free-text documents. For the best of our knowledge, it is the first method to explore concept linking to improve results in this task. C-Rank does not demand training nor other data provided by the user as it performs its linkages through Babelfy using as resources the BabelNet [42], Wikipedia[2] and WordNet [37] knowledge. C-Rank parses the inputted document text, runs Babelfy and constructs a co-occurrence graph with the annotated concepts as vertices. Next, it weights the vertices using their centrality in the graph, selects the top-ranked as candidates and modifies them using heuristic factors. Finally, C-Rank identifies vertices that belong to the same keyphrase and merge them, re-rank all the candidates and outputs the result. We extensively evaluate our approach with distinct gold standard datasets and demonstrate the effectiveness and benefits in our defined solution.

This article is organised as follows: Section 2.2 presents keyphrase extraction related works. Afterwards, Section 2.3 introduces C-Rank, our model to automatically extract keyphrases from documents. Section 2.4 reports on the used benchmark datasets in addition to the achieved results. Whereas Section 2.5 discusses our findings and compares C-Rank with existing methods, Section 2.6 concludes the article exhibiting the final considerations.

2.2 Related Work

This section presents unsupervised keyphrase extraction techniques and compares their approaches to obtain the phrases that best describe the content of a textual document. A survey conducted by Hasan and Ng [22] segmented unsupervised methods in four categories “Graph-based Ranking", that considers the co-occurrence of the phrases in a text as graph edges, in which its vertices represent the keyphrases, that are ranked based on the graph structure; “Topic-Based Clustering", which constructs a graph with the document topics as vertices and its relations as edges, then clusters it to identify the main topics discussed in the analyzed document;“Simultaneous Learning”, considering that keyphrase extraction and text summarization tasks can benefit from each other and be performed simultaneously, combining “Graph-based Ranking" with other summarization techniques to improve results; and “Language Modeling", that uses a background textual set to rank the relevance of a phrase in the analysed document, which is then compared with the same metric gathered in the background set.

Despite achieving some of the best results, “Language Modeling" approaches require external data to be inputted by the user. Therefore, they will not be covered in this paper. In addition to the four categories, Hasan and Ng also observed that many techniques merge their approaches with heuristics to push forward their results. Table 2.1 presents some of the best-unsupervised keyphrase extraction techniques that do not demand a background

[2] http://www.wikipedia.org

textual set to be provided by the user.

Table 2.1: Unsupervised Keyphrase Extraction techniques.

Models	Graph-based	Topic-based Clustering	Heuristics	Year
Text-Rank [35]	X	-	X	2004
BUAP [45]	X	-	X	2010
Topic-Rank [7]		X	X	2013
C-Rank	**X**	**-**	**X**	2019

Mihalcea and Tarau [35] presented the Text-Rank algorithm as a way to represent a text as a graph. First, they tokenize their text and annotate it with part-of-speech tags. Second, Text-Rank creates a syntactic filter and uses the tokens that pass by it as graph vertices, that are connected by undirected and unweighted edges representing their co-occurrence in the text. Third, the technique ranks the graph vertices with a variation of Google's PageRank algorithm [47] and selects the best-ranked as the document keywords. Finally, the algorithm identifies sequences of those tokens in the text and treats them as multi-word keywords, recognized as part of the final result along with the other candidates that are represented by a single token.

On the other hand, instead of constructing a graph with individual words as vertices, Ortiz *et al.* [45] developed *BUAP* that identifies the most frequent sequences of words in a document as vertices of a graph and weights them using the PageRank algorithm. Then, BUAP outputs 15 keyphrases formed by the top-3 multi-term candidates, the top-ranked single-words, and up to 3 of their expanded-forms, if there are acronyms among them.

Bougouin, Boudin and Daille [7] developed the *TopicRank* algorithm. First, it tokenizes, part-of-speech tags the text and clusters it into topics, weighted with the same ranking algorithm used in Text-Rank [35]. In the end, *TopicRank* outputs the most common keyphrases of the principal topics as a result.

The proposed approach in this paper, C-Rank, different from other state-of-the-art unsupervised keyphrase extraction approaches, does not request further data to be provided by the user. It relies on predefined background knowledge to leverage the meaning of terms in the extraction process. Instead of gather knowledge from statistical techniques, which demand a compendium that encompasses domain knowledge, C-Rank extracts information from a wide-coverage semantic-network.

2.3 C-Rank

C-Rank is an unsupervised algorithm that automatically extracts keyphrases from single documents without the support of a background textual collection. In this sense, the user needs to insert only the text from which the system should extract the keyphrases. It combines the knowledge of the document itself and contained inside BabelNet [42], collected through Babelfy [39]. C-Rank works in three stages illustrated in Figure 2.1, and detailed in the following subsections. The first stage pre-processes the input document and annotate it with BabelNet concepts. The second stage takes those concepts as vertices and generates a co-occurrence graph, which is ranked and trimmed based on heuristics

and its centrality, producing candidate keyphrases. The third and final stage identifies candidates that belong to the same phrase, which are merged and re-ranked, generating the final keyphrases list as output.

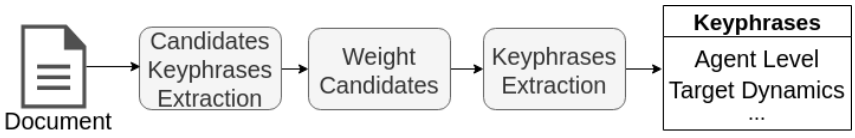

Figure 2.1: C-Rank stages.

2.3.1 Extraction of Candidate Keyphrases

The first stage receives as input a textual document that is initially parsed to have its concepts linked with Babelnet, named here concept linking, resulting in a set of paragraphs annotated with babel synsets as illustrated in Figure 2.2.

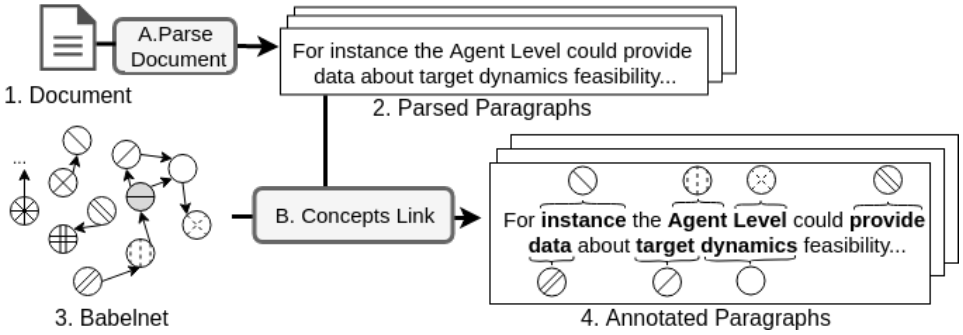

Figure 2.2: C-Rank First Stage: Extraction of Candidate Keyphrases.

Babelfy[3] is the adopted approach to link the document concepts with Babelnet, semantic annotating them. Despite receiving whole texts to process and annotate, some constraints occurred during the use of the Babelfy, as a maximum length of the input text and the service inability to process some special characters.

In order to overcome these limitations, we parsed the input document - process A, Figure 2.2 - segmenting it in sets of paragraphs with at most 5000 words, and removing all non-letter characters, except for “!", “.", “?", “-" and “ ", which are important as they segment sentences and words.

Afterward, the process B - Figure 2.2 - links the parsed paragraphs using Babelfy, which identifies the correspondences between concepts and babel synsets. It can also determine multi-word concepts and its sub-concepts. For example, in “semantic network" the following synsets are linked “semantic", “network", and “semantic network". In our approach, we only use the multi-word concept annotation because the sub-concepts would always appear more in the document and they would be positively biased in C-Rank next stages.

[3] `http://babelfy.org/`

2.3.2 Weight Candidates

The second stage receives the annotated paragraphs as input, generating a weighted directed co-occurrence graph based on it. This is ranked and trimmed with heuristics to output a graph of candidate keyphrases (*cf.* Figure 2.3).

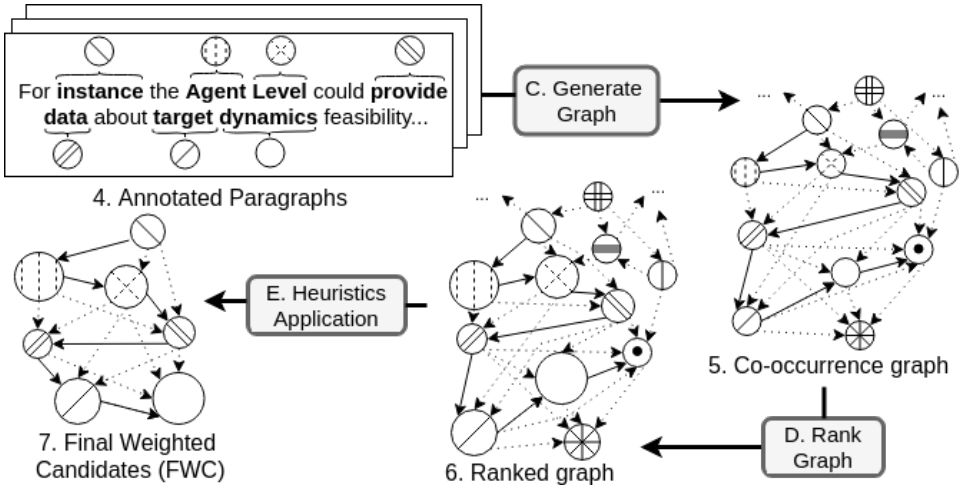

Figure 2.3: C-Rank Second Stage: Weight Candidates.

In order to construct the graph, process C (in Figure 2.3) uses the paragraphs linked concepts as vertices and their co-occurrence to generate the direct edges, that connects directly subsequent vertices represented as solid arrows in Figure 2.3; and indirect ones, linking concepts within a predefined window width, explored in Section 2.4.2, which are represented by dotted arrows.

The graph has their vertices weighted based on the number of times their concepts appear inside the document. This also occurs with the edges, that are weighted based on the distances between the concepts which its vertices represent (*cf.* Equation 2.1), in which $weightEdge_{i,j}$ is the weight of the edge that connects vertice i with vertice j; $In(j)$ refers to the set of edges that arrive at j; $window$ is the predefined window width; and $distance(i, j)$ stands for the co-occurrence distance in text between the concepts that the vertices i and j represent.

$$weightEdge_{i,j} = \sum_{i \in In(j)} 1 - log_{window} distance(i, j) \tag{2.1}$$

Process D (in Figure 2.3) ranks the vertices of the co-occurrence graph to obtain the candidate keyphrases. It uses the centrality degree value normalized by the maximum possible degree of a node. Although being a simple measure, the degree centrality achieves higher results in the identification of keyphrase on graph-based approaches, compared to other traditional ranking techniques [4].

C-Rank second stage also applies four heuristics into the graph, which were studied on a training set and are analyzed in Section 2.4.2. However, one must previously determine how to label each concept of the graph before applying the heuristics, because the same idea can be expressed divergently. As an example, “Artificial Intelligence" and “AI", both represent the same concept, despite being written differently. We label each concept

based on its first occurrence in the document because we understand that it may cover the concept extended form instead of its initials, considering that usually, in a text, a concept is introduced before its abbreviation.

The first heuristic identifies the Part-of-speech (POS) of each candidate label and discards those that have any word different from a noun, a verb or an adjective, which are the most common keyphrase POS tags. The second one cuts the 87% lower-ranked candidates (LRC) if the analyzed document is long - has more than 1000 words - in order to reduce noise. The third heuristic re-ranks the candidates favoring those formed by multiple words, as they are more likely to be chosen to become keyphrases; it uses $c_w = c_w^{\frac{1}{len(c)}}$, in which c_w represents the candidate weight and $len(c)$ its number of words. The fourth and final heuristic discards all candidates that first appeared after a *CutOff* threshold of the text, defined to 18% for long documents, as keyphrases usually are introduced at the beginning of a text.

2.3.3 Keyphrase Extraction

C-Rank third stage (Figure 2.4) receives the Final Weighted Candidates graph (FWC), identifies the concepts that belong together in the same keyphrase and outputs a re-ranked list of the input document keyphrases.

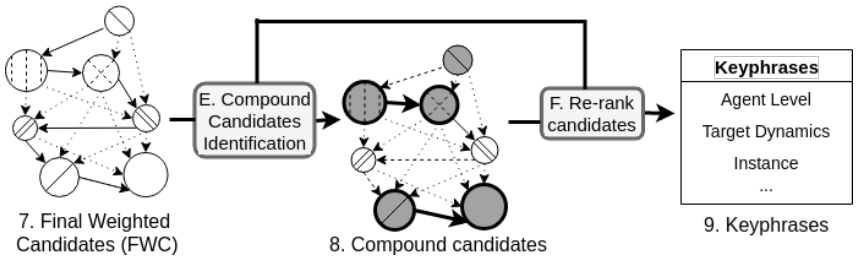

Figure 2.4: C-Rank Third Stage: Keyphrase Extraction.

A “Compound Candidates" is the given definition of the candidates that belong to the same keyphrase, which are formed by the union of two different concepts. As in Figure 2.4, the vertices with dotted patterns, which are labeled as “agent" and “level", despite being subsequently in the text, representing a single thought, were linked separately in stage one, Figure 2.2. The compound candidates identification mitigates this issue and merges these concepts together, allowing them to be multi-word concepts.

The compound candidates identification considers the vertices relations in the graph to determine whether two terms represent a single concept and, therefore, belong together in the same keyphrase. To conclude that two candidates are compound, their subsequent co-occurrence must occur multiple times, which will vary depending on the text length. Therefore, compound candidates must be linked through a direct edge weighting at least $2 + (totalWords_d/1000)$, being $totalWords_d$ the number of words in the document d. This minimum weight ensures that the compound candidates appear at least twice in short texts and require a higher frequency in larger ones.

Next, the third stage weights the compound candidates to have a comparison metric to re-rank them based on the other candidate keyphrases. Process F in Figure 2.4, calcu-

lates the normalized edges weight that connects the compound candidates (*cf.* Equation 2.2), in which $NE_{i,j}$ refers to the normalized edge that links vertice i with vertice j; $w(Out(i))$ is the sum of all edges weights outgoing vertice i; and $w(In(j))$ is the sum of all edges weights incoming vertice j. Then, Process F calculates the ranking weight of the compound candidates as expressed by Equation 2.3. It has $CC_{i,j}$ as the ranking value of the compound candidate formed by the vertices i and j, and v_i, v_j are the weight of the vertices i and j from the FWC. At last step, Process F normalizes the compound candidates by their sum as presented by Equation 2.4, in which $NCC_{i,j}$ is the normalized $CC_{i,j}$; and outputs a sorted list of the input document keyphrases $Keyphrases_d$, generated by the union of the Final Weighted Candidates and the top-ranked Compound Candidates. However, the NCC values difference decrease because of the normalization and after its union with the FWC they tend to cluster, standing out over the rest of the data. To overcome this issue, we join only 6 top-ranked Compound Candidates with the Final Weighted Candidates, a value defined from observations and tests over the keyphrases data-sets, thus $Keyphrases_d = FWC \cup NCC_{1:6}$.

$$NE_{i,j} = \frac{indirectEdge_{i,j}}{w(Out(i)) + w(In(j))} \tag{2.2}$$

$$CC_{i,j} = (v_i + v_j) * (\frac{NE_{i,j}}{\sum_{k \in NE} k}) \tag{2.3}$$

$$NCC_{i,j} = \frac{CC_{i,j}}{\sum_{t \in NE} t} \tag{2.4}$$

2.4 Experimental Evaluation

This section presents the analysis performed for C-Rank development and its evaluation, along with the protocols utilized during the corresponding experiments. We first introduce the used datasets, then report on the refinement performed in the heuristics values followed by the achieved results with *C-Rank* compared to other unsupervised keyphrase extraction techniques discussed in Section 2.2. C-Rank was developed on python and is available online[4].

2.4.1 Datasets

We use two standard benchmark datasets to evaluate the results achieved during and after C-Rank development and compare them with other keyphrase extraction approaches, which explored the same datasets.

The first is the SemEval2010 [28] dataset, divided in a trial, a train and a test set containing 40, 100 and 144 documents, respectively. Each one is an academic article belonging to one of four distinct ACM classifications. All the records are annotated with two sets of keyphrases as the author-assigned, which were part of the original document; and the reader-assigned, that were manually annotated by Computer Science students.

[4] `https://github.com/maurodlt/C-Rank`

The INSPEC is the second dataset [25], composed of 3 sets of documents, a training set containing 3000 text files, a validation set with 1500 and a test set consisting of 500. Despite having a similar number of keyphrases assigned per document, the INSPEC dataset, different from the SemEval, is composed by academic abstracts, which makes its files shorter.

2.4.2 Parameters Refinement

During C-Rank development several variables were defined. To determine the best possible values for those variables, we performed different analyses considering the SemEval training set, which lead us to our defined heuristics and parameters. These were explored to evaluate *C-Rank* in the test set of the datasets (*cf.* Subsection 2.4.3).

The co-occurrence window was the first variable defined in C-Rank development. Table 2.2 shows the results variance when the co-occurrence window changes and highlight in bold its optimal value.

Table 2.2: Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 trainning dataset varying the graph co-occurrence window.

Co-occurrence Window	**Top-5(%)**	**Top-10(%)**	**Top-15(%)**	**Average(%)**
2	15,23	19,98	20,49	18,6
3	**15,83**	**20,58**	**20,76**	**19,1**
4	15,83	20,48	20,81	19,0
5	15,83	20,53	20,95	19,1
10	15,63	20,26	20,95	18,9
100	14,96	19,82	21,08	18,6

Moreover, two heuristic variables values were defined, *LRC* and *CutOff* Threshold. Both of them were varied and provided optimal results when set to 87% and 18% respectively. Another analyzed C-Rank parameter was the centrality measure used to rank the co-occurrence graph. Table 2.3 shows the results when this metric changes.

In order to evaluate the Concept Linking usage and the proposed heuristics, Table 2.4 exhibits the variances of results applying our defined contributions. It clearly shows the improvement achieved when implementing the proposed techniques of concept linking and our elaborated heuristics.

Table 2.3: Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 trainning dataset varying the co-occurrence graph centrality measure.

Centrality measures	**Top-5**	**Top-10**	**Top-15**	**Average**
Closeness Centrality	15,29	18,95	19,62	18,0
Betweenness Centrality	15,36	19,49	20,76	18,5
Eigenvector	12,37	15,95	17,43	15,3
Pagerank	15,83	19,87	20,58	18,8
Degree Centrality	**15,83**	**20,58**	**20,76**	**19,1**

Table 2.4: Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 trainning dataset applying our proposed contributions.

Contributions	**Top-5**	**Top-10**	**Top-15**	**Average**
Using Concept Linking & Heuristics	**15,83**	**20,58**	**20,76**	**19,1**
Using only Concept Linking	8,56	11,58	12,69	10,9
Using only Heuristics	11,15	14,58	15,42	13,7
Without Concept Linking & Heuristics	7,07	9,55	10,68	9,1

2.4.3 Results

A final evaluation was performed to determine the effectiveness of *C-Rank* results achieved in both SemEval and INSPEC test datasets, which allows comparing *C-Rank* among other keyphrase extraction approaches. Table 2.5 presents the obtained results and Table 2.6 shows the comparison with other unsupervised keyphrases extraction techniques.

Table 2.5: Micro-average precision, recall and f-score on the extraction of Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 and the Inspec test datasets.

	Top-5			**Top-10**			**Top-15**		
	P.	**R.**	**F1.**	**P.**	**R.**	**F1.**	**P.**	**R.**	**F1.**
SemEval	28	9,6	14,2	24,2	16,5	**19,6**	20,3	20,7	20,5
Inspec	32	16	21,2	23,1	23,5	**23,3**	17,1	25,9	20,6

Table 2.6: Comparison among micro-average precision, recall, and f-score achieved by extracting 10 keyphrases on the SemEval2010 and the INSPEC test datasets. § indicates statistical significance improvement using a 2-sided paired t-test at $p < 0.05$.

	SemEval			**INSPEC**		
	Precision	Recall	F-score	Precision	Recall	F-score
Text Rank	7,9	4,5	5,6§	14,2	12,5	12,7§
BUAP	17,8	12,4	14,4§	-	-	-
Topic-Rank	14,9	10,3	12,1§	27,6	31,5	27,9
C-Rank	24,2	16,5	**19,6**	23,1	23,5	**23,3**

2.5 Discussion

For the best of our knowledge, the algorithms not relying on user data and yielding the best results were outperformed by C-Rank with statistical significance in the SemEval 2010 dataset.

Our approach explored external background knowledge from Babelnet, which is an important characteristic. We found a relevant impact with the use of the concept linking in the keyphrase extraction (*cf.* Table 2.4). The Concept Linking approach is a novel aspect of our algorithm that might be further explored in the keyphrase extraction and other NLP tasks. It not just brings background knowledge that assists in the keyphrases identification, but further produces intermediate structures with concepts and entities

semantically annotated that can be used to improve domain understatement and enrich other textual representation structures.

During the graph construction, the results varying the maximum co-occurrence distances between concepts (*cf.* Table 2.2) showed that the variance between results is low. Therefore, if performance is an issue, despite the lower f-scores, setting the co-occurrence window to 2 is equivalent of using only the direct-edges during all the algorithm, which would decrease the computational cost without much impact in the resultant values.

Despite the variance of the results, the heuristic values do not impact the algorithm performance. The centrality measure, on the other hand, can significantly decrease the results. As demonstrated by Boudin [6] and corroborated in Table 2.3, despite being simple, the degree centrality achieves higher results than other popular metrics usually used in keyphrase extraction algorithms, as the Pagerank. Considering the achieved results, further investigations on the use of concept linking in related NLP tasks could support and complement current solutions.

2.6 Conclusion

Keyphrase extraction plays a key role in the interpretation and analyses of textual documents. Existing proposals heavily rely on training datasets and external input. In this paper, we introduced *C-Rank*, an unsupervised keyphrase extraction algorithm that explored concept linking and graph-based techniques. Our approach enables the analysis of single documents and does not demand a textual compendium to be inserted by users. Our technique explored background knowledge from a wide-coverage semantic-network in a novel approach to obtain candidate Keyphrase and rank them. It used the concepts linked with the network as vertices of a co-occurrence graph, which is ranked based on heuristics and centrality measures. The conducted experiments showed the benefits of the elaborate features in the technique. The evaluation revealed that *C-Rank* outperformed, with statistical significance, all the unsupervised techniques that do not demand extra information to be provided by users on the SemEval2010 benchmark dataset. As future work, we plan to evaluate C-Rank against different types of data, other than scientific-related articles. Furthermore, we will investigate the C-Rank intermediate semantic structures produced in tasks related to the identification of domain topics based on a set of textual documents.

Chapter 3

SciKGraph: A Knowledge Graph Approach to Structure a Scientific Field

3.1 Introduction

The amount of publications a researcher must absorb has been increasing over the last years [5]. This issue is a regular part of academic life, which has to be updated with the freshest articles and discoveries concerning researcher's knowledge area. Consequently, the coming era of big scholarly data makes it hard for researchers to identify interesting documents to read [62]. Considering that research work is arduous even for experts, it is very quite hard for newcomers to accomplish it in a reasonable time. Therefore, computational mechanisms are essential for assisting scientists in finding what they seek, instead of letting them indiscriminately investigate for information. That is why researches have been conducted to construct and improve recommendation systems for scientific articles [20][62].

Nowadays, state-of-the-art methods usually use classification techniques to segment academic documents in pre-defined areas [65]. 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 pre-defined areas, giving 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 the ideal, leading him/her to waste time studying topics unrelated to his primary goal. This problem is caused by the toughness 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.

Improve 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 manage the interpretation of concepts in

different contexts [8]. To exemplify this, when analyzing a phrase with the word "apple" in it, this word could represent distinct concepts, as a fruit, a brand, or the New York City (The Big Apple), depending on the context it is applied. That is why most of the researches addressing this problem propose 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 the field. Our solution aims to identify segments of a field of knowledge in its sub-areas presenting a short textual description from those. We study to which extent the functioning of the framework behaves with different datasets and its application to distinct knowledge areas.

In our framework, instead of using only meta-data and citation information, we construct a knowledge graph (KG), a knowledge base system that integrates information and applies a reasoner into it to generate new knowledge [14]; 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 appearing 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 their 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 AI documents by gathering 1.018 articles from the *Artificial Intelligence* area. We use it to qualitatively evaluate the topics segmentation and the knowledge extraction, based on the key-concepts and keyphrases identified. Furthermore, we compare the knowledge graphs constructed based on the AI and the 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.

Results reveal up to 84% of accuracy by identifying in which academic area a set of articles belongs to, without relying on annotated data, which expresses the novelty of our proposal. 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 tendencies, which is an evidence that our proposed framework can be used to represent distinct scientific fields.

Our solution is suited to inform 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 3.2 discusses background work. Section 3.3 introduces the proposed framework to structure and analyze a scientific field. Section 3.4 describes the conducted experimental evaluation, including implementation aspects, datasets and reports on the achieved results. We discuss the obtained findings in Section 3.5. Section 3.6 exhibits the final considerations in this article.

3.2 Background

Nowadays, most of the academic textual knowledge have been structured based on three approaches: 1) classification algorithms, which are used to infer in which pre-defined area a piece of text belongs [31]; 2) citation networks, applying clustering methods on citation graphs to identify their main topics [55]; or 3) manually, requesting researchers to assign academic articles to pre-defined categories, as the ones defined in [51].

Kowsari *et al.* [31] state that most of the classification algorithms are divided into four modules: feature extraction, which identifies the main features from a piece of text; dimensionality reduction, that is optional and is used to optimize the algorithm; learning model, the most important step, responsible for determining the machine learning model used to classify the texts; and the evaluation, which determines the metric used to appraise the algorithm.

Concerning the feature extraction module, Kowsari *et al.* [31] describe several common techniques, as Term Frequency-Inverse Document Frequency (TF-IDF) [52], Word2Vec [18], and Global Vectors for Word Representation (GloVe) [48]. However, even identifying similar features based on their distance using Word2Vec or GloVe, those techniques do not disambiguate their elements. Therefore, different words that describe the same concept are represented differently. Moreover, those techniques require training to identify the main features of the text.

Considering this, Tosi and dos Reis [57] proposed C-Rank, a keyphrase extraction technique that, in order to mitigate these issues, uses Babelfy [39] 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 [42], a knowledge graph constructed based on WordNet [36], DBPedia [2], 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.* [31] indicate distinct learning models, as Rocchio classification [50], Naïve Bayes Classifier [32], k-nearest neighbor [1], and Support Vector Machine (SVM) [12]. According to the authors, those models require supervised training, are computationally expensive, or do not achieve satisfactory results. In most of the cases, the usage of supervised training to perform textual classification is not a problem. 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 pre-existing categories.

On the other hand, the citation networks are used to classify documents without requiring supervised training to segment the network areas. Jung and Segev [26] performed this segmentation, and in addition to automatically extract the main communities of a scientific field, their work inferred future changes in these 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 clusterization process identifies groups of related articles belonging to the same area. It is fundamental to understand the network topology and the faced problem to select the better-suited clusterization algorithm for the target application. Although there are different types of clusterization techniques, two main differences between them are the key ones: 1) if they accept overlapping clusters; 2) if 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 smaller sub-clusters inside of it.

Also, to determine the best type of clusterization for each application, it is necessary to choose among several algorithms, which is usually performed based on an evaluation metric. One of the most well-known metrics to evaluate clustered graphs is the modularity, proposed by Newman [43][44]. 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 is considered the segmentation of the network and, therefore, the organization of the clusters. Furthermore, there are several variants of the modularity metric, for example, the ones compared by Chen and Szymanski [10], studying the best metrics to evaluate overlapping clusters.

The definition of the adequate clusterization 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 to the problem, the content of the documents analyzed. In this direction, Silva *et al.* [55] proposed to perform text analysis to identify keywords of their areas in addition to the clusterization of citation networks. Their work aimed to construct the taxonomy of the studied scientific field. Although improving 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 only on metadata information. Besides, it would be relevant to a study to investigate the relationship among these keywords, which was not done.

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

In this work, we originally propose a framework to construct knowledge graphs to represent and segment scientific fields. They are constructed based on features extracted as in document classification tasks, which are modeled as graphs, as in C-Rank. Then, they are clustered similarly to the citation network approaches, identifying the key-areas

of the scientific field and the concepts belonging to them. Moreover, our approach not only considers the textual data from the documents but further addresses its semantics based on links connecting its concepts to existing background knowledge. This may improve the understatement of the analyzed area by researchers and the analysis of the knowledge graphs constructed.

3.3 SciKGraph: Structuring Science as a Knowledge Graph

This section describes SciKGraph, a framework to structure and analyze a scientific field as a knowledge graph. Figure 3.1 illustrates our proposal organized into three key tasks. The “Knowledge Graph Construction" task (cf. Section 3.3.1), based on the C-Rank co-occurrence graph [57], constructs a knowledge graph receiving as input the BabelNet knowledge and the collection of documents that represents the studied scientific field. The “Knowledge Graph Overlap Clusterization" (*cf.* Section 3.3.2) clusters the previously constructed graph, identifying the main topics of the studied scientific field. The “Knowledge Extraction" task (*cf.* Section 3.3.3) extracts relevant information regarding the studied scientific field and its main topics based on the knowledge graph structure and its clusters.

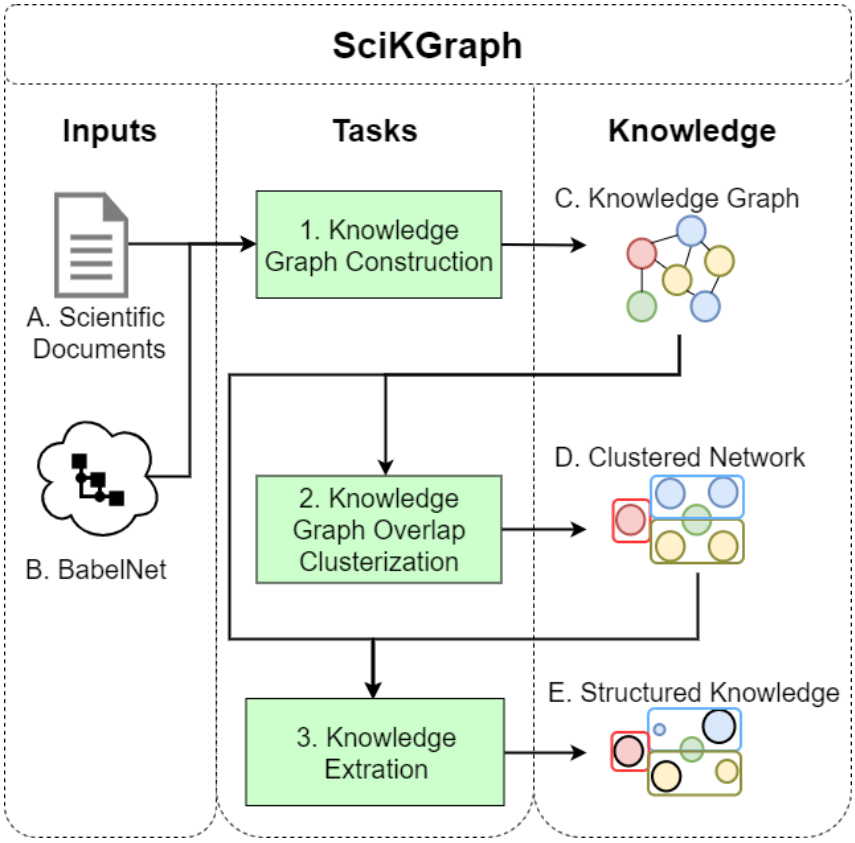

Figure 3.1: SciKGraph pipeline to structure a scientific field as a knowledge graph.

3.3.1 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.* Figure 3.2). First, it parses the input documents to simple texts without images, equations, or citations. Then, it sends those texts to Babelfy HTTP API, that identifies their concepts, disambiguates them, links those 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 that the co-occurrence occurred in the collection. This approach is based on the co-occurrence graph constructed in C-Rank [57], but it structures a collection of documents, instead of single articles.

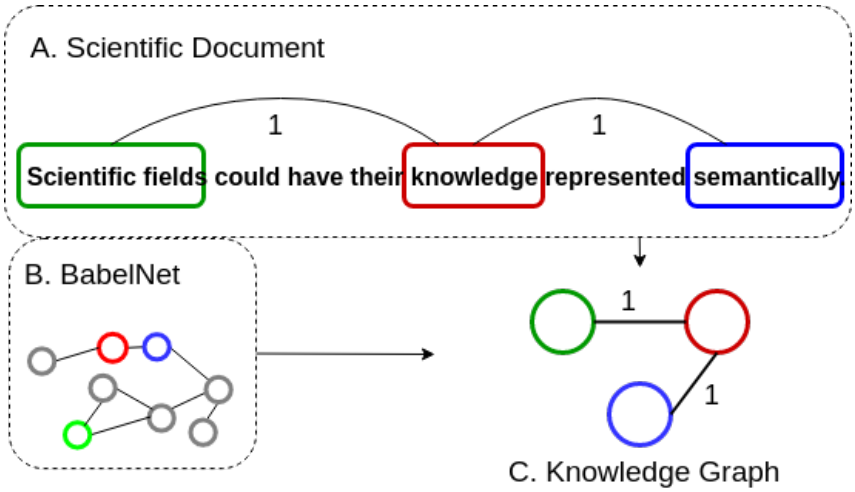

Figure 3.2: Representation of the knowledge graph construction task.

Figure 3.2 represents the construction of an illustrative knowledge graph that received as input the phrase “Scientific fields could have their knowledge represented semantically". This phrase 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 phrase. Therefore, they are linked by an edge, which also occurred to 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 on the collection of documents used to construct the knowledge graph, this weight is updated by the number of times the co-occurrence occurs.

3.3.2 Knowledge Graph Overlap Clusterization

After constructing the knowledge graph, the identification of its sub-areas is the next relevant step. The framework performs it using a clustering technique, which divides the knowledge graph into groups of vertices that are more connected among themselves than

with vertices outside of their group. We propose this structure to segment knowledge, considering that concepts belonging to the same area are more connected among themselves than with other ones. As areas may be interdisciplinary or have concepts in common among themselves, their representation with crisp clusters would not be appropriate, because such 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 pre-processed before the clusterization process. The pre-processing consists of 3 steps.

1. Remove the edges from the knowledge graph 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. Remove the nodes with higher centrality in the network, numbered by the $threshold_{centrality}$. 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 disjoint sub-graphs that were created after the two previous processes.

During the pre-processing, three metrics are explored: 1) the centrality measure; 2) the $threshold_{edges}$; and 3) the $thereshold_{centrality}$. The centrality measure is the metric used to rank the concepts based on their relevance for the network. It is used to identify the most generic concepts, that would negatively impact the clusterization process; and the most relevant ones, used in the Knowledge Extraction task. In C-Rank, Tosi and dos Reis [57] identified that the 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, the degree centrality. It is a simple metric that considers that the higher the number of vertices connected to vertice v, higher is the centrality of v in the graph.

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

The $threshold_{centrality}$ is related to the granularity of the clusters, which means that the higher its value, 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 clusterization 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, as the $threshold_{edges}$, is also found on the basis of the studied dataset.

We recommend a semi-automated model for achieving 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 the user criteria. The results of varying these threshold values are presented and discussed in Section 3.5.

After the pre-processing, the clusterization process segments the knowledge graph in 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 clusterization process must accept overlapping to represent the problem correctly. As the idea behind our approach is to elucidate researchers regarding a scientific field structure, 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 clusterization of the knowledge graph. Among them, only OClustR [49] and SLPA [64] allowed overlapping clusterization without demanding 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 it excludes some vertices during the clusterization process, therefore, we discarded it. Consequently, we adopted the OClustR algorithm during the clusterization process. It is a graph-based clustering technique that allows overlapping and identifies the optimal number of clusters automatically.

We found that as the OClustR identifies the optimal number of clusters automatically, it may segment the knowledge graph in 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 requesting from users the ideal number of clusters, we highlight that this step is optional. The researcher takes his/her decisions based on the already clustered knowledge graph, which may facilitate the task 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 3.4 describes the methodology applied to evaluate and compare the effectiveness of these algorithms in the studied datasets.

Simple-Threshold:

It 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.

Top-Modularity:

Algorithm 1 describes the Top-modularity technique, which compares the modularity that would be obtained 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 the $Clusters$ in descending order based on its elements length. Then, in lines 2 and 3, it segments this set in 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}. At last, 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.
Clusters = descending_sort(Clusters);
C_top = Clusters[: n];
C_small = Clusters[n :];
for j in C_small do
    best_modularity = MAX_FLOAT;
    best_cluster = −1;
    for i in C_top do
        mod_i = calc_Modularity(i);
        mod_j = calc_Modularity(j);
        iUj = i + j;
        mod_iUj = calc_Modularity(iUj);
        if mod_i + mod_j − mod_iUj < best_modularity then
            best_modularity = mod_i + mod_j − mod_iUj;
            best_cluster = i;
    end
    i = merge_clusters(i, j);
    delete j;
end
return C_top;
```

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 its elements length. Then, different from Algorithm 1, between lines 2 and 18, it iterates over clusters $j \in Clusters$ until it reaches the desired number of n clusters, which it checks if 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, it 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 with the cost of a 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.
Clusters = descending_sort(Clusters);
for j in Clusters do
    if length(Clusters) ≤ n then
        return Clusters;
    best_modularity = MAX_FLOAT;
    best_cluster = −1;
    for i in Clusters do
        if i ≠ j then
            mod_i = calc_Modularity(i);
            mod_j = calc_Modularity(j);
            iUj = i + j;
            mod_iUj = calc_Modularity(iUj);
            if mod_i + mod_j − mod_iUj < best_modularity then
                best_modularity = mod_i + mod_j − mod_iUj;
                best_cluster = i;
    end
    i = merge_clusters(i, j);
    delete j;
end
```

3.3.3 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.

The segmented topics are represented by the clusters obtained in the knowledge graph clusterization task (Subsection 3.3.2). Their structure might illuminate scientists regard-

ing the topology of a scientific field; 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. The degree centrality is explored to calculate the relevance of each concept, as Tosi and dos Reis [57] determined it to be the best centrality metric to be used in this structure.

Figure 3.3 presents an example by illustrating a graph with 6 vertices connected among themselves. The vertice labeled as "Image" is connected to another 5 vertices, making its degree centrality equals to 5. It is the vertice with higher degree centrality, which means that it is considered the most relevant vertice of the graph. Therefore, if one would have to describe the graph of Figure 3.3 with one of its vertices, the most relevant one "Image", would be appropriate.

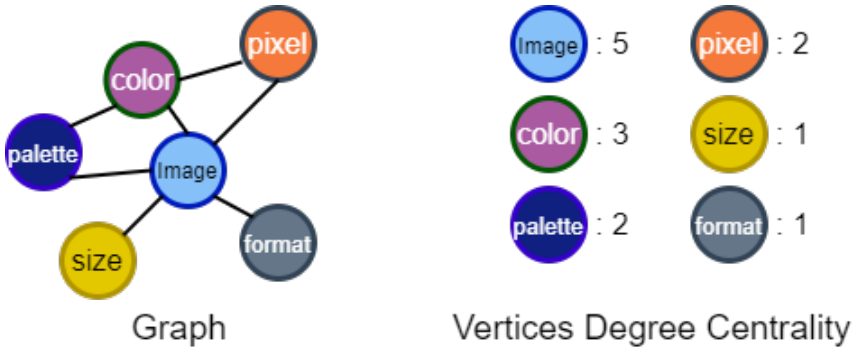

Figure 3.3: Example of degree centrality usage to calculate relevance of concepts.

At last, keyphrases are expressions used to represent the content of textual structures. Different from the most relevant concepts, the 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 trivial 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 [57], which does not demand external data to extract keyphrases. 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, not from single documents, two 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, the graph construction and ranking can now be skipped. Second, the heuristics concerning the exclusion of candidate keyprases appearing in the begging of the document cannot be applied. This heuristic considers that keyphrases usually are 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: 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 by multiple words, powering them by the inverse of their length. Then, one must merge keyphrases formed by multiple concepts [57].

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.

3.4 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 (Subsection 3.4.1). Subsection 3.4.2 describes the performed procedure to evaluate SciKGraph, by providing details of the purpose and organization of the conducted analyses.

3.4.1 Implementation and Datasets

SciKGraph was developed in Jupyter Notebook [29] for facilitate its reproducibility. This allows displaying code along with textual elements and figures, enhancing the interactivity between the user and the 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*[21], a Python package for the study of complex networks. In addition, we implemented the OClustR and the proposed agglomerative methods (*cf.* Subsection 3.3.2), both of them are available online[3][4].

The evaluation of SciKGraph for structuring a scientific field relied on two datasets. We used the WOS-5736 [30] and the AI datasets. In the following, we explain the reason for their choice and present their characteristics.

WOS-5736:

The WOS-5736 (WOS) dataset was constructed using *Web of Science* data and meta-data of published papers to validate document classification methods. It is composed of 5,736 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.

[1] Omitted due to ongoing blind review
[2] `https://github.com/aghie/pybabelfy`
[3] Omitted due to ongoing blind review
[4] Omitted due to ongoing blind review

Our “Knowledge Graph Overlap Clusterization" 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 pre-defined categories (expected answer). This dataset has annotated data and enables us to compare our method with document classification algorithms.

AI:

Different from the 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 [5] using “Artificial Intelligence" as the search term, sorting the results based on their number of citations.

The documents were obtained in PDF, that we converted to XLST using GROBID [19], a machine learning library to parse PDF. Then, we removed their citations, mathematical formulas, images, and converted the documents to simple texts. In this process, 33 articles could not be correctly parsed and were discarded, resulting in the 1,018 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 for results comparison between it and 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.

3.4.2 General Procedure and Organization of Analyses

Table 3.1 presents the conducted quantitative and qualitative analyses with the respective datasets (performed in the WOS and the AI datasets).

To the development of our analyses, we firstly defined the AI and the WOS datasets as representations of the scientific fields we would like to represent. Then, constructed a knowledge graph for each one of them 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. Subsections 3.4.4 and 3.4.5). Based on the obtained results, we determined the *Top-modularity* technique presenting the best trade-off between results produced and computational cost. We applied it to obtain 5 clusters in the AI and 15 clusters in the WOS knowledge graphs (cf. Subsection 3.4.3) 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

[5] `https://ieeexplore.ieee.org/`

Table 3.1: Proposed analyses and used datasets. This table indicates the list of evaluations conducted; for each of them describes the datasets used (WOS, AI, or both datasets (WOS - AI)).

	Accuracy	**Modularity**	**Structure**	**Content**
Knowledge graphs 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	
Overlapping clusters			AI	
Key-concepts and Clusters Keyphrases				AI

200 and 150 concepts each; it will be excluded from the one composed of 150 concepts and maintained on the bigger one. In addition, we experimentally studied the topology of the knowledge graphs (cf. Subsections 3.4.6 and 3.4.7) and extracted knowledge from them, as their key-concepts, keyphrases, overlapping topics and relations among their clusters (cf. Subsections 3.4.8, 3.4.9, and 3.4.10).

In the following, we further explain the conducted analyses.

Knowledge graphs construction and visualization:

This analysis (cf. Subsection 3.4.3) provides the results of the construction of the AI and the WOS knowledge graphs using SciKGraph. It allows the user to visualize the structure obtained applying the proposed framework in a whole-documents and in abstracts-only datasets.

Accuracy comparison of the Agglomerative techniques:

The accuracy of the document classification problem quantifies how related the identified topics are 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. Subsection 3.4.4). 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 demands the usage of an annotated dataset, which is not its case.

Modularity comparison of the Agglomerative techniques:

This analysis (cf. Subsection 3.4.5) compares the clusters' modularities variation after applying the suggested agglomerative techniques. It is performed using both the AI and the WOS knowledge graphs. Moreover, it investigates the variation of the modularity using crisp and overlapping clusters.

Top-modularity accuracy and modularity correlation:

This analysis (cf. Subsection 3.4.6) 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 demands the usage of an annotated dataset, which is not its case.

Knowledge graph size and modularity correlation:

It investigates if there is a direct correlation between the size variation of the proposed knowledge graph and its modularity (cf. Subsection 3.4.7). The size of the knowledge graph varies modifying the $threshold_{edges}$ parameter, which allows updating 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 tendencies using the WOS one.

Knowledge Graphs clusters relations:

This analysis (cf. Subsection 3.4.8) exemplifies how one may analyze the relations among the topics of the studied scientific field using the SciKGraph framework. It is performed using the overlapping clusters identified through the Top-modularity technique in the WOS and the AI knowledge graphs.

Overlapping clusters:

The Overlapping clusters analysis (cf. Subsection 3.4.9) investigates how the clusters overlapping topics can be visualized and which knowledge it may bring to the researchers' analyses. Moreover, it exhibits an example based on the AI knowledge graph by using a Venn Diagram to represent the clusters obtained through the Top-Modularity technique. This analysis could not be performed using the WOS knowledge graph because of the Venn Diagram limitation in illustrating more than 5 overlapping clusters simultaneously.

Key-concepts and Clusters keyphrases:

This analysis (cf. Subsection 3.4.10) shows the key-concepts extracted from the AI knowledge graph. Furthermore, it identifies and presents the keyphrases from its clusters, defined using the Top-Modularity technique. It was not performed in the WOS knowledge graph because of the number of keyphrases its high amount of clusters would generate, which would not be straightforward to analyze.

3.4.3 Knowledge graph construction

We constructed the WOS-5736 knowledge graph and identified 37,591 different concepts and 390,296 connections between them. Before the clusterization, the pre-processing step reduced them to 667 concepts and 665 connections. The clusterization procedure identified 210 different clusters. Since this volume of clusters is hard to analyze, we assessed different agglomerative techniques to reduce it (cf. Subsections 3.4.4 and 3.4.5), 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 pre-processing step reduced them to 2,495 concepts and 6,602 edges, which were clustered in 436 distinct topics. Different agglomerative techniques were analyzed to merge these clusters and favor their interpretation, reaching 5 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 their centrality in the network, its edges represent the co-occurrence of the concepts and have width based on their weight, and the clusters are crisp. This visualization was constructed using Cytoscape [54] software and CoSE [13] algorithm, enabling us to observe the effectiveness of the clusterization process and how the concepts in the network interact one with the other.

Results for the WOS dataset

Figure 3.4 illustrates the knowledge graph created from the WOS dataset with nodes weighted based on their degree centrality and colored based on a crisp clustering.

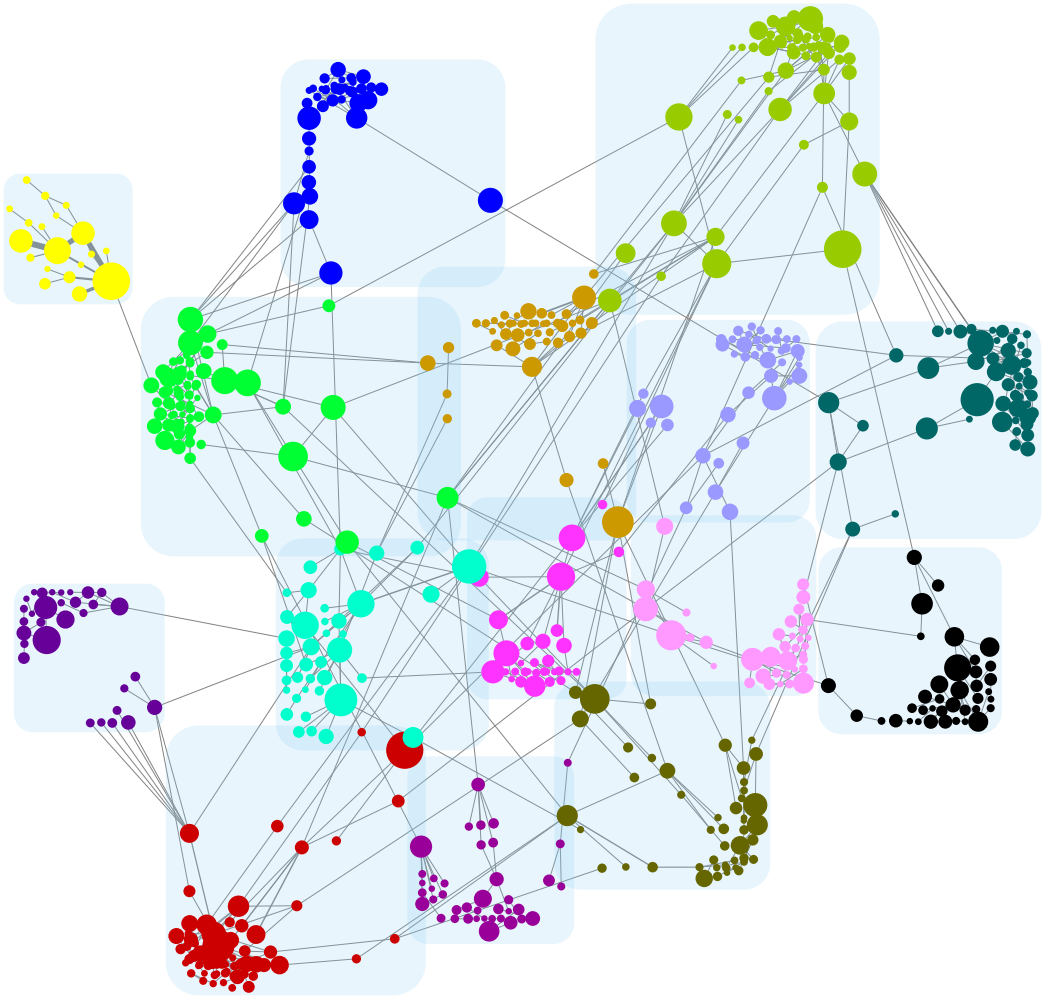

Figure 3.4: WOS knowledge graph.

Figure 3.4 presents the whole WOS knowledge graph, from which one can attest the effectiveness of the clusterization, observing how well-segmented and clustered are the concepts. The yellow cluster, for example, has only a single edge linking it with another cluster. This visualization allows the analysis of the connections among concepts, which is impaired in this example because the image size does not let the concepts' labels to be large enough to be readable.

Results for the AI dataset

Figure 3.5 presents the knowledge graph constructed from the AI dataset with nodes weighted based on their degree centrality and colored based on a crisp clustering.

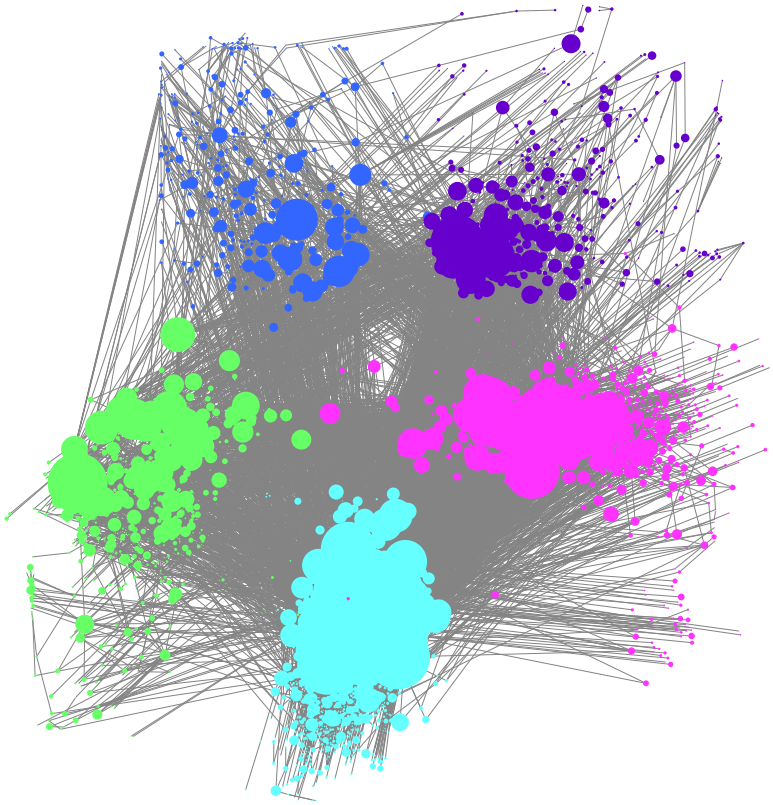

Figure 3.5: Artificial Intelligence knowledge graph.

We observe in Figure 3.5 that, different from Figure 3.4, the segmentation of the clusters is not as well defined, having many connections among clusters. Figure 3.5 illustrates different topics from the same scientific field. Furthermore, the AI knowledge graph visualization highlights the AI topics from a broader perspective, segmenting them in only 5 groups with almost 10 times more connections than the WOS knowledge graph. This explains the vertices overlapping themselves and the edges connecting different clusters.

This analysis enables a broader visualization of the studied areas. The plotted WOS and AI knowledge graphs show the effectiveness of the clusterization process and how the concepts of the network interact with each other. We note that some concepts in both representations could have being clustered differently, as the purple nodes in the left of Figure 3.4, which are segmented in two different groups. We constructed this analysis using the crisp clusterization 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 instead of an image that has only 2 dimensions. Consequently, even not reducing the dimensionality of the problem, we use the crisp representation to diminish the overlap between those dimensions, increasing their disconnection, and improving its visualization by using a simple image.

3.4.4 Accuracy comparison of the agglomerative techniques

The usage of 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 which 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 pre-defined 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 *train* set to train which annotated areas 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, Figure 3.6 exemplifies the identification of which annotated area each cluster represents. It receives the training set and the clusters, identifies the concepts that they have in common, and outputs the clusters/areas correlations percentage. In this example, Cluster 1 shares 6 of 7 concepts with the documents from Area 1, resulting in a correlation of 85%.

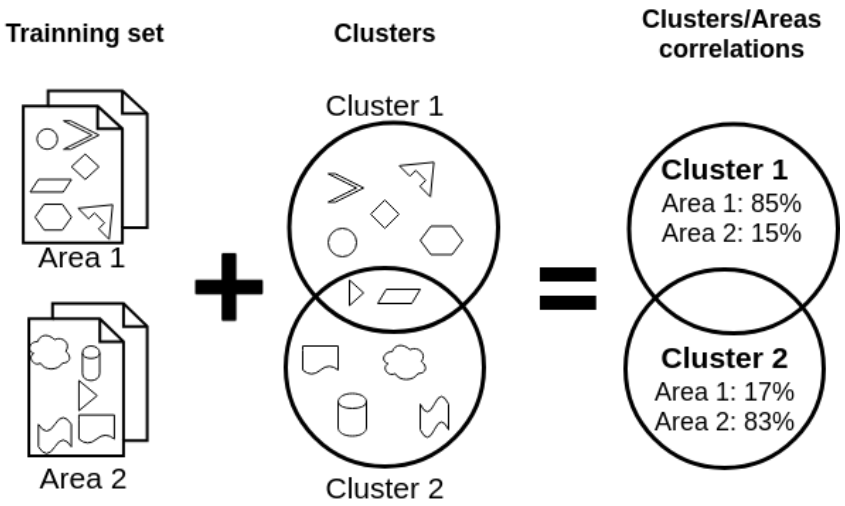

Figure 3.6: Example of the clusters/areas correlations percentage calculation.

Figure 3.7 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.

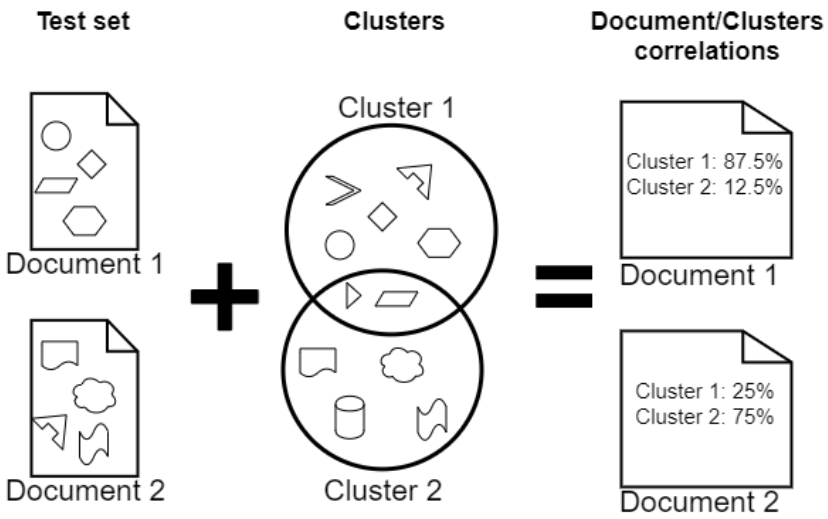

Figure 3.7: Example of the document/clusters correlations percentage calculation.

Furthermore, after identifying the clusters/areas and the document/clusters correlations, we process the probability of the document to belong to each area. For example, in Figure 3.8, Document 1 has correspondence of 0.875 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 to belong 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 which Document 1 has higher probability to belong, we indicate that it is Document's 1 main area.

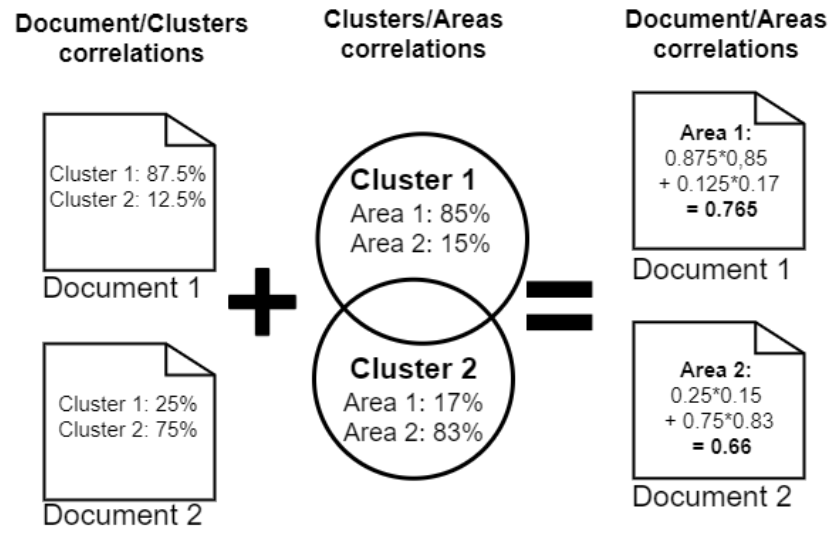

Figure 3.8: Example of the area of a document.

Consequently, considering that the documents from the *test* set are annotated with the areas and sub-areas that they belong, we evaluate the accuracy of the results of our framework, calculating the percentage of document areas and sub-areas that it correctly determined. Accordingly, we can use this metric to compare the accuracy obtained varying the proposed agglomerative techniques and the number of final clusters that they identify.

Figure 3.9 presents the obtained accuracy in the classification of documents in their respective topics.

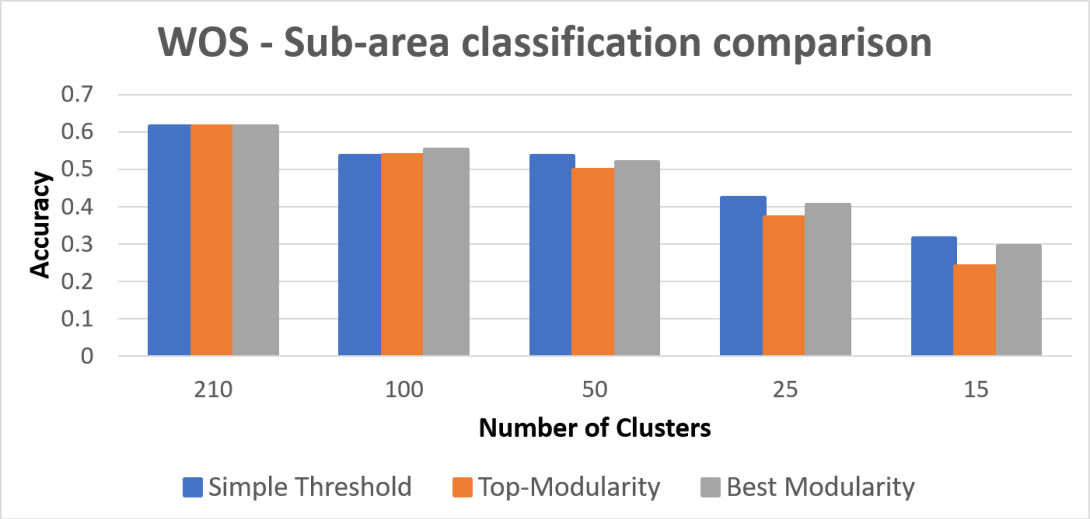

Figure 3.9: Agglomerative methods accuracy comparison classifying WOS documents in pre-annotated sub-areas.

From the results present in Figure 3.9, we observe that the reduction of the number of clusters negatively impacted the document classification accuracy. All three proposed agglomerative techniques followed the same tendency and obtained similar results. Even though Top-Modularity accuracy decreased quicker than the other methods, it obtained similar results segmenting higher amounts of clusters.

Figure 3.10 shows the accuracy in the classification of the documents in their respective areas.

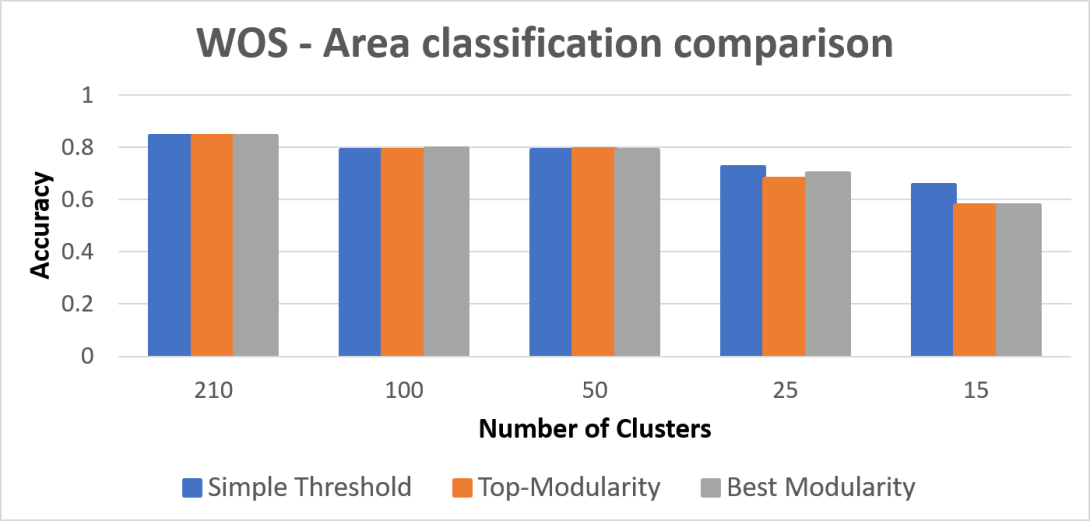

Figure 3.10: Agglomerative methods accuracy comparison classifying WOS documents in pre-annotated areas.

Figure 3.10 exhibits similar results to Figure 3.9. 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 eleven.

Our results reveal that the usage of the agglomerative technique negatively impacted the accuracy in the document classification task. However, this was expected as the OClustR algorithm, theoretically, had already found the optimal number of clusters of the dataset. Thus, the process of merging clusters would only degrade this segmentation. The *Simple Threshold* technique obtained better results, followed by the *Best-Modularity* and at last the *Top-Modularity*. This occurs because the *Simple Threshold* excludes concepts instead of re-classifying them in other clusters, reducing noise and the complexity of the problem. The *Best-Modularity* always choose the best clusters to merge, then it was expected that it would perform better than the *Top-Modularity*. We note that the difference between the accuracy of these two metrics is low compared to the computational complexity difference between them.

3.4.5 Modularity comparison of the agglomerative techniques

This analysis aims to determine how segmented are the obtained clusters and, therefore, how disjoint are the topics identified. For this purpose, we use the Q_{ov}^{L} modularity metric, as suggested by Chen and Szymanski [10]. This metric was calculated not just to determine if the knowledge graph was clustered correctly, but to compare how well segmented were the clusters obtained using the agglomerative methods proposed.

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

Results for the WOS dataset

Figure 3.11 illustrates the modularity variance using the different agglomerative techniques in the WOS dataset. We observe that the Top-Modularity and the 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.

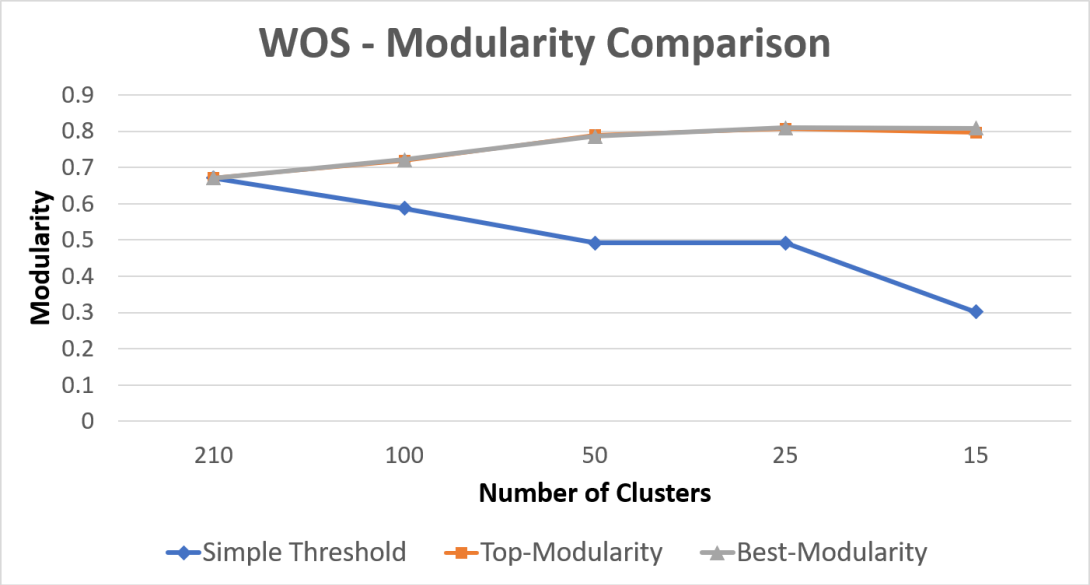

Figure 3.11: Agglomerative methods modularity comparison merging crisp WOS clusters.

Results for the AI dataset

Figure 3.12 presents the modularity variance using the different agglomerative techniques in the AI crisp clusters, whereas Figure 3.13 describe the same variance, but using the overlapping clusters.

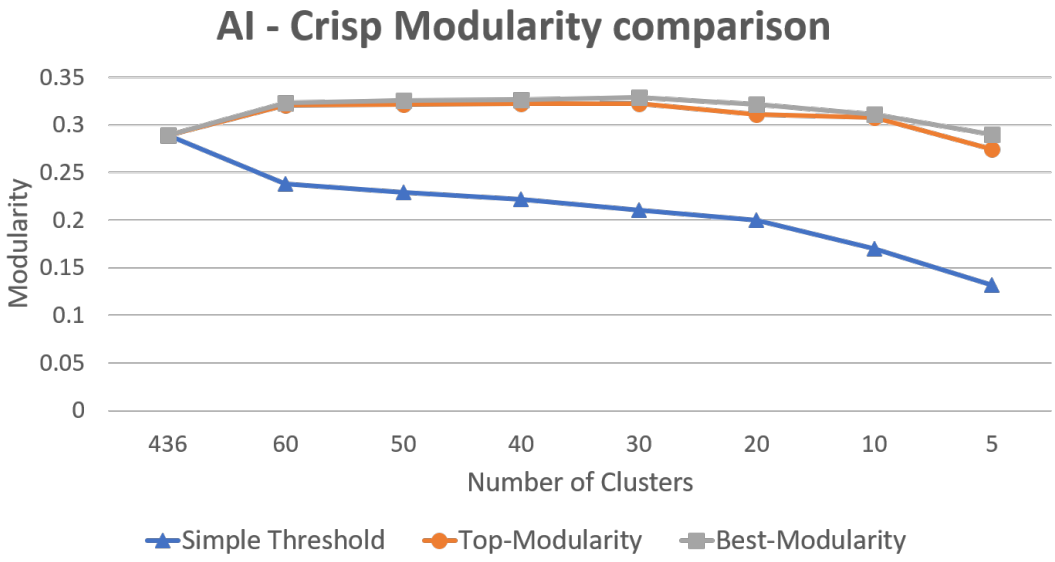

Figure 3.12: Agglomerative methods modularity comparison merging crisp AI clusters.

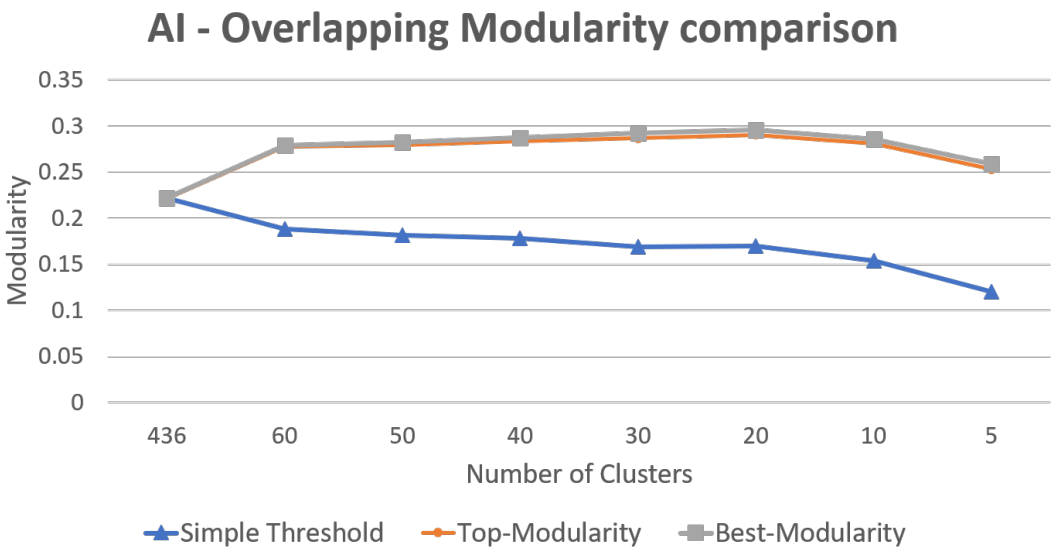

Figure 3.13: Agglomerative methods modularity comparison merging overlapping AI clusters.

We note that Figures 3.12 and 3.13 present the same tendencies as the ones observed in the WOS dataset. However, we highlight that comparing the results from these two analyses, we attest to a difference between the modularity calculated with crisp and overlapping clusters, the latter being the one with the worst modularity.

We found that the overlapping clusterization decreased on average 14% of the clusters modularity. This showed that different from the accuracy analysis, the *Simple Threshold* obtained the worst modularity; the other two techniques improved the modularity metric, achieving almost identical results. The performed analysis using the WOS knowledge graph identified the same tendencies using crisp clusters (cf. Figure 3.11), and overlapping ones.

Furthermore, the AI clusters achieved lower modularity compared to the WOS clusters, but it was already expected as the AI dataset is composed of articles of a single area, and the WOS is composed of three distinct areas, which are easier to be segmented. The fact that the *Top-Modularity* metric and the crisp clusters stayed most of the time above 0.30 indicates that the clusters were well defined.

3.4.6 Top-modularity accuracy and modularity correlation

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

Figure 3.14 presents the modularity and the accuracy obtained varying the number of clusters using the Top-Modularity metric. This analysis merges the accuracy and modularity results, showing no direct correlation between these metrics using the WOS dataset. When we reduce the number of clusters, the modularity increases and the accuracy decreases. We observed a different tendency using the Simple Threshold method, both

metrics decreasing, which indicates that there is no correlation between the modularity and the accuracy.

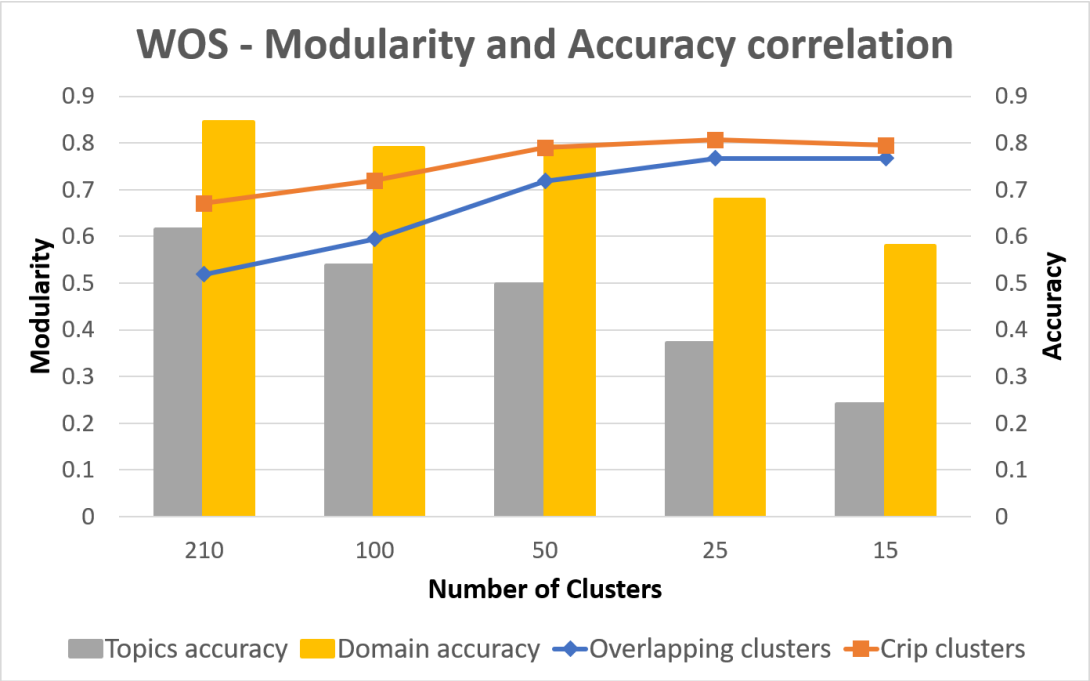

Figure 3.14: Modularity and accuracy correlation analysis using the Top-Modularity agglomerative method to merge crisp and overlapping WOS clusters.

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

3.4.7 Knowledge graph size and modularity correlation

This analysis searches for correlations between the size of the knowledge graph and the modularity of its clusters. It is significant to find out if the size of the knowledge graph impacts the segmentation of the structure or if they are independent. This result can assist researchers to define the $threshdold_{edges}$ value. If observed a correlation between these values, one can modify the $threshdold_{edges}$ to change the size of the knowledge graph and improve its segmentation.

Figure 3.15 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. 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, the 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.

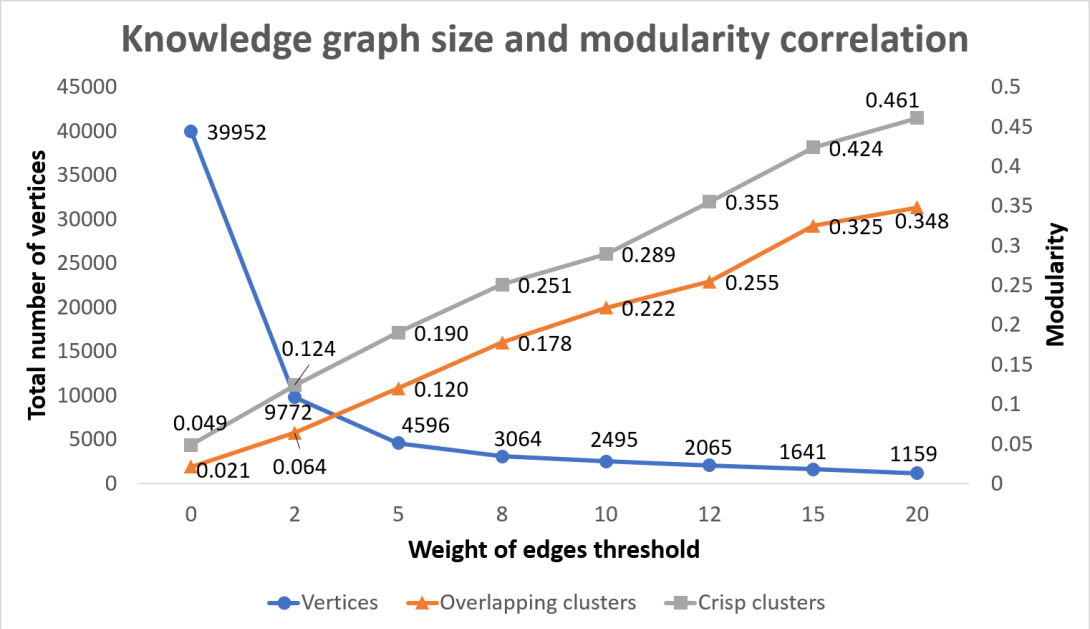

Figure 3.15: Analysis of correlation between Knowledge Graph size and modularity by varying the $threshold_{edges}$ parameter.

The result 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 showed in Figure 3.14 corroborates with this assumption, which illustrates that when increasing the *edges threshold*, the total number of concepts of the knowledge graph exponentially decreases and the modularity linearly increases. Actually, we used this experiment to set the $threshold_{edges} = 10$, so the modularity obtained in the AI knowledge graph would stay close to 0.30, excluding the minimum number of concepts as possible.

3.4.8 Knowledge Graphs clusters relations

After determining the clusterization 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 vertice represents each cluster, and all the connections between two clusters are merged in a single edge. This visualization is constructed using Cytoscape [54], which enables us to graphically observe not only how the clusters are highly connected to themselves, compared to other connections, but which clusters interact more between themselves.

Results for the WOS dataset

Figure 3.16 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.

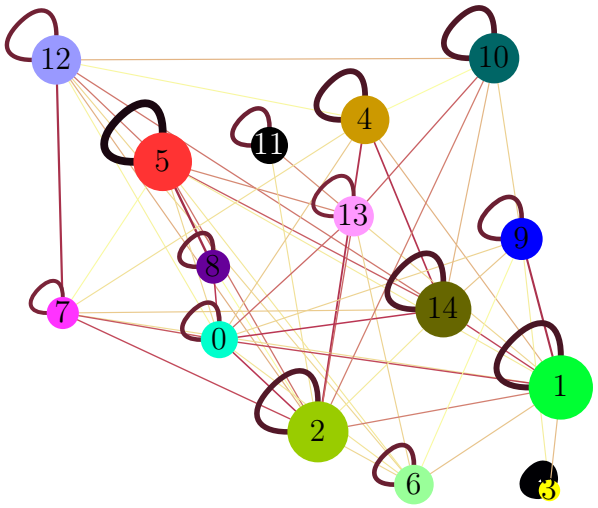

Figure 3.16: Graph of WOS clusters relations.

Results for the AI dataset

Figure 3.17 illustrates the relations between the AI clusters. We observe the same characteristics as obtained with the WOS dataset (Figure 3.16). Take as an example Clusters 3; 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 3 represents a more independent topic.

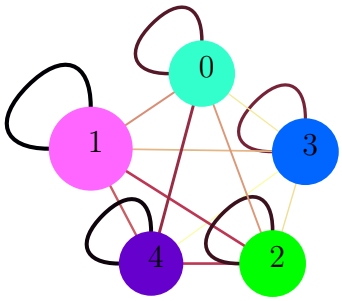

Figure 3.17: Graph of AI clusters relations.

The used representation for plotting the knowledge graphs allows us to confirm the high modularity of the clusters, with self-connections stronger than the other ones. This might assist researchers in identifying the topics that interact more among themselves. For example, in Figure 3.16, the topic *4*, which interacts more with the *13* and the *14* than it interacts with the others.

3.4.9 Overlapping clusters

As SciKGraph represents knowledge through overlapping clusters, not only the edges that link concepts between those clusters must be analyzed but also their overlapping concepts. Therefore, to visualize the number of overlapping concepts and in which clusters they belong, we represented those groups using a Venn Diagram, constructed by InteractiVenn [23]. This visualization allows researchers to observe the intersection regions among areas of a scientific field and quantify their size, which helps them in understanding their correlations and determining how advances in one of the areas can contribute to the others. Due to the limitation of this type of diagram to represent more than 5 groups, we performed this analysis only to the AI dataset, which was agglomerated in 5 clusters for this experiment.

Figure 3.18 presents to which extent the clusters of the AI knowledge graph are overlapped. We observe that the majority of the concepts in the AI knowledge graph belong to a single topic. We can quantify the similarities of the topics based on the number of concepts that they share. In addition, 15 concepts belong to all topics, which indicates their generality and that maybe they should have been deleted in the pre-processing step.

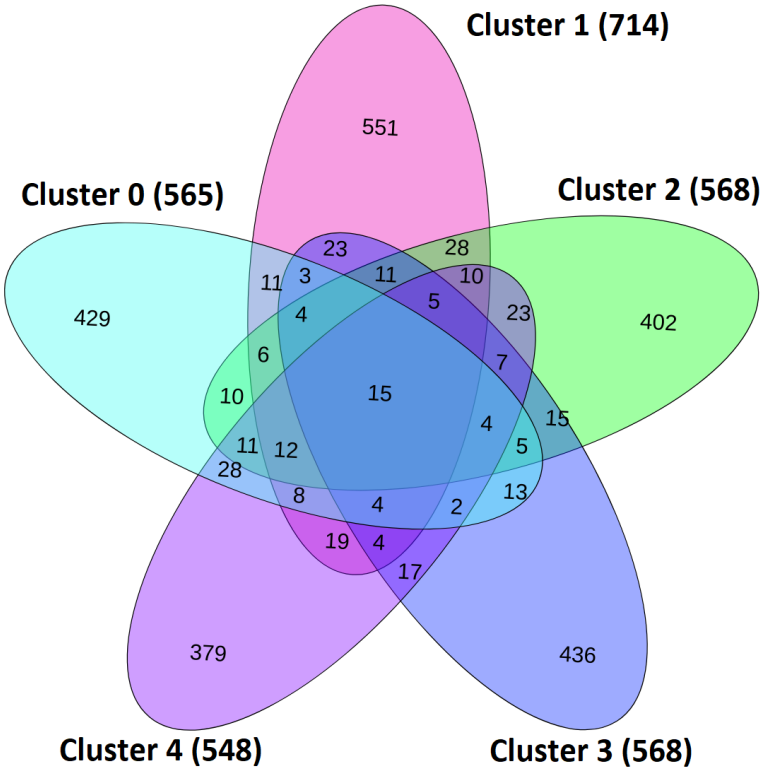

Figure 3.18: Venn Diagram of the AI knowledge graph overlapping clusters.

We also observe in Figure 3.18 that the overlapping rate achieved after the clusterization and the *Top-Modularity* metric application is low, with less than 20% of the concepts belonging to more than one topic. Other than that, this area could have been divided into more topics, but we choose to segment it only in 5 clusters to enable its visualization with a Venn Diagram.

3.4.10 Key-concepts and clusters keyphrases

In order to qualitatively analyze the main concepts from the knowledge graph, we extracted them based on their degree centrality, which was the metric used to extract key-concepts from academic documents in C-Rank [57]. As we constructed the knowledge graph using a structure similar 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 analyzed each cluster as a separate graph and applied C-Rank to extract its keyphrases.

Table 3.2 presents the key-concepts - composed of one or more words - from the AI knowledge graph, identified based on their degree centrality. Table 3.3 presents the keyphrases - composed of one or more concepts - from each of the AI clusters, identified using the C-Rank algorithm.

Table 3.2: AI Knowledge Graph key-concepts sorted by their degree centrality.

1. image	6. framework	11. learn	16. class
2. figure	7. feature	12. dilemma	17. mathematical function
3. algorithm	8. method	13. different	18. approach
4. demonstrate	9. used	14. data	19. value
5. outcome	10. number	15. train	20. performance

We observe in Tables 3.2 and 3.3 that key-concepts and keyphrases identified correspond to relevant concepts in the Artificial Intelligence area. For example, the concept "image" in Table 3.2 is the one with higher centrality, indicating the popularity of artificial intelligence approaches applied to images. Moreover, the keyphrases identified in Table 3.3, despite suffering from some overlapping, can describe the topics that their clusters represent. Take as an example the Cluster 3, which has all keyphrases related to the Machine Learning area as "neural network", "SVM classifier", and "node layer".

Great part of the extracted terms express fundamental concepts of the Artificial Intelligence area, as *learn*, *performance*, and *algorithm*. For example, based on this key-concepts one may understand that *learning* is important for Artificial Intelligence; *algorithms* and its *performance* must be considered. Concepts such as *figure* and *used* 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 the concept *image* (the 51th in our ranking) to be relevant for the scientific field. However, this metric proved to be not 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, the keyphrases of the *Cluster 0* indicate that

Table 3.3: AI clusters keyphrases ranked using C-Rank.

Cluster 0		
1. table reports	2. paper described	3. paper introduces
4. provide	5. paper	6. work
7. table	8. term	9. task
10. information	11. modes	12. technique
Cluster 1		
1. vector input	2. steps distance	3. regions
4. face database	5. researcher areas	6. input space
7. points	8. face	9. input patterns
10. represent	11. vectors	12. changes
Cluster 2		
1. performs best	2. task performs	3. face database
4. input space	5. performs	6. SVM classifier
7. input patterns	8. based	9. steps
10. task	11. techniques	12. face
Cluster 3		
1. node layer	2. SVM classifier	3. position orientation
4. weight connections	5. SVM classification	6. node
7. weights	8. rules	9. based
10. input	11. neural network	12. setting
Cluster 4		
1. percent rate	2. rate recognition	3. table see
4. input space	5. compare	6. achieve
7. improved	8. percent	9. best
10. technique	11. size	12. task

it is formed by a set o generic concepts as *table*, *paper*, and *work*. On the other hand, *Clusters 1* and *2* represent areas related to image processing and computational vision; then, *Cluster 3* is more focused on neural networks and machine learning techniques. At last, *Cluster 4* contains keyphrases related to optimization and result analysis. These results show that despite the AI knowledge graph being segmented in only 5 clusters, this clusterization could divide the network into different topics, which can be represented by automatically identified keyphrases.

3.5 Discussion

The task of understanding relationships among concepts in an area from the reading of scientific articles remains a very challenge 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 the SciKGraph to represent two datasets containing academic documents from different areas and observed the same tendencies in all experiments performed. This implies that the proposed structure can be used to represent different areas expecting similar results. Considering this, we suggest the usage of 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, instead of only their meta-data 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 clusterization 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 the results produced and the computational cost, we recommend the usage of 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 a high segmentation. This was impacted by the $threshold_{edges}$ metric, which can be increased to improve the knowledge graph modularity. We found that the higher this threshold is, the more information is lost during the clusterization pre-processing. On this basis, we recommend increasing this metric as minimum as possible to keep adequate modularity of the obtained clusters.

In the framework, we found that it is possible to describe the topics (identified clusters) based on keyphrases defined based on degree-centrality algorithms executed over the clusters. However, because of the amount of noise data, we recommend to execute 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 the users' experience in understanding the processed documents and the identified topics. Still, our current solution is accurate enough to validate the structure and to give researchers a broader view over a scientific area.

3.6 Conclusion

The amount of scientific information produced has hugely increased throughout the years. This impacts researchers' lives, which have to be up-to-date with discoveries and relevant papers related to their areas of expertise. They must invest much time performing groundwork, but during this process, a significant portion of this time is wasted on unrelated topics 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 the natural language texts from input documents instead of meta-data and citations information to structure knowledge. This approach considered every concept from the studied field, clustering them into topics. The results of our conducted experimental analyses achieved up to 84% of accuracy in identifying the areas of input documents. This finding shows that even not being constructed to segment the knowledge graph in pre-defined areas, it could successfully segment the areas in topics without using annotate 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. For obtaining optimal results, we recommend to use our framework interactively, by taking the threshold values as suggestions to assist users. They can manually fine tuning the recommended thresholds and keyphrases to represent the identified topics better. 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 promote a simplified and representative visualization of the knowledge graph.

Chapter 4

Understanding the evolution of a scientific field by clustering and visualizing knowledge graphs

4.1 Introduction

The process of studying a scientific field or to do a literature review is arduous for both newcomers and expert researchers in a specific area. The increasing amount of publications [5] and, consequently, the coming era of big scholarly data have aggravated this issue, making it more laborious for researchers to search for relevant documents related to their studies [62]. Therefore, they usually go after articles reviews to help them understanding how the field under study is organized. After familiarizing themselves with the state of the art of the studied field, researchers must maintain themselves updated reading novel findings published in prestigious conferences and journals.

Both the processes of studying a new scientific field, or staying up-to-date with fresh discoveries in a previously known field are time-consuming. A common reason for this problem is that survey articles may be outdated or even do not exist for a desired area. Consequently, researchers have to spend a significant part of their research-time reading content, sometimes irrelevant to their studies, until finding the right articles related to their investigations. When keeping themselves updated with fresh discoveries, researchers do not have time to read all newly published articles in their areas.

Recent investigations have developed mechanisms to understand the structure and the evolution of scientific fields. Most of those approaches make inferences based on manually annotated data or metadata information from articles that belong to the studied field[55][24]. However, these characteristics limit the understanding of a field, as they are unfeasible on large-scale or do not consider the semantics of content from its articles. In this context, Tosi and dos Reis [58] proposed SciKGraph, a framework to structure a scientific field as a knowledge graph (KG) in a concept level, considering a KG a structure that integrates information into a knowledge base and applies a reasoner to generate new knowledge from it. Instead of representing a scientific field using only metadata information, SciKGraph uses the concepts extracted from texts on academic articles to represent

and structure it. SciKGraph clusters the constructed knowledge graph by identifying the main sub-areas of the studied field, their relation and the concepts belonging to them. Still, their framework only represents a fixed time-period of the studied field and does not allow us to analyze its evolution over different periods of time.

It is not a trivial task the adequate evolution tracking of a scientific field considering how its structure evolved semantically, examining its concepts, not just its metadata. The study of the evolution of a scientific field at a concept level requires investigating how to track changes in its structure, which is further challenging than only detecting and representing the concepts. Considering that not just the structure itself, but all concepts that shape it can be modified, it is hard to determine if some part of the structure changed, was replaced, or suffered from noise data. For example, one could determine that an sub-area a_1 did not evolve to a_2 because they have only 30% of their concepts in common. However, 90% of the time, a_1 is referred to by one of those common concepts. Thus, the other 70% of not similar concepts, despite being part of a_1, caused a miss classification and should have been considered noise.

In this article, we develop an approach to track the evolution of a scientific field at a concept level. We first define the scientific field periods from which we shall track the evolution. Then, we use SciKGraph [58] to construct knowledge graph representations of those periods, clustered in their main sub-areas. Next, our solution calculates the similarity of all clusters from distinct knowledge graphs. Our method is suited to identify clusters from distinct knowledge graphs that represent the same sub-area of the scientific field. Finally, after determining which clusters represent the same sub-area, we compare them and track their evolution, identifying concepts added or excluded from the analyzed sub-area between the periods analyzed.

In addition, with the aim of computationally assisting researchers understanding how a scientific field organized in a concept level evolves, we provide an application software to facilitate this process. Our solution is a web application designed for researchers without programming skills or background knowledge in the area. Our application contains graphical interfaces to assist users to represent a scientific field as a knowledge graph, analyze its characteristics, and track its evolution. All these features allow graphical representations of the scientific field.

We evaluate the proposed method and application software in two scientific fields from distinct knowledge areas. The first one is the Artificial Intelligence scientific field, represented using 1,002 academic articles. Based on the proposed approach, we found the possibility of tracking the evolution that occurred on the "Image Analysis" and the "Neural Network" sub-areas from 2006. As an example identified with our tool from this period, we observed a drop in "Image Analysis" articles using supervised learning; also, the increase of popularity of "Convolutional Neural Networks". We also explore our tool to analyze the Biotechnology scientific field, represented by 8,964 abstracts. We structure the input textual documents in knowledge graphs and analyze their evolution by comparing articles published between 2014-2016 and 2018-2020. As an example, this analysis allowed us to detect via the software tool that researches using mice to study BRAF-mutation lung cancer decreased.

Our proposal and software application contributes to assist researchers in understand-

ing how a scientific field is organized on a concept level. In addition, our solution enables to track and understand the evolution of the identified topics (represented as clusters) in different knowledge graphs. Our software application provides graphical visualizations for the analyzed fields, their sub-areas, and their relations.

The remaining of this article is organized as follows: Section "Background" introduces background studies. Section "Tracking the evolution of a scientific field" presents the proposed method to track the evolution of the scientific field. Section "Software tool for evolution analysis" fully describes our constructed and available to use application software. Section "Experimental Resuls" reports on our experimental analyses and discusses the obtained findings. Section "Conclusion" concludes this article with our final considerations.

4.2 Background

Scientific knowledge has been structured differently over the past years. Nowadays, publishers and researchers usually use one of four methods to this end — citation networks, manual assignment, classification techniques, or knowledge graphs. Citation networks study the topology of the citations among papers to determine how to structure scientific knowledge [55]. Manual assignments depend on experts to determine how to segment the areas and sub-areas of a scientific-field, like the ones in [51]. Classification techniques determine in which sub-area of a scientific field a document belongs [31], usually considering the sub-areas previously identified by citation networks, or manual assignments. Knowledge graphs structure a scientific-field based only on the textual content of its articles, using them to segment the scientific-field sub-areas [58].

Researchers use some of those methods to identify how the structure of scientific-fields changes over time. Citation network methods are common for this end. Jung and Segev [26], for example, studied not only the evolution of a scientific-field, but inferred future changes in it. Hopcroft *et al.* [24] proposed to track evolving communities in citation networks. They identified groups of communities, also known as covers [33], from citation networks built with documents from distinct time-periods. Then, they classified if clusters from distinct covers were similar enough to be representing the same sub-area. If they were, they could compare those clusters and track sub-areas' evolution.

The use of the Normalized Mutual Information (NMI) metric has been used to train classification methods, or to identify if a clusterization algorithm effectively segmented a citation network in its main sub-areas. This allows comparing the cover generated by the analyzed algorithm with a baseline cover. NMI is a metric that varies from 0 to 1, which evaluates how close two covers are between themselves. The closer the NMI value is to 1, the more similar the covers are. Tanmoy Chakraborty and Abhijnan Chakraborty [9] adopted this approach in which they used a variant of the NMI metric, optimized for overlapping clusters, to evaluate their algorithm results identifying overlapping clusters in citation networks.

This and other citation network approaches study only the metadata of the articles. The tracking of changes concerning the structure of a scientific-field based on its content

remains an open research challenge. If one wants to identify how the structure of a scientific-field evolved in a concept level, the usage of metadata-dependent-only techniques is not adequate. On the other hand, review articles (manual assignment method) could be a solution to this issue. Though many scientific fields do not have review articles, or they are out-dated.

The usage of knowledge graph approaches to represent scholarly data and scientific knowledge is very recent in literature. The study conducted by Vahdati *et al.* [61] tackled the problem of knowledge discovery in scholarly knowledge graphs. They used a knowledge-driven framework able to unveil scholarly communities for the prediction of scholarly networks. Results observed from their evaluations suggested that exploiting semantics in scholarly knowledge graphs enables the identification of previously unknown relations between researchers. However, to the best of our knowledge, there is no work in literature aiming to analyze the evolution of knowledge graphs in the context of scientific knowledge.

Tosi and dos Reis [58] proposed SciKGraph, a knowledge graph framework to structure and represent a scientific-field. SciKGraph receives as input a compendium of textual documents used to represent a scientific-field. Then, SciKGraph identifies and disambiguates the concepts of those documents, and uses them as vertices of a co-occurrence knowledge graph. Next, by clustering the graph, the framework identifies the main sub-areas of the scientific-field. Finally, SciKGraph extracts key-concepts from the knowledge graph representing its topics. Tosi and dos Reis evaluated their framework using collections of documents from distinct fields of science, obtaining satisfactory results. This indicates that SciKgraph can be used to represent scientific knowledge independent from its field. Nevertheless, SciKGraph requires novel features to compare and compute similar clusters as well as to enable identifying change operations from one KG to other.

Our exploratory literature review indicates that there is no current method that can track how a scientific field structure evolves in a concept level. In this investigation, we propose a knowledge graph method to track the evolution of a scientific field. We base our method on the SciKGraph framework [58], and combine techniques usually applied to citation networks to track the evolution of our knowledge graphs.

4.3 Tracking the evolution of a scientific field

This section describes how to structure a scientific field to track its evolution over time. The first subsection introduces how we use SciKGraph to represent a scientific field as a knowledge graph and extract its main sub-areas. The second subsection describes our approach to identify similar sub-areas in different knowledge graphs and how to use this information to track sub-areas' evolution.

Figure 4.1 presents our proposed methodology to track the evolution of a scientific field. First, it receives as input two sets of documents used to describe distinct periods of the same scientific field, 2010 and 2015 (as the example used in Figure 4.1). Next, it uses SciKGraph to construct a knowledge graph for each one of the input sets and extracts their main sub-areas (clusters). Finally, it compares the two knowledge graphs by identifying

common sub-areas in both structures. The output refers to the dissimilarities between detected sub-areas in the different knowledge graphs. These dissimilarities represent what changed in the sub-areas during the time-window between both sets of documents (represented in bold and dotted elements in Figure 4.1). This enables describing concepts added or excluded from an sub-area during a time-window, which we describe here as the sub-area evolution.

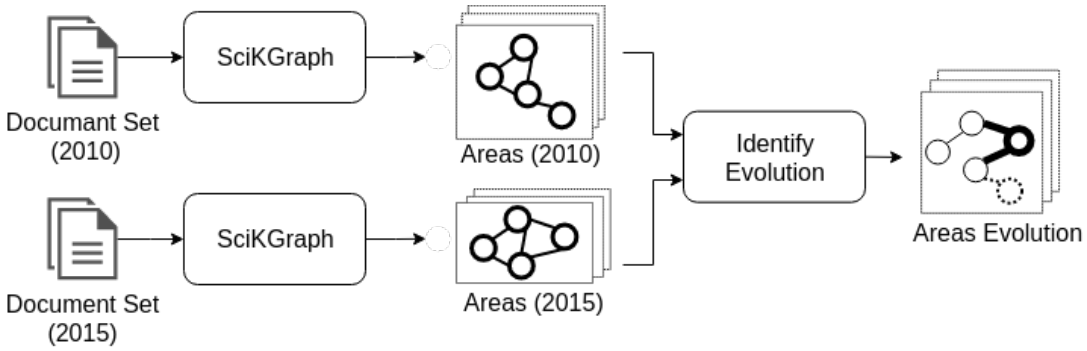

Figure 4.1: Methodology to track the evolution of a scientific field

4.3.1 Representing a Scientific Field as a Knowledge Graph

The representation of a scientific field as a knowledge graph enables the study of knowledge at a concept level. This allows users to understand concepts, their relations and structure. We represent scientific fields as knowledge graphs to track their evolution. To this end, we use the SciKGraph framework [58] (cf. Figure 4.2). First, SciKGraph receives as input a collection of scientific documents used to represent a scientific field. This is used to construct a knowledge graph that represents this collection. Second, SciKGraph clusters the constructed structure by identifying the scientific field's main sub-areas. Finally, SciKGraph extracts knowledge from the structures previously constructed.

SciKGraph performs the "Knowledge Graph Construction" task, which takes as input a collection of scientific documents used to represent a scientific field, which in our case we use to represent a time-period of this scientific field. This task parses those documents, excluding citations, images, and equations. Then, it uses Babelfy [38], a text disambiguation software to perform word sense disambiguation - the task to computationally find the contextual meaning of words [40] - and link corresponding concepts between our documents and BabelNet[41], a multilingual semantic network. This operation returns concept babel synsets, which are their unique identification codes.

Next, this task constructs the knowledge graph by using the identified corresponding concepts as vertices and their co-occurrence in the text as undirected edges, both weighted based on the number of times they appeared in the text. Figure 4.3 presents an example, by receiving as input the phrase "Knowledge graphs can structure knowledge [72].". The solution parses it excluding a citation (pre-processing); identifies the "knowledge graph", "structure", and "knowledge" as correspondent concepts with BabelNet; constructs the knowledge graph taking the three concepts as vertices and their direct co-occurrence as edges, weighted based on the number of times they appeared in the whole collection. We

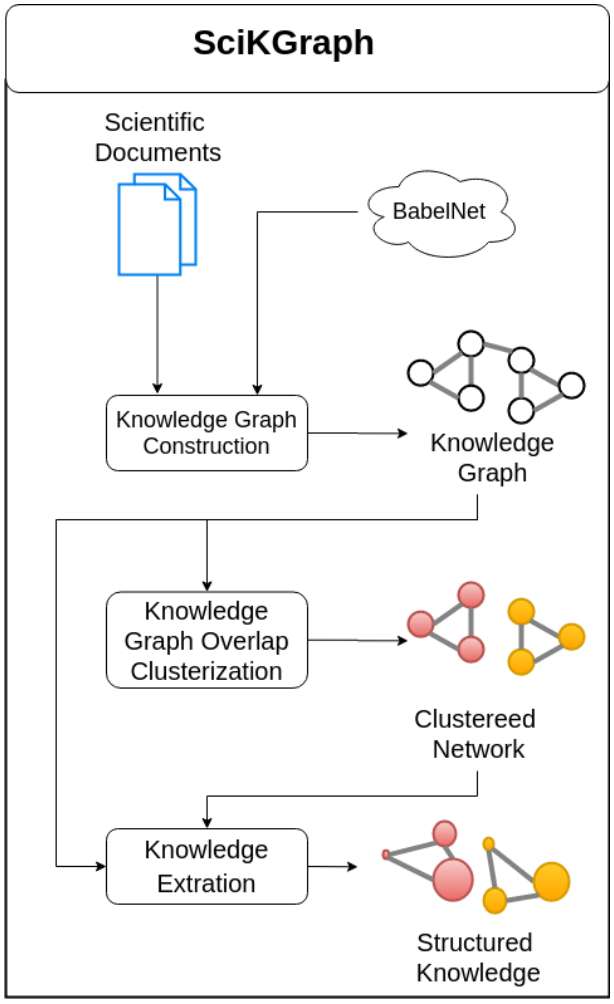

Figure 4.2: SciKGraph framework [58] to structure a scientific collection as a knowledge graph.

note that if those concepts or their co-occurrences appear again in other documents of the collection, their weights in the knowledge graph are updated considering these events.

SciKGraph performs the "Knowledge Graph Overlap Clusterization" task (cf. Figure 4.2), which takes as input the knowledge graph previously constructed and clusters it, identifying overlapping sub-graphs, representing the scientific field sub-areas. The idea behind this task is that concepts that belong to the same sub-area co-occur more than those from different ones because they tend to appear more times in the same context. Therefore, as clusterization techniques divide elements into groups that are more correlated with themselves than with others, SciKGraph states that when clustering a knowledge graph with edges weighted based on concepts co-occurrence, the clusters identified represent distinct sub-areas of the field the knowledge graph represents. Moreover, as a concept can belong to multiple sub-areas simultaneously, it is an intrinsic part of the problem to enable overlapping representations of sub-areas.

In the second step, to cluster the knowledge graph, SciKGraph uses the OClustR algorithm [49]. It is a graph-based clusterization algorithm that automatically identifies the number of overlapping clusters on which it divides the knowledge graph. The user does not need to input the number of sub-areas to organize the scientific field because OClustR

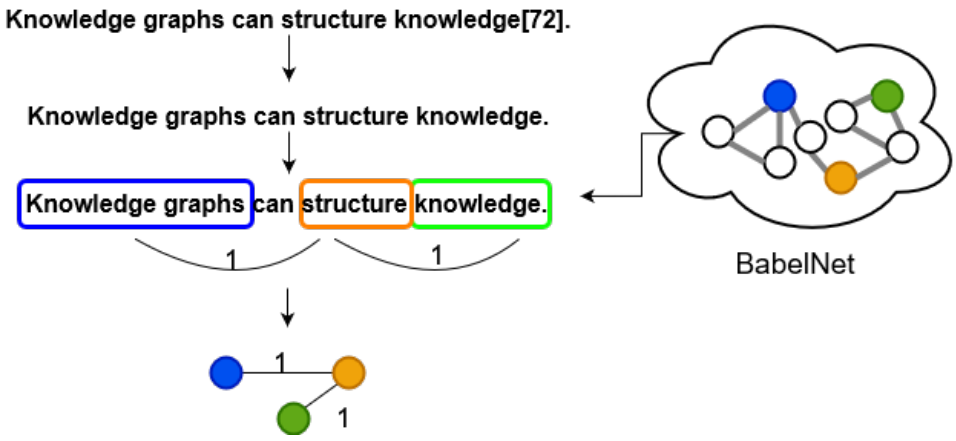

Figure 4.3: Example of generating a knowledge graph from an input sentence.

identifies it automatically. However, if the user desires to fix a number and reduces the number of clusters identified, SciKGraph proposes three agglomerative algorithms for this task [58], which we do not use to track the evolution of the identified sub-areas.

In the third step, the "Knowledge Extraction" task organizes and extracts knowledge from the knowledge graph, and its clusters (sub-graphs). It presents the sub-areas identified through the clusterization process and identifies their adequate labels using the C-Rank algorithm. C-Rank [57] is a non-supervised keyphrase extraction technique that identifies keyphrases from scientific documents receiving as input from the user only the document from which it has to extract the keyphrases. At this stage, keyphrases are not relevant for the identification of similar sub-areas in distinct knowledge graphs, nor to identify their evolution over time. Therefore, we do not further detail their extraction in this work.

Knowledge Graph parameters

To optimize the clusterization, SciKGraph pre-processes the knowledge graph. This pre-processing contains three steps: (1) it removes edges weighted below a certain $threshold_{edges}$, reducing noise; (2) it excludes vertices with higher degree centrality that are inside a $threshold_{centrality}$, which eliminates general concepts that are relevant for the scientific field as a whole, but are too general to be relevant for a specific sub-area; (3) it discards small disjoint sub-graphs created after the previous steps.

In the performance of the pre-processing, two variables must be defined $threshold_{edges}$ and $threshold_{centrality}$; both of them depend on the domain and the size of the input document collection. When one increases $threshold_{edges}$, the clusters further produced are better defined, but contain less information, decreasing the number of vertices in the knowledge graph. By reducing $threshold_{centrality}$, the size of the identified clusters increase, but the number of clusters identified decreases. In order to obtain optimal values, SciKGraph, based on the graph structure, recommends an interactive approach. This approach presents estimated values to the user, which can change them based on the observed generated output.

In this investigation, our objective is to track the evolution of a scientific field by

comparing the structure of its knowledge graph representations in distinct periods. In this sense, the closer the knowledge graph representations are to each other, the less noise to identify the evolution of its sub-areas will be. Thus, we acknowledge the fact that the $threshold_{edges}$ is related to the number of vertices and edges in the knowledge graph, and the evidence that the number of vertices influences the structure of the identified clusters [58]. On this basis, we shall set $threshold_{edges}$ values that result in knowledge graphs with a similar number of vertices.

In our analyses, we observed that representations of the studied scientific fields containing around 1,600 concepts held enough information to be further processed and generated clusters with high modularity. Therefore, we examined which $threshold_{edges}$ values we would need to apply to our representations to produce knowledge graphs with this amount of vertices. Based on these values, we identified a power series relation between the $threshold_{edges}$ and the number of edges in the knowledge graph ($|edges|$) (cf. Figure 4.4).

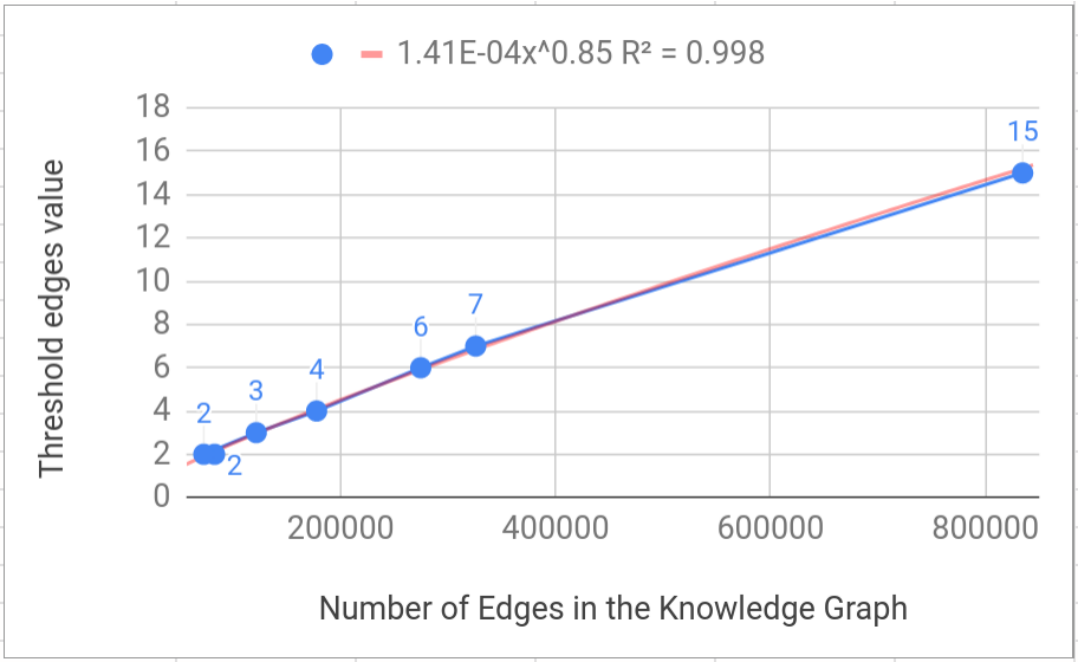

Figure 4.4: Power series relation between the number of edges and the $threshold_{edges}$ value to generate knowledge graphs with the same amount of vertices.

Figure 4.4 shows a chart with the relation between the number of edges of knowledge graphs ($|edges|$), and the $threshold_{edges}$ values we determined to pre-process them in order to obtain their representations with around 1,600 concepts each. This relation does not expect a constant value and is powered to 0.85. Moreover, setting the coefficient of the power series to 0.00014, we obtained the approximate number of vertices we expected. However, one can increase or decrease this coefficient to respectively raise or reduce the number of vertices of the representations in order to identify more specific or general sub-areas of the studied scientific field.

Therefore, considering this relation, when tracking the evolution of a scientific field, we use Equation 4.1 to determine $threshold_{edges}$ value, maintaining a similar amount of

concepts in distinct scientific field representations. However, the equation coefficient is domain-dependent, and, as explained previously, can vary based on the scientific field analyzed and the number of articles used to represent it.

$$threshold_{edges} = |edges|^{0.85} * 0.00014 \tag{4.1}$$

4.3.2 Identifying dissimilarities between Knowledge Graphs

We represent two time-periods of the same scientific field as distinct knowledge graphs. Our goal is to track the evolution of this scientific field during the time-window between both representations. The identification of concepts and connections that were added or excluded from the knowledge graphs during the analyzed time-window would give the researcher a shallow perspective of the evolution of the scientific field. Therefore, to improve the researcher experience, instead of analyzing only the knowledge graph, we track the evolution of the scientific field by identifying the concepts added and/or excluded from each of the scientific field sub-areas (knowledge graph clusters).

In order to track the evolution of a cluster comparing distinct covers, it is necessary to identify the corresponding clusters in both covers. Figure 4.5 presents an example to illustrate the comparison of clusters from distinct knowledge graphs. Even though both covers have three clusters each, it would be inaccurate to compare the clusters labeled with the same numbers; this comparison would result in total dissimilarity between the covers. It requires further considering the meaning of concepts present in the clusters.

In our approach, we analyze the content of each cluster in both covers, so that we may identify that "Cluster #1" from "Cover 1" and "Cluster #3" from "Cover 2" are corresponding clusters in our example (corresponding clusters linked by dotted lines in Figure 4.5). By comparing the elements of the clusters, we can identify that both covers have the same structure, they only had their clusters sorted differently.

We use a similarity measure to determine the correspondence ratio between two clusters c_1 and c_2. It is based on the $match(c_1, c_2)$ equation used by Hopcroft *et. al* in [24] (cf. Equation 4.2), which quantifies how well two clusters match with each other, computing the number of intersection elements between the clusters; and normalizing it by the size of the bigger cluster. We chose this measure because its definition ensures that for clusters to have high similarity, they must roughly have the same size. Therefore, broader clusters will not have a high correspondence ratio with small ones just because they also contain almost all the elements that the small clusters have.

$$match(c_1, c_2) = min(\frac{|c_1 \cap c_2|}{|c_1|}, \frac{|c_1 \cap c_2|}{|c_2|}) \tag{4.2}$$

However, instead of giving the same importance to all concepts in the clusters, our similarity measure weights the concepts based on their degree centralities in the clusters sub-graphs. Therefore, when analyzing a concept n, we consider its weight $weight(n)$ as the average of its degree centrality $central_c(n)$ in the sub-graphs c that it belongs (cf. Equation 4.3).

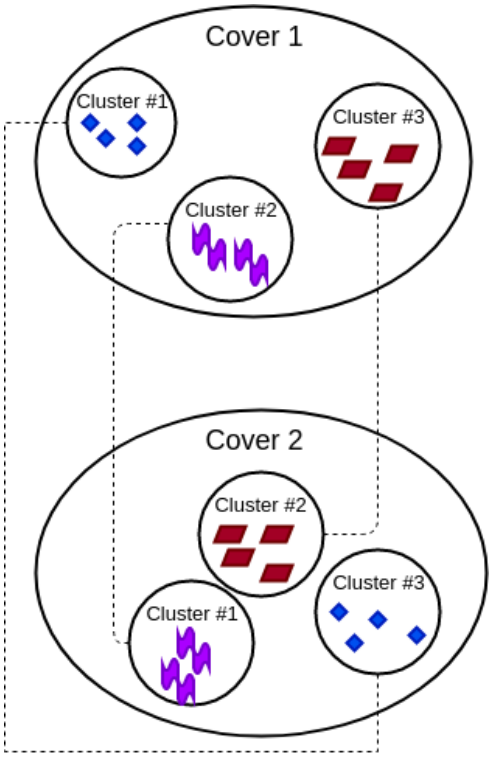

Figure 4.5: Example of covers comparison; correspondent clusters from distinct covers are linked by dotted lines.

$$weight(n) = \begin{cases} central_{c1}(n), & \text{if } n \in c1, n \notin c2 \\ central_{c2}(n), & \text{if } n \notin c1, n \in c2 \\ \frac{central_{c1}(n)+centrality_{c2}(n)}{2}, & \text{if } n \in c1, n \in c2 \end{cases} \tag{4.3}$$

Furthermore, considering the weight of each concept to calculate the similarity between two clusters, our similarity measure $similarity(c_1, c_2)$ is defined as in Equation 4.4. That way, we reduce the relevance of miss classified noisy concepts by using a weighted function based on the centrality of the concepts in the sub-graphs to which they belong.

$$similarity(c_1, c_2) = min(\frac{\sum_{n \in c_1 \cap c_2} weight(n)}{\sum_{n \in c_1} weight(n)}, \frac{\sum_{n \in c_1 \cap c_2} weight(n)}{\sum_{n \in c_2} weight(n)}) \tag{4.4}$$

After defining how to calculate the similarity measure between two clusters, we can identify the more similar ones. To this end, we calculate the similarity between every pair of clusters from different covers. In this procedure, each cluster from one cover has a certain similarity value with all clusters from the other cover. At the final stage, we select the most similar pairs of clusters, which are those that have the similarity value above the $threshold_{similarity}$ value. This threshold value varies from 0 to 1 and determines to which extent the similarity between two clusters must be to consider them correspondent, or related.

Figure 4.6 presents an example. Considering that all concepts in the example have the same degree centrality, “Cluster #1” from “Cover 1” has 0.75 of similarity with “Cluster #2” from “Cover 2”, and 0.25 with “Cluster #1” also from “Cover 2”. Therefore, defining

a $threshold_{similarity}$ (0.5 for example), one may observe that "Cluster #1" from "Cover 1" and "Cluster #2" from "Cover 2" are correspondents. By reducing this $threshold_{similarity}$ number to 0.25, besides the correspondent clusters, one may also observe related ones, as the cluster from "Cover 1" and "Cluster #1" "from "Cover 2". These relations, for example, can describe sub-areas that were merged, or split themselves in distinct ones or have overlapping concepts shared between themselves.

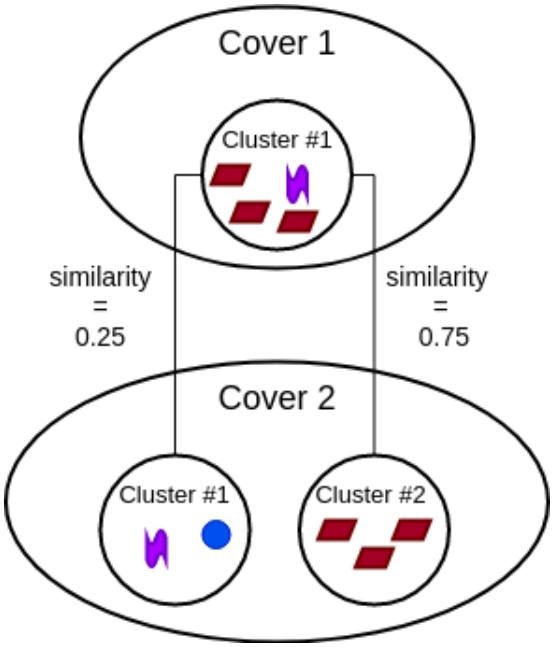

Figure 4.6: Example of similarities among clusters distinct covers.

It is further possible to quantify the similarity of the knowledge graphs and their clusters as a whole. However, it is not simple to compare covers that allow overlapping clusters, and we cannot apply a procedure as used to compare just a pair of clusters. We use a metric proposed by McDaid, Greene, and Hurley [33] to quantify the similarity between the whole covers generated from both knowledge graphs. It is a Normalized Mutual Information metric used to evaluate overlapping clusterization algorithms. Usually, Mutual Information metrics calculate the similarity between a cover of clusters generated by the algorithm it is evaluating, and the control cover, containing correct segmented clusters. In this work, instead of using this metric to evaluate a clusterization algorithm, we use it to compare the covers obtained from clustering the two generated knowledge graphs. By calculating the amount of mutual information between the covers, we can use this normalized value to inform the researcher of the similarity between the generated covers, or, in other terms, the similarity of the scientific field in the analyzed time-periods.

4.4 Software tool for evolution analysis

Our method proposed to track the evolution of a scientific field based on SciKGraph. It allows the analysis of the most diverse scientific areas, identifying concepts added and/or excluded from sub-areas of the studied scientific field. The Jupyter Notebook

[29] interface proposed in SciKGraph limits its usability to researchers with at least a minimum programming knowledge. This limitation negatively impacts the usage of the framework for a significant part of the scientific community. At this stage, our goal is to enable the usage of our method by researchers non literate in programming. For this purpose, we developed a software tool with an user graphic interface. Section "Defined features" presents the software application, its features and interfaces; Section "Technical details" specifics the technologies applied to its construction.

4.4.1 Defined features

Our developed software tool enables the user to examine the evolution of the desired learning field by structuring and analyzing scientific knowledge as proposed by SciKGraph. In this sense, we segment the functionalities into three primary actions in the software.

1. "Create", in which the user creates the knowledge graph representation of the desired scientific field and identifies its main sub-areas.

2. "Analyze", devised to enable the user to obtain quantitative metrics of the previously generated structure and extract knowledge from it.

3. "Track Evolution", in which the user can compare previously generated scientific field representations and track the evolution and similarities between those (the main contribution of this article).

The "Create" interface (cf. Figure 4.7) assists researchers to represent a scientific field as a knowledge graph. To do so, the user input a collection of textual documents that represent the desired scientific field, the language of the documents, and a Babelfy key - obtained by registering in the Babelfy website[1]. In the sequence, the software builds and plots the constructed knowledge graph. In order to better understand the topology, the user can pre-process and cluster it, identifying the main sub-areas of the analyzed scientific field. The pre-processing step consists of choosing the most generic vertices of the graph through a threshold or a list. After defining that, the software can cluster the knowledge graph, identifying and plotting the main sub-areas of the scientific field.

The "Analyze" interface (cf. Figure 4.8) enables the user study the sub-areas previously identified. First, if the researcher assumes that the application identified too many sub-areas, (s)he can choose the number of sub-areas (s)he would consider optimal. Then, our software application uses an agglomerative method [58] to merge the sub-areas until it reaches the number chosen by the user. Moreover, to evaluate how well-segmented are the defined sub-areas, the user can calculate the modularities of the knowledge graph and its sub-areas. Our tool lists the keyphrases of the knowledge graph and the identified sub-areas to assist in the scientific field analysis. At last, the researcher can plot a cluster relation graph to understand how those sub-areas are related among themselves.

The "Track Evolution" interface (cf. Figure 4.9) enables researchers to analyze how sub-areas of a scientific field have modified over time by identifying concepts added and/or

[1] `http://babelfy.org/login`

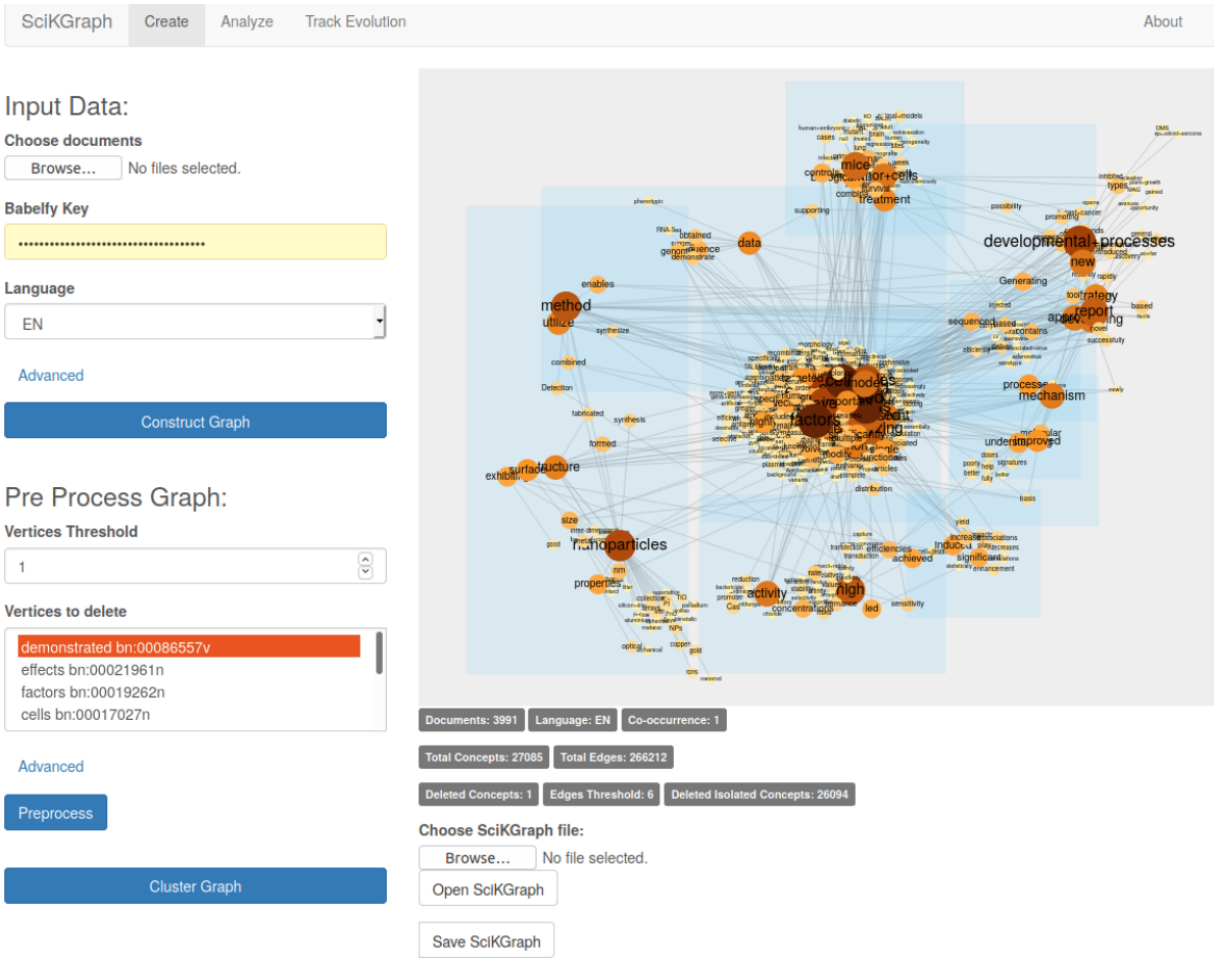

Figure 4.7: "Create" interface used to represent a scientific field as a knowledge graph. This allows to cluster the graph by extracting its main sub-areas.

excluded from them. The user needs to input two knowledge graphs representing the scientific field time-periods for comparison. In the tool, those knowledge graphs are generated by the "Create" feature by inputting only academic articles inside the time-period to be represented. The tool supports the importing of already generated knowledge graphs. As an example, to track the evolution of a Biotechnology area between 2015 and 2019, the user creates a knowledge graph to represent each of those years. One knowledge graph is created by receiving as input biotechnology articles published in 2015, and the other one biotechnology articles published in 2019.

After the knowledge graphs are available, the researcher can compare the similarity between the whole covers, identifying the amount of concepts added and/or excluded from the scientific field clusters in the time-window between the time-periods analyzed. Our software tool can identify correspondent clusters from distinct covers, based on the "Similarity Threshold" configured by the user. Those correspondent clusters are essential because they represent the same sub-areas in different time-periods. Consequently, comparing those, the application can identify how the sub-areas evolved, presenting to the user concepts that were removed, added, or maintained in a specific sub-area. The application provides a graph visualization of this evolution, as presented in the screenshot

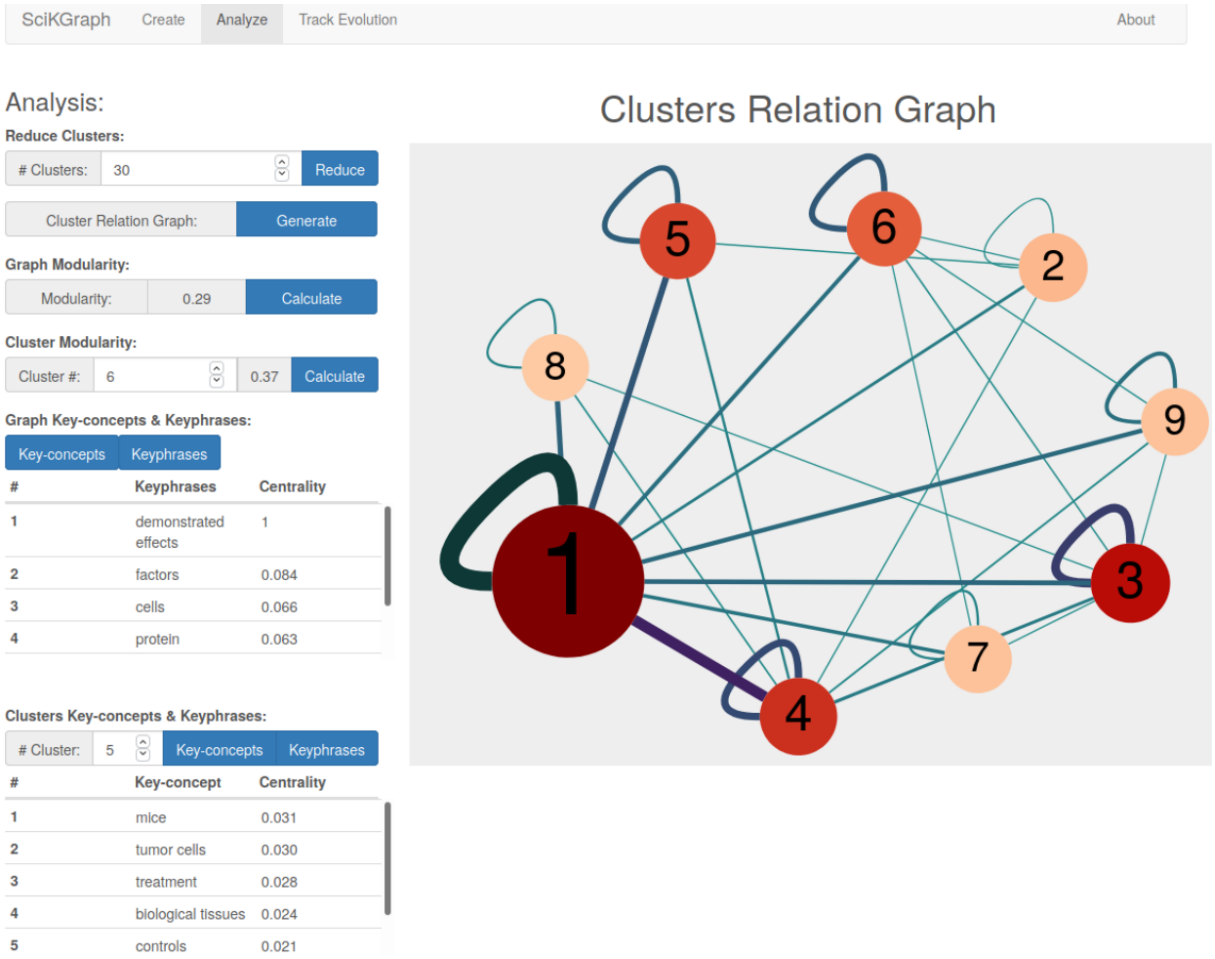

Figure 4.8: "Analyze" interface in our tool used to extract knowledge and presents quantitative metrics from the scientific field previously structured.

of the "Track Evolution" functionality (cf. Figure 4.9) .

4.4.2 Technical details

The software application is available online[2]. It is a web application with its back-end developed in Python 3.7, web server interface in flask 1.1.1, and front-end in HTML 5, CSS 3, Bootstrap 3.3.7, and javascript 6. Furthermore, to plot the constructed knowledge graphs, we used Cytoscape [54], which is an open-source software to visualize complex networks. It has the py2cytoscape library [46] that allows us to link it with our back-end solution. In addition, it has the cytoscape.js library [46] that uses javascript to link py2cytoscape and our web pages. For the construction of the SciKGraph knowledge graphs, we use the pybabelfy library [3], which utilizes the Babelfy HTTP API [4] to link the corresponding concepts between the input documents and BabelNet.

Regarding the used algorithms, we adopted the SciKGraph framework [58] to represent a scientific field as a knowledge graph. Therefore, if the user requires to reduce the number

[2]Omitted due to ongoing blind review
[3]`https://github.com/aghie/pybabelfy`
[4]`http://babelfy.org/guide`

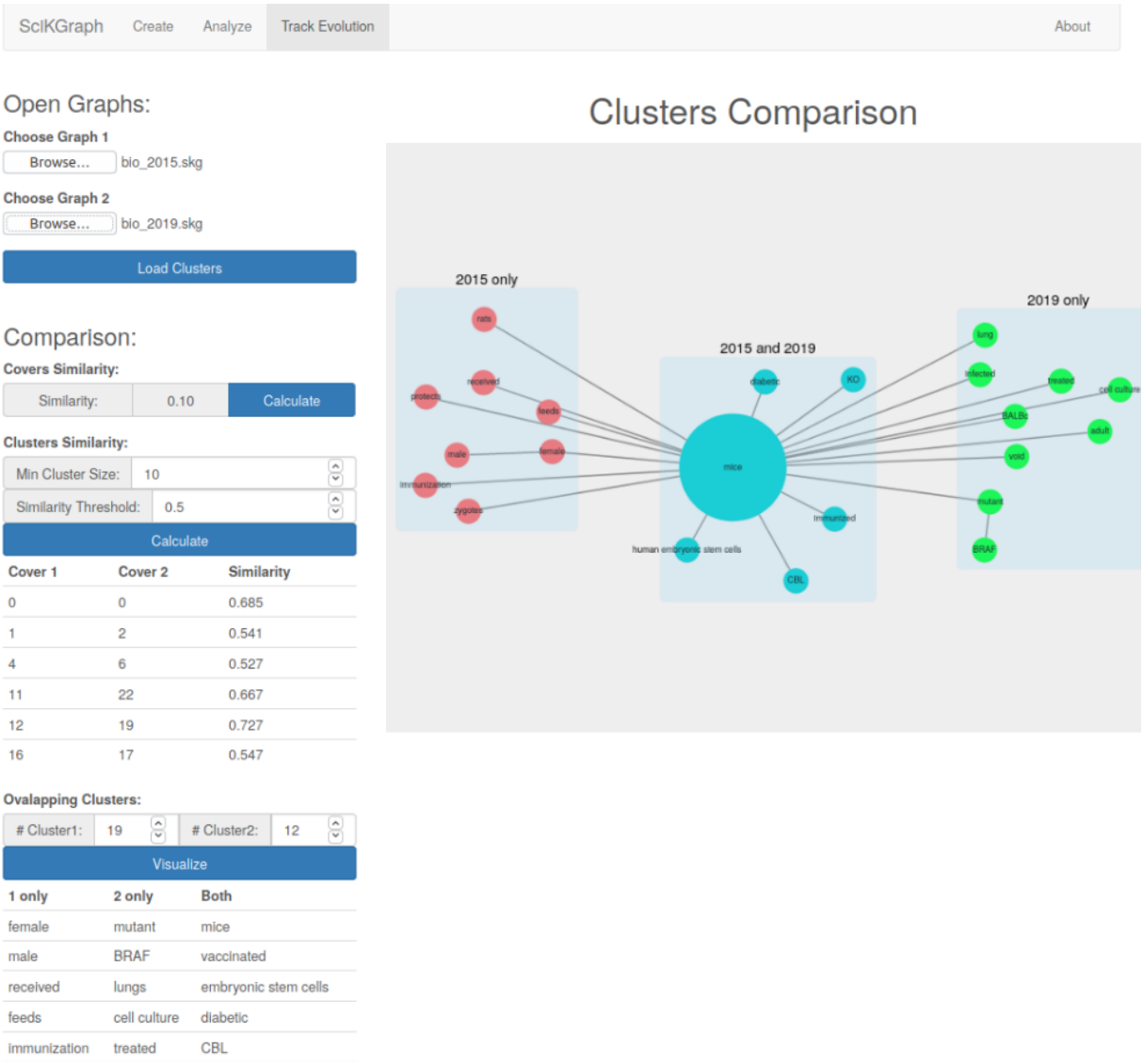

Figure 4.9: "Evolve" interface in our tool used to track the evolution of a scientific field and its sub-areas.

of clusters to n, we use the agglomerative technique recommended in SciKGraph, which selects the n bigger clusters and simulates to merge them with the smallest cluster of the cover. Then, the algorithm permanently merges the smallest cluster with the one that achieved the higher modularity in the simulation. This algorithm repeats this process until the number of clusters is equal to n. The modularity metric used in this algorithm, and over other analyses in our work, must consider the overlapping cluster. To this end, as in SciKGraph, we adopted the Q^L_{ov} metric, suggested by Chen and Szymanski [10]. Our software tool application uses the same algorithms as SciKGraph to cluster the knowledge graph and to extract its keyphrases, which are the OClustR [49] and C-Rank [57] algorithms, respectively. At last, to identify the similarity of distinct covers, we used the NMI metric proposed by McDaid, Greene, and Hurley [33]. Our tool implements a procedure considering the Equation 4.4 to identify the similarity of sub-areas from distinct covers.

Other than the developed graphical user interface (GUI) software application, our

investigation contributed in providing a python library for those who wants to automatize its usage (available online[5]).

4.5 Experimental Results

We present the application of our solution in different scenarios. Subsection "Datasets" introduces two datasets used to represent the Artificial Intelligence (AI) and the Biotechnology (BIO) scientific fields. Then, we show how the proposed application structures and illustrates them. Finally, we demonstrate how our method tracks a scientific field evolution and how a researcher can analyze those results, extracting knowledge from them.

4.5.1 Datasets

This investigation explores the AI and the BIO datasets to, respectively, represent the Artificial Intelligence and the Biotechnology scientific fields. We chose those datasets because they represent distinct branches of science. Our goal is to understand to which extent our devised methods and implemented tool can correctly track the evolution of scientific fields independently of their areas.

Tosi and dos Reis [58] constructed the Artificial Intelligence dataset (AI) to experimentally evaluate the SciKGraph. It contains 1,002 academic documents, and their publishing dates, from the Artificial Intelligence area. Those documents were crawled from the IEEE Xplorer website[6] based on a search using "Artificial Intelligence" as a keyphrase. The results were sorted based on their number of citations by other papers, and the most cited documents were downloaded. At last, our solution parsed the downloaded documents from PDF to text, generating the final dataset with full academic documents, in text format, representing the AI scientific field. To our evolution analyses, we organized AI dataset into two parts: the first one containing 481 documents published before 2006; and the second one containing 521 documents published from 2006 (2006 included).

We constructed the Biotechnology dataset (BIO) to understand how our method would perform tracking evolution in a different science branch. We developed a crawler to download the content of the dataset. Instead of downloading full documents, we downloaded only their abstract and publishing date. In this procedure, we aim to compare the obtained results taking as input full academic articles, from AI, and only abstracts, from the BIO dataset.

Our crawler automatically searched for articles in the nature website[7] selecting only "research" or "reviews" documents having the "biotechnology" subject, sorted by relevance. In order to analyze the evolution of the Biotechnology scientific area in a specific period, we downloaded 3,991 articles abstracts published between 2013 and 2015 and 4,973 articles abstracts published between 2018 and 2020. This allows use to analyze the evolution that occurred between 2015 and 2018 in the Biotechnology scientific field.

[5]Omitted due to ongoing blind review

[6]`https://ieeexplore.ieee.org/Xplore/home.jsp`

[7]`https://www.nature.com/search/advanced`

4.5.2 Case study results

We illustrate the generation and evolution tracking analysis with the use of our software application based on the two cases studies by representing the Artificial Intelligence and the Biotechnology scientific fields. Both case studies are available online[8]

Artificial Intelligence:

Using all 1,002 AI textual documents, our software application can structure and cluster the Artificial Intelligence scientific field (cf. Figure 4.10). Figure 4.10 shows the constructed knowledge graph, which researchers can use to understand how concepts within the same or from sub-areas (clusters) are related. Figure 4.10 highlights a portion of the knowledge graph to facilitate its visualization, which can be accomplished by using the zooming mechanism in the software.

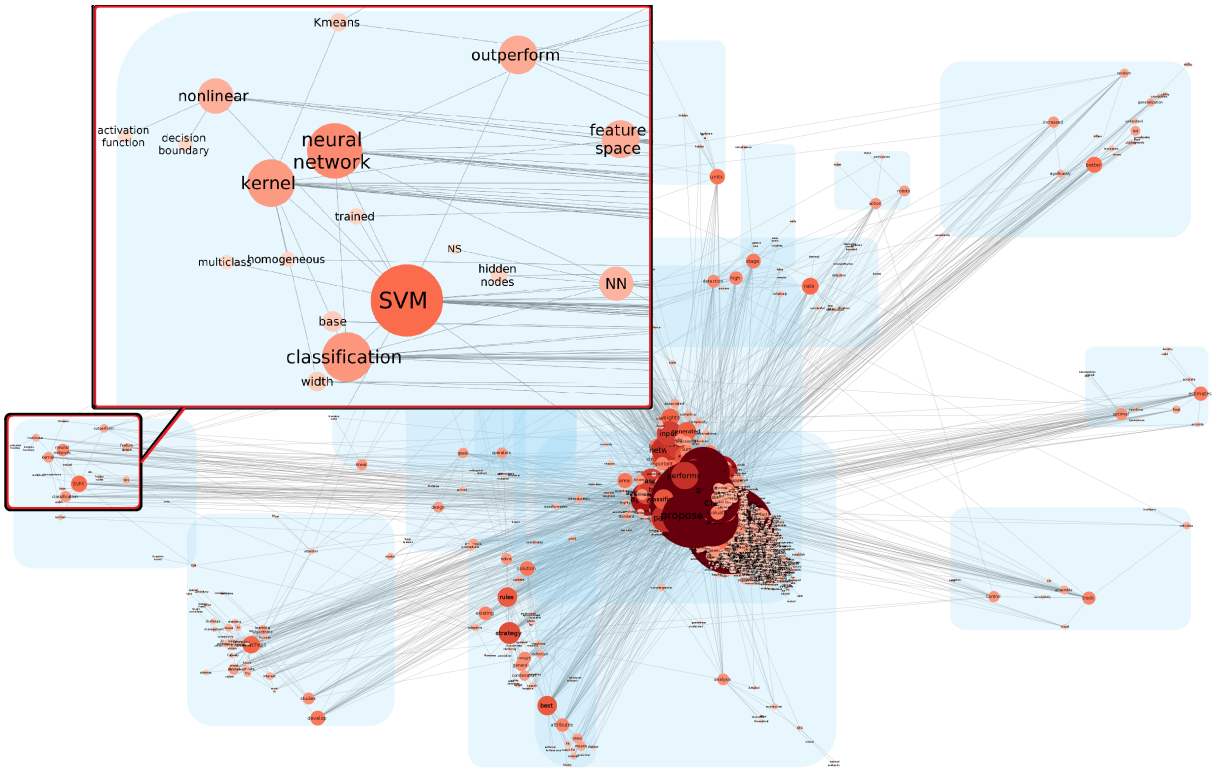

Figure 4.10: Knowledge graph illustrating the Artificial Intelligence scientific field. It highlights a region of peripheral vertices representing specific concepts related to machine learning and classification.

In Figure 4.10, the knowledge graph originally contained 40,343 concepts that the pre-processing step reduced to 2,899 before the clusterization, which identified 18 distinct sub-areas containing at least 10 concepts each. We observed that most concepts are in the same central region of the figure. This occurs with the most generic ones. On the other hand, more specific concepts are positioned at the edge of the knowledge graph; this occurs with the highlighted sub-area In Figure 4.10. It shows specific concepts related to

[8] `https://github.com/maurodlt/SciKGraph/examples`

classification and machine learning techniques, which are central to the AI scientific field. For example, if a researcher wants to understand more about "neural networks", using the visualization proposed, (s)he can observe that it is linked to "classification", "NN" (neural networks), and "trained" concepts. Moreover, it belongs to the same sub-area as "kernel", "hidden nodes", and "feature space", indicating that those concepts are related among themselves and used in similar problems.

By applying our tool, we exemplify how to track the evolution in Artificial Intelligence. We analyzed how its sub-areas change comparing two time-fixed covers of the AI field, one constructed based on articles published before 2006 and the other based on articles published from 2006. Our analysis consists first in identifying the similarities between the clusters from different covers (cf. Table 4.1). To exhibit only the most similar clusters, we set a similarity threshold and display only those clusters with more than 40% of similarity between themselves. Also, clusters with less than 10 concepts are not displayed because those contain insufficient information and would disturb the user analysis. Table 4.1 shows 13 clusters from different covers that have similarities above 40%, indicating that they can represent the same sub-areas in distinct time-periods.

Table 4.1: Similarities between Artificial Intelligence clusters from two distinct periods.

Artificial Intelligence		
# Cluster (Before 2006)	# Cluster (From 2006)	Similarity
0	0	0.586
4	1	0.458
10	10	0.414
11	3	0.415
11	6	0.557
11	8	0.401
12	4	0.666
14	9	0.416
24	32	0.815
26	33	0.450
28	16	0.427
28	37	0.493
33	38	0.514

Figure 4.11 represents clusters 4 and 1, from covers 1 and 2, respectively. It shows that the sub-area the clusters represent is centered around the "image" concept. We assume here that it represents the Image Analysis sub-area. All the concepts that both clusters have in common are related to metrics, characteristics, and domains in which Image Analysis is used, as "grayscale image", "low-resolution", and "iris". Moreover, a researcher analyzing this representation can observe not only concepts of a sub-area, but concepts removed or added to it. For example, before 2006, concepts as "training set", "labeled", and "test image" were part of the analyzed sub-area; and from 2006, concepts as "videos" and "facial" were added to it. By interpreting this information, one may understand that supervised learning - which depends on training sets, labeled information, and test sets - has been less used recently in this context. On the other hand, new domains of problems

acquired more attention in the most recent years, video processing, and facial-related analysis, for example.

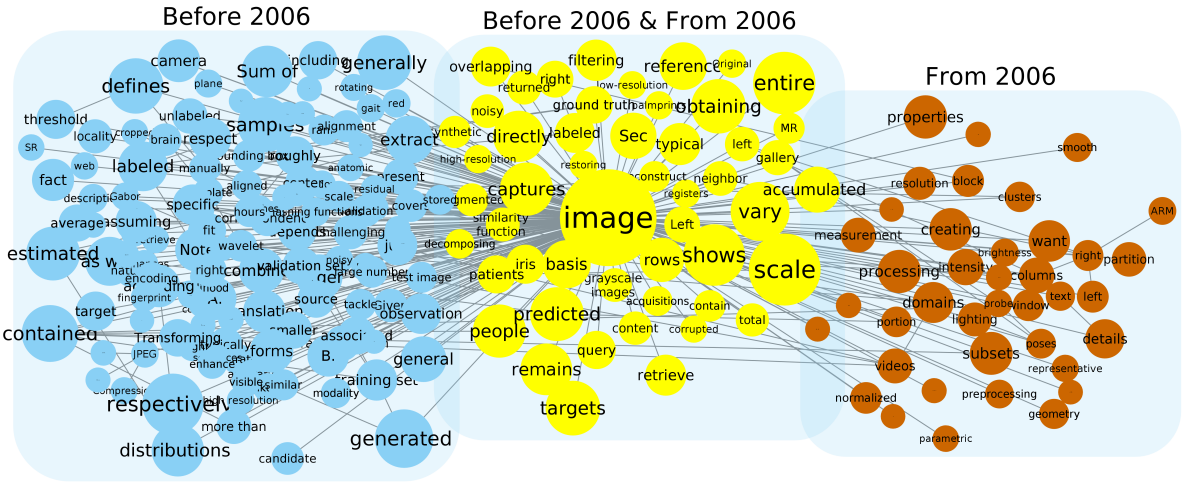

Figure 4.11: Evolution of the "Image Analysis" sub-area comparing its representation by cluster 4 from cover 1 (before 2006) and by cluster 1 from cover 2 (from 2006).

Other than that, it is also possible to analyze how two sub-areas evolved together to form a third one. Figure 4.12 illustrates this, in which the clusters 14 and 19 from cover 1 evolved to form cluster 5 from cover 2. We note that in this case, the similarity between the cluster from cover 1 and the cluster from cover 2 is smaller than the 40% we determined before. This occurs because the third cluster is formed not only by concepts from the other two clusters, but by newer concepts added to it. In this sense, cluster 5 from cover 2 has 35% of similarity with cluster 14 and 33% of similarity with cluster 19, both from cover 1. Therefore, Figure 4.12 illustrates three groups, clusters 14 and 19 from cover 1, and cluster 5 from cover 2. Those we represent by orange, blue, and yellow concepts, and we assume that they represent "Neural Networks", "Neural Networks Structures", and "Convolutional Neural Networks" sub-areas, respectively. We assumed this to facilitate the overall comprehension of the example. This assumption relies on the concepts that belong to each of those clusters.

Figure 4.12 representation allows researchers to understand how two clusters evolved into a single one. It shows all the concepts from the three clusters and links the corresponding ones. A researcher can observe which concepts the clusters share in common. For example, we mentioned that the similarity between the orange and the yellow clusters is higher than the blue and the yellow one. Though, this visualization allows us to observe that the last pair of clusters have more concepts in common between themselves. In addition, it is interesting to note how the "Convolutional Neural Networks" sub-area was created. Most of its concepts are not in the two clusters from cover 1, not because they were miss classified, but because they were not relevant enough to be part of the scientific field representation. From 2006, concepts related to neural networks and its structure were used several times together with concepts as "convolution" and "CNN", showing their relevance for the Artificial Intelligence field. Therefore, both sub-areas from

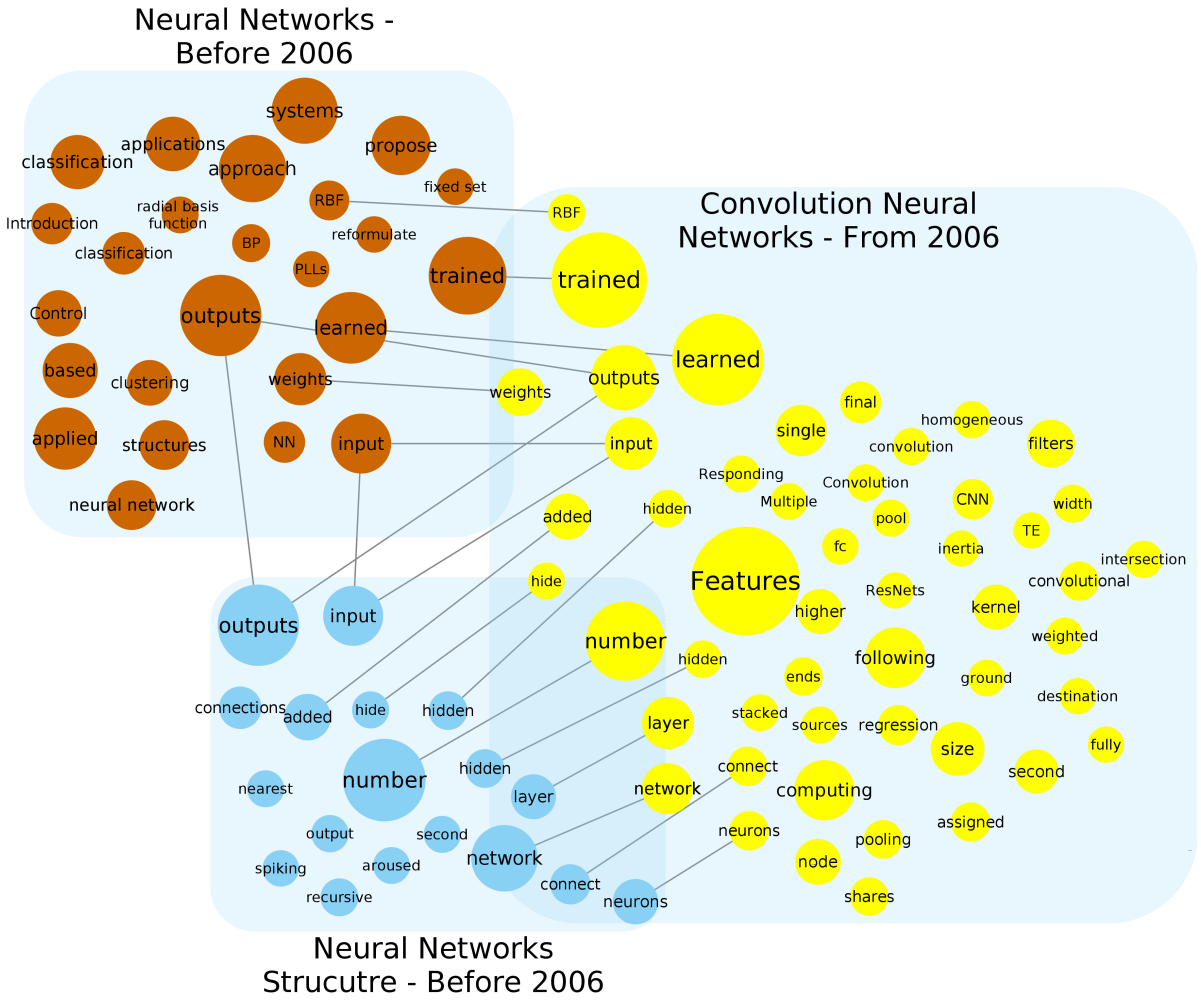

Figure 4.12: Evolution of the "Convolution Neural Networks" sub-area comparing its representation by cluster 14 and 19 from cover 1 (before 2006) and cluster 5 form cover 2 (from 2006).

cover 1, before 2006, were merged with other newer concepts, creating the "Convolutional Neural Networks" sub-area.

Biotechnology:

We used our developed software tool to analyze, structure and cluster the Biotechnology field based on the 8.964 abstracts from the BIO dataset (cf. Figure 4.13). It shows the most relevant concepts of the biotechnology field and how they are correlated. We zoomed in a region of Figure 4.13 to highlight how correlated concepts are plotted close to each other in this visualization.

Similar to the structure of the Artificial Intelligence knowledge graph representation, Figure 4.13 exhibits a central group of vertices containing generic concepts and peripheral vertices representing more specific ones. We observe this pattern in the zoomed part of

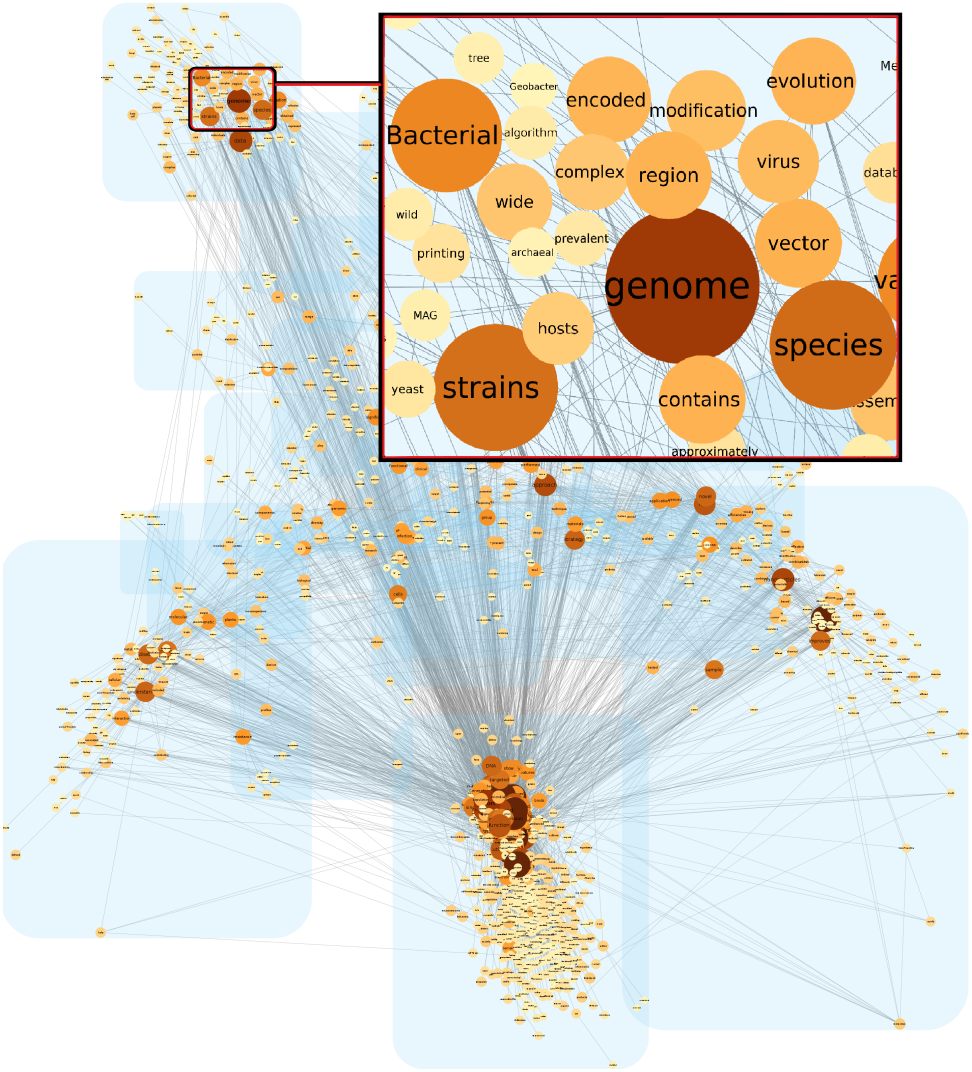

Figure 4.13: Knowledge graph illustrating the Biotechnology scientific field. It highlights a region of peripheral vertices representing specific concepts related to the microbiology sub-area.

Figure 4.13, in which biotechnology specific concepts related to microbiology, as "Bacterial", "strains", "genome", and "virus" are close to each other in the knowledge graph. Therefore, if a biologist would like to understand more about bacterial related researches, for example, (s)he could study other concepts connected to the "bacterial" and those that are in its surroundings.

In order to study the evolution of the Biotechnology field, similar to what we performed to the AI knowledge graph, we analyzed the evolution of its sub-areas comparing their representations in two distinct time-periods. In this case, our first representation contains 3,991 abstracts published between 2013 and 2015; and the second containing 4,973 abstracts published between 2018 and 2020. We are tracking the evolution of the Biotechnology field that occurred between these two time-windows. Accordingly, we represent the two time-periods as knowledge graphs, cluster both of them, and identify the similarities among their clusters. Afterwards, we set a threshold similarity to display only the most similar clusters, which was defined as 50%, a higher value compared to the one

used in the Artificial Intelligence analysis, increased to reduce the number of sub-areas displayed to the user (cf. Section "Tracking the evolution of a scientific field").Table 4.2 presents the clusters that have more than 50% of similarity with each other, and therefore, we assume that they represent the same sub-area of the analyzed scientific field.

Table 4.2: Similarities between Biotechnology clusters from distinct periods.

Biotechnology		
# Cluster (Before 2016)	# Cluster (From 2018)	Similarity
0	0	0.685
1	1	0.541
4	6	0.527
11	22	0.667
12	19	0.727
16	17	0.547

We further analyzed an sub-area, represented by a pair of clusters listed in Table 4.2 with the aim of tracking and understanding the evolution of its concepts. Figure 4.14 illustrates how cluster 12 from cover 1 changed into cluster 19 from cover 2. Figure 4.14 shows the concepts belonging only to cluster 12 colored in orange; the concepts belonging only to cluster 19 colored in blue; and belonging to both of them colored in yellow. We observed that both of these clusters are centered around the "mice" concept. We assume that they represent an sub-area of "Researches using mice", in two distinct periods. On this basis, we notice that concepts as "mutant", "BRAF" (cancer-related gene), and "lung" are present only on the representation of the field before 2016, which indicates that researches using mice focused on studying BRAF-mutation lung cancer decreased. Our method to track the changes of a scientific field can be used by researchers to understand better how its sub-areas evolved in a concept level.

4.6 Conclusion

It is of utmost relevance the proposal of methods and tools to help researchers to better understand how scientific sub-areas evolved in a concept level. Textual documents as research papers provide rich information, but they are not structured. In this article, we proposed an approach to track changes in a scientific field, identifying how it evolves at a concept level. Our approach used knowledge graphs to structure the periods of a scientific field. It considered relations among concepts of a studied field to identify its main clusters, representing their relevant sub-areas to track their evolution. We used a similarity metric, based on the relevance of the concepts within the clusters, to determine if two clusters from distinct periods represent the same sub-area. In the conducted experimental case studies, we found that such technique is valuable to track sub-areas represented in different periods and to compare them, which by our analyses, occurred successfully. In particular, we analyzed the evolution of the Artificial Intelligence and the Biotechnology fields. Their representation and the evolution observed in both scientific fields were befitting with

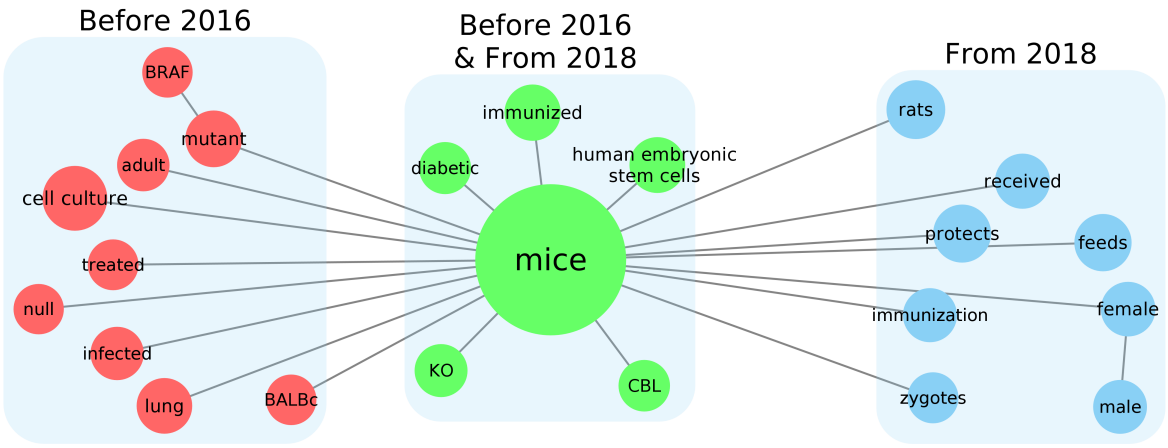

Figure 4.14: Evolution of the "Researches using mice" sub-area comparing its representation as cluster 4 in cover 1 (before 2016) and cluster 6 from cover 2 (from 2018).

the studied fields. The analysis of our solution occurred in distinct knowledge areas by assessing the results in a case with full academic articles and other considering only abstracts. Obtained results attest that our approach is domain-independent and, despite being further effective based on full articles, can structure a scientific field using only academic abstracts. As future work, we aim to help end-users to deal with noise data and with the configuration of the required parameters in the tool. We plan to study further techniques to reduce the number of noise concepts plotted to the user. Further interactive mechanisms to navigate over the graph can help users to explore the concepts in the detected clusters.

Chapter 5

Discussion

The main research goals of this MS.c. thesis are: (1) develop a framework to structure, represent, and analyze a scientific field over-time at a concept level; and (2) construct a software tool to facilitate the framework usage by end-users. Considering the complexity of those goals, we derived them into four more straightforward objectives to reduce their complexity, facilitating their management. Section 5.1 discusses how the outcomes generated by the last three chapters fulfill each of our five objectives. Next, Section 5.2 lists the limitations we observed in the proposed framework. Then, Section 5.3 presents a synthesis of our findings and recommendations.

5.1 Discussion on the Individual Research Objectives

This section discusses how we fulfill each of our four straightforward objectives during the development of this work.

1. **To structure and represent scientific textual data at a concept level.**

   The structure to represent scientific textual data at a concept level must enable not only to represent the input data, but to extract knowledge from it. The construction of the proposed structure occurred along with the development of the keyphrase extraction algorithm - objective (2). We explored this structure and the extraction of its keyphrases in Chapter 2, presenting C-Rank, which structures scientific articles and abstracts and extracts their keyphrases.

   As it is intrinsic to our goal to represent knowledge at a concept level, we studied strategies that would enable us to represent the scientific data that way. We observed that our approach to structure the scientific data should be able not just to identify concepts in the input texts, but further disambiguate them. That is why we explored the external background knowledge from Babelnet and the concept-linking of Babelfy. This strategy reduced the number of noisy concepts in our structure, as Babelfy identifies only concepts that indeed exist and belong to Babelnet. Our solution leveraged Babelnet background knowledge to improve the concepts disambiguation, which would be limited using only data input by the user. However, one has to admit that, despite the benefits of using Babelnet, further investigations should examine the usage of other semantic networks in this context.

After identifying the concepts to be used in the scientific data representation, we had to determine how to structure these concepts. Considering the keyphrase extraction task, we observed that co-occurrences graphs are commonly used to structure data in this context. Based on this information, we developed a co-occurrence graph, which, instead of using words, used concepts as its vertices. Therefore, we could extract knowledge from our scientific textual data using complex network analyses to, for example, recognize the most relevant vertices for the network and identify its clusters.

After structuring scientific documents and successfully extracting their keyphrases, we experimentally attested that the proposed structure can represent scientific textual data at a concept level.

Furthermore, to investigate the usage of this structure to represent broader textual data, we studied its usage to represent a scientific field. Accordingly, as scientific data is organized based on publications, we consider here that we can represent a scientific field with a collection of academic articles that belong to this field. Then, we could structure this collection with the approach previously developed, obtaining a knowledge graph of the scientific field. Accordingly, we fulfilled this objective in Chapter 3.

Moreover, the idea behind this knowledge graph representation is to assist researchers in understanding a scientific field. However, the proposed structure, because of its volume of data, is not simple to be analyzed without computational approaches. Therefore, we studied how to mitigate this issue and found out that the degree centrality is useful to identify the most relevant concepts in the clusters. Accordingly, we let the user choose if he/she wants to visualize only the most relevant concepts from the academic field, which reduces the volume of data to be analyzed by him/her.

The usage of the degree centrality reduced the amount of data to be processed by the user. However, some of the relevances identified were biased by the structure being constructed based on scientific articles. This bias occurred because, in academic articles, specific concepts are usually less mentioned than generic ones. Therefore, when visualizing only the most relevant concepts (key-concepts), the most specific ones will probably be excluded from the representation. So, the researcher must consider this characteristic when defining the relevance of the concepts to be represented.

2. **To automatically extract keyphrases from the previously mentioned structure without supervision.**

   As mentioned above, the development of C-Rank, our keyphrase extraction algorithm, occurred along with the development of the structure to represent scientific textual data - objective (1). We evaluated this algorithm and obtained state-of-the-art results extracting keyphrases from academic documents in the SemEval 2010 dataset. We also obtained competitive results extracting keyphrases from academic abstracts in the INSPEC dataset.

The idea behind developing a keyphrase extraction algorithm to extract keyphrases from a previously constructed structure was to do not request from the user background knowledge to base its extraction. Different from most keyphrases extraction algorithms, we could not request from the user external data. In this sense, we decided to base our keyphrases on the correspondent concepts identified in Babelnet. We discovered that we could use it to assist in the keyphrase extraction task, reducing noise and confusion between concepts not disambiguated. This strategy, as we show in Table 2.4, improved our results in almost 20%.

Other than the concept-linking approach, we designed four heuristics, which, apart from the one that discards candidates that appeared after a certain threshold of the text, can all be applied directly to our input structure. Hence, we attested that C-Rank obtains satisfactory results extracting keyphrases from academic articles structured as we proposed. Accordingly, we considered that, if we structure whole academic collections using the same strategy, C-Rank would extract its keyphrases likewise. This objective was attained in Chapter 2.

3. **To automatically identify sub-areas of a scientific field at a concept level.**

   The idea behind this objective was to assist researchers in understanding the topology of a scientific field, observing its sub-areas, and analyzing the concepts that form them. To this end, we used the knowledge graph representation of the scientific field previously proposed.

   In order to fulfill this objective, we clustered our knowledge graph representation, identifying its main sub-areas. Here we assumed that concepts that belong to the same sub-area, usually appear close to each other in texts. We considered that concepts that appear adjacent to each other in the academic articles are represented close in the knowledge graph. In general, clusterization algorithms tend to group elements close to each other in networks and divide those that are far. On this basis, we hypothesized that the clusterization of the knowledge graph would correctly segment the scientific field in its sub-areas.

   Based on the results of the document classification in Section 3.4, we confirmed our hypothesis that we could identify sub-areas of a scientific field by clustering its knowledge graph representation. Even without being constructed to classify documents based on previously defined groups, using our methodology to identify the sub-areas of a scientific field, we obtained satisfactory results identifying the sub-areas of a collection of academic documents. This objective was attained in Chapter 3.

4. **To track the evolution of a scientific field and its sub-areas.**

   Fulfilling this objective, we can let researchers understand how an academic field changes over time. To this end, we proposed a method to track the evolution that occurred in a scientific field between two time-periods.

   Considering this, we represented both time-periods using ScikGraph. Then, we thought in comparing those representations, identifying what changes in them. We

could not only output all concepts and relations that differed from the representations of both time-periods. In that way, we would be generating too much data for the researcher to analyze, which, without a refinement, would be meaningless. Therefore, we understand that to better track the evolution of a scientific field, we have to identify how its sub-areas change over time.

To track the evolution of sub-areas represented as clusters of distinct covers is not simple. This complexity happens because before comparing two clusters, one has to identify which clusters, from distinct covers, represent the same sub-area. Therefore, a similarity metric must be defined to determine if two clusters are similar enough to be considered correspondents from distinct covers.

Based on the metric proposed by Hopcroft in [24], we set our similarity metric, but, different from him, we did not consider that all concepts have the same weight for the similarity value. We calculated this similarity considering the relevance of each concept in the knowledge graph, favoring clusters formed around the same relevant concepts to be considered correspondents. Consequently, the importance of less relevant concepts for the similarity value is low, reducing the noise we observed that they generate in the identification of the correspondent clusters.

By using the similarity metric proposed, we successfully identified representations of the same sub-area in distinct time-periods. Their comparison was effective to support researcher to observe what changed in a particular sub-area, identifying, for example, sub-areas merged, segmented, and concepts added or excluded. Accordingly, we completed this objective in Chapter 4.

5.2 Limitations

This section describes the main limitations we observed in this work.

- We only evaluated C-Rank in the extraction of academic data. Therefore, we cannot affirm if it performs similarly in other domains, extracting keyphrases from, for example, news articles, books, and tweets.
- We did not investigate the usage of other semantic networks to replace the Babelnet in the concept linking task.
- We did not consider possible semantic relations between concepts to weight the edges that link them in the knowledge graph.
- We studied only the extraction of the key-concepts and keyphrases from the clusters of the knowledge graphs (representing the sub-areas of the scientific fields), not identifying how to label them with a single term or concept.
- We did not study how to use the scientific field concepts and their relations for summarising it textually.

- We did not evaluated the proposed software application with final users, which could have brought novel insights to the application.
- The results obtained extracting keyphrases from the knowledge graphs are limited by the C-Rank algorithm. Despite achieving state-of-the-art performance, C-Rank still produces noisy keyphrases along with the correct ones.

5.3 Synthesis of Findings and Recommendations

This Ms.c. obtained the following findings:

We designed the structure to represent scientific textual data at a concept level to represent a compendium of documents. However, we would not be able to validate if we structured the input knowledge coherently with a compendium of documents as input. We also notice that the keyphrase extraction validation from whole scientific fields would be impractical. Therefore, we decided to reduce the scale of what our knowledge graph would represent and used it to firstly structure and extract keyphrases from single academic documents or their abstracts, which would be simpler to validate.

It was possible to use the proposed structure to represent single articles or abstracts.

We could extract keyphrases without training or requesting from the user background data, achieving state-of-the-art results in this task.

We were successful adopting the approach of coherently representing scientific articles and automatically extracting their keyphrases to larger datasets containing a set of documents

Afterwards, we knew that because of the volume of data, analyzing whole scientific fields, composed of thousands of concepts, would be impracticable. Therefore, we studied how to identify the sub-areas of a scientific field automatically. This identification would allow researchers to fragment the data and analyze only the sub-areas in which they are interested, which they could identify based on the keyphrases identified with C-Rank.

However, there were three points we had to consider. First, a researcher may not have prior knowledge in the scientific field he is studying and, therefore, will not know the optimal number of sub-areas to divide it. Second, an abstract concept is commonly related to multiple sub-areas simultaneously. Third, the high amount of noisy concepts, which appeared too many or too few times in the whole collection of articles, would hinder the clusterization process.

Thus, to solve the first two points, we chose a clusterization algorithm that: allows overlap clusters and automatically identifies the number of groups to divide the knowledge graph. Then, we mitigated the third point by identifying the less relevant and the most generic concepts in the knowledge graph and excluding them. However, as we observed minor flaws during this approach, we recommended to perform it with a prior validation from the user.

After validating the methodology to identify sub-areas of a scientific field at a concept level, we studied how to track correspondent clusters over distinct covers. After proposing our similarity metric to determine if two clusters are correspondents, we could focus on the development of the application software to display the evolution of the sub-areas.

We constructed a web-based software tool to allow researchers that do not have programming skills to use the framework and functionalities proposed in this Ms.c. thesis. Our solution has not just a graphical interface, in addition a graphical representation of the structures developed to represent the studied scientific field.

Chapter 6

Conclusion

The amount of scholarly data produced has been increasing throughout the years. This fact impacts researchers' routines, which have to be up-to-date with discoveries and relevant papers related to their areas of expertise. Current automated mechanisms developed to assist researchers in dealing with this volume of scholarly data focus in aiding researchers in analyzing scientific content based on metadata information extracted from academic articles. Despite mitigating the problem, most of these approaches do not consider the concepts under the scientific knowledge.

This MS.c. thesis proposed a framework to structure, analyze, and track the evolution of scientific fields at a concept-level. To represent a scientific field, our framework explored a knowledge graph structure constructed using Babelnet concepts. It analyzes the representation based on its topology, identifying, for example, its key concepts and its main sub-areas. We confirmed obtaining satisfactory outcomes that our proposal is suited to adequately segment scientific fields in their sub-areas by using the identified sub-areas from our solution in a document classification task.

Our solution tracks the evolution of scientific fields by comparing two knowledge graphs, representing distinct time-periods of a scientific field. Relying on qualitative analyses, we achieved consistent results by tracking the evolution of sub-areas in the Artificial Intelligence and the Biotechnology scientific fields. Moreover, we performed all the previously mentioned experiments in multiple datasets, formed by data from distinct fields of science. We observed similar outcomes in all of them, indicating that datasets do not bias the proposed framework and, therefore, our approach is domain-independent.

Furthermore, we developed an application software with a web interface to let researchers without programming skills to use our proposed framework.

6.1 Summary of Contributions

This MS.c. thesis contributes to the enhancement of the scientific knowledge management and its visualization, obtaining state-of-the-art results in this area. In the following, we summarize the main contributions achieved by the development of this work.

1. The development of C-Rank [57] (cf. Chapter 2), an approach to perform unsupervised keyphrase extraction. It automatically extracts keyphrases from textual

documents without demanding training nor external data as input by the user. In this context, it obtained state-of-the-art results for extracting keyphrases from scientific documents in the SemEval 2010 dataset. Moreover, for not requiring external data from users, it is suited for adoption to extract keyphrases from other textual instances, like abstracts or whole compendiums of documents.

2. A technique to automatically identify the main sub-areas of a scientific field at a concept level, without training nor manually annotated data [58] (cf. Chapter 3). Experiments have shown that this technique can satisfactorily classify documents to which sub-area they belong. Our solution can adequately segment the sub-areas of a scientific field. Researchers can use this technique to understand how the sub-areas of a scientific field are organized at a concept-level. Through this, they can, for example, understand the relevance of concepts and how their relations might impact multidisciplinary studies.

3. A framework to structure and analyze scientific knowledge at a concept-level [58] (cf. Chapter 3). This framework allows researchers to observe how sub-areas of a scientific field are correlated, which concepts they have in common, how concepts of a sub-area are organized, and the key-concepts from a scientific field and its sub-areas.

4. An approach to track the evolution of a scientific field and its sub-areas at a concept-level [59] (cf. Chapter 4). This approach defined how to identify the similarity between groups of concepts based on their relevance. Our experimental analyses identified by using our approach, we could, for example, track concepts that were added or removed from a sub-area, and identify multiple sub-areas that merged themselves, generating a new one.

5. An online web-based software tool that allows researchers to graphically use the proposed framework to structure, analyze, and track the evolution of a scientific field [59] (cf. Chapter 4).

6. The online knowledge graph representations of the Artificial Intelligence and the Biotechnology scientific fields. [59] (cf. Chapter 4).

6.2 Disseminating our findings

In order to promote the dissemination of our research and its findings, during the development of this MS.c. thesis, we participated in academic events and wrote articles published and submitted to international conferences and journals. We list those below:

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. Combining complex networks and semantic annotations to analyze scientific literature. Oral presentation on the XIII Workshop de Teses, Dissertações e Trabalhos de Iniciação Científica, 2018.

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. C-rank: A concept linking approach to unsupervised keyphrase extraction. In Research Conference on Metadata and Semantics Research, pages 236–247. Springer, 2019 [57]

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. A Knowledge Graph Approach to Structure and Visualize a Learning Field. Oral flash presentation on the VI CCES Workshop of the Center for Computing in Engineering & Science, 2019.

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. A Knowledge Graph Approach to Structure and Visualize a Learning Field. Poster presentation on the VI CCES Workshop of the Center for Computing in Engineering & Science, 2019.

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. A Knowledge Graph Approach to Structure and Visualize a Learning Field. Poster presentation on the the XIV Workshop de Teses, Dissertações e Trabalhos de Iniciação Científica, 2019.

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. SciKGraph: A knowledge graph approach to structure a scientific field. Submitted to international journal, 2020. [58]

- Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. Understanding the evolution of a scientific field by clustering and visualizing knowledge graphs. Submitted to international journal, 2020. [59]

6.3 Future Work

This section presents perspectives of future investigations to expand the studies and contributions achieved in this work.

- Evaluate C-Rank against different types of data, as news articles, tweets, and books. This process would define if C-Rank suits to extract keyphrases from domains other than the scientific one.

- Study alternative ways to identify, disambiguate, and link concepts in a text with a background knowledge base. This research line might investigate other alternatives to reduce the dependence that our framework has with Babelnet and Babelfy.

- Investigate how to weight the correlation between concepts in the knowledge graph considering their semantic relation. This investigation could better describe the relations among concepts, reducing noisy concepts in the scientific knowledge representation.

- Study the labeling of groups of concepts that belong to the same knowledge graph automatically, considering the relevance of each concept for the groups. This study could be used to name the sub-areas of the scientific fields automatically, which would improve how users would understand them.

- Investigate how to textually summarise groups of concepts that belong to the same knowledge graph automatically. This investigation could be used to produce summarization texts of the identified sub-areas of a scientific field.
- Further apply, with the support of domain subjects, the proposed framework to represent, analyze, and track the evolution of a specific scientific field. In a user study, researchers could use the results generated by this application to understand the structure of the chosen scientific field better. Thus, basing further investigations on these results, or using them as a baseline to produce reviews in the studied field.

6.4 Final Considerations

This MS.c. thesis studied how to automatically structure, analyze, and track the evolution of a scientific field at a concept-level. It presented several findings on the scientific knowledge representation and visualization areas, achieving state-of-the-art results in the extraction of keyphrases from academic articles without demanding external data from the user. This research produced three articles published or submitted to international conferences and journals; along with a framework of our proposals, several python libraries, and an application software with a graphical interface to enable the usage of our framework by researchers without programming skills. We obtained satisfactory results evaluating our framework representing scientific data in distinct fields of knowledge, indicating that our findings are domain-independent. To conclude, with our findings and applications, we expect to assist researchers in understanding how scientific fields are organized, reducing the time they invest in this process.

Bibliography

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

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[3] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555. Association for Computational Linguistics, 2017.

[4] Slobodan Beliga, Ana Meštrović, and Sanda Martinčić-Ipšić. An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*, 39(1):1–20, 2015.

[5] Lutz Bornmann and Rüdiger Mutz. 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, 2015.

[6] Florian Boudin. A comparison of centrality measures for graph-based keyphrase extraction. In *International Joint Conference on NLP*, pages 834–838, 2013.

[7] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 543–551, 2013.

[8] Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2):48–57, 2014.

[9] Tanmoy Chakraborty and Abhijnan Chakraborty. Overcite: Finding overlapping communities in citation network. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pages 1124–1131. IEEE, 2013.

[10] Mingming Chen and Boleslaw K Szymanski. Fuzzy overlapping community quality metrics. *Social Network Analysis and Mining*, 5(1):40, 2015.

[11] Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

[12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[13] Ugur Dogrusoz, Erhan Giral, Ahmet Cetintas, Ali Civril, and Emek Demir. A layout algorithm for undirected compound graphs. *Information Sciences*, 179(7):980–994, 2009.

[14] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48, 2016.

[15] Howard Eichenbaum and Neal J Cohen. Can we reconcile the declarative memory and spatial navigation views on hippocampal function? *Neuron*, 83(4):764–770, 2014.

[16] Christina Feilmayr and Wolfram Wöß. An analysis of ontologies and their success factors for application to business. *Data & Knowledge Engineering*, 101:1–23, 2016.

[17] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.

[18] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[19] Grobid. `https://github.com/kermitt2/grobid`, 2008 — 2020.

[20] Shashank Gupta and Vasudeva Varma. Scientific article recommendation by using distributed representations of text and graph. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1267–1268. International World Wide Web Conferences Steering Committee, 2017.

[21] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[22] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers)*, volume 1, pages 1262–1273, 2014.

[23] Henry Heberle, Gabriela Vaz Meirelles, Felipe R da Silva, Guilherme P Telles, and Rosane Minghim. Interactivenn: a web-based tool for the analysis of sets through venn diagrams. *BMC bioinformatics*, 16(1):169, 2015.

[24] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5249–5253, 2004.

[25] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics, 2003.

[26] Sukhwan Jung and Aviv Segev. Analyzing future communities in growing citation networks. *Knowledge-Based Systems*, 69:34–44, 2014.

[27] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *5th International Workshop on Semantic Evaluation*, pages 21–26, 2010.

[28] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742, 2013.

[29] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90, 2016.

[30] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, , Matthew S Gerber, and Laura E Barnes. Hdltex: Hierarchical deep learning for text classification. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE, 2017.

[31] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[32] Melvin Earl Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

[33] Aaron F McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011.

[34] Meta | expand your research. `https://www.meta.org/`, 2020.

[35] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[36] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[37] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[38] Andrea Moro, Francesco Cecconi, and Roberto Navigli. Multilingual word sense disambiguation and entity linking for everybody. In *International Semantic Web Conference (Posters & Demos)*, pages 25–28, 2014.

[39] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Computational Linguistics*, 2:231–244, 2014.

[40] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[41] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, 2010.

[42] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[43] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[44] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[45] Roberto Ortiz, David Pinto, Mireya Tovar, and Héctor Jiménez-Salazar. Buap: An unsupervised approach to automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 174–177. Association for Computational Linguistics, 2010.

[46] David Otasek, John H Morris, Jorge Bouças, Alexander R Pico, and Barry Demchak. Cytoscape automation: empowering workflow-based network analysis. *Genome biology*, 20(1):1–15, 2019.

[47] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[48] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[49] Airel Pérez-Suárez, José F Martínez-Trinidad, Jesús A Carrasco-Ochoa, and José E Medina-Pagola. Oclustr: A new graph-based algorithm for overlapping clustering. *Neurocomputing*, 121:234–247, 2013.

[50] Joseph Rocchio. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323, 1971.

[51] Bernard Rous. Major update to acm's computing classification system. *Communications of the ACM*, 55(11):12–12, 2012.

[52] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[53] Semantic scholar | ai-powered research tool. `https://www.semanticscholar.org/`, 2015.

[54] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[55] Filipi N Silva, Diego R Amancio, Maria Bardosova, Luciano da F Costa, and Osvaldo N Oliveira Jr. Using network science and text analytics to produce surveys in a scientific topic. *Journal of Informetrics*, 10(2):487–502, 2016.

[56] Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

[57] Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. C-rank: A concept linking approach to unsupervised keyphrase extraction. In *Research Conference on Metadata and Semantics Research*, pages 236–247. Springer, 2019.

[58] Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. Scikgraph: A knowledge graph approach to structure a scientific field. *Submitted to international journal*, 2020.

[59] Mauro Dalle Lucca Tosi and Julio Cesar dos Reis. Understanding the evolution of a scientific field by clustering and visualizing knowledge graphs. *Submitted to international journal*, 2020.

[60] Mauro Dalle Lucca Tosi and Guilherme Alberto Wachs Lopes. Sistema de gerenciamento automático de artigos. In *VII Simpósio de Iniciação Científica, Didática e de Ações Sociais de Extensão na FEI*, 2017.

[61] Sahar Vahdati, Guillermo Palma, Rahul Jyoti Nath, Christoph Lange, Sören Auer, and Maria-Esther Vidal. Unveiling scholarly communities over knowledge graphs. In Eva Méndez, Fabio Crestani, Cristina Ribeiro, Gabriel David, and João Correia Lopes, editors, *Digital Libraries for Open Knowledge*, pages 103–115, Cham, 2018. Springer International Publishing.

[62] Feng Xia, Haifeng Liu, Ivan Lee, and Longbing Cao. Scientific article recommendation: Exploiting common author relations and historical preferences. *IEEE Transactions on Big Data*, 2(2):101–112, 2016.

[63] Feng Xia, Wei Wang, Teshome Megersa Bekele, and Huan Liu. Big scholarly data: A survey. *IEEE Transactions on Big Data*, 3(1):18–35, 2017.

[64] Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 ieee 11th international conference on data mining workshops*, pages 344–349. IEEE, 2011.

[65] Chyi-Kwei Yau, Alan Porter, Nils Newman, and Arho Suominen. Clustering scientific documents with topic modeling. *Scientometrics*, 100(3):767–786, 2014.

Appendix A

Springer Copyright Clearance

This Agreement between Mr. Mauro Dalle Lucca Tosi ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	4754191313364
License date	Jan 22, 2020
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	C-Rank: A Concept Linking Approach to Unsupervised Keyphrase Extraction
Licensed Content Author	Mauro Dalle Lucca Tosi, Julio Cesar dos Reis
Licensed Content Date	Jan 1, 2019
Type of Use	Thesis/Dissertation