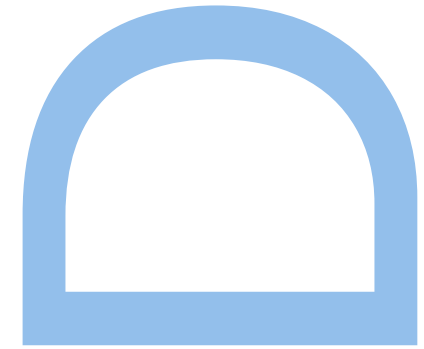
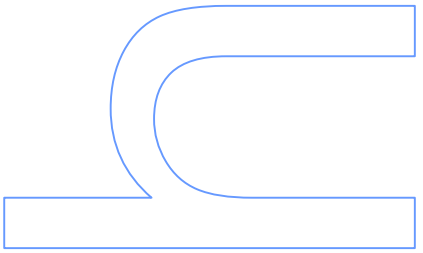
Towards Measuring Scientific Impact Using Network Science

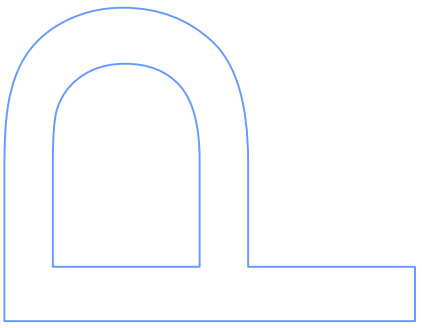


Jorge Miguel Barros da Silva
Programa Doutoral em Informática das Universidades do Minho, Aveiro e Porto
Departamento de Ciência de Computadores
2021

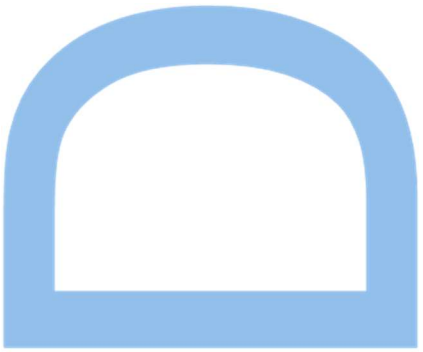


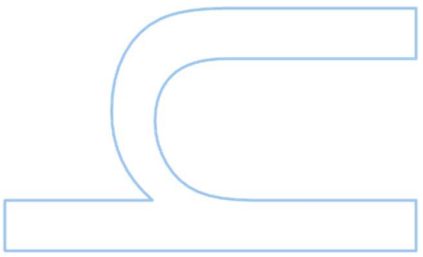
Orientador
Fernando Manuel Augusto da Silva, Professor Catedrático
Faculdade de Ciências da Universidade do Porto

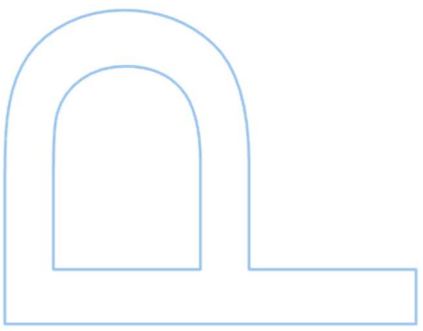
Coorientador
Pedro Manuel Pinto Ribeiro, Professor Auxiliar
Faculdade de Ciências da Universidade do Porto



Universidade do Minho







Para a Patrícia

Acknowledgements

First and foremost, I want to thank my advisors, who guided me through this journey. They were always ready to propose and discuss new ideas. In particular, I am most grateful for the intellectual liberty they always gave me, which made me a more autonomous researcher and helped me grow as a person. I want to thank Professor Fernando Silva for his guidance and for the interest he showed in my scientific and personal development long before this PhD started. I started working with Professor Fernando Silva in 2014. When I started I had no expectations to do a PhD but through his counselling, encouragement and enthusiasm I was drawn to this great adventure. Through his connections I was also able to travel to different places, meet new people and work with different research groups which were very enriching experiences. I am really grateful for all of his help and trust. I also want to thank Professor Pedro Ribeiro for his guidance. He is one of the most passionate professors that I ever had and I will always be grateful for his help throughout this thesis.

I want to thank MAP-i's organisation for providing me with a good research environment. This allowed me to share and discuss my ideas with many other PhD students and experienced researchers. I also want to thank MAP-i for awarding me a four year FCT Doctoral grant [PD/BD/128157/2016].

Throughout my PhD I had the opportunity to visit the Centre for Science and Technology Studies (CWTS) research institute at the Leiden University. I want to thank the whole CWTS team for their hospitality. They were always very kind to me and were always ready to discuss new ideas. In particular, I want to thank Dr. Nees Jan van Eck for giving me the opportunity to work with a prestigious research institute such as the CWTS. Furthermore, I want to thank him for helping in my research during and after my visit. His knowledge and experience had a huge impact of the work presented in this thesis. I will miss our weekly Skype calls.

I met many colleagues during my PhD. I want to thank my office mates for the countless discussions and the fun times, namely, Pedro Paredes, Daniel Loureiro, Ahmad Naser, Sadeeq and Shamsuddeen. In particular, I want to thank David Aparício one of the brightest minds in research and definitely the best researcher to come up with acronyms for new algorithms. I am sorry that I did not have time to implement more algorithms so we could complete our plan to create a full-circle around the sea lion animal. Our collaborations were undoubtedly one of the highest points of my PhD, I will always remember how our discussions and plans always ended up with some jokes. I am really glad that I had the opportunity to meet and work with you. Most of all, I am happy to call you a friend. I also want to thank my friends that were not in the same office as me but still were very important for my PhD. Namely, I want to thank Marcelo Santos, Fábio Carvalho and Vladir Vicente for always being ready for the *"lets not talk about work"* coffee and for some football which massively helped me during my PhD.

I want to thank my friends Joana Gonçalves, Tânia Rodrigues and Rui Fonseca for keeping me sane throughout my PhD. It does not matter how many lockdowns there are or the physical distance between us, they are always there for me to cheer me up. I love you guys. In particular, I want to thank Nuno Guimarães. I intentionally delayed writing your name just to keep messing with you (as always). The truth is that it is amazing to meet someone almost 15 years ago, instantly feel a connection and end up more than 10 years later doing a PhD at the same time and in the same office. You have been always there for me and I am very grateful for all the times we spend together. I will end this now before you start making fun of me, but just so you know, you are like a brother to me.

Finally, I want to thank my family for always supporting my decisions. In particular, I want to thank my parents and grandparents for always motivating and pushing me to pursue my dreams. They always did everything for me and to make this possible. I also want to thank Patrícia Santos the love of my life. During this PhD we finally completed our dream of living together and it was the best decision I ever made. You always support my choices and show confidence in me to succeed no matter the challenge. This PhD and many other things in my life would have not been possible without you. I was very lucky to meet you and still am for being with you. I will always love you.

Abstract

Bibliometrics are algorithms that resort to statistical analysis of scientific literature to track the output and impact of research. For several years bibliometrics have been used to help researchers understand the evolution of science and aid with the decision making processes that pave the way for future research. In particular, measuring the scientific impact of researchers is one of the most important tasks in the bibliometrics area. The output of this task is often used in peer-reviewing processes and funding allocation decisions. Despite being a widely studied problem, measuring the scientific impact of researchers still faces several challenges such as: (1) criteria flexibility, (2) incomplete citation information, (3) citation boosting, (4) comparing the scientific impact of different research areas, and (5) discriminating the different contributions of the authors of a publication.

The goal of this thesis is to contribute towards the creation of a tool that addresses the previous challenges in measuring the scientific impact of researchers. In more detail, in this thesis we present solutions for each individual challenge and also conceptualise the integration of these methods into a single framework that measures the scientific impact of researchers. We present solutions to the challenges 1, 2 and 3 by proposing a novel PageRank-based algorithm to measure scientific impact in incomplete citation networks, and by presenting a penalty system to decrease the impact of the citations that result from the abuse of citation boosting techniques. Regarding challenges 4 and 5 our solutions require addressing two other problems from the bibliometrics area, namely, publications clustering and expertise profiling.

Publications clustering consists in grouping publications according to their scientific areas. Being able to group publications according to their scientific area aids with challenge 4. In more detail, the clusters of publications make it possible to measure the scientific impact of researchers for a scientific area, and then apply normalisation

strategies to create comparable rankings between different scientific areas. For the publications clustering problem we propose a new publication similarity estimator algorithm that analysis the relations between publications and their metadata to measure the similarity of publications.

Expertise profiling consists in assessing the areas of expertise of researchers. With respect to measuring the scientific impact of researchers, knowing the scientific areas in which the researchers have the most expertise aids with challenge 5. More concretely, the expertise profiles of authors make it possible to discriminate the credit given to the authors of a publication according to their expertise on the topics discussed in the publication. For the expertise profiling task we propose a method that automatically constructs a multi-typed topical hierarchy and maps the knowledge of researchers into this hierarchy in order to obtain their hierarchical expertise profile.

Throughout this thesis we address three bibliometric problems: author ranking (name used in literature to refer to the task of measuring the scientific impact of researchers), publications clustering and expertise profiling. For each bibliometric problem we contribute with new algorithms that tackle existing problems in these areas. We evaluate all the developed solutions with real-world data and the results show that our methods outperform comparable methods from the state-of-the-art. With respect to our end goal, the creation of a complete tool to measure the scientific impact of researchers, we present a concept that integrates all the developed work. We believe that our work helps the development of the bibliometrics area and we present several other possible research directions for future work.

Resumo

As bibliometrias recorrem a análise estatística da literatura científica para estimar a quantidade e impacto da investigação. Durante os últimos anos as bibliometrias tem sido essencialmente utilizadas para ajudar os investigadores a perceberem a evolução da ciência e a tomar decisões sobre a mesma. As bibliometrias tem sido particularmente essenciais para medir o impacto científico dos investigadores. Esta informação é frequentemente utilizada na tomada de decisões como alocação de fundos financeiros e promoções dentro de instituições. Apesar da medição do impacto científico ser um problema bastante estudado, existem ainda alguns desafios nesta área. Nomeadamente (1) a flexibilidade dos critérios, (2) informação incompleta sobre as citações, (3) uso indevido de citações, (4) comparar o impacto científico entre investigadores de diferentes áreas e (5) atribuir diferente mérito pelas contribuições dos autores da mesma publicação.

O objectivo desta tese é contribuir para o desenvolvimento de uma ferramenta para medir o impacto científico de investigadores que consiga precaver todos os desafios anteriormente mencionados. Com este objectivo em mente, nesta tese começamos por apresentar soluções para cada um dos desafios anteriores e no fim, apresentamos um desenho conceptual para integrar todas as soluções numa única ferramenta para medir o impacto científico dos investigadores. Para os desafios 1, 2 e 3 as nossas soluções consistem no desenvolvimento de um novo algoritmo baseado no PageRank para medir o impacto científico dos investigadores em redes de citações incompletas e de um sistema para penalizar o uso indevido de citações. Para os desafios 4 e 5 as nossas soluções requerem estudar outros dois problemas de bibliometrias: o clustering de publicações e a criação de perfis de conhecimento. O clustering de publicações consiste em agrupar publicações pelos seus tópicos. Esta informação ajuda com o desafio 4 porque torna possível separar as publicações por diferentes tópicos, medir o impacto dos investigadores em cada tópico e no fim utilizar estratégias

de normalização para tornar os valores comparáveis entre diferentes áreas. Para o clustering de publicações desenvolvemos um algoritmo para calcular a similaridade entre as publicações que analisa as relações entre as publicações e os seus metadados para calcular a similaridade. O problema de criação de perfis de conhecimento consiste em identificar as áreas nas quais um investigador tem conhecimento. Este problema pode ajudar com o desafio 5 uma vez que saber o conhecimento dos autores de uma publicação ajuda a criar estratégias que associam maior mérito a autores que tenham mais conhecimento sobre a publicação. Para o problema de criação de perfis de conhecimento, desenvolvemos um novo algoritmo que constrói uma hierarquia de tópicos através da análise dos metadados das publicações e em seguida mapeia o conhecimento dos investigadores nesta hierarquia de forma a criar um perfil de conhecimento hierárquico.

Nesta tese estudamos três problemas de bibliometrias: o ranking dos autores (nome normalmente utilizador para referir o problema de medir o impacto cientifico de investigadores), o clustering de publicações e a criação de perfis de conhecimento. Para cada um destes problemas contribuímos com novos algoritmos que propõe soluções para desafios existentes. Todas os algoritmos desenvolvidos são avaliados com dados do mundo real e, de uma forma geral, os nossos resultados mostram que os algoritmos desenvolvidos tem melhores resultados que outros algoritmos semelhantes. Em relação ao nosso objectivo de construir uma ferramenta para a medicação do impacto científico dos investigadores, apresentamos um desenho conceptual de uma ferramenta para integrar todo o trabalho desenvolvido durante esta tese. Acreditamos que este trabalho ajuda a desenvolver a área das bibliometricas e apresentamos algumas ideias que podem ser seguidas como trabalho futuro.

Contents

List of Tables

List of Figures

LIST OF FIGURES

List of Algorithms

List of Acronyms

ACM	**A**ssociation for **C**omputing **M**achinery
ALAR	**A**uthor-**L**evel **A**uthor **R**anking
APT	**A**uthor-**P**ersona-**T**opic
AT	**A**uthor-**T**opic
ATC	**A**uthor-**T**opic-**C**ommunity
CPM	**C**onstant **P**otts **M**odel
DN	**D**angling **N**odes
FOCAS	**F**riendly **O**nly **C**itations **A**naly**S**er
GA	**G**ranularity x **A**ccuracy
HEPHIN	**H**ierarchical **E**xpertise **P**rofiles using **H**eterogeneous **I**nformation **N**etworks
HIN	**H**eterogeneous **I**nformation **N**etwork
LDA	**L**atent **D**irichlet **A**llocation
MeSH	**M**edical **S**ubject **H**eadings
MRR	**M**ean **R**eciprocal **R**ank
NDCG	**N**ormalized **D**iscounted **C**umulative **G**ain
OTARIOS	**O**pTimizing **A**uthor **R**ankings using **I**nsiders/**O**utsiders **S**ubnetworks
PR	**P**age**R**ank
PMI	**P**ointwise **M**utual **I**nformation
PURE-SIM	**PU**blication **R**elatedness **E**stimator using **S**tar-schema **I**nformation networks of **M**etadata
RR	**R**andom **R**estart
ST	**S**core **T**erm
TAT	**T**emporal-**A**uthor-**T**opic

Chapter 1

Introduction

The term bibliometrics was introduced by Pritchard in 1969 and it was defined as *"the application of mathematical and statistical methods to books and other media of communication"* [6]. In the same year, Nalimov and Mulchenko introduced the term scientometrics which was defined as *"the application of those quantitative methods which are dealing with the analysis of science viewed as an information process"* [7]. In general, the term bibliometrics was originally defined as a broader topic that encapsulates all forms of communications, while the term scientometrics is restricted to measurements in scientific literature. Nowadays, in the literature related to the study of science, both terms are used interchangeably. In this thesis, we categorise our work in the area of bibliometrics. The algorithms discussed in this document are applied in the context of scientific literature, however they can be used in other forms of communications such as social media posts and books. Thus, the term bibliometrics is more adequate to describe our work.

The initial efforts of statistical studies on scientific literature can be traced back to 1926 when Alfred J. Lotka[8] published his pioneering study on the frequency of distributions of scientific research. In 1934, Bradford [9] published a similar study this time analysing the distribution of papers over journals. This work was fundamental for the development of the famous Bradford's law which states that for any given topic, journals are divided in three zones. The top third (zone 1) which represents the journals most cited about that topic, the middle third (zone 2) which includes the journals with an average amount of citations, and the bottom third (zone 3) with the remaining journals. Bradford's law predicts that the number of journals in zone 2

and 3 are n and n^2 times larger than the first zone. Thus, it is possible to estimate the total number of journals about a topic once the number of journals for zone 1 is known [10].

Despite some initial efforts, bibliometric studies did not have much impact until the early 1960s when Derek Price presented his vision about science and its place in society in the book *"Little Science, Big Science"*. Price promotes the ideas that science has shifted from *"small science"* to *"big science"* and that studying science is important. Throughout his career, Price continuously paved the way to modern bibliometrics by bringing attention to questions dealing with the quantitative aspects of research, promoting the use of the Science Citation Index database and presenting the first systematic approach to study science [11]. During this period the number of bibliometric studies increased, however they were very limited (in terms of data used) by the technology available at the that time (i.e., no access to large bibliographic databases and most statistics were calculated by humans). In the 1990s, the new developments in computer science granted easier access to large bibliographic databases and computational power to automatically process more data. As a result, the bibliometrics area grew so much in importance that it became a standard tool for science police and research management. In more detail, most of the science indicators produced heavily relied on publications and citations statistics, and other more sophisticated bibliometric techniques [11].

In the early 2000s a new challenge emerged on the bibliometrics field. Up to this point, bibliometrics focused on the study of science at the macro level (i.e., oriented towards the study of publications, journals and research fields). For example, some popular studies were related to: identifying the most predominant papers/journals in terms of received citations, clustering publications according to their research area and discovering interdisciplinary areas. In 2005, Jorge Hirsch proposed the h-index [12] which was a new indicator to measure the scientific impact of individual researchers. Thus, introducing the idea of using bibliometrics to study science at the micro level (i.e., oriented towards individual researchers). The idea of measuring the scientific impact of scientists quickly became one of the hot topics in bibliometrics and still today it is one of the most studied problems in bibliometrics with several new algorithms proposed every year.

Nowadays, bibliometrics still are a popular topic which importance has been growing mainly due to the current exponential growth in scientific research. With the evolution of computer science, it is possible to store and make available large bibliographic databases, such as Google Scholar, AMiner, CiteSeerX, among others [13]. These

databases store millions of records of published research as well as several metadata associated to each publication. For example, in the AMiner database, each publication is associated with its abstract, author's names, keywords, journal and references. This overflow of information along with the number of scientific documents being published each year, make it impossible to obtain a good understanding of the development of science using only human analyse. Thus, algorithms to extract information from large corpus of documents, i.e., bibliographic databases, have grown in importance. Some current hot-topics in the bibliometrics field are: measure scientific impact, development of classification systems and expertise retrieval [14, 15, 16]. Measuring scientific impact estimates how much an entity contributes to the development of a scientific area. These entities can range from individual researchers to research groups and universities. This information is particularly important for policy decisions such as research funding and allocation or promotions within institutions. Classification systems assign a publication to a research field. Thus, these algorithms are critical for collection organisation and help define the boundaries of science fields which is also important to obtain a big picture of the evolution of science. Expertise retrieval consists in identifying experts on a specific research field. This task is particularly important to categorise personal in large institutions, identify possible collaborations and to allocate human resources in institutions.

Network science is an interdisciplinary field which studies network representations of physical, biological and social phenomena leading to predictive models of these phenomenons [17]. A network represents entities as nodes and their corresponding relations as edges. Network science algorithms analyse different properties of the network and obtain valuable information. For example, let us consider a social media network where nodes represent social profiles and edges represent the friendship relation between the profiles, i.e., an edge exists between two nodes if the social profiles are *"friends"* in the social media platform. A community detection algorithm which identifies groups of closely connected nodes in networks can be used to identify different groups of persons that share similar interests [17]. Furthermore, centrality measures which determine the most central nodes in the network can be used to identify the biggest *influencers* in the social media [17]. These algorithms along with many others provide knowledge about the social media system which help us to better understand the system as a whole.

Similarly to social media data, bibliographic data can also be modelled as a network in order to improve our knowledge about science. The first attempt to use network science in the field of bibliometrics traces back to 1965, when Price constructed a

network of papers and their citation relations to study the patterns of references in science [18]. Nowadays, paper-level citation networks (as the one constructed by Price) are frequently used in several bibliometric studies such as the creation of classification systems, identification of research front and measurement of publications/journals scientific impact [14, 15, 19]. Another type of network that is commonly used in bibliometric studies is the author-level network where nodes represent authors or researchers. In these networks there are mostly two types of relations that are studied: citation relations between authors which are frequently used in studies to measure the scientific impact of researchers [20] and co-authorship relations between authors which are commonly used to study research collaboration [21].

In conclusion, bibliometrics are necessary to understand science and they aid the decision process that leads to the evolution of science. Network science is a powerful tool that along with the large bibliographic databases that are available, can extract important knowledge about science and help develop new bibliometric studies. Thus, improving our understanding of science.

1.1 Thesis motivation

This thesis focuses on the problem of measuring the scientific impact of researchers. This task consists in assigning a value to a researcher that reflects his impact or relevance with respect to a certain scientific area. The general assumption is that the quality of the publications of a research is directly correlated to his scientific impact. The better the publications are (which is often estimated through the number of citations received) the more scientific impact the researcher has. Despite being a problem that has been widely studied and with multiple metrics proposed, measuring the scientific impact of researchers still presents many challenges. Some of these challenges are:

1. **Criteria flexibility.** The best criteria to measure scientific impact is not clear and often depends on the application. Current metrics do not allow the users to easily change the evaluation criteria which makes it difficult to find an approach for all applications. Furthermore, there are some important criteria that are not considered on current approaches.

2. **Incomplete citation information.** Measuring the scientific impact of researchers is often achieved through the analyse of the citation network. One

problem of these approaches is that they assume that the citation network is complete (i.e., it contains all the received citations for all the nodes). Obtaining the complete citation network is unfeasible and even if the complete citation network is obtained it would require a huge computational power to extract knowledge from it. Thus, strategies that use the citation network make an incorrect assumption that leads to unfair measurements of scientific impact.

3. **Citation boosting.** Researchers can resort to the use of certain citation patterns (e.g., self-citations and reciprocal citations) to boost their number of citations and consequently, their perceived scientific impact. Current metrics to measure scientific impact are not able to address this problem and consider that citations are equally deserved which again leads to unfair measurements of scientific impact.

4. **Comparing the scientific impact of different research areas.** The amount of citations received by a publication depends on the research area. For example, some research areas have more publications than others thus it is more likely for a publication to receive more citations just by chance. Current metrics to measure scientific impact disregard this fact and offer no solution to compare the scientific impact of researchers from different research areas which limits the use of current approaches (i.e., it is not possible or ideal to compare the scientific impact of researchers that work on different research areas).

5. **Discriminating contributions of the authors of a publication.** In publications with multiple authors the traditional scenario is the one where a small portion of the authors are responsible for the most contributions in the publication. Current metrics to measure scientific impact often assign the same credit for all the authors in a publication. This is a problem since by equally dividing the credits for all the authors, the authors that contribute the most are receiving less credit for their publications. Thus, their scientific impact is not being accurately estimated.

The goal of this thesis is to contribute towards the creation of a tool to measure scientific impact of researchers that addresses all the previous challenges. To achieve this, we propose and evaluate solutions for each individual challenge, and also conceptualise how these methodologies could be integrated into a single framework. For the challenges 1, 2 and 3 our solutions consists of proposing novel algorithms to measure the scientific impact of researchers and to penalise citations resulting from citation boosting practices. For the challenges 4 and 5 our solutions require tackling other

bibliometric problems. Namely, the problems of publications clustering and expertise profiling. The problem of publications clustering consists in dividing publications in multiple clusters in a way that publications in the same cluster are strongly related to each other. The general idea is that each cluster represents a research area. The publications clustering problem aids with the challenge 4 since it defines the boundaries between research areas. With this information it is possible to separate researchers by their research areas, measure the impact of authors in their area and normalise the impact within an area in order to obtain comparable measurements of scientific impact from researchers in different research areas. The expertise profiling problem consists in identifying the areas of expertise of a researcher. With respect to measuring scientific impact, knowing the areas of expertise of researchers and the topics discussed in the publication (which can be obtained from the publications clustering problem) can aid with challenge number 5. In more detail, one can assign more credit to the authors of a publication that, at the time of publishing, had more expertise in the topics discussed in the publication. Thus, promoting a fairer distribution of scientific credits through the authors.

The current available approaches for publications clustering and expertise profiling present some drawbacks that prevent us from using existing approaches to tackle the problem of measuring scientific impact. More concretely, publications clustering methods rely on citation or textual information of publications which is often not available in bibliographic databases, and expertise profiling methods often fail to present the detailed knowledge of a researcher at multiple granularity levels. For these reasons, in addition to the problem of measuring scientific impact, we also study the problems of publications clustering and expertise profiling. Thus, this thesis is divided in the discussion of three bibliometric problems, author ranking (name used in the literature to refer to the problem of measuring the scientific impact of researchers), publications clustering and expertise profiling. In the following sections we further motivate each problem.

1.1.1 Author ranking

In the research community it is often necessary to evaluate and rank the quality of the work produced by researchers. Decisions such as funding allocation, scientific committees creation, or faculty promotions are heavily based on the estimated contributions of a researcher with respect to a given scientific area (also known as the researcher scientific impact). Nowadays, these decisions are still done mostly by peer-review (i.e.,

evaluation of work by experts with similar competencies as the producers of the work). Although this process still relies on human intervention, more and more the experts use bibliometrics to aid in their decision [22, 23].

Author ranking approaches define a set of rules (i.e., an algorithm) that evaluate an author (or researcher) based on his scientific output (e.g., published documents). The most frequently used indicators for this evaluation are: the quantity of publications, the quality of the journals in which documents are published and the quality of publications (measured by the number of received citations) [23]. In general, most of the author ranking approaches rely on the study of citations received by a publication to evaluate its quality and then, the quality of an author is directly associated to the quality of his publications [12, 20, 24].

Traditional author ranking approaches rank individual entities (e.g., publications, authors or journals) by directly analysing the citations received by the entity. One of the drawbacks of using traditional approaches is that they do not credit indirect citations. For example, consider that authors A, B and C publish in the same scientific area and A cites B, and B has previously cited author C, the more likely scenario is that C's work has had influence in A's and B's publications. However, traditional approaches do not give any credit to C from A's indirect citation. To overcome this limitation, more recent author ranking algorithms use network science concepts to estimate scientific impact through the analysis of citation networks [19, 20, 25, 26].

Centrality measures, namely modifications of the PageRank algorithm [27] can be effectively used to rank authors in citation networks [19, 20, 26]. The original idea of the PageRank algorithm developed in the context of webpages is that it is good to be referenced by any webpage, however it is better to be referenced by important webpages. This idea extends naturally to author ranking in citations networks since more credit is given to researchers that are cited by more important researchers. Another advantage of using citation networks and centrality measures to rank authors is that one can easily change the criteria used to rank authors by changing citations weights in the network [19, 24, 28] and/or by considering different node initialisation and score diffusion strategies while estimating node centrality [19, 20, 29]. This is particularly important since different applications of the author ranking may require different bibliometric indicators to be considered [22, 23]. There are two limitations with the current approaches. First, they disregard the fact that citation networks are often incomplete. More concretely, in real-world data it is often impossible to obtain all the citations received by an author. Thus, in order to promote fair author ranking results, for these authors it is necessary to consider other strategies than PageRank to

estimate their score. Second, current approaches fail to combine important features that are present in citation networks. For example, there are methods that consider that the venue of publication and the year of a publication are important to measure scientific impact. Meanwhile, there are no methods that consider that being cited by a publication published in a prestigious venue in recent years is more important than being cited by a publication published in a mediocre venue a decade ago.

In the scientific community the use of author ranking approaches is also associated to some controversy [30]. An author ranking approach clearly defines the criteria that determines the scientific impact of the researchers. Most of the times, these criteria are related to the number and quality of the citations received by the researchers. Thus, it is very enticing for researchers to use citation boosting strategies to increase their number of citations and consequently, their perceived scientific impact. Although this is a well known problem in the research community and it has been proven by several studies [31, 32, 33] author ranking approaches often disregard this practice which leads to unfair measurements of scientific impact.

We aim to develop new algorithms for author ranking in citation networks that are more accurate than current strategies. We intend to improve the current methods by taking into account that information in citation networks is often incomplete, by making use of more features present in the citations and by penalising citations that originate from citation boosting techniques.

1.1.2 Publications clustering

A comprehensive analysis and understanding of the organisation of the scientific knowledge is necessary in the research community to promote further advances in science and research [34]. The organisation of the scientific knowledge is mostly accomplished through classification systems that assign journals or individual publications to research fields. In the past, some classification systems were completely created using only the expert knowledge [35]. However, this technique requires great effort in terms of time and the number of experts needed to accomplish the task. For that reason, approaches to automatically create classification systems started to emerge. These approaches are often referred as publications clustering techniques. A publications clustering strategy is divided in two independent tasks: estimating publications similarities and applying a cluster algorithm. In this thesis, we focus on the task of estimating publications similarities (also referred as estimating publications relatedness in the literature).

Current publication similarities estimators are based on citation (direct citation, co-citation and bibliographic coupling [36, 37]) or co-word [38] analysis. One limitation of these current approaches is the fact that several bibliographic databases do not contain (or at least for a significant amount of publications) references information and/or the text content of the publications [37]. As a result, it is impossible to estimate the publication similarity for all the documents in the database and consequently, it is impossible to cluster the complete set of publications. Thus, it is necessary to come up with new strategies to estimate publications similarity, in particular strategies that couple with missing information in the bibliographic database. This new development would allow any bibliographic database to organise their scientific knowledge using the information that they have available.

We aim to develop a new algorithm to estimate publications similarity that is more accurate than current strategies that require citation or textual information of the publications. Furthermore, we intend that the algorithm is capable of dealing with missing information in the sense that it can be used in cases where the bibliographic database does not contain citation or textual information for all the publications. For this purpose, we use a network to model the relations between metadata and publications in bibliographic databases, and then we use a stochastic process to estimate publications similarities.

1.1.3 Expertise profiling

With the increasing number of researchers in the academic world, it is important to develop tools that can automatically summarise the competences of each individual in a profile [14]. One way to obtain these profiles is to analyse the published literature of researchers. Text mining based approaches are widely used for this purpose [14, 39, 40, 41] while there exist a small number of strategies available that rely on network based approaches [42, 43]. Both approaches follow a general workflow to tackle the problem. First, the algorithms need to define the topics (or research areas) where the competences (or knowledge) of an expert are assessed. For the purpose, the whole corpus of publications is considered and a topic modelling strategy is utilised. For text mining based approaches, generative statistical models are used to estimate the distribution of publications over words and the distribution of words over topics [14, 44, 45]. For network based approaches, usually community detection approaches are utilised on publications citation networks [46] or keywords co-occurrence networks [43]. Similarly to the publications clustering problem, the general assumption is that publications in

the same community address the same topic. After identifying the topics discussed in a corpus of publications, expertise profiling approaches map the knowledge of researchers into the topics. Traditionally, approaches identify the topic of publications and use the authorship links to map the knowledge to their authors (i.e., the researchers).

Despite being a widely studied topic with several existing proposed approaches in the literature, creating expertise profiles still is a challenging task. Mainly, because even for humans it is difficult to quantify knowledge [14]. One of the problems that is frequently reported while assessing the quality of profiles obtained with automatic algorithms is that profiles are redundant (in the sense that there are several topics that address the same research area) and they are either too general or too specific (i.e., they either fail to detailed explain the knowledge of the researcher or fail to provide a global picture of his knowledge) [16]. One way to tackle this problem is to construct hierarchical expertise profiles where the knowledge of an expert is mapped along different granularity levels, from broader areas to more specific ones [16, 47, 48]. An hierarchical expertise profile requires that the topic modelling step also identifies hierarchical topics (this problem is referred as topical hierarchy construction). In the literature there are some algorithms proposed for the creation of topical hierarchies [49, 50]. However, these algorithms are intended only for topic modelling and mapping the experts into these structures is not trivial. Thus, these algorithms are not ideal for expertise profiling tasks. The few studies available in the literature that create hierarchical expertise profiles, use topical hierarchies that were manually created with the intent of easily mapping publications and authors into the topical hierarchy [47, 48].

We aim to develop a new algorithm for expertise profiling that creates hierarchical expertise profiles and does not require human intervention. In more detail, we aim to develop a network based algorithm for the creation of multi-typed topical hierarchies and present strategies to map authors into the topics, thus creating their hierarchical expertise profile. Furthermore, we intent that our algorithm is also capable of creating hierarchical expertise profiles for other entities such as publications, theses, journals and institutions.

1.2 Main contributions

This work consists of the design, implementation and evaluation of bibliometric methods. Namely, we develop methods to measure the scientific impact of authors, to estimate publications similarity and to create expertise profiles. Our methods are

more accurate than similar state-of-the-art solutions when compared in real-world scenarios and are scalable in the sense that they can be used in huge bibliographic databases. Bibliometric studies often require different evaluation criteria depending on the application. For example, for some applications it may be important to rank authors according to their citations received in more prestigious venues, while for others it may be important to rank them according to their most recent citations. As a result, our methods also target flexibility and allow the user to easily redefine the criteria used by the algorithms. Finally, our methods are capable of dealing with missing information. Most previous work in bibliometrics field, namely in the three previous tasks mentioned, assume (1) that bibliographic databases contain citation information and/or the text of publications and (2) that information is complete for all the publications. In the real-world scenario it is often the case where databases are not able to gather the same information for all the publications and end up having publications with partial information compared to others. Thus, the use of current bibliometrics methods in some real-world systems is limited. We make our tools available for practitioners, which are of great use to anyone that aims to study bibliographic databases. Next, we give a more detailed descriptions of our contributions.

OTARIOS. We propose a PageRank-based measure for author ranking named OTARIOS. OTARIOS analyses author citation networks to determine the scientific impact of authors. Compared to other author ranking methods, OTARIOS divides the network in two subnetworks, insiders and outsiders. The *insiders* consist of authors whose full scope of received citations is known, while the outsiders represent the group of author whose received citations are unknown. Thus, OTARIOS provides an efficient representation to deal with the problem of missing information in citation networks. Furthermore, OTARIOS combines multiple publication and citation criteria to rank authors giving the user the freedom to select the criteria that best represents his needs. We verify that OTARIOS outperforms other state-of-the-art author ranking algorithms in terms of creating a rank more similar to one created using human judgement. Our tests consisted of 5 different real-world citation networks from the computer science area.

FOCAS. we propose a penalty system for *friendly* citations named FOCAS. FOCAS aims at decreasing the impact of citations that result from the abuse of citation boosting strategies. Citation boosting is a well-documented problem in the literature, however author ranking algorithms do not address this problem (besides removing self-citations which is the most basic form of citation boosting). FOCAS combines

the citation and co-authorship networks to measure author proximity and penalises citations between *friendly authors*. FOCAS is an independent method that can be used along with any PageRank-based author ranking algorithm. FOCAS also presents a set of different criteria to measure author's proximity. Our experiments show that FOCAS improves the accuracy of the predicted ranks by author ranking algorithms. Our tests consisted of 8 different author ranking approaches, a real-world citation network from the computer science area and a ground-truth based on best paper awards.

PURE-SIM. We propose a new methodology to estimate publications similarities named PURE-SIM. PURE-SIM builds a network to analyse the relations between metadata and publications, and uses a stochastic process to estimate publications similarity. In contrast to other state-of-the-art approaches, PURE-SIM allows the user to control the information used as input (i.e., the metadata analysed) to measure similarity. This feature is particular important to tackle the problem of missing information in bibliographic databases. Additionally, PURE-SIM contains two other user-defined parameters that control the computational cost of the process and assign different weights to the network in order to differentiate the importance of certain metadata relations. We tested PURE-SIM on a bibliographic database that consisted of more than 4 million publications in the medicine area. We observed that PURE-SIM similarities lead to more accurate clusters when compared to other 11 state-of-the-art approaches.

HEPHIN. We propose an algorithm to create hierarchical expertise profiles named HEPHIN. Compared to other state-of-the-art approaches HEPHIN is the only algorithm that constructs hierarchical expertise profiles without human intervention. Our solution is divided in two parts. First, we developed a methodology to create multi-typed topical hierarchies through the analysis of the relations between publications and their metadata. Second, we introduced a strategy to map the knowledge of an entity (e.g., a researcher) into this structure in order to create hierarchical expertise profiles. Our experiments show the benefits of having hierarchical expertise profiles over the traditional ones (i.e., profiles consisting of only a flat line of topics) in a series of real-world applications. Namely, personal categorisation, temporal profile analysis and profile summarising. Furthermore, HEPHIN is also capable of creating hierarchical expertise profiles for publications, thesis and any other entities. We show that this is an important feature to tackle problems such as the expert recommendation one.

1.3 Thesis organisation

This thesis is structured into six major chapters. A brief description of each is provided below.

1. **Introduction** presents the main context of this thesis. This chapter introduces the areas of bibliometrics and network science. In more detail, we describe how networks can be used to aid bibliometric problems such as author ranking, publications clustering and expertise profiling. This chapter also enumerates the main contributions of this work, namely the proposed methods to measure the scientific impact of researchers, decrease the impact of citation boosting in author ranking, estimate publications similarities and create hierarchical expertise profiles. Finally, this chapter presents an overview of the thesis' contents and a bibliographic note.

2. **Background** introduces necessary network science concepts and terminology used throughout this thesis. In more detail it focus on three relevant concepts for this thesis: centrality measures, community detection and heterogeneous information networks. This chapter also provides a formal description and an analysis of the related work of the three main problems addressed in this thesis, namely author ranking, publications clustering and expertise profiling.

3. **Author ranking** motivates the problem and presents the advantages of our method over current related work. This chapter presents two proposed methods. First, we present OTARIOS which is a new feature enriched author ranking algorithm for incomplete networks. Second, we present FOCAS which is a penalty estimator algorithm that decreases the impact of friendly citations in author ranking. This chapter also presents the experiments made for OTARIOS and FOCAS which were conducted in real-world citation networks.

4. **Publications clustering** motivates the problem and presents the advantages of our method over current related work. This chapter introduces PURE-SIM, our proposed method that estimates publications similarities through the analysis of the relations between metadata and publications in bibliographic databases. It also presents our experiments with PURE-SIM on a real-world database and the comparison against other state-of-the-art approaches.

5. **Expertise profiling** motivates the problem and presents the advantages of our method over current related work. This chapter introduces HEPHIN, our

proposed method to create hierarchical expertise profiles. HEPHIN is presented in two parts. First, we describe the process to construct a multi-typed topical hierarchy through the analysis of the bibliographic database. Second, we present two strategies to map the knowledge of different entities into the topical hierarchy in order to create the hierarchical expertise profiles. This chapter also presents the HEPHIN experiments conducted in a real-world bibliographic database of Portuguese researchers. Our experiments are divided in two parts. First, we show that HEPHIN is able to identify coherent topics and that its expertise profiles are accurate. Second, we show the advantages of having hierarchical expertise profiles built on top of multi-type topical hierarchies on three other applications, namely temporal expertise profiling, expert recommendation and profile summarisation.

6. **Conclusions and future work** discusses the research done, summarises contributions, and gives directions for future work. This chapter also presents a conceptual design to integrate all the developed work into a single tool to measure scientific impact.

1.4 Bibliographic note

Parts of the work of this thesis have already been published in international conferences and journals. A list of those publications is provided next:

- **Expertise profiling.** A new algorithm named HEPHIN that constructs multi-typed topical hierarchies through the analysis of metadata relations between publications and maps the knowledge of the authors into the topical hierarchy to obtain their hierarchical expertise profile. We evaluated our approach on a dataset consisting of Portuguese researchers. Our experiments show that the multi-typed topical hierarchy identifies coherent topics. Furthermore, the hierarchical expertise profiles generated are accurate in the sense that they discriminate the knowledge of individuals at different granularity levels. This work was published in the 21st International Conference on Discovery Science (DS 2018) [51].

- **Author ranking.** A new method to estimate the scientific impact of authors named OTARIOS. We present the problem of estimating the impact of authors when the citation network is incomplete and propose OTARIOS as a solution

to the problem. We evaluate OTARIOS in citations networks and compare its accuracy against PageRank and three other baseline from citation counting methods. This work was published in the 7th International Conference on Complex Networks and Their Applications (Complex Networks 2018) [52]. An extended version of this document was published in Applied Network Science journal [53]. This extended work states that the current state-of-the-art author ranking approaches fail to efficiently combine features present in citations network. Furthermore, it shows that OTARIOS is more accurate than other current state-of-the-art approaches in the task of predicting author rankings more similar to one obtained using human judgement.

- **Author ranking.** A penalty estimator measure named FOCAS that aims to decrease the impact of citation boosting patterns in author ranking. We verify that *friendly* citations are frequent in real-world citation networks and that they are unevenly distributed among authors with different scientific impact. More concretely, more impactful authors (according to a human-based ground-truth) receive fewer *friendly citations*. We also show that adding FOCAS to author ranking algorithms improves their ability in creating more accurate ranks for the authors. This work was published in the 35th Annual ACM Symposium on Applied Computing (SAC 2020) [54].

- **Publications clustering.** A publication similarity estimator named PURE-SIM. PURE-SIM builds a network through the analysis of the relations between different elements in bibliographic databases and uses a stochastic process to estimate the similarities between publications. We evaluate the similarities estimated by PURE-SIM against 11 other state-of-the-art approaches in the task of publications clustering. We verify that PURE-SIM similarities lead to more accurate clusters than other approaches, while also being more flexible with the data necessary to compute publications similarities. This work was submitted to the Quantitative Science Studies journal and is currently under revision [55].

Chapter 2

Background

The aim of this work is to use network science concepts in large bibliographic databases to develop new tools for bibliometric studies. This chapter presents the basic concepts that are necessary for understanding the remaining content of this thesis. The first part of this chapter presents the reader with basic network science concepts with more emphasis on the tasks of community detection and node centrality. In the second part, we formally define and present related work for the three bibliometric tasks that we address: author ranking, publications clustering and expertise profiling.

2.1 Network concepts and terminology

This section introduces relevant network science concepts and notation used throughout this thesis.

Network (or graph). A mathematical structure that models pairwise relations between entities. Formally, a network is defined as an ordered pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of vertices of the network and $\mathcal{E}$ represents the set of edges of the network. The notation $\mathcal{V}(G)$ and $\mathcal{E}(G)$ is used throughout this thesis to represent the set of vertices and edges of a network G.

Vertices (or nodes). Set of entities that are represented in a network. A vertex v represents a component of a complex system that is modelled through a network. A vertex can also have attributes associated to it. For example, in the case of a

citation network, a vertex typically represents a publication and each vertex can have attributes such as authors, venue and keywords. The notation $|\mathcal{V}|$ is used throughout this thesis to represent the number of vertices of a graph G. This value is often used to indicate the size of a network.

Edges (or links). Set of relations between the vertices in a network. An edge $e \in \mathcal{E}$ is a pair of nodes (u, v) which indicates that a relation between nodes u and v exists. More formally, u is adjacent to v. The interpretation of a relation depends on the network. For example, in citation networks a relation represents a citation between publications, while in co-authorship networks a relation represents a co-authored publication between two authors. Depending on the network, edges could be directed or undirected. In case of a directed network the order of the pair (u, v) indicates that vertex u has a relation with vertex v. If a network is undirected the order of the pair is irrelevant since relations are mutual (i.e, if u is connected to v then v is also connected to u). A citation network is an example of an directed network since citations are directional, while co-authorship networks are an example of undirected networks since collaboration relations are mutual. Similar to vertices, edges may also have attributes associated to them. The most frequent attribute used in edges is the weight which is typically used to discriminate the strength of the relations in the network. For example, in co-authorship networks, higher weights in the edges indicate that two authors have co-authored more papers. The notation $u \rightarrow v$ is used throughout this thesis to represent the edge that starts at vertex u and ends in vertex v. Additionally, $|\mathcal{E}|$ is used to denote the number of edges of a graph G.

Subnetwork (or subgraph). A graph whose set of vertices and edges are a subset of other graph. A subgraph of G is represented as S_G, where $\mathcal{V}(S_G) \subseteq \mathcal{V}(G)$ and $\mathcal{E}(S_G) \subseteq \mathcal{E}(G)$.

Neighbourhood. The neighbourhood of a vertex v, denoted as $N(v)$, represents the set of vertices that are adjacent to v.

Vertex degree. Any vertex $v \in G$ has a certain degree denoted $deg(v)$. In case of an undirected network, the degree of a vertex v is the total number of edges that connect v to any other vertices. In case of directed networks, there are two degrees. The out-going degree $deg_{out}(v)$ which is the number of edges that start in v and end at any other node, and the in-going degree $deg_{in}(v)$ which is the number of edges that start at any other node and end in v.

Paths (or walks). A path $p(u \rightarrow v)$ in a network defines a set of edges that can be used to start at vertex u and travel to vertex v. The distance of a path numerical

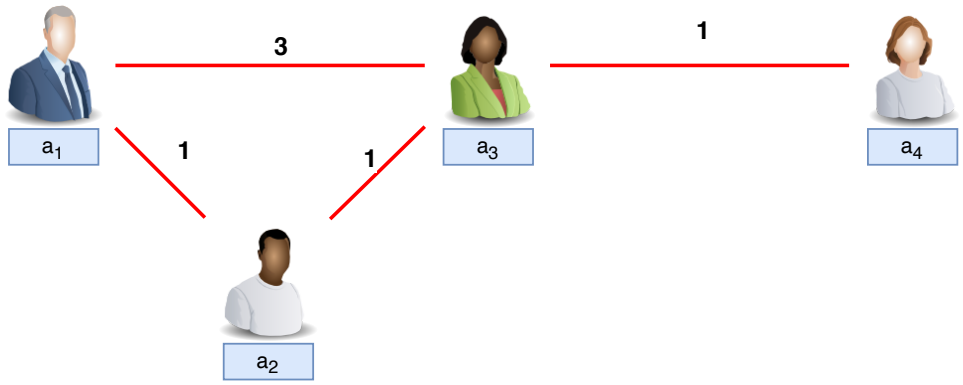


Figure 2.1: Example of a co-authorship network.

quantifies how far two vertices are from each other. Typically, the distance is the number of hops (i.e., the number of edges) that are present in the path. However, in networks where the edges have the weight attribute (i.e., weighted networks) the distance of a path is often estimated as the sum of the weights of the edges along the way. In network science, researchers are particularly interested in the shortest-path which refers to the minimum cost path between a pair of vertices. Throughout this thesis every time we use the notation $p(u \rightarrow v)$ to refer to a path we are referring to the shortest path between nodes u and v. Additionally, we use the notation $d(p(u \rightarrow v))$ to refer to the numerical value interpreted as the distance of the path.

Random walk. A random walk in a network represents a path that is constructed through a succession of random steps. Random walks start at a vertex v and then l random steps are taken. At each random step a neighbour of vertex v' is randomly selected. Note that in the beginning $v' = v$ and then at each random step v' is updated as the vertex selected in the previous step. In some cases the neighbours of v' are selected with equal probability, while in other cases the weight of the edges can be used as a probability distribution to select the next vertex. The number l is commonly referred as the length of the random walk.

Cluster coefficient. The cluster coefficient is a property of the network that indicates how connected the network is. The cluster coefficient measures the proportion to which the neighbours of a given vertex, are connected to each other. This value is estimated as the number of triangles (a triangle exists in a network if for any given nodes u, v and t, (u, v),(v, t) and (t, u) exist in $\mathcal{E}$) present in the network dividing by the total number of possible triangles. The cluster coefficient value ranges between 0 and 1, where 0 indicates that there are no triangles while 1 indicates that the network is complete (i.e., every vertex is connected to all other vertices).

To further clarify the reader, we use Figure 2.1 to exemplify the basic network concepts just described. Figure 2.1 represents a co-authorship network where vertices represent authors and edges indicate that two authors have co-authored at least one publication. The network is undirected since author a_1 co-authoring with author a_2 also implies that a_2 has co-authored with a_1. Each edge in the network contains the number of publications that two authors have co-authored together. For example, the value 3 on the edge (a_1, a_3) compared to the value 1 on the edge $(a1, a2)$ indicates that a_1 has a stronger co-authorship relation with author a_3 than with author a_2. The neighbourhood of vertex a_3 consists of vertices: a_1, a_2 and a_4. We also observe that a_3 presents the highest degree with the value of 3 (i.e., a_3 is the author with the most collaborations in the network). Additionally, the network has an average vertex degree of 2 (resulting from the dividing the sum of the individual vertex degrees, 3+2+2+1, by the number of vertices, 4). With respect to paths, the average short-path distance in the network is 1.33 which is relatively low but comes from the fact that only a few pair of vertices are not adjacent to each other. The largest short-path distance in the network (also known as network diameter) is 2 which is represented by the path from node a_4 to either a_1 or a_2. The cluster coefficient is 0.25 since the network has one triangle between vertices a_1, a_2 and a_3 out of the 4 possible triangles to form with 4 vertices.

2.1.1 Centrality measures

In network science, centrality measures rank the importance of individual nodes in a network. Using these metrics is often necessary to understand and identify important elements of a complex system. For example, centrality measures can be used to identify the most influential persons in a social network, key infrastructure nodes in power lines or internet and super spreaders in case of a pandemic [17].

The standards that define the importance of a node in a network are not universal. For example, a common measure for power in networks is the node degree. However, considering the example of Scott Adams, the creator of a web comics named Dilbert [1]:

> "The power a person holds in the organisation is inversely proportional to the number of keys on his keyring. A janitor has keys to every office, and no power. The CEO does not need a key: people always open the door for him.".

[1] http://dilbert.com/

In this scenario, the degree (i.e., the number of keys in the keyring) is not a good representation of the distribution of power in the system. In fact, the higher the degree, the lesser important the person is in the company hierarchy. The definition of importance in a network depends on the nature of the relations and on what the researcher is looking for [56]. As a result, several centrality measures have been proposed in the literature. Next we review some of the most frequently used ones:

- **Degree Centrality (DC):** Nodes with higher degree have more *connections* therefore they are more important in the network. The degree centrality of a node v is its own degree $deg(v)$. For directed networks, two types of degree centrality can be estimated using the in-coming edges ($deg_{in}(v)$) or the out-going ones ($deg_{out}(v)$).

- **Closeness Centrality (CC):** Nodes that are central to the network are more important. The closeness centrality of a node v is calculated as follows:

$$CC(v) = \frac{|\mathcal{V}|}{\sum\limits_{u \in \mathcal{V}(G)} d(p(u \to v))} \tag{2.1}$$

- **Betweenness Centrality (BC):** Nodes that act as bridges for communications over the network are more important. The betweenness centrality of a node v is calculated as follows:

$$BC(v) = \sum_{u,t \in \mathcal{V}(G)} \frac{\sigma(u,t|v)}{\sigma(y,z)} \tag{2.2}$$

  where $\sigma(u,t)$ is the number of shortest-paths between u and t (i.e., the number of possible paths between u and t where the distance is minimal), and $\sigma(u,t|v)$ is the number of shortest-paths that pass through node v.

- **PageRank (PR):** Nodes that are more important receive (in the sense of in-coming edges) more connections from important nodes. PageRank was originally developed by Google and it was used to rank web pages in search engine results. The general assumption is that it is important for a web page to have many in-coming links (i.e. other web pages with hyperlinks to it) but it is even more important if those in-coming links come from other important pages. This PageRank idea quickly became important for many other types of networks and

systems. Nowadays, PageRank is one of the most widely used algorithms to estimate node centrality. The PageRank of a node v is calculated as follows:

$$PR(v) = (1 - d) + d \times \sum_{u \in N_{in}(v)} \frac{PR(u)}{|N_{out}(u)|} \tag{2.3}$$

where d is the damping factor, $N_{in}(v)$ is the set of nodes that have in-coming edges to node v and $|N_{out}(u)|$ is the number of out-going edges that start on node u.

2.1.2 Community detection

In network science, communities represent groups of nodes that are more likely to share relations with each other than with other nodes from other communities. Detecting communities in a network is important to understand factors that divide the network into denser groups. For example, studying the communities of the cellphone calls network in Belgium, led to the conclusion that language differences is what divides the network [17]. Communities are particular important for social and biological networks since they make it easier to identify the latent reasons that divide the groups of nodes assigned to each community.

2.1.2.1 Community definition

A community is defined as a locally dense connected subgraph. This is a loose definition, thus there are a large number of subgraphs in a network that fit in this definition. To restrict the number of potential communities in a network, some other community definitions were proposed in the literature. One example is the clique community where a subgraph is a community only if it is a clique subgraph (i.e., nodes are all connected to each other). Clique subgraphs are frequent in real-world networks, however these subgraphs are of small size. Consequently, community detection based on the clique rule, results in a large number of communities but all of them are of small size. This type of community partition is probably not very useful. To overcome this problem, softer community definitions were also proposed.

Consider a subgraph $S(G)$, where the internal degree k_u^{int} of a node u represents the number of edges between u and any other node in $S(G)$, and the external degree k_x^{out} represents the number of edges between u and any other node outside of $S(G)$. Two other possible definitions of community are [17]:

- **Strong Community:** every node in $S(G)$ has more internal edges than outside ones. In particular:

$$k_u^{int}(S(G)) > k_u^{ext}(S(G)),\ \forall u \in S(G) \tag{2.4}$$

- **Weak Community:** the total internal degree of $S(G)$ exceeds its total external degree. In particular:

$$\sum_{u\ \in\ S(G)} k_u^{int}(S(G)) > \sum_{u\ \in\ S(G)} k_u^{ext}(S(G)) \tag{2.5}$$

Strong communities allow nodes that are not totally connected to the community, but still are more connected to nodes in the community than to nodes outside of it. On the other hand, weak communities allow nodes that are more connected to nodes outside of the community, as long as the total number of connections within the community is greater than the total number of connections outside of it. The rankings of the three discussed community definitions according to their restrictiveness (increasing order) are: weak, strong and clique. The process of studying communities in a network starts with the definition of a community to use. This decision should take into consideration the real system that is being studied and the information that the user expects to retrieve from the communities. For example, in a co-authorship networks, weak communities are more likely to identify groups of authors that work in the same topic while clique communities identify groups of authors that have published together.

2.1.2.2 Community detection algorithms

Even with a clear definition of a community, identifying communities in a network is a hard problem. The straightforward solution consists of testing every possible network partition (i.e. all possible communities with different size and combinations of nodes) to determine which one leads to the best communities according to the selected community definition. However, this is impracticable even for small networks since the number of possible partitions of a network grows exponentially with the number of nodes. Thus, making community detection a NP-hard problem. The literature on community detection algorithms is a vast topic with several different strategies proposed. In this thesis we do not develop new community detection algorithms, instead we use and adapt existing ones in order to tackle bibliometric problems. Therefore, in this section we only discuss the algorithms used throughout this study. For a more detailed overview of community detection algorithms in networks we

recommend the study of Javed et al. [57] which presents an interesting taxonomy of the algorithms available.

In this thesis, we use a category of community detection algorithms named greedy optimisation. These algorithms utilize a greedy strategy to search for a partition of the network that maximises a certain quality function. Therefore, they have two essential components: quality function and searching strategy. First, we address the quality function component. In the context of community detection, a quality function measures whether a group of nodes (i.e., a partition) is a good candidate to from a community. The most widely used quality function in community detection algorithms is modularity. The general idea of modularity is that a partition represents a community if the number of internal edges exceeds the number of edges that one expects to find in the partition if the network were randomly wired (i.e., edges were randomly distributed over the network) [58]. The modularity of a certain partition of the network is given by:

$$\mathrm{M} = \frac{1}{2|\mathcal{E}|} \sum_{c \in \mathcal{C}} \left(e_c - \gamma \frac{K_c^2}{2|\mathcal{E}|} \right) \tag{2.6}$$

where $\mathcal{C}$ is the set of communities in the partition, e_c is the number of internal edges in community c, $\frac{K_c^2}{2|\mathcal{E}|}$ is the expected number of edges considering that K_c is the sum of degrees of the nodes in community c and γ is a resolution parameter. $\gamma > 0$ with higher resolutions leading to more communities, while lower resolutions resulting in fewer communities. The higher the value of M the better the community partition is. Although modularity is widely used in community detection, it presents some well-known limitations:

- **Limited resolution.** In several real-world systems it is common to have a small community that it is also part of a bigger community. In most cases, merging the smaller communities leads to a greater modularity score. Thus, community detection algorithms that aim to maximise modularity often fail to identify smaller communities. In the literature, this problem is referred as the modularity limited resolution [59].

- **Modularity maxima.** As the search for the optimal community partition progresses, it becomes increasingly more difficult for the community detection algorithm to select the next communities to merge or separate since the variations in the quality function score among the possible solutions is minimal. At each

step this selection can lead to very different results in the final solution. For example, at a certain step merging community c_1 with c_2 instead of merging community c_1 with c_3 can lead to large differences in the modularity score of the final communities partition. Another drawback using modularity as the quality function is that it is not possible to estimate an approximation of the maximum modularity of a network. Thus, it is difficult to detect problems in the search process of the community detection algorithm and to assess if the produced solution is the best one[17].

The Constant Potts Model (CPM) [60] is an alternative quality function that was proposed to overcome the limited resolution problem of modularity. The general idea of CPM compared to modularity is that the network should be compared to a constant factor instead of to a random null model (as is the case for modularity). The CPM of a certain partition of the network is given by:

$$\text{CPM} = \sum_{c \in \mathcal{C}} \left[e_c - \gamma \binom{n_c}{2} \right] \tag{2.7}$$

where $\mathcal{C}$ is the set of communities in the partition, e_c is the number of internal edges in community c, n_c is the number of nodes in community c and γ is the resolution parameter. Similarly to modularity, higher resolutions result in more communities while lower resolutions result in fewer communities. The community detection algorithms utilised in this thesis either use modularity or CPM as quality function.

With respect to the searching strategy we utilise the ones presented in two well-known community detection algorithms: Louvain [61] and Leiden [62]. The Louvain algorithm (Algorithm 2.1) optimises the quality function by repeatedly executing two steps: local moving of nodes and network aggregation. In the first step, individual nodes are moved to the community that yields the largest increase in the quality function (lines 6 and 7). In the second step, the communities formed in step one are treated as nodes of an aggregate network (line 8) and individual nodes are again moved to the community that yields the largest increase in the quality function (lines 9 and 10). Note that during the second step, an individual node in the aggregate network represents a group of nodes in the original network. Thus, the second moving of the nodes represents the merging of two or more communities. The Louvain algorithm starts with every node belonging to its own community (line 1) and step one and two are repeated until it is not possible to improve the quality function with new partitions (lines 5 and 11-13).

Algorithm 2.1 Louvain community detection algorithm.

Input: Network $G = (\mathcal{V}, \mathcal{E})$ and a quality function θ.
Output: Set of communities $\mathcal{C} = \{c_1, c_2, ..., c_k\} : v \in \mathcal{V}$, $v \in c_i$ and $c_i \in \mathcal{C}$.

1: $\mathcal{C} = \{c_i = v_i : \forall\ v_i \in \mathcal{V}\}$
2: $\mathcal{C}'' = \mathcal{C}$
3: **while** True **do**
4: $\quad \mathcal{C} = \mathcal{C}''$
5: $\quad \text{Current}_{quality} = \theta(\mathcal{C})$
6: $\quad$ **for** $v \in \mathcal{V}$ **do**
7: $\quad\quad \mathcal{C}' = \text{MoveToMaxQuality}(v, \mathcal{C}, \theta)$
8: $\quad \mathcal{V}' = \{v_i = c_i : \forall\ c_i \in \mathcal{C}'\}$
9: $\quad$ **for** $v \in \mathcal{V}'$ **do**
10: $\quad\quad \mathcal{C}'' = \text{MoveToMaxQuality}(v, \mathcal{C}', \theta)$
11: $\quad \text{New}_{quality} = \theta(\mathcal{C}'')$
12: $\quad$ **if** $\text{New}_{quality} \leq \text{Current}_{quality}$ **then**
13: $\quad\quad$ break
14: **return** $\mathcal{C}$

In 2019, Traag et al. [62] showed that the Louvain algorithm may find arbitrarily badly connected communities and proposed the Leiden algorithm (Algorithm 2.1) to overcome this problem. The Leiden algorithm still uses step one and two from the Louvain algorithm (lines 6-7 and 10-12) however it introduces the refinement step in between them (lines 8-9 and 13-14). The idea of the refinement step is to identify nodes whose community membership is not clearly defined (i.e., moving the node to a certain community presents approximately the same gain in the quality function as moving it to another one) and split them from their communities in order to broader the search space in the next algorithm iteration. According to the authors, the refinement step guarantees that no badly connected communities are part of the final community partition solution.

2.1.2.3 Overlapping communities

The algorithms discussed in the previous section aim at disjoint community detection. More concretely, nodes are exclusively assigned to a single community. Yet, in real-world systems it is often the case where an element (i.e., a node) is part of multiple communities. For example, consider a co-authorship network of researchers, a senior researcher can work with several different graduating students, with his research group and have collaborations with external research groups. As a result he is a member

Algorithm 2.2 Leiden community detection algorithm.

Input: Network $G = (\mathcal{V}, \mathcal{E})$ and a quality function θ.
Output: Set of communities $\mathcal{C} = \{c_1, c_2, ..., c_k\} : v \in \mathcal{V}$, $v \in c_i$ and $c_i \in \mathcal{C}$.

$\mathcal{C} = \{c_i = v_i : \forall\ v_i \in \mathcal{V}\}$
$\mathcal{C}''' = \mathcal{C}$
while True **do**
    $\mathcal{C} = \mathcal{C}''$
    $\text{Current}_{quality} = \theta(\mathcal{C})$
    for $v \in \mathcal{V}$ **do**
        $\mathcal{C}' = \text{MoveToMaxQuality}(v, \mathcal{C}, \theta)$
    for $c \in \mathcal{C}'$ **do**
        $\mathcal{C}' = \text{SplitCommunities}(c, \mathcal{C}')$
    $\mathcal{V}' = \{v_i = c_i : \forall\ c_i \in \mathcal{C}'\}$
    for $v \in \mathcal{V}'$ **do**
        $\mathcal{C}'' = \text{MoveToMaxQuality}(v, \mathcal{C}', \theta)$
    for $c \in \mathcal{C}''$ **do**
        $\mathcal{C}'' = \text{SplitCommunities}(c, \mathcal{C}'')$
    $\text{New}_{quality} = \theta(\mathcal{C}'')$
    if $\text{New}_{quality} \leq \text{Current}_{quality}$ **then**
        break
return $\mathcal{C}$

of several communities. Figure 2.2 illustrates an example of non-overlapping and overlapping communities. The phenomenon of overlapping communities was first studied by Palle et al. [63]. Since then several algorithms to detect overlapping communities have been proposed [57].

In this thesis, we use the concept of overlapping communities because it is common in bibliographic data that nodes are part of multiple communities. However, we do not develop new algorithms for overlapping community detection. Instead, we develop methodologies that use network projection to use algorithms such as Louvain and Leiden community detection to detect overlapping communities. For this reason we do not provide a detailed overview of existing overlapping community detection algorithms in the literature.

2.1.2.4 Testing communities

The communities of a network could lead to interesting discoveries about a complex system. However, a single community detection algorithm fails to perform the best

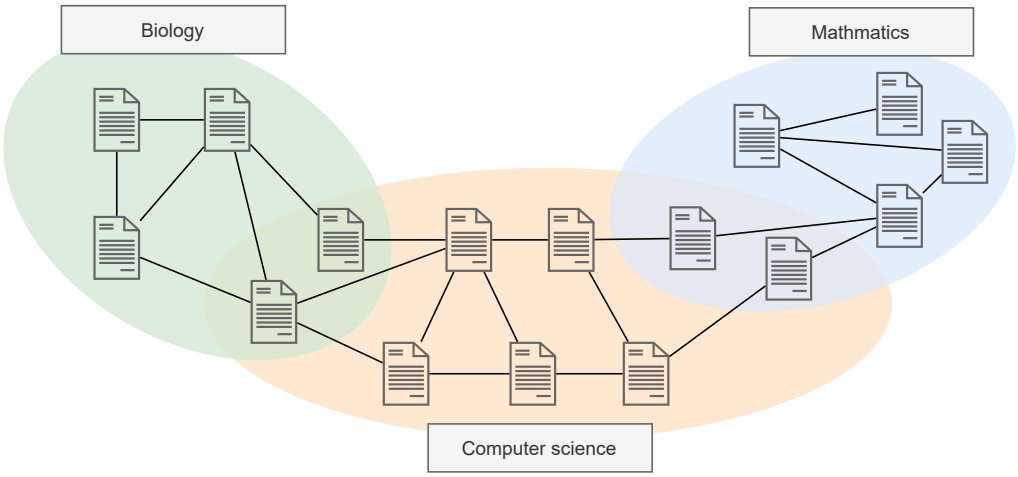


Figure 2.2: Example of overlapping communities.

in all networks due to the variety of networks that can be generated from different systems [57]. With so many community detection strategies, it is not trivial to assess which one performs the best for a certain network. Ideally, we would estimate the number of nodes that are correctly assigned to the communities versus the number of incorrect assignments to benchmark community detection algorithms. However, in real systems the *"correct"* communities are unknown, hence they cannot be used as a comparable ground-truth. Furthermore, quality functions such as modularity or CPM are also not option since the *"correct"* communities are not necessarily the ones with the highest values of modularity or CPM.

In the literature, studies that propose new community detection algorithms often recur to models that generate random networks with known community structure [17]. The main goal of these models is to generate random networks with similar properties to real systems but where the distribution of nodes per communities is known a priori. In this thesis we also require comparing the communities generated from our approaches with the ones obtained from state-of-the-art algorithms. However, in our comparisons, communities represents topics of a traditional publications clustering or topic modelling problem and can be evaluated using other metrics. For this reason, we do not present a detailed description of methods to test communities.

2.1.3 Heterogeneous information networks

Several real-world systems that are modelled by networks consist of multiple types of elements (or nodes) and relations (i.e., edges that connect different types of nodes).

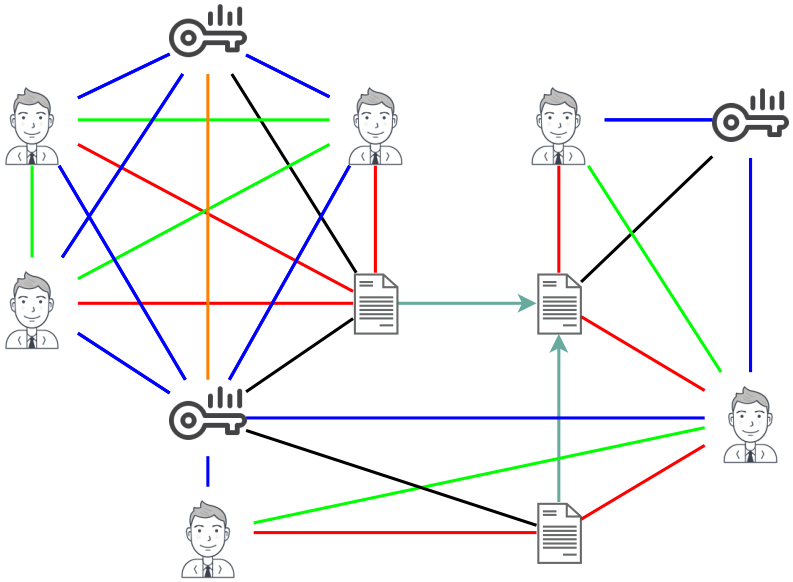


Figure 2.3: Example of a bibliographic heterogeneous information network. Different symbols represent different entities (i.e., node types) and different edge colours (i.e., edge types) represent different relations between entities.

This heterogeneity of components is ignored by traditional networks (also referred as homogeneous information networks) that represent the entire system using a single type of nodes and edges. Thus, leading to the loss of potentially valuable information.

In 2009, Sun et al. [49] introduced the term Heterogeneous Information Network (HIN). On this type of network, nodes and edges have a type associated to them. As a result, it is possible to take into account the heterogeneity of real-world systems while modelling their information. Figure 2.3 illustrates an example of a HIN modelling the relations in a bibliographic database. On this example we have three different types of nodes: publications, authors and keywords. Additionally, we have six different types of edges: publication to publication, publication to author, publication to keyword, author to author, author to keyword and keyword to keyword. Compared to homogeneous information networks, HINs can efficiently model more information and contain richer semantics which allows new developments in data mining and network science strategies [3].

2.1.3.1 Basic concepts

An information network represents abstractions of the real-world, focusing on the elements and their relations. Information networks are defined as follows [3]:

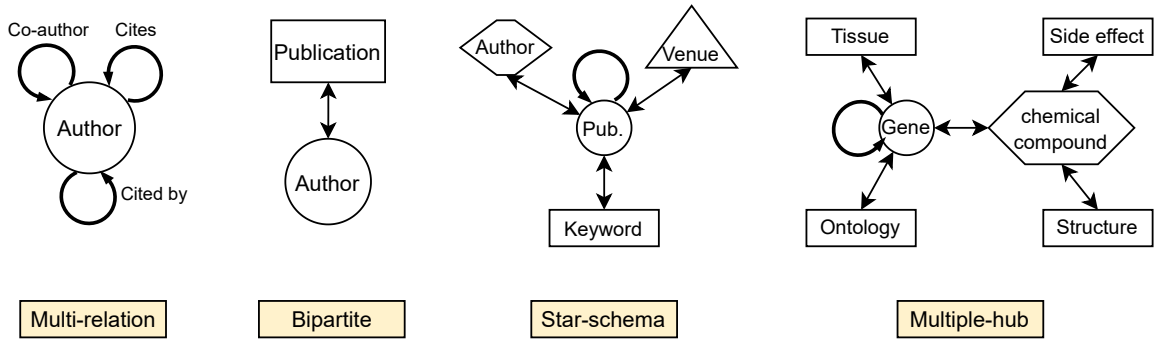


Figure 2.4: Some examples of information network schema. This image was adapted from [3].

Definition 2.1 *Similarly to the previous definition of a network (or graph), an information network is also represented as an ordered pair* $G = (\mathcal{V}, \mathcal{E})$. *However, information networks also contain an object type mapping* $\psi : \mathcal{V} \rightarrow \mathcal{A}$ *and a link type mapping* $\varphi : \mathcal{E} \rightarrow \mathcal{R}$. *Each node* $n \in \mathcal{V}$ *belongs to an object type* $a : \psi(n) \in \mathcal{A}$. *Furthermore, each edge* $e \in \mathcal{E}$ *belongs to a relation type* $r : \varphi(e) \in \mathcal{R}$. *If two edges share the same relation type, they both start at a node with type* a' *and end at a node with type* a''.

An information network is heterogeneous if $|\mathcal{A}| > 1$ or $|\mathcal{R}| > 1$. In order to better understand the elements and their relations, heterogeneous information networks have a meta-level description named network schema [64].

Definition 2.2 *A network schema* $T_G = (\mathcal{A}, \mathcal{R})$ *is a meta template for an information network* $G = (\mathcal{V}, \mathcal{E})$ *with the node type mapping* $\psi : \mathcal{V} \rightarrow \mathcal{A}$ *and an edge type mapping* $\varphi : \mathcal{E} \rightarrow \mathcal{R}$ *where the nodes represent the different node types in* $\mathcal{A}$ *and the edges represent the different possible types of edges in* $\mathcal{R}$. *Thus, enumerating the possible relations existing in the information network.*

A network schema specifies the possible relations among different nodes in the information network. Figure 2.4 illustrates some examples of commonly used network schema. The multi-relation shows a case where an information network consists only of a single type of node (in this case an author) but where the nodes can have multiple different relations (in this case an author can be connected to another one through the relations of co-authorship, citing or being cited). Bipartite networks consist of two types of nodes and an unique relation that relates nodes from type a' with nodes from type a'' or vice-versa. In bibliographic data, some frequently used examples of

bipartite networks are publications-words and publications-authors networks. A star-schema assumes that one type of node (referred as star type) is the common link between the other types of nodes (referred as attributes). In this schema, all the edges must either start and/or end in a star type node. The star-schema is a very popular schema to model bibliographic databases due to the fact that publications are often the central element that connects information such as: authors, keywords and venues. A multiple-hub schema represents a more complex system where two or mode types of nodes act as stars (or hubs). Each star has a similar function to the star-nodes in star-schema networks (i.e., can connect to other attribute-node types that can only connect to their star type) and star type nodes can connect to each other even if they have different types. This type of network schema is often used in bioinformatics data and has not been used to model bibliographic data yet.

In an heterogeneous information network two types of nodes can be connected through several different types of nodes. These paths are named meta-paths [3].

Definition 2.3 *A meta-path P is a path defined on a network schema $T_G = (\mathcal{A}, \mathrm{R})$ and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} ... \xrightarrow{R_l} A_{l+1}$, which defines a walk trough between node types $A_1, A_2, ..., A_l + 1$ through edge types $R_1, R_2, ..., R_3$.*

In an HIN that models bibliographic data, different meta-paths are useful to discriminate relations. For example in the star-schema illustrated in Figure 2.4, two different meta-paths can be used to model relations among authors. The Author-Publication-Author represents a co-authorship relation, while the Author-Publication-Venue-Publication-Author represents two authors that published in the same venue. Modelling interactions between objects using different relations is an important characteristic of HINs. This concept has been widely used in data mining tasks such as similarity measure, community detection, and classification [3].

2.1.3.2 Community detection in heterogeneous information networks

Similarly to homogeneous networks (i.e., networks with only one type of nodes and one type of edges), studying the community structure of HINs also reveals valuable information about the system that is being modelled. The literature on community detection algorithms for homogeneous networks is vast, however these algorithms are not adequate for HINs since they disregard nodes and edges heterogeneity which leads to the loss of valuable information about the system. To overcome this problem some

studies have proposed community detection algorithms for HINs. In this thesis, we do not use any community detection algorithm for HINs. Still, they can replace the community detection methods used in some of our proposed algorithms. For this reason we present a brief discussion of the algorithms available. Throughout this section (and this thesis overall) the terms community detection and clustering are used to describe the same task.

In 2009, Sun et al. [65] proposed a ranking based algorithm to detect communities in bipartite HINs named RankClus. With respect to HINs, a ranking function aims to discriminate the importance that different nodes or types of nodes have in the network structure. The idea of ranking based clustering consists in improving the communities detected through the use of the conditional ranking distribution of nodes and their types. More concretely, Given a bipartite network with node types a' and a'', and a random partition of a' in k communities a'_i, each node $n \in a''$ can be defined as a mixture model over k conditional ranks of a''. The RankClus algorithm is described as the following steps:

1. Randomly initialise k clusters for nodes.
2. Based on current clusters, calculate conditional rank for type a'' and a', and within cluster rank for type a'.
3. Estimate parameter θ in the mixture model, get new representations for each node and centre for each cluster.
4. Calculate distance from each node to each cluster and assign the node to the nearest cluster.
5. Repeat steps 2, 3 and 4 until there is no significant change in the clusters or a maximum number of iterations is reached.

Following the idea of ranking based clustering, Chen et al. [66] proposed the GPN-RankClus which adapts the ranking function used in RankClus to a gamma distribution within each cluster. Thus, allowing high ranking values for every cluster. In 2014, Shi [67] proposed the HeProjI algorithm which extends the concept of ranking based clustering from bipartite HINs to arbitrary schema. For any HIN, the authors presented the idea of decomposing the network into k-partite subnetworks, where each bipartite subnetwork models a different relation in the HIN. Then, community detection algorithms for bipartite networks and mechanisms for information transfer can be used to detect communities in the entire HIN. In the case of HeProjI, the algorithm uses a random walk model to select the best nodes that can be used for

ranking and clustering the entire network, but other approaches followed the authors ideas and presented new algorithms [68, 69].

In the same year as RankClus was proposed, a similar algorithm named NetClus [49] was also presented. The main difference between both algorithms is that NetClus targets star-schema networks. The idea of NetClus is to define the probability of a star-node n belonging in a community c_i as the product of the probability of the attribute-nodes connected to n belonging in community c_i. Similarly to RankClus, the algorithm also starts with a random partition of nodes but uses a different ranking function to adjust the communities detected.

In addition to the discussed strategies there are some innovative ideas to solve the problem. Gupta et al. [70] identifies the most relevant meta-paths according to the cluster goal and use them to select a set of seed nodes for each community. Then, the initial communities are expanded until every node is part of a community. Wu et al. [71] use tensors to restructure the network and non-negative tensor decomposition to detect communities.

2.2 Author ranking

Input. A set of authors $\mathcal{A}$ and a set of publications $\mathcal{P}$ authored by authors in $\mathcal{A}$.

Problem description. Analyse the publications in $\mathcal{P}$ and produce a score for each author $a \in \mathcal{A}$ that represents his scientific impact.

Output. A produced rank Pr, which is a vector of the positions and scores of each author in the ranking, ordered from highest to the lowest scores.

The author ranking task analyses the scientific output of authors in order to estimate their scientific impact. Author ranking is often necessary to identify the most prominent authors working in a certain scientific area. The common agreement in the research community is that the impact of an author is directly related to the citations received by his documents. In simple terms (which do not hold truth for all the metrics available in the literature), the more citations an author receives, the higher his impact is. These metrics are often referred as citation-based. More recently, a new group of metrics named Altmetrics have emerged [72]. The goal of Altmetrics is to measure and monitor the impact of published documents beyond the scope of citations. For the purpose, Altmetrics can use data such as references on online encyclopedias (e.g., Wikipedia), research blogs, media coverage, bookmarks on reference managers

(e.g., Mendeley) and mentions on social media (e.g., Twitter). Altmetrics were not proposed to replace citation-based metrics but to complement them [73]. In this work, we address the problem of citation-based problems since they are still the standard on author ranking problems.

The most widely used citation-based metric in the research community is the h-index [12]. According to the h-index, the impact of an author is given by the number of his published documents with more than h citations. Despite being widely used, traditional citation-based metrics such as h-index fail to capture the nature of scientific development since they disregard the fact that a new discovery is not solely due to previous work directly referenced. As an alternative, graph-based metrics that analyse the citation graph have been proposed. These metrics consider multi-step citation relations instead of only considering one-step relations (i.e., direct citations). Thus, they can effectively spread the credit of a citation to previous works that paved the way [24]. In this thesis we focus on graph-based author ranking methods.

2.2.1 Graph-based author ranking

There are two groups of graph-based author ranking methods: paper-level and author-level [24]. Paper-level strategies use the papers' citation network to diffuse scientific credit to cited papers, and then authors' scientific impact is derived from the credit of their papers [19, 29]. Conversely, author-level strategies use the authors' citation network to diffuse scientific credited to cited authors. As a result, the authors' credit is directly obtained from the network [20, 26, 74]. Figure 2.5 illustrates the differences between paper and author citation networks. Independent of the type of citation network, graph-based author ranking consists in determining the more powerful nodes in a network. As a result, any centrality measure for networks can be used to tackle the problem. In particular, PageRank [75] is one of the most widely used measures in author ranking. The definition of power in the network according to PageRank is based on the principle that it is good to have links, but it is better to have links to important nodes. This principle naturally fits the idea of author ranking in the sense that it is better to be cited by other authors with high scientific impact than by authors with low scientific impact. Thus, PageRank is the core of most graph-based author ranking methods available.

There are several graph-based author ranking methods proposed in the literature which are based on the PageRank algorithm. Their key differences are the citation network used, the score initialisation and the citation weight. The citation network can be

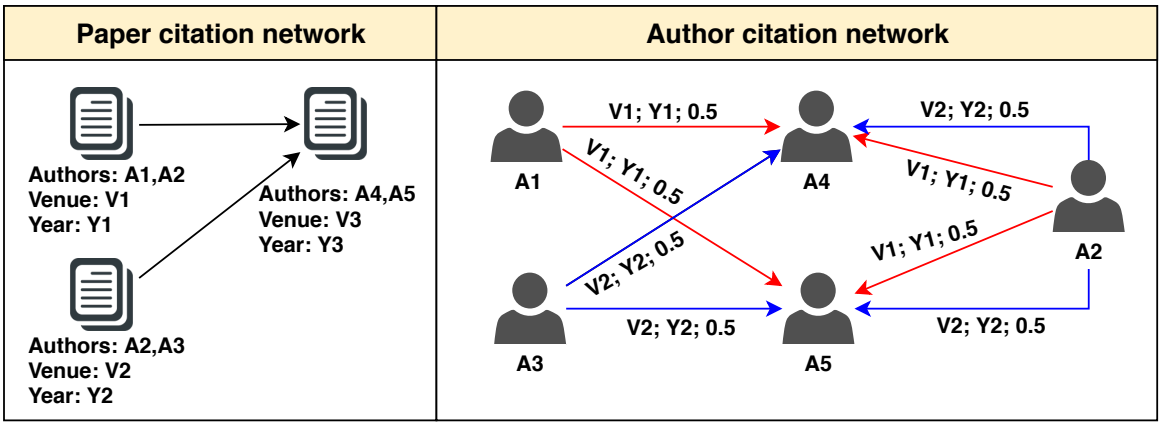


Figure 2.5: Comparison of paper-level and author-level networks.

either paper-level [19, 29] or author-level [20, 25, 26, 28]. The score initialisation defines an initial score for the nodes (i.e., authors or papers) before considering citations relations in the network. Some methods assume that initially all the nodes have an uniform score [25, 28], while others assign higher score to authors with more published work [20, 26], to more recent papers [29, 19] or to papers published in more prestigious venues [19]. The citation weight defines the importance of the citations in the network. More important citations give more credit to the authors receiving the citation. Author-level methods assign more importance to citations where the document that received the citation has fewer authors [20, 25, 26, 28]. With respect to paper-level methods, some assign more importance to more recent citations [29] while others consider more recent citations and citations coming from more prestigious venues [19]. In Chapter 3 we introduce terminology for our method and we present the state-of-the-art methods in more detail.

2.2.2 Evaluation

The quality of an author ranking algorithm can be measured by comparing the produced ranking Pr against a ground-truth ranking Gr which is created using human judgement. The goal of the author ranking algorithm is to produce a rank as similar as possible to the ground-truth one, without requiring human effort and without knowing the ground-truth rank a priori. Some of the most popular data used as ground-truth ranking are: research awards (e.g., awards given by research institutions), journal awards (e.g., best paper awards in conferences) and peer-review systems (e.g., research funding allocation decisions).

We compare the rankings Pr and Gr using the Normalised Discounted Cumulative

Gain (NDCG) [76] and the Mean Reciprocal Rank (MRR) [77]. Both NDCG and MRR are estimated in relation to the top-n authors, denoted by NDCG@n and MRR@n, respectively. For example, NDCG@10 represents the computation of the NDCG measure for the top 10 highest ranked authors of the produced ranking Pr. NDCG is based on the DCG measure which is computed using the following equation:

$$\text{DCG@}n = \sum_{p=1}^{n} \frac{Gs(a)}{log(p+1)} \tag{2.8}$$

where, $Gs(a)$ is the ground-truth score of author a. For each position $p \in \{1..n\}$, we find the author a ranked in position p in the produced ranking Pr. Then, we use the ground-truth score $Gs(a)$ as an indicator of the quality of an author. This value is divided by $log(p+1)$ to discriminate the importance of placing bad or good authors in the top positions. For example, placing an author with low $Gs(a)$ in the 1st position of Pr presents an higher penalty than placing the same author in the 3rd position. In general, a Pr that puts authors with high Gs in the top-n positions has high DCG.

NDCG is a normalisation of the DCG, thus NDCG@$n \in [0, 1]$. NDCG is obtained using the following equation:

$$\text{NDCG@}n = \frac{\text{DCG@}n}{\text{IDCG@}n} \tag{2.9}$$

where, IDCG@n is the ideal DCG score, i.e., the DCG@n score obtained when the top-n positions of the produced ranking Pr, perfectly match the ones of the ground-truth ranking Gr. The IDCG@n is also obtained using Equation 2.8. However, to compute IDCG, for each position $p \in \{1..n\}$, we find the author a ranked in position p in the ground-truth ranking Gr.

MRR@n is obtained using the following equation:

$$\text{MRR@}n = \frac{1}{n} \times \sum_{p=1}^{n} Gr(p) \tag{2.10}$$

where, $Gr(p)$ is the position of an author in the ground-truth ranking GR. Similarly to the NDCG measure, for each position $p \in \{1..n\}$, we find the author a ranked in position p in the produced ranking Pr. MRR@n is the average ground-truth rank of the top-n authors in the produced ranking Pr. The MRR is not normalised, thus MRR@$n \in [\frac{\sum_{p=1}^{n} p}{n}, \frac{\sum_{p=1}^{n} |A|-p+1}{n}]$, where $|A|$ is the number of authors to rank.

Table 2.1: Toy example of five authors and their corresponding produced ranking (Pr), ground-truth ranking (Gr) and ground-truth score (Gs).

a	$Pr(a)$	$Gr(a)$	$Gs(a)$
1	#5	#2	4
2	#4	#3	2
3	#3	#4	2
4	#2	#1	5
5	#1	#5	1

Consider a toy example of five authors a with a certain ground-truth score $Gs(a)$, ground-truth rank $Gr(a)$ and produced rank $Pr(a)$. Table 2.1 shows this toy example. In this example:

$$\text{NDCG@3} = \frac{\frac{1}{log(2)} + \frac{5}{log(3)} + \frac{2}{log(4)}}{\frac{5}{log(2)} + \frac{4}{log(3)} + \frac{2}{log(4)}} \approx \frac{17.12}{28.32} \approx 0.61$$

$$\text{MRR@3} = \frac{5+1+4}{3} \approx 3.33$$

$$\text{MRR@3} = \frac{5+1+4+3+2}{5} \approx 3$$

Produced rankings with high NDCG and low MRR are better.

2.3 Publications clustering

Input. A set of publications $\mathcal{P}$.

Problem description. Measure the similarity $s_{p,q}$ between every pair of publications $(p, q) \in \mathcal{P}$. Partition the publications into n clusters based on the publications similarity.

Output. The cluster c_p of each publication $p \in \mathcal{P}$.

Publications clustering refers to the task of dividing a corpus of publications into clusters based on the principle that *"publications that are more similar should be part of the same cluster"*. The general idea of publications clustering is that each cluster c represents a different scientific topic t, and that all the publications in cluster c address the topic t.

The publications clustering task consists of two independent subtasks: estimating publications similarities and clustering publications. The former consists in estimating the similarity between publications, while the latter consists in grouping publications according to their similarities. The work developed in this thesis focus on the task of estimating publications similarities which, by itself, is already a problem with some other potential applications (e.g., searching for similar publications in bibliographic databases). However, we present our strategy in the context of the publications clustering task since evaluating the clusters obtained using different similarities is a reliable strategy to compare publication similarity approaches [37].

In the research community, the publications clustering task is also referred to as community detection in publications networks. In general, using one term or the other often depends on the data used to estimate publications similarities, how the similarities are modelled and the type of algorithm utilised to cluster publications. Studies that use the term clustering often resort to textual evidence to estimate the publications similarities, use a similarity matrix to represent the similarities and use clustering algorithms (e.g., K-means). Conversely, studies using the term community detection typically use citation evidence to estimate the publications similarities, use a network to model the similarities and use community detection algorithms (e.g., Leiden and Louvain).

Regarding estimating publications similarity there are several algorithms that have been proposed. Depending on the data used to estimate the publications similarities, approaches are divided in three groups: textual-based, citation-based and hybrid. Next, we review each group in more detail.

2.3.1 Textual-based approaches

Textual-based approaches use the title and/or abstract of the publications to identify relevant terms for every publication. The similarity between publications is then estimated under the assumption that more similar publications have more relevant terms in common or their terms are more semantically related. A key step in text-based approaches is defining a term-weighting scheme which is the process of assigning a numeric value to each term in order to measure its contribution in distinguishing a particular publication from other publications. The term frequency-inverse document frequency (tf-idf) is one of the most popular term-weighting schemes in the literature [78]. The idea of tf-idf is to assign higher values to terms that occur frequently in certain publications but only occur in a relatively low number of publications. For

example, an high tf-idf value for a term t in publication p implies that t does not frequently occur in the publications but it is frequently used in p. The tf-idf algorithm does not measure publication similarity but it models publications as numerical vectors which then can be used to estimate the publications similarities by any methods that measures the distance between two vectors, as for example the cosine similarity [37].

The word2vec algorithm [79] is another alternative to represent publications using numerical vectors. Word2vec uses a neural network model to learn the word associations from a larger corpus of text. Once the model is trained, it can be used for word embedding. This means it can be used to convert words to numerical vectors. The main idea of word2vec is that the distance between the numerical vectors of words is proportional to their semantic distance. For example, it is expected that the distance between the vectors of "network" and "graph" is smaller than the distance between the vectors of "network" and "bibliometrics" since the words "network" and "graph" are used as synonyms in several publications. Text-based approaches that use the word2vec algorithm typically estimate the numerical vector of a publication as the average or the sum of the vectors of the words that occur in the publication. One example of this strategy is the study by [80] in which the word2vec model and the k-means algorithm were used to cluster publications. Building on the idea of the word2vec algorithm, the doc2vec algorithm [81] was proposed for document embedding. The doc2vec model adds a paragraph vector to the word2vec model which is unique for each publication. Consequently, during the training phase the model also learns the numerical representation of publications based on the semantic distance of the words. The distance between the resulting vectors can then be used to estimate the similarity between publications.

Another textual-based approach consists of modelling publications as probabilistic distributions of terms. The main idea is that a term belongs to a publication with a certain probability and that similar publications have similar term distributions. Examples of this approach are the Latent Semantic Analysis (LDA) [44], Kullback-Leibler Divergence (KLD) [82] and BM25 [83, 84]. According to experimental results obtained in [85], BM25 is one of the most accurate textual-based measures for identifying clusters of publications. The main difference between BM25, and its competitors is that it considers the length of the publications while computing the probabilistic distributions of terms.

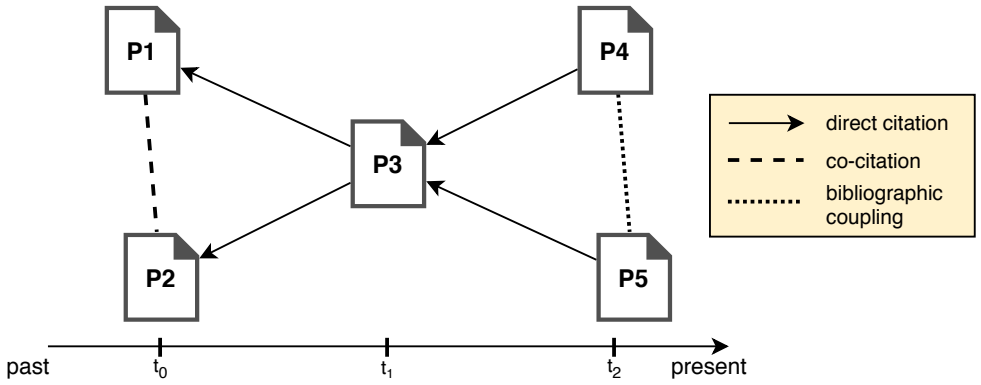


Figure 2.6: Citation patterns studied in the literature.

2.3.2 Citation-based approaches

Citation-based approach rely on the citation relations between publications to estimate publications similarities. There are three types of citations that are typically considered in these studies: direct citation, co-citation and bibliographic coupling relation. A direct citations represents a relation where paper i cites paper j, a co-citation represents a relation where paper i and j are cited by a paper k, and a bibliographic coupling relation represents a case where paper i and j cite paper k. Figure 2.6 illustrates the different types of citation relations.

The use of citation relations to measure publications similarities and to identify communities of publications have been studied in several works in the literature [37]. The main difference between these studies is the types of citations considered. More concretely, some studies consider only a single type of citation relation [86, 87, 88] while others combine different citation relations [37, 89, 90, 91]. Another aspect that distinguishes these approaches from each other is the weighting scheme used to differentiate the importance of citations. With respect to direct citations, Persson [89] uses a normalised weighting scheme where citations are more important if there are fewer of them. For example, the fewer publications a certain publication cites, the more important these citations are and the higher weights assigned to these citations. Chu et al. [88] discriminate the importance of citations depending on their location in a publication. For example, citations in the *"introduction"* section have a different weight than citations in the *"results"* section. Fujita et al. [91] study the effect of publication years, frequency of citations, reference similarity and keyword similarity on the weighting of the links in citations networks. In the case of co-citation relations, the used weighting scheme is often related with the distance in the text between the citations. Consider a publication k that cites publications i and j, thus creating a

co-citation relation between i and j. The weight of the co-citation is higher if the citations to i and j appear closer to each other in the text of k [90]. For example, citing i and j in the same sentence has a higher weight than citint them in different sections. The assumption is that the closer the two citations are located in the text, the stronger the two cited publications are related to each other [92].

2.3.3 Hybrid approaches

The idea of hybrid approaches is that textual and citation evidence of publications can be used together to improve the accuracy of similarity estimation for publications. The most straightforward strategy to combine both approaches is to sum the similarities produced with a textual-based approach with the ones obtained using a citation-based approach. For example, Ahlgren et al. [37] combines publication similarities obtained based on direct citation with both tf-idf and BM25 textual-based similarities in their study that compares over 10 similarity measures for clustering publications. Another hybrid approach is the SimCC algorithm [93] which utilises citation information to enhance traditional textual weighting schemes such as tf-idf. Zhang et al. [94] proposed the extended citation model which combines the similarities obtained using direct citation, co-citation and bibliographic coupling with the ones obtained using BM25.

2.3.4 Evaluation

There are several strategies to evaluate the quality of a clustering solution in general. For example, the aspects of cohesion (elements in one cluster should be as similar to each other as possible) and separation (elements in different clusters should be as dissimilar to each other as possible) are often considered to evaluate the quality of a clustering solution. In this thesis, we address the problem of publications clustering where each cluster represents a set of publications that are related (in terms of scientific area) to each other. Therefore, we disregard traditional measures to evaluate clustering solutions and we follow an evaluation framework that has been used in previous similar studies [37, 46].

Typically, the evaluation of clusters of scientific publications requires that a ground-truth similarity (which is created using human judgement) exists for the dataset. The principle of an evaluation strategy like this is to reward clustering solutions that place highly similar publications (according to the ground-truth) in the same cluster. For this purpose, a matrix S^{GT} where the value S^{GT}_{ij} represents the ground-truth similarity

between publications i and j is necessary. The higher the value of S_{ij}^{GT} is, the more similar or related two publications are. Conversely, a value of 0 indicates that two publications are not related.

Assuming that a ground-truth similarity matrix exists, one possible way to compare the quality of different clustering solutions is to sum the ground-truth similarities of the publications in the same clusters. In more detail, sum the ground-truth similarities for all the possible pairs of publications in a cluster and repeat the process for all the other clusters. This value is often referred as the accuracy of the clustering solution. In these type of evaluation, the best clustering solutions is the one with the highest accuracy (i.e., the one that placed the most pairs of publications with the highest similarities in the same clusters). One problem of this evaluation is that the best clustering solution is the one that only creates a single cluster. In this case, the accuracy would be the sum of all the values in the ground-truth similarity matrix (since all the publications are in the same cluster) which is the maximum value that is possible to obtain. However, having a single cluster is not a valid or useful solution for the publications clustering problem. To overcome this problem, the granularity of the clustering solutions should be considered along with their accuracy [37, 46]. In clustering solutions, the granularity measure provides information about the number of the clusters in the solution. For the same dataset (i.e., number of elements to cluster in the clustering problem), the lower the value of granularity for a clustering solution, the smaller is the number of clusters. Conversely, the higher the granularity of the clustering solution, the greater is the number of existing clusters. The granularity of a clustering solution X is given by:

$$\text{granularity} = \frac{1}{|\mathcal{P}|} \sum_{c \in X} |c|^2 \tag{2.11}$$

where c is a cluster in solution X and $|c|$ is the number of publications in cluster c. In the publications clustering problem, the desired granularity values for a solution are not known a priori and even for the same set of publications different values of granularity may be necessary depending on the applications. For this reason, a publications clustering approach, M, typically have user-defined parameters that help regulate their clustering solutions granularity which makes it possible to generate different clustering solutions with the same approach.

In order to compare two publications clustering approaches M_1 and M_2, n clustering solutions are generated for both approaches. Each clustering solution has different values of granularity and accuracy. These values can then be used to compare the two

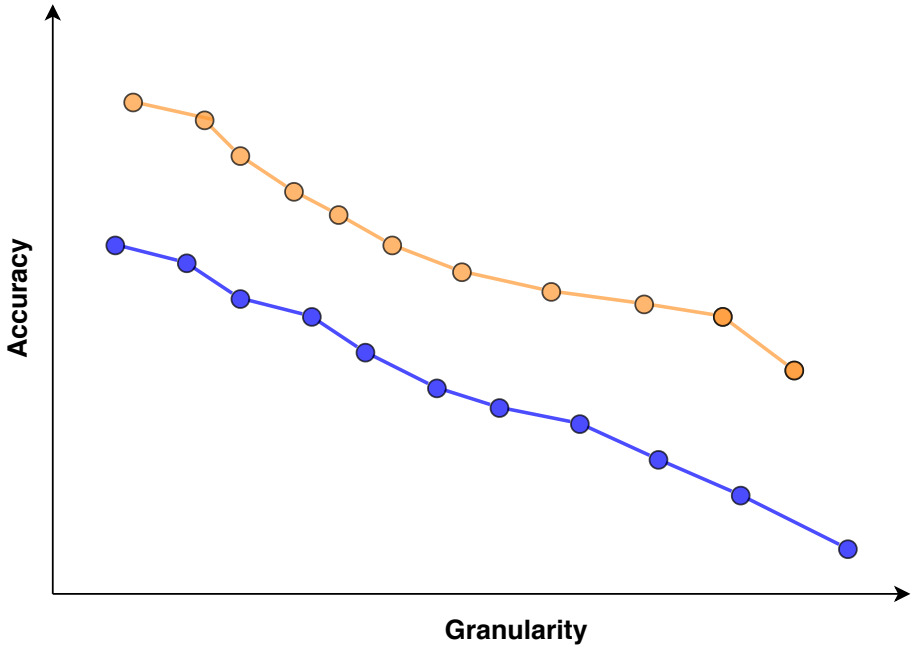


Figure 2.7: Example of a GA plot. Each dot represents the granularity and accuracy of a clustering solution. The yellow and blue lines represents these values for approaches M_1 and M_2, respectively. In this example, M_1 is a better approach since it presents higher values of accuracy for approximately similar values of granularity.

approaches using Granularity × Accuracy (GA) plots. In a GA plot the horizontal axis represents the granularity and the vertical axis the accuracy. The higher the granularity of a clustering solution, the greater is the number of clusters identified. In terms of research areas identification, lower values of granularity represent finding more broad areas (e.g. computer science) while higher values represent identifying more specific ones (e.g., classification problems - a specific problem in the machine learning area). With respect to accuracy, the higher the accuracy is, the more publications with a certain similarity (according to the ground-truth) are placed in the same clusters. Figure 2.7 illustrates an example of comparing two methods using a GA plot analysis. In this example, 11 clustering solutions are generated for each method. Furthermore, a line is drawn to connect all the different solutions of the same method. Lines that are consistently above the others represent better methods since the solutions provide higher accuracy for approximately similar values of granularity.

2.4 Expertise profiling

Input. A set of persons $\mathcal{R}$ and a set of publications $\mathcal{P}$ authored by persons in $\mathcal{R}$.

Problem description. Identify the set of topics $\mathcal{T}$ discussed in publications $\mathcal{P}$. Use the publications of each person $r \in \mathcal{R}$ to assess his knowledge about the topics $t \in \mathcal{T}$.

Output. A vector k_r for each person $r \in \mathcal{R}$ with length $|\mathcal{T}|$ where each position $k_r[t]$ reflects the knowledge of a person with respect to a certain topic.

Expertise profiling is the task of the knowledge or interests of a person. The goal of this task is to create an expertise profile that answers the following questions: *"which topics does a person has knowledge about?"* and *"which topics is a person interested on?"*. The expertise profiles have several applications such as categorising personal in institutions, searching for similar persons and studying the evolution of knowledge.

Despite being a widely studied problem, creating accurate expertise profiles still is a challenging task, mostly because even for humans it is not easy to quantify expertise [14]. The expertise profiling task is divided in two subtasks: topic modelling and topic to person association. Topic modelling is the task of identifying the topics discussed in a set of publications. The topic modelling task is necessary because the general assumption in the expertise profiling task is that the authored documents of a person represent their knowledge or interests [4, 16, 95]. More concretely, if a person is an author of a certain document, then the person has knowledge or interest on the topics discussed in the document. The topic to person association task consists in associating the topics identified in the publications to the persons in order to create their expertise profile. Solutions for this task often rely on the authorship links between the persons and publications to solve the problem.

In the literature there are several expertise profiling approaches which combine different topic modelling with different person to topic association strategies. The algorithms are mostly divided in two groups: author-topic and network models. Next, we review each group in more detail. Note that in this section, the terms person, author and expert are used interchangeably.

2.4.1 Author-Topic models

In text mining, language models are used to represent documents as mathematical models that are interpretable by machines. Their general idea consists in modelling topics as multinomial distributions over words, and documents as multinomial distributions over topics. The Latent Dirichlet Allocation (LDA) model [44] is one of the most widely used language models and became very popular due to its usefulness in

organising, classifying and searching for documents. The LDA model is described in three steps:

1. Randomly assign each word w to a topic t. This creates an initial distribution of documents over topics since documents already have a distribution over words.
2. For each document d estimate:
   2.1. $p(t|d)$ which is the probability of words in document d being assigned to topic t.
   2.2. $p(w|t)$ which is the probability of documents being assigned to topic t based on word w.
3. Reassign word w to a new topic t' according to the probability of the w being generated by topic t': $p(t'|d)p(w|t')$.

In 2004, Rosen-Zvi et al. [4] studied the use of the LDA algorithm in the context of the expertise profiling problem. The authors initial approach suggested that considering groups of authors that write a document is a straightforward modification of the LDA algorithm that allows it to model authors as a distribution over topics. Thus, obtaining expertise profiles. This initial approach provided information about the expertise of the authors but failed to identify the topics discussed in the documents. In their following effort, Rosen-Zvi et al. extended the LDA model to include the authorship information of the documents. This extension allowed the approach to simultaneously model documents and authors as a distribution over topics. Thus, the model can identify the expertise of the authors and the topics addressed by the documents. This model is known as the Author-Topic (AT) model. Figure 2.8 illustrates the differences between the original LDA algorithm, the first attempt of authors in [4] and their final proposed AT model.

The AT model gained a lot of attention due to its ability to associate persons to topics (i.e., create expertise profiles) and the flexibility of the algorithm (in the sense that other sources of information can be easily included in the original AT model). As a result, following the ideas of the AT model several new models were proposed in the literature. In 2007, Mimno and McCallum [96] proposed the Author-Persona-Topic (APT). The APT model considers that each author consists of multiple personas, each characterising different combinations of his expertise areas. The APT model divides the publications of an author into multiple clusters. Then, the algorithm processes each cluster individually which results in multiple topic distributions for an author, each one representing a different persona.

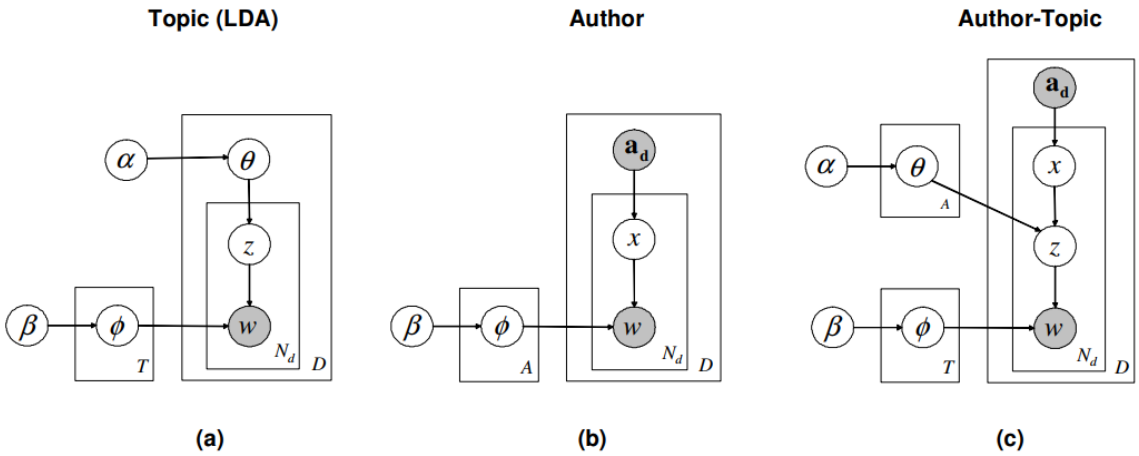


Figure 2.8: Evolution of the LDA algorithm to the Author-Topic model. Image taken from the original paper [4]. The model represented in (a) is informative about the content of document but provides no information about the expertise of the authors. The model represented in (b) is informative about the expertise (or interests) of the authors but fails to identify the topics discussed in the documents. Finally, the model represented in (c) simultaneously identifies the expertise of the authors and the topics discussed in the documents.

In 2008, Tang et. al [97] proposed the Author-Conference-Topic model where each topic is characterised by a distribution over words and also a multinomial distribution over the conferences. Following this work, Wang et. al [98] proposed the Author-Conference-Topic-Connection model which adds the subject of the conference and the latent mapping information between subjects and topics.

In 2012, Daud [40] proposed the Temporal-Author-Topic (TAT) which models the topic distribution of an author over time. A similar effect could be achieved using AT in chunks of publications divided by years. However, Daud argues that considering the entire interval of years improves the results since it can handle topics exchanges along the years. In this work, authors are still represented as a distribution over topics, but topics are modelled as a multinomial distribution over words and timestamps. Following the ideas of TAT, in 2016, Jeong et. al [41] proposed the Author-Topic-Flow (ATF). According to the authors, the advantages of ATF over TAT is that the former allows each author to directly have a temporal pattern of research interests, while each author in TAT model only has a topic proportion covering all the time spans.

2.4.2 Network-based models

The network-based models aim to improve the topic to person association step by using networks to obtain evidence besides the direct relation of authorship. The topic to person association in author-topic models is based on the one-step relation of authorship. More concretely, the knowledge of an expert is entirely built by his authored documents. Network-based models extend this concept by analysing multi-step relations to associate knowledge. The general assumption is that the knowledge of an expert can be estimated by considering his authored documents and the knowledge of the other experts *"around"* him[2]. Under these circumstances, the closer the authors are in the network the more similar their expertise profiles are [14].

Most of the network-based models proposed in the literature focus on the task of expert finding which is a related problem of expertise profiling (more details about this problem 2.4.6). Still, there are some network-based models developed with the goal of constructing expertise profiles. Sun et al. [99] proposed the iTopicModel which introduces a two-layer model that simultaneously integrates text evidence (i.e., the terms present in the documents) and citation relations. The top-layer in the model uses a citation network to relate documents to each other using random walks. Meanwhile, the bottom-layer represents each document as a multinomial distribution of topics in a similar strategy as used by the LDA algorithm. Deng et al. [100] proposed the Topic Model with Biased Propagation (TMBP) algorithm which uses an heterogeneous information network consisting of documents, authors and venues. The TMBP algorithm initially estimates the topics of the documents using the a language model. Then, the topics are propagated to the authors and venues through two different bias strategies.

Neshati et al. [101] proposed a network-based model which does not create expertise profiles but aims at being used as a complementary source of information for the expertise profiling problem. The authors considered the problem of crediting all the authors of a document with the same merit/knowledge gain. The authors proposed a data-driven model that estimates different contributions of authors in the same document. Some of the features used in their model are centrality and random-walk metrics obtained from the co-authorship network of the authors. The different contributions of authors to different documents can be used to further improve the process of associating the topics documents to the authors.

[2]Note that the term *"around"* depends on the context on the network. For example, in co-authorship networks the term *"around"* relates an author to his co-authors.

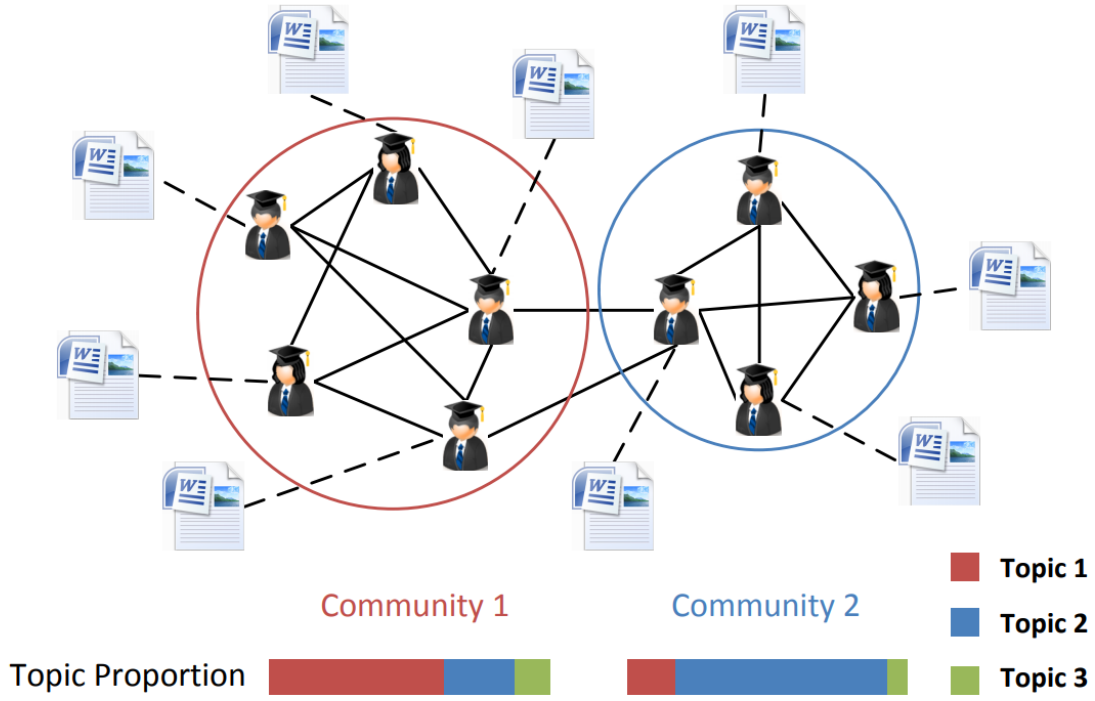


Figure 2.9: Image from [5]. Illustration of the group profiling problem. In this example, community detection in the co-authorship network is used to estimate the groups of experts with similar expertise profiles, then text evidence is used to estimate the knowledge topics of each group.

2.4.3 Group profiling

Network-based and author topic models are often used (or combined) to estimate the expertise profiles of a group of persons instead of an individual one. In the literature, this is often referred as the task of group profiling. With the increasing size of data, namely in the case of social and bibliographic networks, identifying communities and the topics discussed within these groups have become an important task to better understand these systems. Compared to the traditional expertise profiling strategies, group profiling adds an extra step which consists in identifying groups of persons with similar expertise profiles or interested in the same topics. Typically, these groups are identified using a community detection algorithm over a network (e.g., a co-authorship network). Figure 2.9 illustrates an example of the group profiling problem where the groups of experts are identified using community detection on the co-authorship network and then the topics are estimated through the documents of each expert.

The previous example shows a straightforward strategy to solve the group profiling

problem. However, in this case, communities and topics are independent from each other. More concretely, communities are formed considering the network structure, while topics are determined using textual evidence of the documents. Thus, this strategy does not take advantage of all the information available [5]. An ideal approach for the problem is one where the community detection enhances the results of topic modelling and vice-versa [5, 102, 103].

A pioneer effort to tackle this problem was proposed in 2009 by Liu et al. [104]. The authors general assumption was that links between documents in the network represent a combination of topic similarity and community closeness. Still, this study treated communities and topics as the same latent variable. More concretely, each community represents a single topic and this topic is exclusive to that community. In the real-world, communities are often represented by more than one topic which can be shared by other communities.

In 2012, some approaches that handled topic modelling and community discovery in the same framework, while allowing a community to consist of multiple topics and topics to be part of multiple communities were created. Duan et al. [5] proposed the Mutual Enhancement Infinite community-topic model and Yin et al. [102] the Latent Community Topic Analysis. Later, in 2015, Li [103] proposed the Author-Topic-Community (ATC) model which simultaneously models expertise profiles and experts' communities. ATC is an extension of the AT model where the link between two experts is created based on the community structure of the co-authorship network. A different strategy was introduced by Revelle et al. [105] who proposed the Seeded Estimation of Network Communities (SENC). The SENC algorithm initialises the communities as a seeded subgraph and then grows each one while updating its topics. The initial subgraph is a lower-bound to the communities. Furthermore, a limiting upper-bound is also estimated using the topic distribution (represented as nodes in the network) and network properties such as clustering coefficient and network diameter.

2.4.4 Hierarchical expertise profiles

The task of expertise profiling is most of the times intertwined with topic modelling. Most of the algorithms for expertise profiling either address this aspect by analysing textual evidence (author-topic models) or the community structure of the network (network-based approaches). Regarding the topics discovered, the outcome of these approaches is typically a set of independent topics. In more detail, experts may have knowledge about topics t_i and t_j, however there is no relation between both topics.

Studies about the expertise profiling problem have shown that profiles generated over topics that are organised in hierarchies, i.e., the topics discovered are related to each other with relations such as "t_i is a sub-topic of t_j", provide the best results [99, 16, 48]. Organising the topics discovered in an hierarchy allows the algorithm to map the knowledge of experts at different granularity levels. These profiles are referred as hierarchical expertise profiles and they provide a more detailed overview of the expert's knowledge.

In the literature there are only a few methods that are capable of creating hierarchical expertise profiles and they have some drawbacks which we briefly explain here and introduce in more detail in Chapter 5. In this section, we take a step back and present the general problem of creating hierarchies of topics and the reason why they are difficult to integrate with expertise profiling strategies.

In huge collections of information, creating an hierarchical organisation of data at different levels of granularity, provides an easier and faster way to analyse information. This methodology has been widely used for documents categorisation in the context of bibliographical databases. The general idea is that documents are separated according to their similarities and each group of documents represents a topic. Moreover, at each level of the hierarchy the strength of the similarity between documents in the same topic is different. In more detail, the topic specificity that determines if two documents discuss the same topic changes over different levels of the hierarchy. Typically, the top level represents more broad topics while the bottom levels represent more specific topics. As a result, documents in the same topics at bottom levels are more similar than documents in the same topic at the top level. For example, one document discussing parallel computing and another discussing machine learning can be in the same topic at the top level (since both documents are from the computer science area) but will be in different topics at the bottom levels. In the literature, a structure that organises topics like this is referred as topical hierarchy or ontology.

In the community there are some publicly available topical hierarchies. Two examples of popular topical hierarchies are the DBpedia [3] and the Association for Computing Machinery (ACM) classification system [4]. The DBpedia topical hierarchy describes the hierarchical organisation of the concepts defined in its pages. The ACM topical hierarchy describes the organisation of the topics discussed in the ACM journals which is used to organise documents. Figure 2.10 illustrates a sample of the ACM topical hierarchy. The problem with these topical hierarchies is that they were mostly created

[3] http://mappings.dbpedia.org/server/ontology/classes/
[4] https://www.acm.org/publications/class-2012

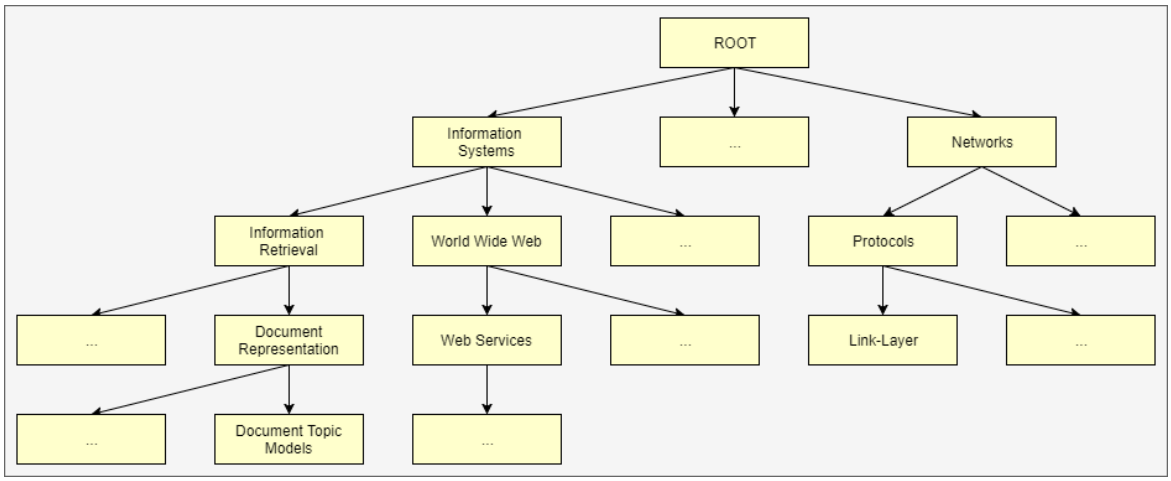


Figure 2.10: Sample of the topical hierarchy of the ACM computing classification system.

manually, thus requiring a lot of human effort. Moreover, topical hierarchies often change through time, as a result this is not a one time task and requires frequent updates.

In the literature, the problem of automatically or semi-automatically creating topical hierarchies has received a lot of attention. However, this still is a challenging task with several algorithms being proposed [106]. Most of the strategies focus on textual evidence present in documents. Typically the task consists of extracting terms from the text recurring to natural language processing techniques and then, using pattern-based or statistical-based techniques to inter the relations among terms. Pattern-based techniques search for patterns in the text such as *"x is a y"* to infer the relations. Meanwhile, statistical-based techniques resort to statistics such as the number of times terms co-occur to infer the relations.

The literature on this topic is vast and there is a considerable amount of algorithms that have been proposed to generate topical hierarchies. However, going into the details of these methods is behind the scope of this thesis. The reason for that is the fact that the topical hierarchies constructed using these methods typically consists of topics that are described by term(s) (similar to the topical hierarchy represented in Figure 2.10). Using topical hierarchies like these in the expertise profiling problem is impracticable because it is difficult to associate the experts to the topics represented in the hierarchy. Daud [40] used the ACM computation classification system to profile experts. The author mapped the knowledge of experts into the ACM topical hierarchy based on their documents that have been published on ACM journals. Every one of these documents contains manually assigned labels that place them in the ACM

topical hierarchy, thus providing a straightforward strategy to map the authors into the topical hierarchy and obtain their hierarchical expertise profile. However, in this study, we have to consider that the ACM topical hierarchy as well as the labels on the publications were manually created and only address part of the computer science topic. Furthermore, the method presented disregarded all the documents that were not published in ACM journals but still contain evidence of the expertise of the authors. As a result, the study presented is not practicable solution for all the expertise profiling situations.

Regarding methods to automatically construct a topical hierarchy, we would like to point out the study of Wang et al. [50] which motivated our work in hierarchical expertise profiling. The authors proposed the CATHYHIN algorithm that constructs topical hierarchies using a generative model in heterogeneous information networks. The main difference between CATHYHIN and other hierarchical topic modelling approaches is the fact that each topic is represented by a list of attributes instead of word(s) (referred as multi-typed topic). CATHYHIN is not an expertise profile strategy therefore is not a direct competitor for how work. Nevertheless, we extend their idea of having multi-typed topics to map knowledge in the expertise profiling task.

2.4.5 Evaluation

In general there are two aspects of an expertise profiling strategy that can be evaluated: the topic modelling and the profiles generated. The former focus on the evaluation of the topics discovered and on their overall ability to represent the topics discussed in the documents. Meanwhile, the latter focus on assessing the quality of the topics associated to the experts and how they reflect their knowledge.

For the topic modelling part there are several metrics that have been proposed in the literature. In this thesis, we use one of the most popular ones which is based on Pointwise Mutual Information (PMI) metric. PMI was originally developed to assess if the co-occurrence of words in text is meaningful or not. The idea of PMI is to quantify the likelihood of co-occurrence of two words, taking into account the fact that it might be caused by the frequency of the singles words. The PMI of words w_i and w_j is given by:

$$\text{PMI}(w_i, w_j) = log\left(\frac{p(w_i, w_j)}{p(w_i)p(w_j)}\right) \tag{2.12}$$

where $p(w_i, w_j)$ represents the joint probability of words w_i and w_j (i.e., their probability of co-occurring in text) and, $p(w_i)$ and $p(w_j)$ represent the probability of finding words w_i and w_j in the text, respectively. When the value of PMI between two words is 0, it means that the two words together are not likely to form a concept (i.e., a word consisting of two words). Conversely, the higher the value of the PMI of two words, the more likely they are to form a unique concept.

PMI can be adapted to the context of topic modelling in documents. Consider an author-topic model that generates topics based on a LDA approach. Each topic consists of a set of terms which were extracted from the documents. In this case, PMI can be used to evaluate the likelihood of two terms t_i and t_j in the same topic to be relevant to each other by defining $p(t_i, t_j)$ as the probability of finding both terms in the same document and, $p(t_i)$ and $p(t_j)$ as the probability of finding terms t_i and t_j in a document, respectively. In order to assess the quality of a topic discovered in the topic modelling approach the average value of PMI for the top-k terms of the topics can be considered. The higher the PMI value of the topic is, the more likely are that the top-k terms form a topic by themselves (i.e., the better the topic is).

Evaluating the profiles generated for the experts often requires human judgement. There are two approaches that are most frequently used in expertise profiling studies: self-assessment and expert comparison. Self-assessment relies on experts self-evaluating the profiles that were generated for them. This is typically achieved through a survey that presents the expert with several questions with the end goal of evaluating how well does the profile represent the knowledge of the expert [16]. The expert comparison evaluation is based on the assumption that experts with similar interests (or that work in similar areas) should have similar profiles. A commonly used strategy consists on using external data sources to group experts with similar interests and then compare the profiles of these experts [14]. An example of an external data source are the Google Scholar pages of researchers. Figure 2.11 presents an example of such profile where the author has manually added his interests to the profile. These interests can be used to find other researchers whose profile should be similar to this one.

Either strategy discussed, and the others discussed in the literature, are not capable of numerically quantify the quality of an expertise profiling strategy. As a result, comparing expertise profiling strategies is a difficult task which most relies on the discussion of advantages and disadvantages of methods, and a case-by-case evaluation of profiles generated from different strategies.

Jorge Silva

CRACS - INESC-TEC & Computer Science Department, FCUP, University of Porto, Portugal

Email confirmado em inesctec.pt

Data Mining Network Science Bibliometrics

Figure 2.11: Example of a Google Scholar profile. The interests of the researcher are highlighted in red.

2.4.6 Related problem: expertise finding

The expertise retrieval area addresses the problem of linking experts to knowledge areas and vice-versa [14]. This area is divided in two subtasks: expertise profiling and expertise finding. Expertise finding addresses the problem of identifying persons with knowledge about a topic (in contrast to the problem of identifying the knowledge of a person addressed by expertise profiling). Despite addressing different questions, expertise profiling and finding share the same core task: quantify the knowledge of a person with respect to a certain topic. Following the example provided in [14], consider a skill matrix that represents the outcome of an expertise retrieval solution (Table 2.2 presents an example of this matrix). A cell in such matrix presents the knowledge of an expert with respect to a topic. The task of expertise profiling consists in filling a row in the matrix, while the expertise finding task consists in filling a column. There are also other differences between both tasks. In the case of the expertise profiling task it is common that the set of topics (i.e., the number of columns in the skills matrix) is unknown a prior, thus the strategies require topic modelling. In the case of expertise finding, the task often involves identifying the expert with the most knowledge about a topic (i.e., the highest value of a column), thus the strategies requires some sort of knowledge ranking.

In this thesis, our focus is on the expertise profiling task and for this reason we do not provide a detailed overview of expertise finding methods. However, some of the

Table 2.2: Skills matrix

	topic 1	**topic 2**	**topic 3**	**...**	**topic n**
expert 1	x		x		x
expert 2		x	x		
expert 3		x			x
...					
expert m	x		x		x

methods developed in this thesis can be used to tackle the problem of expertise finding. This topic is discussed in Section 5.5.3 with an application case and in Chapter 6 with future research directions.

Chapter 3

Author ranking

Deciding where (or to whom) to allocate research funding is a problem that affects all scientists directly. This is typically done by estimating the scientific impact of a scientist. More concretely, determine *how much of his research work has contributed to the evolution of his scientific field*. Scientific impact is also commonly used to chose scientific committees, attribute research grants, or decide faculty promotions. Traditionally, more impactful scientists tend to have access to more funding which supports the creation of more quality work. Thus, estimating the scientific impact of scientists has a direct impact on the evolution of science. For more important decisions, the process is mostly done via peer review. However, more and more, bibliometrics (i.e., measures to determine scientific impact without human intervention) have been proposed to assist the peer review process.

In this chapter we propose new methods to measure the scientific impact of scientists through the analysis of their published work. We have two goals in this chapter. First, we present OTARIOS (OpTimizing Author Rankings using Insiders/Outsiders Subnetworks) which is an author ranking algorithm that combines different author features and handles incomplete networks. Second, we present FOCAS (Friendly Only Citations AnalySer) a method which that is applied to author-level author ranking algorithms to decrease the impact of citations received through the abuse of citation boosting patterns. Thus, providing a fairer ranking for the scientists.

We measure the scientific impact of authors in real-world author-level citation networks. To compare different approaches we create a ground-truth based on best paper

awards from multiple conferences. In our experiments, approaches that produce an author ranking more similar to the ground-truth ranking are better. The goal of our experiments is two-fold. First, we aim to show that OTARIOS is capable of producing better author rankings than other approaches. Second, we aim to show that FOCAS improves the produced rankings by author ranking algorithms.

3.1 OTARIOS

3.1.1 Terminology

Graph-based methods for author ranking analyse citations either by creating a paper-level or a author-level network. These networks model the relations between citing and cited publications or authors. Along with the relations modelled by the edges in the network, there are several features that can be used in the edges and nodes of the network. In this section, we discuss some of these features and the terminology that we use to refer to them.

A paper (or publication) $P_j \in \mathcal{P}$ is co-authored by authors $\mathcal{A}_{P_j} \subseteq \mathcal{A}$. Likewise, an author $A_i \in \mathcal{A}$ is (one of) the author(s) of papers $\mathcal{P}_{A_i} \subseteq \mathcal{P}$. In *paper-level networks*, graph $G = \{\mathcal{V}, \mathcal{E}\}$ comprises a set $\mathcal{V}$ of nodes that represent papers and a set $\mathcal{E}$ of edges that represent paper citations, written as $P_{j'} \rightarrow P_j$. In *author-level networks*, nodes represent authors and edges represent citations between authors, written as $A_{i'} \rightarrow A_i$.

With respect to some features that can be present in the nodes, papers have publication metadata which can be used as features, namely the year, venue prestige, and the number of references, represented by $y(P_j)$, $v(P_j)$ and $r_{out}(P_j)$, respectively. The recency of a paper, represented by $\delta(P_j)$, is the difference in years between the year of the paper and the most recent paper in the dataset (e.g., $\delta(P_j) = 2017 - 2015 = 2$ if P_j is a paper from 2015 in a dataset where the most recent paper is from 2017). $\delta(P_j)$ is estimated using the following equation:

$$\delta(P_j) = \Big(\max_{P_{j'} \in \mathcal{P}} y(P_{j'}) \Big) - y(P_j) \tag{3.1}$$

Similarly, the recency of an author, represented by $\delta(A_i)$, is simply the recency of his most recent paper (i.e. the number of years that have passed since his last publication).

$\delta(A_i)$ is estimated using the following equation:

$$\delta(A_i) = \min_{P_j \in \mathcal{P}_{A_i}} \delta(P_j) \tag{3.2}$$

The venue prestige of a paper P_j, represented by $v(P_j) = \lambda(V_k, y)$, depends on the venue $V_k = v(P_j)$ where it was published and the year $y = y(P_j)$ when it was published. In this thesis, we use the *CiteScore* metric to estimate venue prestige since this is a widely used metric [107]. $\lambda(V_k, y)$ is calculated using the following equation:

$$\lambda(V_k, y) = \frac{c(V_k, y)}{\sum_{x=1}^{3} p(V_k, y - x)} \tag{3.3}$$

where $p(V_k, y)$ is the number of papers published in V_k in year y and $c(V_k, y)$ is the number of citations that all papers published in V_k in year y received. The underlining assumption in the *CiteScore* metric is that venues with more citations per paper have higher prestige.

Regarding the features that can be used in the edges of the network, in paper-level networks edges are traditionally unweighted and simple. More concretely, two papers are connected by a single edge with weight equal to 1 [19, 29]. In author-level networks, edges are often weighted and multiple. For example, two authors can be connected by multiple edges with different weights. These multiple edges concern different edge features that depend on the publication P_j where author $A_{i'}$ cites author A_i. The recency of edge (also referred as citation recency), represented by $a(A_{i'} \rightarrow A_i, P_j)$, gives more importance to recent citations. $a(A_{i'} \rightarrow A_i, P_j)$ is calculated using the following equation:

$$a(A_{i'} \rightarrow A_i, P_j) = e^{\frac{-\delta(P_j)}{\tau}}, A_{i'} \in \mathcal{A}_{P_j} \tag{3.4}$$

where τ is a decay factor which defines the rate at which the value of a citation (i.e., edge) decreases as the time passes. We set $\tau = 4$ since this is the recommended value used by the authors that proposed the recency concept [19]. This value highly favours citations received in the time window of the last four years. According to Equation 3.4, a citation with recency 0 has a maximum edge weight of 1.0, while citations with recency 4 and 8 have edges with weights 0.37 and 0.13, respectively.

The venue prestige of an edge (also referred as citation prestige), represented by $v(A_{i'} \rightarrow A_i, P_j)$, gives more importance to citations in more important venues. $v(A_{i'} \rightarrow A_i, P_j)$ is estimated using the following equation:

$$v(A_{i'} \rightarrow A_i, P_j) = v(P_j), A_{i'} \in \mathcal{A}_{P_j} \tag{3.5}$$

According to this equation, citations coming from the most prestigious venues in the dataset have a maximum edge weight of 1.0, while citations coming from the least prestigious venues have a value close to 0.0. Finally, the individuality of an edge (also referred as cited individuality), represented by $w(A_{i'} \rightarrow A_i, P_j)$, gives more importance to citations received in papers where author A_i has few (or no) co-authors. $w(A_{i'} \rightarrow A_i, P_j)$ is calculated using the following equation:

$$w(A_{i'} \rightarrow A_i, P_j) = \frac{1}{|\mathcal{A}_{P_j}|}, A_i \in \mathcal{A}_{P_j} \tag{3.6}$$

According to this equation, if an author has a publication P_1 with 2 authors and a publication P_2 with 4 authors, the importance of citations (i.e., the edges weight) coming to P_1 is double the ones coming to P_2 for that author. Thus, $w(A_{i'} \rightarrow A_i, P_j)$, unlike $a(A_{i'} \rightarrow A_i, P_j)$ and $v(A_{i'} \rightarrow A_i, P_j)$, depends on the cited author A_i and not on the citing author $A_{i'}$.

The author's feature total out-edge weight is obtained by summing all of its out-edges. For example, considering citation recency, $a_{out}(A_i) = \sum_{(A_i \rightarrow A_{i'}, P_j)} a(A_i \rightarrow A_{i'}, P_j)$. The out-edge weights for cited individuality (w_{out}) and citation prestige (v_{out}) are obtained in the same way.

3.1.2 Related work

In this section, we use the introduced terminology to discuss in more detail the PageRank-based algorithms for author ranking that have been proposed in the literature. PageRank consists of two main steps: score initialisation and score diffusion. The score initialisation step creates a vector R that defines an initial score for every node using *a priori* information. In the simplest case, every node (i.e., paper or author) is considered equally important, thus an uniform distribution is used (i.e., $R[A_i] = \frac{1}{|\mathcal{A}|}$) [28, 74, 75]. Paper-level approaches typically assign higher initial scores to more recent papers [19] or favour a combination of recent papers and papers published

in venues with high impact factor (or prestige) [29]. Author-level approaches typically assign higher initial scores to authors that publish many papers [26] or favour authors that publish many papers but with few co-authors [20].

The score diffusion step updates the nodes scores by taking into consideration the network structure. Score diffusion is an iterative process which computes three addends: random restart, dangling nodes, and score term. Random restart (RR) estimates the likelihood of reaching a certain node by moving randomly in the network. PageRank defines a value q as the random restart probability, and q is multiplied by the node's initial score R (thus, nodes with higher initialisation score receive higher random restart score). Dangling nodes (DN) is a process where the score of nodes that do not have any out-links is split by all the other nodes. This is a necessary step to prevent the existence of nodes that do not disseminate their credit. In the same manner as the random restart addend, this division takes into consideration the initialisation vector R (thus, nodes initialised with higher values receive higher dangling nodes score). Score term (ST) updates the score of a node A_i, according to the score of his in-links (i.e., nodes citing A_i). In the simplest case, scores are evenly split by co-authors of the cited publication. For example, if the paper has two authors, the score is divided by the two authors, if it has three authors, the score is divided by the three authors. An

Table 3.1: Comparison of graph-based methods for author ranking. N_i represents a node in the network, i.e., $N_i = A_i$ in author-level networks, and $N_i = P_i$ in paper-level networks. Score diffusion $S(N_i)$ is equal to $ST(N_i) + RR(N_i) + DN(N_i)$. For all methods, $RR(N_i) = q \times R(N_i)$ and $DN(N_i) = (1-q) \times R(N_i)$, thus we omit them from the table.

	Method	**Initialisation: $R(N_i)$**	**Score term: $ST(N_i)$**
Author-level	RLPR [74]	$\frac{1}{\lvert\mathcal{A}\rvert}$	$(1-q) \sum_{(A_{i'} \to A_i, P_j)} \frac{S(A_{i'}) \times w(A_{i'} \to A_i, P_j)}{w_{out}(A_{i'})}$
	SARA [20]	$\frac{\sum_{(P_j \in \mathcal{P}_{A_i})} \frac{1}{\lvert\mathcal{A}_{P_j}\rvert}}{\sum_{(A_{i'} \in \mathcal{A})} \sum_{(P_j \in \mathcal{P}_{A_{i'}})} \frac{1}{\lvert\mathcal{A}_{P_j}\rvert}}$	
	ALEF [26]	$\frac{\lvert\mathcal{P}_{A_i}\rvert}{\lvert\mathcal{P}\rvert}$	
	SCEAS [28]	$\frac{1}{\lvert\mathcal{A}\rvert}$	$\frac{(1-q)}{a} \sum_{(A_{i'} \to A_i, P_j)} \frac{(S(A_{i'})+b) \times w(A_{i'} \to A_i, P_j)}{w_{out}(A_{i'})}$
Paper-level	YetRank [29]	$v(P_i) \times \frac{e^{\frac{-\delta(P_i)}{\tau}}}{\tau}$	$(1-q) \sum_{(P_{i'} \to P_i)} \frac{S(P_{i'}) \times R(P_i)}{r_{out}(P_{i'})}$
	NewRank [19]	$e^{\frac{-\delta(P_i)}{\tau}}$	

important fact to note is that in the latter case (the paper with three authors) each author receives less credit than each author of the paper with two authors [20, 26, 74]. SCEAS [28] adds a constant value b to every score received by nodes and divides the total score received by another constant a in order to make the algorithm converge faster. YetRank [29] and NewRank [19] take into consideration the vector R in the score distribution (i.e., if a paper cites a paper P_j from 2015 and another paper P'_j from 2010, P_j receives a bigger chunk of the score since it is a more recent paper). In case of the YetRank, the distribution of score also takes into consideration the impact factor of the venues where P_j and P'_j where published, favouring papers published in venues with higher prestige. Table 3.1 summarises the methods discussed in this section and highlights their differences.

3.1.3 Motivation

A simple way to measure the scientific impact of an author is to use a centrality measure in a author-level citation network. In this context, the more important the author is in the network the higher his scientific impact is. For example, a highly cited author (which is often associated with someone with high scientific impact) has several edges in the network, thus he is more likely to be an important node. In citation networks, edge direction is important since the actions of being cited (having an in-coming edge) and cite (having an out-going edge) are different, and only the first one is asserted as a proof of scientific merit.

The PageRank algorithm is one of the most popular centrality measures used in networks. PageRank takes into account edge direction while measuring node importance and its definition of power naturally fits the problem of author ranking. More concretely, being cited by authors with high scientific impact is more important than being cited by author with less scientific impact. As a result, PageRank is a prime candidate for author ranking and several PageRank-based approaches have been proposed in the literature [19, 20, 26, 28, 74]. The state-of-the-art author ranking algorithms adapt the PageRank and introduce modifications to favour different types of authors. For example, some methods assign higher scientific impact to authors that are cited in important venues by increasing the weight of citations that come from more prestigious venues. Another example are the methods that favour authors that are being cited more recently by decreasing the weight of citations based on their age (the weight of older citations is decreased). These methods introduce features outside of the topology of the network and for that reason we call them *feature enriched methods*,

in contrast with the traditional PageRank algorithm which is a *topology-based method*.

We found that state-of-the-art approaches were lacking in two aspects. First, they do not adequately combine publications features (e.g., author's productivity, the venues prestige of where an author usually publishes, and how recent the papers of an author are) with citation features (e.g., the prestige of the venue where the citation is coming from and how recent the citations are). Second, these methods assume that the full network is known. In real-world cases, it is not possible to obtain the complete citation information. Let us assume that we want to rank a set of authors $\mathcal{A}$. First, we need to expand the network by obtaining all authors $B_i \in \mathcal{B}$ that cite any $A_i \in \mathcal{A}$ such that $B_i \notin \mathcal{A}$. Then, we need to also extract all authors $C_i \in \mathcal{C}$ that cite any $B_i \in \mathcal{B}$ such that $C_i \notin \{A \cup B\}$, to correctly determine the scores of all $A_i \in \mathcal{A}$, i.e., C_i does not cite A_i directly but C_i cites some B_i which cites A_i, thus C_i cites A_i indirectly. Ideally, this should be performed recursively until the complete set of authors (and their citations) with seed $\mathcal{A}$ is obtained. Due to memory and time constraints, only a sample of the citation network can be obtained. As a result, current state-of-the-art author ranking algorithms estimate scientific rankings based on incorrect information. More concretely, the authors in the periphery are not being adequately taken into account since their citations are not in the network. Although there is no ideal solution for this problem, one can be more careful in estimating the rank of nodes in the periphery.

3.1.4 Overview of our contribution

Here we present OTARIOS (OpTimizing Author Ranking with Insiders/Outsiders Subnetworks) which is a new feature enriched author ranking algorithm for incomplete networks. OTARIOS efficiently combines different publication/citation features in a multi-edge weighted network, conversely to the traditional simple unweighted network used by other approaches. OTARIOS is also a flexible algorithm in the sense that publication/citation features can be personalised to fit the ranking criteria decided by the user. For example, users may select only the venue prestige feature if they aim to rank authors according to the ones that publish on the most prestigious venues. OTARIOS also handles incomplete networks by dividing the citation network in two subnetworks, *insiders* and *outsiders*. Then, only the insiders are considered for ranking (since we have their full citation information) while outsiders contribute to the ranks of the insiders, not being themselves ranked. Table 3.2 summarises the features utilised by the state-of-the-art methods and the ones used by OTARIOS. OTARIOS is the only method that efficiently combines multiple features and deals with incomplete networks

Table 3.2: Comparison of state-of-the-art methods with OTARIOS. OTARIOS tries to combine all features efficiently and is also the only method that adequately deals with incomplete networks by using insiders/outsiders subnetworks.
*ALEF gives higher score to authors with many publications but ignores the number of authors in the publications.

	Publications			**C**itations			**I**ncomplete
Method	**V**olume	**R**ecency	**V**enues	**I**ndividuality	**R**ecency	**V**enues	**N**etworks
RLPR				✓			
SARA	✓			✓			
ALEF	✓*			✓			
SCEAS				✓			
YetRank		✓	✓	✓			
NewRank		✓		✓			
OTARIOS	✓	✓	✓	✓	✓	✓	✓

which makes it a valuable contribution to the author ranking problem.

We find that on five networks belonging to different areas of Computer Science, OTARIOS is $> 20\%$ more accurate than other state-of-the-art methods in creating a predicted rank more similar to one created using human judgement. OTARIOS obtained the best results when considering (i) the author's publication volume and recency, (ii) how recently the author work is being cited by outsiders, and (iii) how recently the author work is being cited by insiders and how individual his work is (i.e., publishing papers with few co-authors).

3.1.5 Problem formalisation

We formalise the problem of author ranking as the task of receiving a set of authors $\mathcal{I}$ and ranking them according to their scientific impact based on a set of user-defined criteria. First, we obtain all citations between all authors $I_i, I_{i'} \in \mathcal{I}$. More concretely, we obtain the complete citation network for $\mathcal{I}$. Second, for each author I_i, we obtain all of his received citations coming from authors $O_i \notin \mathcal{I}$. The process stops here. In more detail, we do not obtain all received citations for authors $O_i \in \mathcal{O}$. Doing so iteratively is unfeasible in practice because the number of authors added at each step grows very rapidly. Thus, we divide the citation network into two groups of nodes: *insiders* ($\mathcal{I}$) and *outsiders* ($\mathcal{O}$), i.e., $\mathcal{A} = \{\mathcal{I}, \mathcal{O}\}$. An important point to highlight is that no outsider can also be an insider, and vice-versa. Edges connect insiders ($\mathcal{E}_\mathcal{I}$) or outsiders to insiders ($\mathcal{E}_\mathcal{O}$), but no edges exist from insiders to outsiders nor between outsiders, i.e., $\mathcal{E} = \{\mathcal{E}_\mathcal{I}, \mathcal{E}_\mathcal{O}\}$. Figure 3.1 illustrates an example of dividing the network into these two subnetworks.

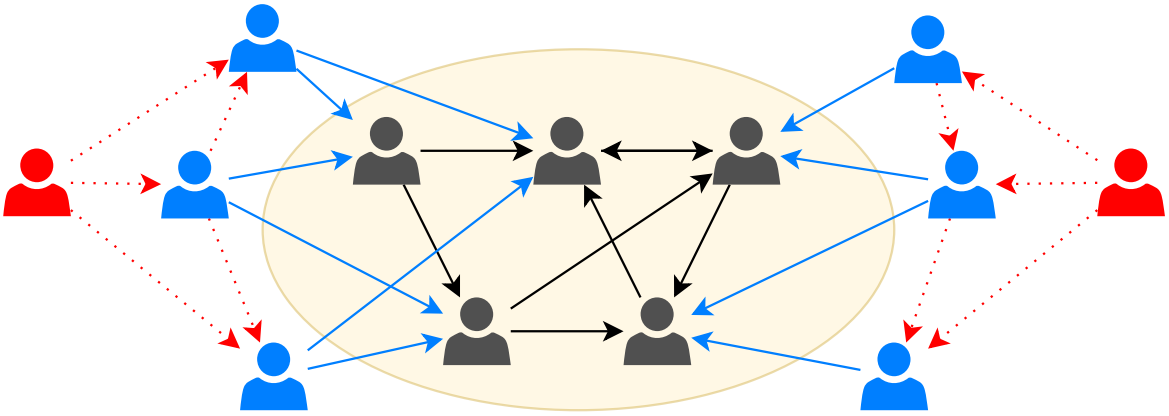


Figure 3.1: Example of insiders and outsiders subnetworks. Insiders are nodes/authors coloured in black and outsiders are coloured in blue. Note that no links between outsiders exists (dashed red lines). Furthermore, no information exists of outsiders that do not cite any insiders (coloured in red).

The outsiders are authors that were not in the initial set of authors $\mathcal{I}$, thus they are not ranked. Instead they are used to mitigate the problem of incomplete networks and improve the insiders' ranks. Before calculating the ranks of the insiders we estimate outsiders' prestige (λ). We use the outsider's history of publications and give higher prestige to authors with more citations ($c(A_i)$) in fewer publications ($p(A_i)$) (Equation 3.7). The outsiders' prestige is then used along the links between outsiders and insiders to improve the initial rankings of the insiders.

$$\textbf{Outsider prestige} \qquad \lambda(A_i) = \frac{c(A_i)}{p(A_i)} \qquad (3.7)$$

3.1.6 Methodology

OTARIOS is a graph-based algorithm for author-level citation networks. Its aim is to rank authors based on their publication and citation history. OTARIOS uses the notion of insider/outsider subnetworks to adequately estimate authors scores in a network with limited information. Furthermore, OTARIOS is a flexible algorithm that allows the user to decide the publication/citation features (i.e., the criteria) used to rank the authors.

In the first step, OTARIOS computes an initial score for each author, represented by $R(A_i)$. OTARIOS calculates $R(A_i)$ by taking into account multiple features that favour different author characteristics (Table 3.3). We divide the features into two

Table 3.3: List of features used for OTARIOS' author rank initialisation: $R(A_i)$. OTARIOS considers both the authors' productivity and the direct influence of outsiders on the authors. We create different variants of these criteria, e.g., $PV + V$ uses volume (P) and venue prestige (V) to measure author productivity, and uses venue prestige (V) to measure the direct influence of outsiders. Indivi. stands for Individuality.

	Feature	**I**nitialisation: $\boldsymbol{R(A_i)}$	**D**escription
Productivity	Volume (P)	$\frac{\sum_{(P_j \in \mathcal{P}_{A_i})} \frac{1}{\lvert\mathcal{A}_{P_j}\rvert}}{\sum_{(A_{i'} \in \mathcal{A})} \sum_{(P_j \in \mathcal{P}_{A_{i'}})} \frac{1}{\lvert\mathcal{A}_{P_j}\rvert}}$	Favours publishing many papers with few co-authors.
	Recency (A)	$e^{\frac{-\delta(A_i)}{\tau}}$	Favours publishing recently.
	Venues (V)	$\Big(\sum_{(P_j \in \mathcal{P}_{A_i})} v(P_j)\Big) \times \lvert\mathcal{P}_{A_i}\rvert^{-1}$	Favours publishing in prestigious venues.
Outsiders Influence	Indivi. (W)	$\sum_{(A_{i'} \to A_i, P_j)} \frac{\lambda(A_{i'}) \times w(A_{i'} \to A_i, P_j)}{w_{out}(A_{i'})}, A_{i'} \in \mathcal{O}$	Favours being cited by outsiders that cite few authors.
	Recency (A)	$\sum_{(A_{i'} \to A_i, P_j)} \frac{\lambda(A_{i'}) \times a(A_{i'} \to A_i, P_j)}{a_{out}(A_{i'})}, A_{i'} \in \mathcal{O}$	Favours being cited by outsiders more recently.
	Venues (V)	$\sum_{(A_{i'} \to A_i, P_j)} \frac{\lambda(A_{i'}) \times v(A_{i'} \to A_i, P_j)}{v_{out}(A_{i'})}, A_{i'} \in \mathcal{O}$	Favours being cited by outsiders in prestigious venues.

categories: productivity and outsiders influence. Productivity measures the value of the author's publications, while outsider influence measures the value of the author's citations coming from outsiders. Regarding productivity, OTARIOS takes three factors into account: volume, recency and venues. Regarding outsiders influence, OTARIOS takes another three factors into account: individuality, recency and venues. We compute the author's initial score $R(A_i)$ as the sum of the two products of the factors in each group. In more detail, productivity ($volume \times recency \times venues$) + outsiders influence ($individuality \times recency \times venues$).

Then, on the second step, OTARIOS improves author scores in an iterative process. Outsiders are removed from the network since their presence degrades the score diffusion step. In each iteration, OTARIOS updates an author's score $S(A_i)$ as $ST(A_i) + RR(A_i) + DN(A_i)$. We compute $RR(A_i)$ and $DN(A_i)$ as a function of the initial rank of each author (discussed in Table 3.3), and compute $ST(A_i)$ as a function of the author's citations coming from other insiders. OTARIOS considers three different features to assess score term $ST(A_i)$: individuality, recency and venues (Table 3.4). The $ST(A_i)$ at each iteration is the product of every feature. In more

Table 3.4: List of features used for OTARIOS' author score term calculation: $ST(A_i)$. Combined with author initialisation (Table 3.3), we create different variants, e.g., PV+V+A combines initialisation PV+V with score term A, i.e., using citation recency. All variants use $RR(N_i) = q \times R(N_i)$ and $DN(N_i) = (1-q) \times R(N_i)$, thus we omit them from the table.

Feature	Score term: $\boldsymbol{ST(A_i)}$	Description
Individuality (W)	$\sum_{(A_{i'} \to A_i, P_j)} \frac{S(A_{i'}) \times w(A_{i'} \to A_i, P_j)}{w_{out}(A_{i'})}, A_{i'} \in \mathcal{I}$	Favours being cited by insiders that cite few authors.
Recency (A)	$\sum_{(A_{i'} \to A_i, P_j)} \frac{S(A_{i'}) \times a(A_{i'} \to A_i, P_j)}{a_{out}(A_{i'})}, A_{i'} \in \mathcal{I}$	Favours being cited by insiders more recently.
Venues (V)	$\sum_{(A_{i'} \to A_i, P_j)} \frac{S(A_{i'}) \times v(A_{i'} \to A_i, P_j)}{v_{out}(A_{i'})}, A_{i'} \in \mathcal{I}$	Favours being cited by insiders in prestigious venues.

detail, score term (*individuality* × *recency* × *venues*). Like PageRank, OTARIOS stops when it reaches low variation in the node scores. Figure 3.2 illustrates the three feature categories used in the OTARIOS algorithm.

Here we do not assume that every feature should be used for author ranking. The features' importance depends greatly on the dataset. For instance, venue prestige might be very important to rank some communities (i.e., top authors publish in top conferences of that scientific area, e.g., machine learning) but irrelevant in some other community because we are studying a specific conference (i.e., all authors publish in the same venue, e.g., KDD). OTARIOS is parameterisable, i.e., users can define by which features authors are ranked. For example, for a certain application, we may

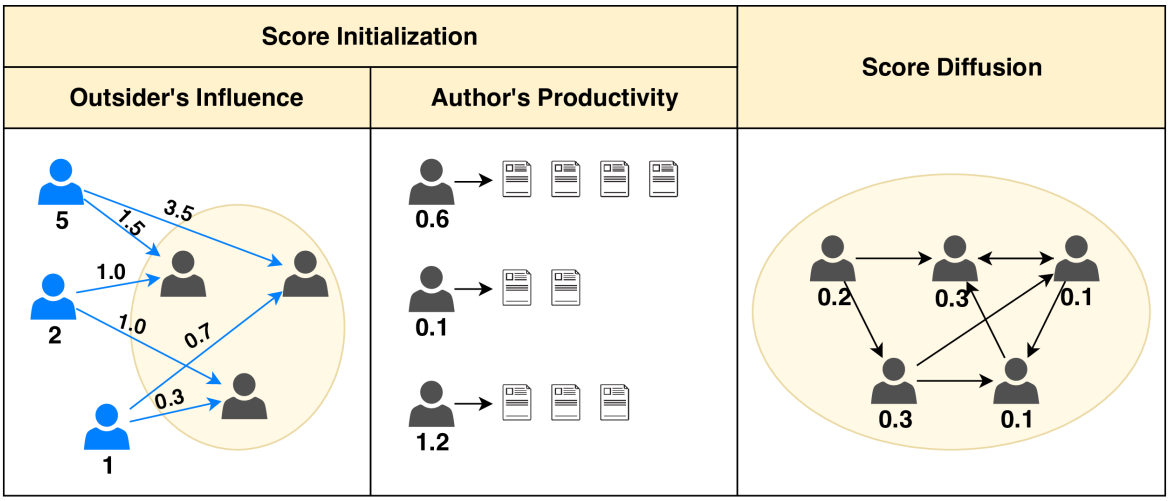


Figure 3.2: Illustration of the three different feature categories used in OTARIOS to rank authors.

want to rank authors taking into account recent publications and the venue prestige of citations coming from both insiders and outsiders. We define the OTARIOS variants using notation APV+AVW+AVW, where the addends define the features used at each group. the first productivity, the second for outsiders influence and the last for score term. In the previous example, the OTARIOS variant nomenclature is A+V+V.

3.1.7 Experimental evaluation

In this section we compare OTARIOS against state-of-the-art methods. We create a test scenario using a snapshot from December of 2017 of the DBLP dataset which is a bibliographic database for computer science. This dataset contains over 3 million publications and for each one we have: title, authors, abstract, venue, year, number of citations and references. Using the publications' references we obtain the author-citation network of 26 top-tier computer science venues. In order to prevent the impact of citation manipulation in the rankings we do not consider self-citations in the networks [108]. Furthermore, for each conference we create a ground-truth ranking using the best paper award information[1]. We counted each paper award as a unit of prestige which is equally divided by its authors. Thus, we are assuming that authors that have won more awards with fewer co-authors should be ranked higher. We use the ACM taxonomy[2] in order to group the conferences into five networks, each representing a different computer science area. Table 3.5 shows the conferences considered for each network, the number of awarded authors, and the number of nodes and edges of the insiders and outsiders subnetworks.

In our experiments, we compare a predicted ranking (e.g., one produced by OTARIOS or any other state-of-the-art approach) against the ground-truth ranking defined based on the best paper awards given by the conferences. Methods that produce rankings more similar to the ground-truth one are considered better. We use NDCG and MRR metrics (explained in more detail in Section 2.2.2) to compare predicted rankings against the ground-truth ranking. Furthermore, for a detailed analysis, we calculate NDCG and MRR for the top 5, 10, 20, 50 and 100 authors.

[1]Awards information obtained from: `https://jeffhuang.com/best_paper_awards.html`
[2]`https://www.acm.org/about-acm/class`

Table 3.5: Set of networks used for experimental evaluation. Data was taken from [1, 2]. The full DBLP dataset contains over 3M publications from 1936 to 2018. Each network contains publications from only a set of conferences, e.g., networks TC contains publications from FOCS, SODA and STOC. For each network we show the number of: awarded authors (AA), insider and outsider nodes ($|\mathcal{I}|$ and $|\mathcal{O}|$ respectively), and insider and outsider edges ($|\mathcal{E}_\mathcal{I}|$ and $|\mathcal{E}_\mathcal{O}|$ respectively).

Network	**C**onferences	# AA	**N**odes		**E**dges	
			$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{E}_\mathcal{I}\rvert$	$\lvert\mathcal{E}_\mathcal{O}\rvert$
CM	AAAI, IJCAI, ICML, ACL, ICCV, CVPR	380	35.6k	224.9k	4.6M	4.9M
TC	FOCS, SODA, STOC	440	5.0k	82.4k	0.5M	0.8M
NET	INFOCOM, NSDI, SIGCOMM, MOBICOM, SIGMETRICS	95	15.2k	138.8k	2.1M	3.7M
IS	KDD, CIKM, PODS, SIGMOD, VLDB, WWW, SIGIR	752	28.3k	190.9k	4.0M	5.1M
SE	PLDI, FSE, ICSE, OSDI, SOSP	349	10.8k	99.9k	1.0M	2.1M

3.1.7.1 Finding the best OTARIOS variants

OTARIOS does not define a strict set of rules to rank authors, since the criteria (i.e., features) used depends on many factors (e.g., scientific area, preferences of the entity ranking authors). Instead, OTARIOS gives the freedom to personalise the features used to rank authors. In the particular case of our test scenario, we did not know a priori which features would be the most important, so we did an exploratory analysis to find the best OTARIOS variants. However, there are more than 500 variants that we can create by combining different features. In order to estimate the best variant without testing a large number of variants, we performed a greedy search for each network.

We start with simple variants (i.e., one with a single feature) and progressively add more features to the more promising variants. We illustrate this process for the network NET on Table 3.6. We begin by comparing OTARIOS variants that only contain outsiders influence (e.g., $\varnothing$ + A + $\varnothing$). For the best ones, we added the productivity (e.g., AP + A + $\varnothing$). In general, we see that results improve when merging outsiders influence with productivity. Finally, we add the score term calculation to the best variants (e.g., AP + A + A). Again, we see that overall the results improve when we add this feature to the score term. For the NET network, we see that AP + A + AW is the best variant with a mean NDCG of 0.330 and a mean MRR of 606. This variant uses recency and volume to measure author productivity, uses recency to calculate outsiders influence, and uses recency and individuality on the score diffusion step.

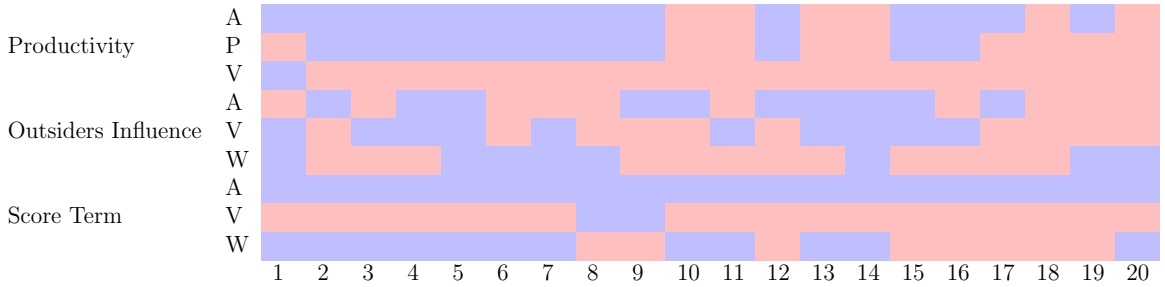
Table 3.7 presents the features used by the 20 best OTARIOS variants according to the average NDCG across the five networks created. We observe that the top-9 variants always use a mix of productivity, outsiders influence and score term, thus revealing the importance of considering multiple aspects of publications and citations

Table 3.6: Comparison of OTARIOS variants on network NET (from Table 3.5). For each OTARIOS variant, we measure its ranking's NDCG and MRR for the top-5, top-10, top-20, top-50 and top-100 authors, as well as the metric mean value. In bold we highlight the highest score for each metric. The best OTARIOS variant is coloured in blue.

OTARIOS	**NDCG**						**MRR**					
variant	5	10	20	50	100	Mean	5	10	20	50	100	Mean
∅ + A + ∅	0.269	0.233	0.207	0.186	0.174	0.214	443	1125	903	1526	2066	1213
∅ + V + ∅	0.269	0.233	0.207	0.186	0.185	0.216	412	1108	916	1522	2096	1211
∅ + AV + ∅	0.269	0.233	0.207	0.186	0.177	0.215	419	1109	902	1511	2074	1203
AP + A + ∅	0.288	0.246	0.259	0.218	0.241	0.250	350	500	440	1121	1502	783
AP + V + ∅	0.288	0.246	0.258	0.218	0.239	0.250	344	489	439	1134	1527	787
AP + AV + ∅	0.288	0.246	0.259	0.218	0.240	0.250	345	494	439	1143	1523	789
AP + A + A	0.380	0.297	0.283	0.282	0.280	0.304	385	647	472	1111	1416	806
AP + A + V	0.350	0.261	0.217	0.203	0.203	0.247	255	726	617	1251	1615	893
AP + A + AV	**0.407**	0.345	0.291	0.291	0.274	0.322	242	614	473	1116	1455	780
AP + A + AW	0.381	**0.369**	**0.313**	**0.302**	**0.288**	**0.330**	**219**	**386**	**328**	**879**	**1219**	**606**

information. Of the top 20 variants, only 6 do not use productivity features and only 1 does not use the outsiders influence; score term features are present in all top-20 variants. Regarding specific features, we observe that recency (A) seems to be the most important feature for all three categories: productivity, outsider influence and score term. In fact, recency is used in the score term of all top-20 variants. This indicates that most of awarded authors are still actively publishing and/or being cited. Individuality (W) and volume (P) seem to be more important to measure productivity and score term than to measure outsiders influence. This indicates that awarded authors publish more papers and also that publish with fewer co-authors. Venue prestige (V) seems to be more relevant when measuring outsiders influence than productivity and insiders score term. This is expected because, due to the nature of the two subnetworks, insiders tend to publish in the same venues, while outsiders cite

Table 3.7: Features considered on the top 20 OTARIOS variants on the NDCG metric. The rows represent different features and the columns the variants that ranked at position n. The blue colour in a column indicates that the feature is considered on the variant, while the red colour indicates its absence.



insiders in any venues, thus the venue prestige of outsiders citations varies greatly.

3.1.7.2 Comparing OTARIOS against other approaches

We compare OTARIOS against the state-of-the-art methods discussed in Section 3.1.2 and a baseline method named CountRank (CR) which counts the citations received by each author. We create three CR variants: uniform, individuality, and position. For each citation received, uniform assigns the same merit to all of the authors in publication (i.e., merit = 1), individuality equally divides the merit for all the authors (i.e., merit = $\frac{1}{|\mathcal{A}|}$), and position gives more credit to authors whose name appears first in the publication (first author: merit = 1, second author: merit = $\frac{1}{2}$, third author: merit = $\frac{1}{3}, ...$). Table 3.8 shows the results obtained for all state-of-the-art methods and 5 OTARIOS variants over all networks. For each network, we calculate NDCG and MRR for the top-5, 10, 20, 50 and 100 authors, and compute their mean values. Furthermore, we compute the mean NDCG and MRR across all networks.

In our experiments, SCEAS is the best state-of-the-art method, obtaining the highest mean NDCG (0.208) and the lowest mean MRR (691). The $\text{CR}_{position}$ method obtained the lowest NDCG mean (0.154), while NewRank obtained the highest mean MRR by a considerable margin (4091). An important aspect to highlight is that $\text{CR}_{individuality}$, despite being a baseline strategy, obtained the second highest NDCG and MRR across

Table 3.8: Comparison of state-of-the-art (STOA) methods against OTARIOS over all networks. The value of each cell is the metric's mean value for that network (e.g., the mean NDCG and MRR of AP+A+AW for network NET is highlighted in Table 3.6). In bold we highlight the highest score for each metric. The best STOA method (i.e., SCEAS) is colored in red and the best OTARIOS variant is colored in blue. Inside parentheses we show the gain of OTARIOS versus SCEAS, i.e., G_{NDGC} and G_{MRR}, respectively.

		NDGC						MRR					
	Method	CM	TC	NET	IS	SE	Mean	CM	TC	NET	IS	SE	Mean
State-of-the-art	$CR_{position}$	0.097	0.049	0.189	0.176	0.261	0.154	1427	463	1009	892	324	823
	YetRank	0.128	0.028	0.206	0.157	0.271	0.158	2083	673	1047	846	491	1028
	ALEF	0.152	0.020	0.182	0.129	0.323	0.161	1260	561	670	803	310	721
	$CR_{uniform}$	0.138	0.045	0.278	0.189	0.222	0.174	1659	516	1066	1067	387	939
	RLPR	0.180	0.032	0.231	0.176	0.338	0.191	1203	508	817	720	356	721
	SARA	**0.193**	0.035	0.232	0.156	0.354	0.194	1122	461	738	**668**	303	658
	NewRank	0.115	0.004	0.297	**0.319**	0.266	0.200	5057	3112	3597	6637	2050	4091
	$CR_{individuality}$	0.129	0.043	0.247	0.211	0.372	0.200	1171	438	878	744	289	704
	SCEAS	0.143	0.035	0.275	0.255	0.335	0.208	1154	493	776	752	279	691
OTARIOS	∅ + AVW + AW	0.143	0.081	0.323	0.213	0.315	0.215	1161	324	664	707	289	629
	∅ + V + AW	0.148	0.080	0.321	0.214	0.314	0.215	1169	325	671	709	294	634
	AP + VW + AW	0.150	**0.087**	0.330	0.268	**0.383**	0.244	**1070**	273	**604**	680	**207**	**567**
	AV + VW + AW	0.143	0.085	**0.356**	0.264	**0.383**	**0.246**	1333	285	618	676	215	626
	AP + A + AW	0.152	**0.087**	0.330	0.273	**0.383**	0.245 (+18%)	1079	**272**	606	688	**207**	570 (+21%)

the five networks, among the state-of-the-art methods.

With respect to OTARIOS variants, we tested 53 variants and 21 of them obtained higher mean MRR and mean NDCG than the best state-of-the-art method, SCEAS. The best mean NDCG and mean MRR that OTARIOS variants obtained were 0.246 and 567, respectively. Assuming that both NDCG and MRR measures have the same weight (i.e., are equally important), the best OTARIOS variant is (AP+A+AW), which uses (a) recency and volume to measure productivity, (b) recency to measure outsiders influence, and (c) recency and individuality to measure the score term. This variant obtained a mean NDCG of 0.245 and a mean MRR of 570. We compared the gain of this variant with respect to state-of-the-art (STOA) methods, using equations 3.8 and 3.9. Compared to RLPR, a topology-based author ranking algorithm, we achieved a gain of 28% in terms of NDCG and 27% in terms of MRR. With respect to the best feature-enriched author ranking method (SCEAS), we achieved a gain of 18% in terms of NDCG and 21% in terms of MRR.

$$\text{Gain}_{NDGC} = \frac{\text{OTARIOS}_{<NDGC>} - \text{STOA}_{<NDGC>}}{min(\text{OTARIOS}_{<NDGC>}, \text{STOA}_{<NDGC>})} \tag{3.8}$$

$$\text{Gain}_{MRR} = \frac{\text{STOA}_{<MRR>} - \text{OTARIOS}_{<MRR>}}{min(\text{OTARIOS}_{<MRR>}, \text{STOA}_{<MRR>})} \tag{3.9}$$

3.1.7.3 Using the outsiders to compute author ranking

In our previous experiments, state-of-the-art methods only used the author citation network of the insider authors (i.e., outsiders were not part of the network). However, for the OTARIOS variants, since we require outsiders to calculate outsider influence features, we used a network consisting of insiders and outsiders. In order to demonstrate that we were not unfairly comparing our variants with other methods with less information, we tested the state-of-the-art algorithms using the complete network (i.e., outsiders + insiders) and compared those results with the ones obtained using only the insiders network. Table 3.9 shows the results of this comparison [3]. The results indicate that, on average, the state-of-the-art methods obtained a negative gain of -17% for NDCG and -25% for MRR when using the full network. The NewRank and SCEAS methods were the ones that presented the highest losses (-54% and -30% on NDCG, and -63% and -37% on MRR, respectively). These methods were

[3]Gains estimated using equations 3.8 and 3.9

Table 3.9: Gain of using outsiders as part of the network in the score diffusion step. The *fullnet* versions incorporate outsiders in the network, i.e., they convert outsiders in insiders. Note that OTARIOS does not use outsiders as part of the network in the score diffusion step, only in the initialisation step. The mean of both NDCG and MRR is highlighted, showing that, overall, STOA methods' performance degrades when they use outsiders as insiders.

	NDCG						MRR					
Method	CM	TC	NET	IS	SE	Mean	CM	TC	NET	IS	SE	Mean
SCEAS	**0.144**	**0.036**	**0.275**	**0.255**	**0.335**	**0.209**	**1154**	**493**	**776**	**753**	**279**	**691**
Fullnet SCEAS	0.106	0.024	0.224	0.198	0.250	0.160	1517	845	929	999	433	945
Gain	-36%	-51%	-23%	-29%	-34%	-30%	-31%	-71%	-20%	-33%	-55%	-37%
SARA	**0.194**	**0.036**	**0.232**	0.157	**0.355**	**0.195**	**1123**	**461**	**739**	**668**	**303**	**659**
Fullnet SARA	0.181	0.030	0.227	**0.177**	0.300	0.183	1146	602	885	719	408	752
Gain	-7%	-20%	-2%	+13%	-18%	-6%	-2%	-31%	-20%	-8%	-35%	-14%
RLPR	**0.181**	**0.032**	**0.231**	**0.177**	**0.338**	**0.192**	**1203**	**508**	**817**	**721**	**357**	**721**
Fullnet RLPR	0.174	0.027	0.227	0.162	0.276	0.173	1274	728	864	757	436	812
Gain	-4%	-18%	-2%	-9%	-22%	-11%	-6%	-43%	-6%	-5%	-22%	-13%
ALEF	**0.152**	0.021	0.183	0.130	**0.323**	**0.162**	**1261**	**561**	**670**	804	**310**	**721**
Fullnet ALEF	0.125	**0.024**	**0.203**	**0.151**	0.299	0.160	1373	608	930	**735**	432	816
Gain	-22%	+13%	+11%	+16%	-8%	-1%	-9%	-8%	-39%	+9%	-39%	-13%
NewRank	**0.116**	0.004	**0.297**	**0.320**	**0.267**	**0.201**	**5057**	**3113**	**3598**	6638	2050	**4091**
Fullnet NewRank	0.089	**0.020**	0.180	0.191	0.170	0.130	11277	6951	6877	**6541**	**1651**	6659
Gain	-29%	+381%	-66%	-68%	-57%	-54%	-123%	-123%	-91%	+1%	+24%	-63%
YetRank	0.128	0.028	0.206	**0.158**	**0.272**	0.158	2084	**673**	**1048**	846	**492**	**1029**
Fullnet YetRank	**0.157**	**0.029**	**0.224**	0.149	0.259	**0.163**	**2031**	874	1200	**836**	561	1101
Gain	+22%	+1%	+9%	-6%	-5%	+3%	+3%	-30%	-15%	+1%	-14%	-7%

among the top state-of-the-art methods when considering only the insiders network, as a result the complete network had a higher negative impact when compared to other methods that obtained worse results when considering only the insiders (e.g., NewRank). The only method that presented an overall positive gain was YetRank in terms of NDCG. This test demonstrated that adding more authors to the citation network decreases the overall performance of the state-of-the-art methods when there is incomplete information about the new authors (i.e., their received citations are unknown) and they are treated equally as those authors whose full citation network is known. Thus, this further corroborates our hyphotesis that incomplete networks should be carefully divided into fully known nodes and partially known nodes.

3.1.8 Summary

In this section, we presented OTARIOS a new feature-enriched author ranking algorithm, and compared it against (a) traditional author ranking algorithms, (b) topology-based author ranking algorithms, and (c) feature enriched author ranking algorithms. Previous author ranking methods did not combine relevant information effectively, such as the author's productivity and the citations' relevance. Furthermore, previous

methods assume that the full network is known, which is not true for most real cases. We thus divided the network into insiders (i.e., the authors that we want to rank) and outsiders (i.e., the authors that cite insiders but we do not rank). In our experiments, we analysed which publication/citation information is more relevant and how it can be efficiently combined.

We obtained the best results when OTARIOS considers (i) the author's publication volume and publication recency, (ii) how recently his work is being cited by outsiders, and (iii) how recently his work is being cited by insiders and how individual his work is (i.e., publishing with few authors is better). This evaluation was performed on a set of five networks where the ground-truth was the number of best awards in the conferences belonging to the specific network. Our tests showed that OTARIOS is >30% more efficient than topology-based author ranking methods, namely RLPR, and is >20% more efficient than other feature-enriched author ranking methods. We demonstrated that OTARIOS efficiently uses outsiders (i.e., authors whose received citations are not fully known) on the score initialisation process. Furthermore, we showed that adding outsiders to the score diffusion process decreases the performance of the state-of-the-art algorithms. These results indicate that current methods have poor results on networks where some nodes have missing information (which is true for most real cases), while OTARIOS is able to use nodes with limited information adequately.

3.2 FOCAS

3.2.1 Motivation

Regardless of using traditional, topology-based or featured enriched methods for author ranking, citations are the main source of scientific impact evidence. Undoubtedly, the quality of the author's work is correlated with his number of citations [109]. However, other factors such as the author's co-authorship network [101] and his social behaviour [31, 110] also have a big effect on his citation count. As a result, not all citations should be equally valued and a further analysis to determine the value of a citation depending on the context and motive is necessary to promote fairer author rankings.

In this section, we focus on the aspect of social behaviour of authors. Due to the importance of citations in estimating scientific impact, authors can abuse certain

citation patterns to boost someone's (or their own) citation count, thus increasing the author's perceived scientific impact. There are three types of patterns often used to boost citations, namely (i) self-citations, when an author cites his own work, (ii) reciprocated citations, when authors or groups of authors interchangeably cite each other, and (iii) co-author citations, when authors cite works of their co-authors. Note that there may be nothing inherently malicious in using these citation patterns since in certain cases it makes sense for an author to cite his previous work, cite other people that have cited him, or cite the work of his co-authors, as long as the publications are in the same research line. However, by abusing these practices, authors can unreasonably boost their number of citations and consequently their perceived scientific impact, as shown in several studies [31, 32, 111, 110]. Thus, it is important to provide the scientific community with tools to mitigate the effect of citation boosting in author ranking algorithms.

Despite several studies showing that the abuse of citation boosting practices leads to undeserved scientific impact [31, 109, 110], author ranking algorithms do not address their negative effect in ranking estimation. At most, some methods remove self-citations from the data in a pre-processing step [20]. We should point out the existence of the c-index [32], an h-index based algorithm that counts the number of citations that an author has received from a distance bigger than c in his co-authorship network. However, the c-index does not analyse the citation network, thus it has the same drawbacks of other traditional author ranking methods (Section 2.2).

3.2.2 Overview of our contribution

In this section, we merge the citation boosting patterns self-citations and co-author citations into a group named *friendly citations* and we propose Friends-Only Citation AnalySer (FOCAS). FOCAS is a penalty estimation algorithm that analyses the co-authorship and citation networks in order to decrease the effect of friendly citations in author ranking algorithms. We present three different criteria used to capture friendly citations: authors' distance, co-authorship frequency and co-authorship recency. FOCAS does not rank authors by itself, instead it is designed to be easily integrated with any existing graph-based Author-Level Author Ranking (ALAR) algorithms. We select ALAR algorithms due to the closer similarities between author-level and co-authorship networks (i.e., the nodes represent authors in both networks).

We evaluate FOCAS on eight ALAR algorithms (five variants of our OTARIOS algorithm and three approaches from the state-of-the-art). Our results show that FOCAS

improves the author rankings on an average of 25% and in the best case on 46%. The eight algorithms were tested in a network consisting of the top conferences in the area of Information Retrieval and evaluated by their ability to produce a ranking close to one produced using human judgement (the same ground-truth ranking used in Section 3.1). We also observed that FOCAS obtained the best results when considering the distance criterion to penalise friendly citations.

3.2.3 Methodology

FOCAS is a citation penalty estimator for author-level citation networks that aims to reduce the effect of friendly citations in author ranking. FOCAS, by itself, does not rank authors; instead it estimates a penalty $p \in [0, 1]$ for every citation. Thus, ALAR algorithms can easily incorporate this information in their ranking estimations. Furthermore, FOCAS is a flexible algorithm that estimates different penalties depending on user-defined criteria.

While ALAR algorithms differ in the criteria used to rank authors, they are similarly divided into two main steps. On the first one a vector R is defined as the initial score for all the authors, this process is referred as *score initialisation*. On the second step a vector S, containing the scores of the authors, is continuously updated using the edges' weights until convergence, thus leading to the authors final ranking. This step is known as *score diffusion*. Note that at the first iteration, $R = S$.

We present our method in the following sections. First we describe how friendly citations are penalised. Then, we detail how an initial version, named FOCAS-naive, applies the penalties before score initialisation. Finally, we put forward FOCAS, which iteratively applies the penalties during the score diffusion step.

3.2.3.1 Penalising friendly citations

We use citation and co-authorship networks to calculate penalties for friendly citations. Table 3.10 presents the notation used throughout this section. The citation network has authors as nodes and citations as edges. In more detail, author a' cites author a''. Each citation $(a' \to a'')$ is made in year $(a' \to a'')_y$ and has weight $(a' \to a'')_w$. The weight is computed by an ALAR algorithm (e.g., OTARIOS) and measures the impact of the citation. Citation with higher weights have higher impact on the ranking of the cited author.

Table 3.10: Notation table.

Notation	Description
$a' \rightarrow a''$	author a' cites author a''
$(a' \rightarrow a'')_y$	year of the citation $a' \rightarrow a''$
$(a' \rightarrow a'')_w$	weight of the citation $a' \rightarrow a''$
$(a' \rightarrow a'')_p$	penalty of the citation $a' \rightarrow a''$
$a' \leftrightarrow a''$	author a' and a'' are co-authors on a publication
$(a' \leftrightarrow a'')_y$	year of the collaboration $a' \leftrightarrow a''$
$C_{(a' \leftrightarrow a'')}$	set of collaborations between a' and a''
$\Delta(C_{(a' \leftrightarrow a'')}, y)$	year of the most recent collaboration between authors a' and a'' prior to year y
$\Phi(C_{(a' \leftrightarrow a'')}, y)$	number of collaborations between authors a' and a'' prior to year y
$p(a' \leftrightarrow a'')$	path between a' and a'' in the co-authorship network
$P(a' \leftrightarrow a'')$	all paths between a' and a'' in the co-authorship network
R	authors' initial score
S	authors' estimated score
$S(a' \rightarrow a'')$	score of citation $a' \rightarrow a''$
$R_{a'}$	a' initial score
$S_{a'}$	a' estimated score

The co-authorship network has authors as nodes and collaborations as edges. More concretely, author a' co-authors a publication with author a''. Each collaboration $(a' \leftrightarrow a'')$ is published in year $(a' \leftrightarrow a'')_y$. Note that authors a' and a'' may collaborate multiple times, thus we define $C_{(a' \leftrightarrow a'')}$ as the set of collaborations between them. Additionally, $\Delta(C_{(a' \leftrightarrow a'')}, y)$ is the year of their most recent collaboration prior to year y and $\Phi(C_{(a' \leftrightarrow a'')}, y)$ is the number of times they collaborated prior to year y.

We now describe how citations penalties are computed (Algorithm 3.1). We iterate over all citations $(a' \rightarrow a'')$ in the citation network N_c (line 1). Initially, we assign no penalty to the citation (line 2). In more detail, if authors a' and a'' never co-authored a paper together, and there is not a path between a' and a'', the citation has no penalty. Otherwise, we find set $Q \subseteq P(a' \leftrightarrow a'')$ which contains all paths between a' and a'' in the co-authorship network N_a constrained by $(a' \rightarrow a'')_y$, i.e., only co-authorships previous to the citation are used to calculate penalties (line 3). This means that citations from author a' to author a'' can have different penalties in different years. For example, two authors have no penalty in year y because they never collaborated; then, if they co-author a paper in year y, they will have a penalty in year $y + n : n > 1$. For efficiency purposes, we only consider paths with distance $p(a' \leftrightarrow a'')_d \leq 3$. Our co-authorship network is a small-world network, which is typical

Algorithm 3.1 Penalty estimation.

Input: Co-authorship network N_a, citation network N_c, criteria θ.
Output: Penalties $P_c = \{(a' \rightarrow a'')_p : \forall (a' \rightarrow a'') \in N_c\}$.

```
1:  for a' → a'' ∈ N_c do
2:      (a' → a'')_p = 0
3:      Q = getCoAuthorPaths(a', a'', (a' → a'')_y, N_a)
4:      for p(a' ↔ a'') ∈ Q do
5:          p_q = 1
6:          for a^i ↔ a^j ∈ p(a' ↔ a'') do
7:              p_q = p_q × calculatePenalty(a^i ↔ a^j, θ)
8:          if p_q > (a' → a'')_p then
9:              (a' → a'')_p = p_q
10:     P_c = P_c ∪ (a' → a'')_p
11: return P_c
```

for collaboration networks between scientists [112]. Thus, even for small distances, we find paths between many authors that are not co-authors. We then calculate the penalty p_q for all $p(a' \leftrightarrow a'') \in Q$ using criteria θ (e.g., frequency) and store the largest penalty found as the final penalty $(a' \rightarrow a'')_p$ for the citations between a' and a'' in year $(a' \rightarrow a'')_y$ (lines 4-9). Finally, we add the penalty to the list of penalties P_c. Note that we calculate penalties for direct co-authors and the total penalty is the product of the penalties in the path. For example, if a is a co-author of b with penalty 0.5 and b is a co-author of c with penalty 0.7, the penalty of path a to c is 0.5×0.7 (lines 6-7). If multiple paths exist between a and c, the final penalty between them is the maximum penalty found. For example, if we found three paths with penalties 0.5, 0.7×0.3, and 0.2, the final penalty is 0.5 (lines 8-9).

We use three different criteria to compute the penalties: distance, frequency and recency. The three criteria capture different properties of collaborations and give higher penalties to different types of citations. In more detail, the distance criterion (D-FOCAS) gives higher penalties to citations between authors that are closer in the co-authorship network. For example, if author a is at distance 1 to author a' (i.e. they have co-authored at least one publication) and has no path to author a'' in the co-authorship network, then the citation $a \rightarrow a'$ has an higher penalty than $a \rightarrow a''$ (i.e., $(a \rightarrow a')_p > (a \rightarrow a'')_p$). The distance penalty is calculated using the following

equation:

$$D(a' \to a'') = d = \begin{cases} 0.75, & \text{if } \Phi(C_{(a' \leftrightarrow a'')}, y) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

The distance criterion applies a penalty of $d = 0.75$ to co-authors and 0 otherwise. We use the value of 0.75 since it is approximately the highest expected penalty between co-authors using the any of the other two criterion. The frequency criterion (F-FOCAS) gives higher penalties to citations between authors that collaborate more frequently. For example, if author a has collaborated 5 times with author a' and 3 times with author a'', then the citation $a \to a'$ has an higher penalty than $a \to a''$ (i.e., $(a \to a')_p > (a \to a'')_p$). The frequency penalty is calculated using the following equation:

$$F(a' \to a'') = 1 - e \left(\frac{\Phi(C_{(a' \leftrightarrow a'')}, (a' \to a'')_y)}{\lambda} \right)^{-1} \tag{3.11}$$

The recency criterion (R-FOCAS) gives higher penalties to citations between authors that have collaborated more recently. For example, if author a has collaborated with author a' 1 year ago and with author a'' 5 years ago, then the citation $a \to a'$ has an higher penalty than $a \to a''$ (i.e., $(a \to a')_p > (a \to a'')_p$). The recency penalty is calculated using the following equation:

$$R(a' \to a'') = e \left(\frac{(a' \to a'')_y - \Delta(C_{(a' \leftrightarrow a'')}, (a' \to a'')_y)}{\lambda} \right)^{-1} \tag{3.12}$$

Note that the frequency and recency equations use a decay parameter λ that regulates the function's slope. In our experiments we set $\lambda = 4$. Furthermore, the letter y in all the equations represents the year of the citation.

Note that it is possible to combine the different criteria. For example, we can combine frequency with recency, thus decreasing the weight of the citations between authors that co-authored multiples times recently. Our nomenclature for that variation is FR-FOCAS. Other possible criteria combinations are: DF-FOCAS, DR-FOCAS and DFR-FOCAS. In total, we have seven variations. We should note that FOCAS handles self-citations as a special case. More concretely, independently of the criteria used, penalty $(a' \to a'')_p = 1$ when $a' = a''$. Thus, self-citations have weight $(a' \to a'')_w = 0$ and are removed from the citation network.

Table 3.11: Penalties using three different criteria for citation $a1 \rightarrow a4$ in 2016 from the co-authorship network of Figure 3.3. Penalties for co-authors (i.e., direct connections) are calculated using Equations 3.10, 3.11, and 3.12. Penalties for indirect connections are the product of penalties of the co-authors chain (e.g. $(a1 \rightarrow a2 \rightarrow a4)_p = (a1 \rightarrow a2)_p \times (a2 \rightarrow a4)_p$). η is the number of collaborations between two co-authors, δ is the difference in years between the citation and the most recent collaboration of two co-authors (e.g., 2016 - 2009). Bold values indicate the path from a1 to a2 with the highest penalty for the respective criteria.

Path	Distance (D)	Frequency (F)	Recency (R)
$a1 \leftrightarrow a2$	0.75	$(\eta = 4)$ 0.63	$(\delta = 7)$ 0.17
$a2 \leftrightarrow a4$	0.75	$(\eta = 3)$ 0.53	$(\delta = 5)$ 0.29
$a1 \leftrightarrow a3$	0.75	$(\eta = 1)$ 0.22	$(\delta = 1)$ 0.78
$a3 \leftrightarrow a4$	0.75	$(\eta = 1)$ 0.22	$(\delta = 3)$ 0.47
$a1 \leftrightarrow a2$	**0.75**	$(\eta = 1)$ 0.22	$(\delta = 9)$ 0.10
$a1 \leftrightarrow a2 \leftrightarrow a4$	$0.75 \times 0.75 = 0.56$	$\mathbf{0.63 \times 0.53 = 0.34}$	$0.17 \times 0.29 = 0.05$
$a1 \leftrightarrow a3 \leftrightarrow a4$	$0.75 \times 0.75 = 0.56$	$0.22 \times 0.22 = 0.05$	$\mathbf{0.78 \times 0.47 = 0.37}$

To further clarify the reader, we show an example in Table 3.11 of how penalties are calculated using different criteria for a given citation of the co-authorship network presented in Figure 3.3. This example highlights how different the penalties are when using different criteria. In this case we are considering different paths between the same citing and cited author. However, if we consider a case where the only path available is the direct one (e.g., $a1 \leftrightarrow a2$ in the illustrated network), then the penalty applied to the citation varies from a maximum of 0.75 (D-FOCAS) to a minimum of 0.10 (R-FOCAS). Thus, one most carefully decide which criteria to use.

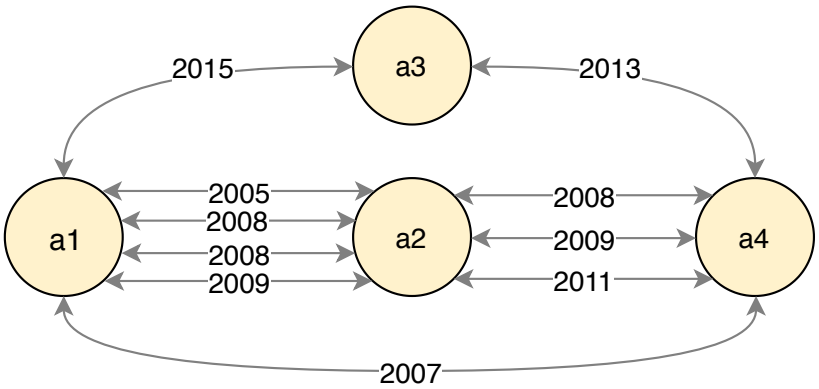


Figure 3.3: Example of a co-authorship network.

3.2.3.2 FOCAS-naive

We now describe FOCAS-naive, an initial version of FOCAS which applies penalties, calculated as described in Section 3.2.3.1), before score initialisation. Thus, FOCAS-naive can be used as a pre-processing step of ALAR algorithms.

FOCAS-naive (Algorithm 3.2) iterates over all citations $(a' \rightarrow a'')$ in the citation network N_c (line 1) and calculates the new citation weight $(a' \rightarrow a'')'_w$ based on the original weight $(a' \rightarrow a'')_w$ and the penalty $(a' \rightarrow a'')_p$ (line 2). The citation network with new citation weights is then used by the ALAR algorithms during the score diffusion step; thus, they will obtain different author rankings.

In each iteration of the score diffusion step of ALAR algorithms, every citing author divides his score (obtained from the previous iteration) and distributes it among his cited authors according to his citations weights. Therefore, cited authors with higher weights receive more score. For example, if $(a1 \rightarrow a2)_w = 0.6$, $(a1 \rightarrow a2)_w = 0.3$, and the score from the previous iteration $S_{a1} = 0.8$, then $a2$ receives $0.8 \times \frac{0.6}{0.6+0.3} \approx 0.53$ and $a3$ receives $0.8 \times \frac{0.3}{0.6+0.3} \approx 0.27$. FOCAS-naive decreases the weight of friendly citations. Consequently, whenever a citation $a1 \rightarrow a2$ is penalised, the score received by other authors cited by $a1$ is increased. Considering the previous example, if $(a1 \rightarrow a2)_p = 0.5$ and $(a1 \rightarrow a2)_p = 0$, then the new weight $(a1 \rightarrow a2)_{w'} = 0.6 \times (1 - 0.5) \approx 0.3$ which is the same as $(a1 \rightarrow a3)_{w'}$, thus both $a2$ and $a3$ received a score of 0.4 from $a1$'s citation. After applying the penalty, $a2$ decreases the score received from $a1$'s citation from 0.53 to 0.4 and $a3$ increases the received score from $a1$'s citation from 0.27 to 0.4. Therefore, FOCAS-naive not only decreases the score/impact of authors with (many) friendly citations but it also increases the impact of authors without (many) friendly citations.

This idea fits our goal, but FOCAS-naive fails to penalise friendly citations in certain scenarios. Let us consider that $a1$ only cites $a2$ and $a3$ with respectively citation weights $w2$ and $w3$; if the same (or a similar) penalty is calculated for both citations,

Algorithm 3.2 FOCAS-naive.

Input: Citation network N_c, citation penalties $P_c = \{(a' \rightarrow a'')_p : \forall (a' \rightarrow a'') \in N_c\}$.
Output: Citation network N_c with redefined weights.

1: **for** $a' \rightarrow a'' \in N_c$ **do**
2: $\quad (a' \rightarrow a'')_{w'} = (a' \rightarrow a'')_w \times (1 - (a' \rightarrow a'')_p)$;
3: **return** N_c

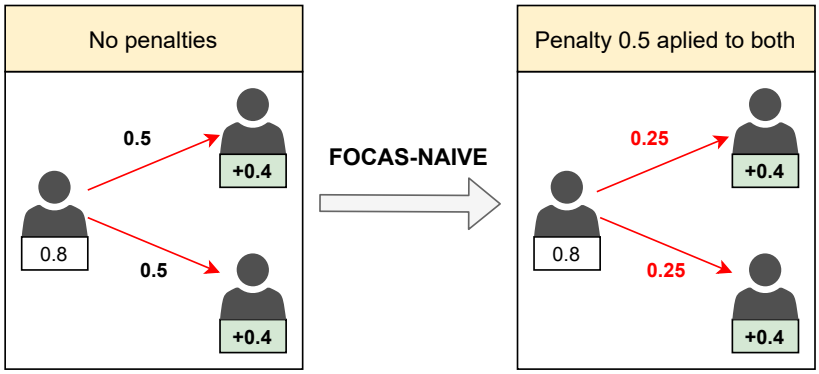


Figure 3.4: Applying the FOCAS-NAIVE penalty in cases where all the cited authors have similar citation weights and penalties. In these cases, FOCAS-NAIVE fails to penalise any of the cited authors and the received scores are nearly the same.

then the authors still receive (nearly) the same score from $a1$ as if their citations were not penalised. Furthermore, in cases where $a1$ only cites $a2$ (once or many times), $a2$'s receives a score from $a1$ that is independent of the penalty assigned to the citations. Figure 3.4 presents an illustration of this scenario. To overcome this limitation, we propose FOCAS in the next section.

3.2.3.3 FOCAS

We now describe FOCAS, an improved version of FOCAS-naive, which applies penalties during the score diffusion step of ALAR methods. Therefore, FOCAS is integrated during runtime with ALAR methods.

FOCAS (Algorithm 3.3) calculates penalties $(a' \rightarrow a'')_p$ for every citation as described in Section 3.2.3.1. Initially, the ALAR method calculates vector R which contains the authors' initial score (line 1; we do not specify parameters for the function since they depend on the ALAR method). Then, the authors' scores are initialised with the initial scores (line 2) and the score diffusion step begins (line 3). At each iteration, all authors' scores S and their total penalised score ($\text{total}_{penalty}$) are initialised 0 (lines 4 and 5, respectively). Then, the ALAR method iterates over all citations $a' \rightarrow a''$ in the citation network N_c (line 6). For every citation, the ALAR algorithm calculates the score $S(a' \rightarrow a'')$ given from author a' to a'' based on the properties of the citation $a' \rightarrow a''$ and the previous iteration score $S'_{a'}$ (line 7; we do not specify the calculation of this step since it is ALAR dependent). Then, the ALAR method adds the score $S(a' \rightarrow a'')$ to the cited author score S''_a and moves on to the next citation (this would imply skipping lines 8-11 in Algorithm 3.3). When FOCAS is integrated with

Algorithm 3.3 FOCAS.
Input: Citation network N_c, citation penalties $P_c = \{(a' \to a'')_p : \forall (a' \to a'') \in N_c\}$.
Output: Author scores S.

1: $R =$ ALAR_Initialisation()
2: $S' = R$
3: **while** True **do**
4: $\quad S = \{S'_a = 0 : \forall a' \in R\}$
5: $\quad \text{total}_{penalty} = 0.0$
6: $\quad$ **for** $a' \to a'' \in N_c$ **do**
7: $\quad\quad S(a' \to a'') = ALAR\_\text{Score}(a' \to a'', S'_{a'})$
8: $\quad\quad S_{a''} += S(a' \to a'') \times (1 - (a' \to a'')_p)$
9: $\quad\quad \text{total}_{penalty} += S(a' \to a'') \times (a' \to a'')_p$
10: $\quad$ **for** $a' \in S$ **do**
11: $\quad\quad S_{a'} = \text{total}_{penalty} \times R_{a'}$
12: $\quad$ **if** converged(S, S') **then**
13: $\quad\quad$ break
14: $\quad S' = S$
15: **return** S

the ALAR method there is a new step, before adding the score to S''_a, where friendly citations are penalised. FOCAS removes a portion $(a' \to a'')_p \in [0, 1]$ from $S(a' \to a'')$ (line 8). However, in order to maintain the scores stable (i.e., the sum of all the authors' scores equal to 1) and to guarantee that the ALAR method will eventually converge, FOCAS cannot simply remove scores. Thus it stores the total penalised score ($\text{total}_{penalty}$) to reallocate it at a later stage (line 9). After iterating over all citations, FOCAS iterates over all the authors (line 10) and gives a portion of the total penalty $\text{total}_{penalty}$ to each one according to their initial score (line 11; i.e., authors with higher initial score receive an higher portion of the penalised score). Finally, the normal process of ALAR algorithms resumes, namely, (i) checking if the stopping criteria is met (i.e., if the scores have converged, calculated by comparing if the scores S are *too similar* to the scores from the previous iteration S') (lines 12 and 13) and (ii) updating the scores for the next iteration, if necessary (line 14). At the end of the process, FOCAS obtains the authors' scores calculated by an ALAR method and penalising friendly citations. For further clarification, lines 5 and 8-11 in Algorithm 3.3 are specific to FOCAS while the remaining lines are general to ALAR methods. Figure 3.5 shows the score received from authors in the previous example where an author cites two other authors with the same weight and the same penalty. By contrast to FOCAS-naive, FOCAS can efficiently penalise friendly citations for any scenario in the citation network.

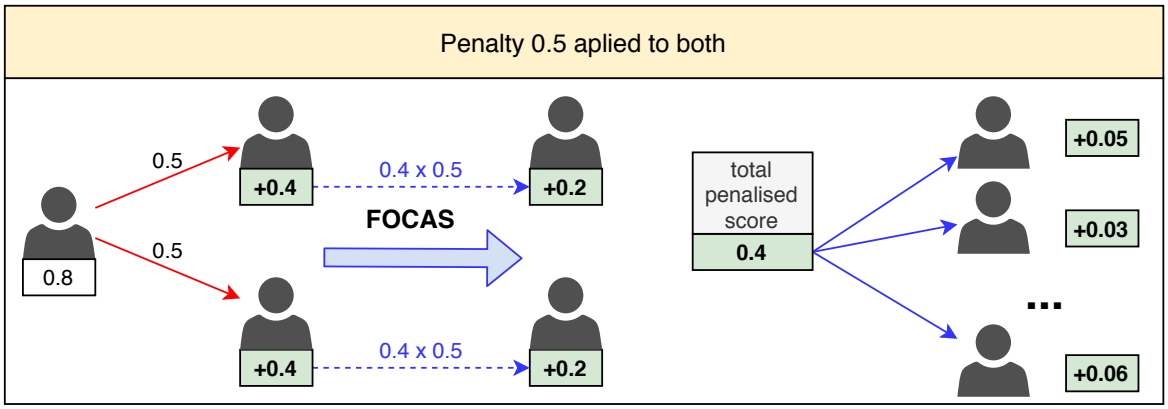


Figure 3.5: Applying the FOCAS penalty in cases where all the cited authors have similar citation weights and penalties. FOCAS successfully penalises the scores that come from friendly citations for all scenarios.

3.2.3.4 FOCAS-naive *versus* FOCAS

FOCAS decreases the score/impact given through friendly citations, consequently, FOCAS decreases the score of authors that have many friendly citations. However, by reducing the score of an author, and in order to keep the ranking system stable, ALAR methods automatically benefit some other authors. More specifically, scores cannot be removed, instead they are reallocated to other authors. FOCAS-naive and FOCAS differ on how they reallocate the penalised score to other authors. FOCAS-naive benefits authors that are not being cited by friendly citations, but that the citing author is using friendly citations to cite other authors. For example, consider that b has a friendly citation from a and c has a non-friendly citation from a, FOCAS-naive penalising the citation $a \rightarrow b$ results in higher scores given from a to c. On the other hand, FOCAS reallocates the penalised scores through the authors according to the score initialisation of authors (i.e., authors with higher score initialisation receive a larger part of the penalised score) which is calculated by the ALAR method.

3.2.4 Experimental setup

In this section we study a real-world co-authorship and citation networks and we integrate FOCAS-naive and FOCAS with existing ALAR methods. Our aim is to show that friendly citations are frequent in real-world datasets and that both FOCAS-naive and FOCAS improve the authors rankings of ALAR methods.

3.2.4.1 Evaluation scenario

In order to create a test scenario, we build a citation and co-authorship network using the publications extracted from the the DBLP dataset [1] for 7 top-tier conferences (KDD, CIKM, PODS, SIGMOD, VLDB, WWW, SIGIR) in the area of Information System from Computer Science. There are a total of 28,266 different authors in these publications. This value corresponds to the number of nodes in each network. Furthermore, there are 5.77 million citations and 0.15 million collaborations. These values correspond to the number of edges in citation and co-authorship networks, respectively. Similarly to the ground-truth presented in Section 3.1.7, we create a ground-truth author ranking based on the best paper awards given by the 7 conferences. Again, in our ground-truth ranking we are assuming that authors that have won more best paper awards with fewer co-authors should be ranked higher. In our experiments, rankings produced by ALAR or ALAR + FOCAS methods are compared to the ground-truth ranking. We use the NDCG metric for the top 5, 10, 20, 50 and 100 authors to compare both rankings. To ease results interpretability and discussion, we only show the average NDCG obtained for these values.

We evaluate FOCAS-naive and FOCAS on 8 different ALAR algorithms: RLPR [25], SCEAS [28], SARA [20] and five variants of our proposed author ranking algorithm, named OTARIOS (Section 3.1). OTARIOS variants are numbered from 1 to 5 according to their respective criteria *(_ + A + AW)*, *(_ + AVW + AW)*, *(AP + _ + AW)*, *(AP + A + AW)* and *(AV + VW + AW)*. We decided to not use the NewRank and YetRank algorithms (used in the OTARIOS experiments) in this experiment since they were originally proposed for paper-level citation networks. We run each ALAR on the citation network and calculate the average NDCG as our baselines to beat. The goal of our experiment is not to determine which is the best ALAR algorithm, instead we aim to measure the improvement of the produced rankings (i.e., the average NDCG) after adding FOCAS-naive and FOCAS to the ranking process. To measure the gain of the methods we use the following equation:

$$gain = \frac{NDCG_{FOCAS} - NDCG_{ALAR}}{min(NDCG_{FOCAS}, NDCG_{ALAR})} * 100\% \tag{3.13}$$

3.2.4.2 Frequency of friendly citations in real-world data

In order to show the frequency of friendly citations in real citation networks and how they diverge for different groups of authors, we measure the co-authorship distance

Table 3.12: Distribution of the co-authorship distance of the citations. *L-X* represents the level of distance with *L-0* corresponding to auto-citations and *L-N* corresponding to 4 or more. Network represents the citations for all the authors while *T* represents the ones incoming to authors with best paper awards. *T@N* represents the top *N* authors with the most awards.

	# Cits	L-0	L-1	L-2	L-3	L-N
Network	5.77M	2.08%	16.63%	**64.45%**	8.93%	7.92%
T@397	0.84M	1.91%	7.56%	12.98%	21.83%	**55.73%**
T@100	0.33M	2.04%	7.17%	13.27%	22.11%	**55.42%**
T@50	0.21M	2.23%	7.60%	13.16%	22.18%	**54.82%**
T@10	0.05M	2.94%	8.68%	11.15%	20.51%	**56.71%**
T@1	0.01M	5.69%	17.47%	15.42%	**37.27%**	24.15%

between citing and cited author in the citation network. Table 3.12 presents this analysis. We observe that $> 92\%$ of the citations are friendly citations and that most of them (i.e, $> 64\%$) have a co-authorship distance of 2. We filter the citations to compare the differences between the friendly citations received in general (i.e., the whole network) and the most prestigious authors (i.e., the ones with at least one best paper award). We observe that the distribution is similar within these groups of best authors across different levels of prestigious authors (i.e., T@397, T@100, T@50, T@10) and that the distribution is very different from the whole network. For the awarded authors, $> 55\%$ of their citations have a co-authorship distance higher than 3. The case of citations coming towards the most awarded author (T@1) is the exception. In this case his citations are on average closer to his co-authors when compared to other awarded authors, but they still are much farther away when compared to the whole network. We should note that the small-world network effect [112] is a justification for the high number of friendly citations in the network. However, it does not explain the different distribution between the whole network and the awarded authors, since we are just filtering citations and not recalculating their co-authorship distances.

Our exploration suggests that friendly citations are frequent in the network. Additionally, authors do not receive the same amount of friendly citations. In particular, authors that (on average) receive the most friendly citations are placed (on average) lower in our ground-truth ranking which is created based on human judgement. These facts further corroborate the necessity of penalising friendly citations in order to obtain more reliable author rankings.

Next we give an example of how friendly citations penalties affect the citation network. For this purpose we chose to single-out Ryen White (the best author according to our

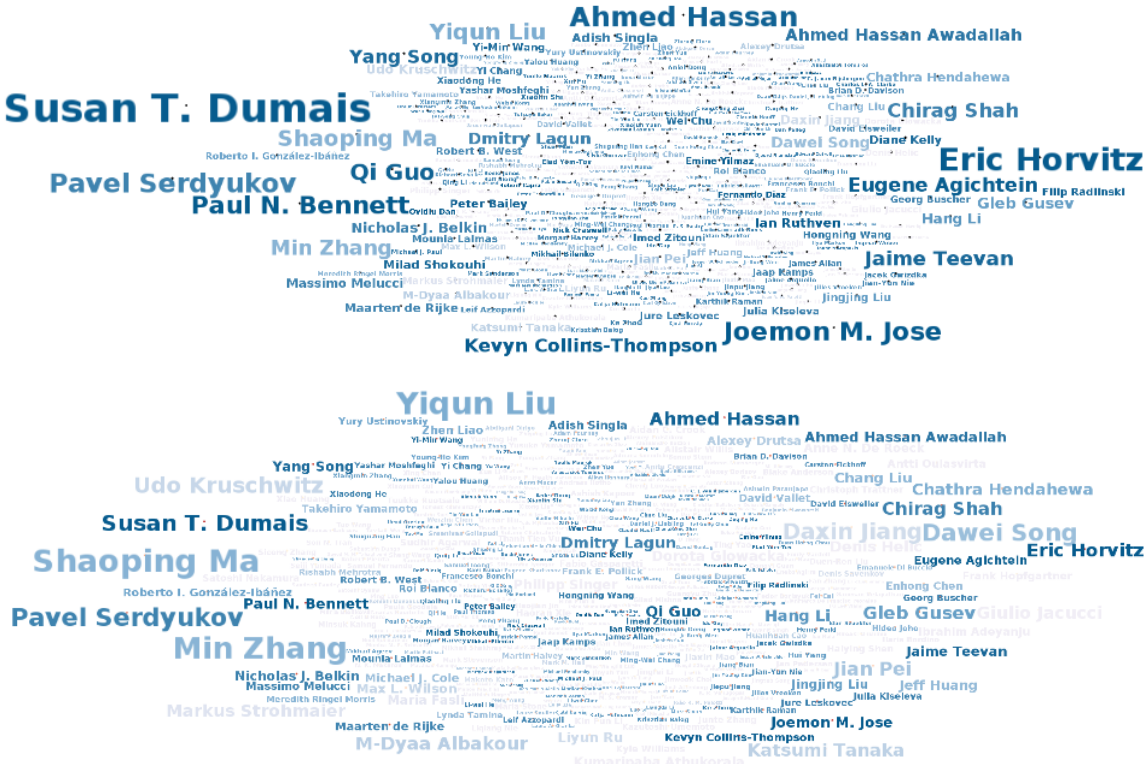


Figure 3.6: Ego-networks of the citations received by Ryen White (the best author according to the ground-truth) without any penalties (top figure) and with D-FOCAS penalty applied to the citation weights (bottom figure). Larger author names indicate that they have higher weights in Ryan White's citation network. Additionally, darker colours indicate that the author is close to Ryen White in his co-authorship network.

ground-truth) and we created two ego-networks of the citations he receives (Figure 3.6). The first ego-network uses a traditional approach [28, 20, 25] to calculate citation weights. The second ego-network uses the same approach to calculate citation weights and also applies the D-FOCAS penalties. More concretely, friendly citations are penalised based on co-authorship distance.

There are 1614 different authors citing Ryen White in our dataset. To ease visualisation, for both ego-networks, we removed citing authors whose citation weight (towards Ryen White) is lower than the average weight in the ego-network. The ego-network without D-FOCAS (top figure) and the ego network with D-FOCAS (bottom figure) have an average weight of 1.27 and 0.86, respectively, and there are 392 and 447 authors with weights above the average weight (i.e, visible in the figure), respectively. The effect of D-FOCAS is very noticeable by looking at the difference between the average weight in both ego-networks. D-FOCAS decreases the average weight by 0.41, which is a total loss of $\approx 32\%$ on the citation weights. We also observe that, without

any penalties, there are only a few citing authors that contribute the most to the citation weight (i.e., in-weight) received by Ryen White. These authors are identified by larger author names in the figure. Furthermore, these authors are close to Ryen White in the co-authorship network. This is represented by darker colours in the authors' names in the figure. Overall, the in-weight received by Ryen White is heavily based on a small set of authors which are close to him in the co-authorship network. After applying the D-FOCAS penalties, we now observe an increase on the number of authors that contribute the most to Ryen White's citation weight (i.e., there are more authors with larger names) and these authors have different co-authorship distances to Ryen White (i.e., the larger names now have a wider range of darker colours). Overall, the new weights are more evenly distributed by citing authors and less dependent on friendly co-author relations when no penalties are applied. As an example, for the case without penalties, citing authors Susan T. Dumais and Eric Horvitz were the ones with the highest contribution to Ryen White's in-weights with values of 33.73 and 26.67 respectively. After applying D-FOCAS penalties, their contribution decreased to values of 8.65 and 6.92 and they are now the 6th and 10th authors that contribute the most to Ryen White's in-weights.

There is a high correlation between the citation weights received by authors and their score in ALAR algorithms. Frequently co-authors (or authors close in the co-authorship network) are the ones that contribute the most for an author received citation weight. Although it is normal for an author to cite his co-authors, abusing this practice can lead to undeserved (perceived) scientific impact. In these cases, FOCAS tries to bring fairness to the author ranking process by making the authors' score calculation based on a more evenly distributed citation weight network. More specifically, authors do not benefit as much from their co-authors.

We should point that although Ryen White citation weights are penalised in our example, that does not mean that his citing authors abuse the co-author citation pattern. As we will see in more detail in the next section, due to the nature of FOCAS penalty estimation, the common case in the citation network is that authors lose citation weight and consequently score. However, this does not indicate that their final ranking position decreases.

3.2.4.3 The impact of FOCAS on author ranking

Here we combine FOCAS-naive and FOCAS with eight different ALAR methods and measure their improvements when compared to the original ALAR method.

Table 3.13: Results of the average NDGC @(5,10,20,50,100) for the STOA methods.

	Baseline	No self-citations
RLPR	0.176	0.4%
SCEAS	0.261	-2.1%
SARA	0.160	-2.3%
$OTARIOS_1$	0.213	-3.6%
$OTARIOS_2$	0.212	0.2%
$OTARIOS_3$	0.265	4.9%
$OTARIOS_4$	0.267	2.7%
$OTARIOS_5$	0.238	11.0%
Average gain		1.4%

Table 3.13 shows our baselines, i.e., the average NDCG obtained for the rankings produced by each ALAR method. The results show that $OTARIOS_4$ (0.267), $OTARIOS_3$ (0.265) and SCEAS (0.261) are the best methods, while RLPR (0.176) and SARA (0.160) are the worst. Removing self-citations is a common practice used by ALAR methods. As a result, we measure the gain in NDCG obtained by removing self-citations from the citation network when compared against the baselines. $OTARIOS_5$ is the algorithm that benefits the most from removing self-citations (11% gain). On the other hand, there are three ALAR methods that have negative gain: $OTARIOS_1$ (-3.6%), SARA (-2.3%) and SCEAS (-2.1%). Furthermore, we observe that removing self-citations only has a gain of 1.4% on average.

Table 3.14 shows the gains of combining ALAR methods with FOCAS-naive using different penalty criteria. There are three ALAR methods (RLPR, SCEAS, and SARA) that systematically have negatives gains regardless of the penalty criteria used by FOCAS-naive. We should point out that these methods all share the same method to calculate citation weights in the network. In the worst case, R-FOCAS-naive has a gain of -13.3% for SARA. On the other hand, the five OTARIOS variants consistently improve their rankings with FOCAS-naive. However, for four out of the five variants, the gains are only significantly high for the distance criteria (D). Overall, $OTARIOS_5$ is the only method that has significantly high gains ($> 10\%$) for all criteria. We also observe that distance is the only criteria that has significantly high gains (13.2% average) and recency is the worst one (-1.3% average). The remaining criteria have gains comparable to simply removing self-citations.

Table 3.15 shows the gains of combining ALAR algorithms with FOCAS using different penalty criteria. We observe that seven out of the eight ALAR methods (i.e., all except

Table 3.14: Gain on the average NDCG obtained by the ALAR algorithms after combining them with FOCAS-NAIVE using 7 different criteria. Bold value per row represents the criterion with the most gain.

	D	F	R	DF	DR	FR	DFR
RLPR	-8.3%	**2.3%**	-7.8%	**2.3%**	-3.9%	0.5%	0.5%
SCEAS	-9.9%	-2.4%	-9.2%	-2.3%	-7.7%	-2.4%	**-2.1%**
SARA	-9.7%	-5.9%	-13.3%	-6.0%	-6.9%	**-3.2%**	**-3.2%**
$OTARIOS_1$	**36.1%**	0.5%	5.2%	-0.3%	-2.5%	-3.6%	-3.6%
$OTARIOS_2$	**34.0%**	-0.2%	1.9%	0.2%	-2.5%	0.0%	-0.2%
$OTARIOS_3$	**22.6%**	6.7%	2.3%	6.9%	4.9%	5.7%	5.7%
$OTARIOS_4$	**16.0%**	-0,1%	-1.9%	0.0%	4.5%	2.9%	2.7%
$OTARIOS_5$	24.8%	20.1%	12.1%	**29.9%**	14.1%	12.0%	12.0%
Average gain	**13.2%**	2.5%	-1.3%	3.8%	0.0%	1.5%	1.5%

SCEAS) have positive gains for FOCAS regardless of the criteria used to calculate penalties. Six of the ALAR methods have gains $\geq 30\%$ for some criteria. $OTARIOS_2$ has the highest gain of 46.4%. SCEAS is the only method that does not have positive gains for any criteria and has the lowest gain of -8.7% for D-FOCAS and R-FOCAS. Overall, all FOCAS criteria present significantly high gains across ALAR algorithms. D-FOCAS has the highest average gain with 25.4% and FR-FOCAS has the lowest average gain with 3.8%.

Comparing FOCAS-naive against FOCAS, our results indicate that FOCAS is considerably better than FOCAS-naive. FOCAS improves more different ALAR algorithms

Table 3.15: Gain on the average NDCG obtained by the ALAR algorithms after combining them with FOCAS using 7 different criteria. Bold value per row represents the criterion with the most gain.

	D	F	R	DF	DR	FR	DFR
RLPR	**11.3%**	3.1%	4.5%	3.2%	7.8%	2.1%	1.2%
SCEAS	-8.7%	-3.3%	-8.7%	-4.1%	-2.3%	-3.0%	**-2.0%**
SARA	**30.0%**	5.5%	16.3%	2.9%	5.6%	2.3%	-0.9%
$OTARIOS_1$	32.0%	**39.3%**	-2.1%	39.0%	2.2%	2.1%	3.3%
$OTARIOS_2$	**46.4%**	39.0%	2.7%	38.6%	2.4%	0,7%	3.4%
$OTARIOS_3$	**32.5%**	8.8%	26.9%	8.7%	13.1%	8.8%	7.1%
$OTARIOS_4$	**23.3%**	21.6%	11.4%	21.7%	16.0%	2.0%	3.2%
$OTARIOS_5$	**36.6%**	32.0%	17.8%	31.7%	28.8%	15.5%	26.0%
Average gain	**25.4%**	18.6%	8.6%	17.7%	9.2%	3.8%	5.2%

with different criteria than FOCAS-naive, and FOCAS also improves them more than FOCAS-naive (i.e., higher average gains). Furthermore, our results show that the gains of FOCAS-naive are highly dependent on the process that estimates citation weights. RLPR, SCEAS, and SARA all share the same strategy to calculate citation weights and they all have very similar gains for all the criteria when FOCAS-naive is used. On the other hand, we observe that FOCAS gains are dependent on the quality of the score initialisation. RLPR and SCEAS are the only methods that use an uniform score initialisation strategy and, as a result, they are the ones with the smallest gains when FOCAS is used. We should also point out that SCEAS is the only method that does not have positive gains for neither FOCAS nor FOCAS-naive. This is not surprising because SCEAS converges in fewer iterations, meaning that the effect of the penalties (which grows as the iterative process of PageRank continues) are not noticeable.

Regarding the best criteria to penalise friendly citations, we observe that measuring the co-authorship distance between citing and cited authors and/or the frequency of their collaborations (i.e., D-FOCAS, F-FOCAS, and DF-FOCAS) yields the highest gains. Measuring how recent a collaboration is prior to a citation and its combinations with other base criteria (R, DR, FR and DFR) also yields positive gains; however they are much smaller.

In order to demonstrate the effect of FOCAS when measuring author impact, we compare the rankings and scores of authors on the OTARIOS_2 variant before and after applying the D-FOCAS penalties. We select OTARIOS_2 with D-FOCAS because, overall, this is the combination that produces the best results, with an average NDCG of 0.351. For brevity, we restrict our analysis to the top-10 authors from the ground-truth. Table 3.16 shows this analysis. We must first highlight the difficulty of ALAR methods in producing rankings similar to the rankings created using human judgement, in this case, using best paper awards. Only one of the top-10 authors of the ground-truth is placed in the top-10 of the ranking produced by OTARIOS_2 (with or without D-FOCAS) and six of the top-10 authors are placed outside the top-350 predicted authors. Regarding the differences in authors' scores after applying D-FOCAS, we observe that the top-10 authors lose 14% of their score on average. Ryen W. White loses the most score with a gain of -44% and Edo Liberty was the only one that presents a positive gain of 8%. The loss of score for the top-10 authors is not surprising; if we consider that due to their impact they are more likely to be cited (not only in quantity but also by different authors) and that due to the small-world effect of the co-authorship network (i.e., there is a small distance between most pair of authors)

Table 3.16: Impact of FOCAS with criterion distance (D) on the $OTARIOS_3$ baseline on the top 10 most awarded authors. Author names are sorted from the most awarded author to the lowest awarded one. *BR*: Baseline Rank, *PR*: Penalty Rank, *RI*: Rank Improvement, *BS*: Baseline Score, *DFSG*: D-FOCAS Score Gain and *# CIT*: number of citations received. The number of citations only considers citations received from publications from the 7 conferences of our dataset.

Author	BR	PR	RI	$10^{-2}\times$ BS	DFSG	# CIT
Ryen W. White	31	28	+3	0.225	-44%	5749
Pedro M. Domingos	24	14	+10	0.246	-13%	9202
Marcelo Arenas	381	372	+9	0.044	-20%	2602
Leonid Libkin	607	483	+124	0.029	5%	1433
Gerhard Weikum	29	18	+11	0.228	-17%	11566
Georg Gottlob	628	601	+27	0.029	-7%	2329
Edo Liberty	751	598	+153	0.025	8%	244
Ian Ruthven	531	675	-144	0.033	-35%	681
Jan Van den Bussche	2347	2192	+155	0.008	13%	554
Thorsten Joachims	2	1	+1	0.619	-30%	10984

they are more likely to receive a friendly citation, it is expected that their score is negatively affected. Our results also show that despite the fact that the top-10 lose score after applying D-FOCAS penalties, these authors actually improve their ranking position on an average of 35 places. Jan Van den Bussche has the highest ranking improvement, jumping 155 positions, while Ian Ruthven is the only author whose rank position decreases, going down 144 positions. We should point out that the variations on ranking positions after applying FOCAS are more volatile for authors ranked lower because the difference between their scores and authors at the same level are smaller. More concretely, a smaller variation of the score is required to change ranking position.

3.2.4.4 So, authors shouldn't collaborate?

FOCAS only penalises self-citations and co-author citations. As a result, one might be mislead into thinking that methods like FOCAS encourage authors to avoid collaborations since having co-authors makes you closer to everyone else in the community, and thus it will result in higher penalties for your citations. Part of this assumption is correct since FOCAS is less likely to penalise authors with less co-authors. However, it does not necessarily mean that these authors are going to obtain higher rankings. There are studies that have shown that collaboration is key to achieve career success

in research [113]. As a result, an author that does not collaborate is more likely to obtain fewer citations which would result in a lower author ranking compared to having collaborations and having some of his citations penalised. FOCAS does not aim to discourage collaborations, instead it aims to identify and mitigate the abuse of friendly citations patterns on author ranking.

3.2.5 Summary

Despite the problem of citation boosting leading to undeserved scientific impact being a pertinent theme in several studies, we found that ALAR methods did not address this problem. In this section, we present FOCAS, a method that penalises friendly citations based on (i) authors' distance, (ii) citation frequency, and (iii) citation recency. FOCAS is a flexible algorithm that allows integration with existing ALAR methods. Thus, providing the community with a simple tool to decrease the impact of some citation boosting patterns.

We assessed if FOCAS improved author ranking methods on a citation and co-authorship network comprised of seven Information Retrieval top-conferences. In our experiments, we verified that FOCAS improved state-of-the-art author ranking methods by 25% on average and 46% at best. The most important criteria to improve rankings was the distance between authors, highlighting the importance of graph-based methods since traditional author ranking methods can not capture this information. Our experiments also suggested that the frequency of the citations seems to be more important than the recency of the citations. Another relevant result obtained in our study was that the traditional approach of removing self-citations has minimal gains ($\approx$ 1% on average, and 11% at most for one of the tested methods). This latter result highlights why current state-of-the-art is lacking and the importance of our approach.

Chapter 4

Publications clustering

Due to the large number of scientific research documents available, automatic tools to analyse large corpus of documents are growing in importance. Consequently, the task of clustering of scientific publications has been receiving a significant amount of attention. This task consists of grouping publications into clusters of related or similar publications. The general idea is that publications belonging to the same cluster should be strongly related to each other, while publications belonging to different clusters should be weakly related. Being able to accurately group publications into clusters has important applications such as the classification of science (classifying publications into research areas or scientific fields), information retrieval (obtaining publications similar to a given one), research front detection (identifying research areas and their sub-areas) and topic evolution (detecting which topics are emerging or declining). These applications are particularly important for researchers, publishers, funding agencies and universities since they help to provide an informed overview of science, to get a better understand of how scientific fields are organised and to keep track of the important scientific developments [37, 94, 114].

Clustering scientific publications is essentially divided in two independent tasks. First, a similarity estimator is utilised to estimate the similarities between all publications (i.e., how similar or related the pairs of publications are). Then, a clustering or a community detection algorithm is utilised to construct the clusters based on the publications similarities. In this thesis, we focus on the task of estimating publications similarities. In the context of clustering of scientific publications, the term similarity or relatedness refers to a value that indicates whether two publications address the

same research topics. The higher the value, the more likely it is that two publications address the same topic.

In this chapter we present PURE-SIM (PUblication Relatedness Estimator using Star-schema Information networks of Metadata) an approach to estimate the similarities of publications. PURE-SIM uses a Heterogeneous Information Network (HIN) to model the relations between publications and their metadata. This network is then used to estimate publications similarities. The general idea is that the higher the amount of shared metadata elements between two publications, the higher is the similarity between them.

We compare PURE-SIM against other approaches from the literature in the task of clustering of publications. There are two major goals of our experiments. First, we aim to show that considering metadata such as authors, journals and keywords is necessary to deal with the problem of incomplete information in bibliographic databases (i.e., some publications do not have textual or reference information). Second, we aim to show that considering additional metadata sources from the publications leads to more accurate similarities which are capable of producing better clusters.

4.1 Motivation

The literature for the topic of estimating the similarity of publications is mostly dominated by either text-based or citation-based approaches [85]. The former uses textual evidence (e.g., counting how many terms two publications have in common) to estimate the similarity of publications while the latter considers citation relations between publications (e.g., counting how many times two publications are cited together). More recently, some studies have combined both approaches to create hybrid strategies [37, 94]. One drawback of the current approaches is that they are unable to cope with missing information in bibliographic databases. For example, consider a bibliographic database where 50% of the publications do not have information on their references. In this case, a citation-based approach would not be able to estimate the similarity for a significant portion of the publications in the database since there are no citation relations to analyse. In the same manner, in cases where textual data is missing (i.e., the abstract or full-text of the publication) it is not possible to use text-based approaches. Hybrid approaches combine textual evidence and citation evidence to compute the similarity of publications. It is thus possible to estimate the similarity when either textual data or citation data is missing. However, hybrid

approaches still offer no solution for cases where both textual data and citation data are missing. An alternative to this drawback is to consider additional information to estimate publications similarities.

Bibliographic databases contain metadata beyond abstracts and references. For example, information is often available on the authors of a publication, their affiliations, the journal in which a publication has been published, and the keywords provided by the authors. These metadata elements provide other sources of evidence to compute similarities of publications (e.g., publications that are published in the same journals and by the same authors are expected to have some similarity) which can be used along with textual evidence and citation evidence in order to (i) improve the similarities estimated and (ii) provide sources of information for cases where textual and/or citation data is not available. In the literature, only a few works have explored the use of different combinations of metadata elements to detect clusters of publications [49, 50]. However, these studies disregard citation data. Hence, the use of textual data and citation data along with information on other metadata elements such as authors, journals and keywords to estimate the similarity of publications still remains an unexplored area.

4.2 Overview of our contribution

Here we present PURE-SIM, a novel algorithm to estimate publications similarities. PURE-SIM uses a network to model the relations between metadata and publications, and estimate publications similarities based on the principle that the more similar two publications are, the more metadata attributes they share. In contrast to other approaches presented in the literature. PURE-SIM is a flexible approach that allows users to define different combinations of metadata elements to estimate publications similarities. Consequently, PURE-SIM helps users to overcome the problem of missing information in bibliographic database. For instance, consider the previously mentioned case where 50% of the publications in the database have no information on their references. In that case, PURE-SIM is still capable of estimating similarities for all publications by not only relying on the incomplete information on the references, but to combine this information with the information on other metadata elements like authors or journals. The flexibility of PURE-SIM in the metadata elements that can be combined to estimate the similarities of publications also offers another advantage. Evaluating the quality of clustering solutions is not trivial [46] and it is often necessary to visually compare different solutions in order to identify the approach that is the most suited for the problem. This can be easily achieved using PURE-

SIM by defining multiple similarity measures that are based on different metadata combinations. PURE-SIM also contains two other additional user-defined parameters that allow the user to control the computational cost of the process and the weighting scheme of the network.

We compare PURE-SIM against other publication similarity estimation approaches in a fair evaluation scenario with almost 3 million publications from the PubMed database. Each publication in the PubMed database has the Medical Subject Headings (MeSH) which is a controlled vocabulary thesaurus used for indexing articles. Following a related study [37], we use a MeSH embedding scheme and the cosine distance measure to estimate the similarities for the publications in the dataset. These similarities are then used as an independent evaluation criterion to measure the accuracy of the clusters produced by a cluster algorithm when fed with the similarities of a publication similarity estimation approach. In general, a publication similarity estimation approach is better if its similarities lead to clusters that, on total, place more similar pairs of publications (according to the similarities used as evaluation criterion) in the same cluster. We first compare PURE-SIM variants (that only change from each other in the user-defined parameters) in order to determine the best parameters and to rank the importance of different metadata elements. Then, we compare the best PURE-SIM variant against a total of 11 text-based, citation-based and hybrid approaches from the literature. Our results show that PURE-SIM similarities lead to the best clusters according to the ground-truth defined. PURE-SIM obtained the best results while using the metadata combination of journals, keywords, bibliographic coupling relations and direct citations. This variant of PURE-SIM presented an average improvement of 36% compared to other 11 textual, citation and hybrid approaches and an improvement of 3.3% compared to the best approach which was based on extended direct citations. An important point to highlight is that we had to exclude 1,250,644 publications from our experiment because they did not have textual or citation information. Thus, PURE-SIM competitors were not capable of computing the similarities of these publications. The excluded publications contain a link to a journal and for most of them authors and keywords information is also available. This means that the best PURE-SIM variant not only outperforms the best approaches from the literature but would also is able to estimate the similarities of these publications thus solving an important problem in the publications clustering task.

4.3 Problem formalisation

We formalise the publications clustering task as partitioning a set of publications $\mathcal{P}$ into a set of clusters $\mathcal{C} = \{c_1, c_2, ..., c_k\}$ such that a quality function Λ is maximised. Λ is defined by the clustering algorithm[1] which produces the clusters by receiving as input a similarity matrix S with dimensions $|\mathcal{P}| \times |\mathcal{P}|$, where $|\mathcal{P}|$ is the number of publications in set $\mathcal{P}$, and s_{ij} represents the similarity between publications i and j. In this thesis, we address the problem of estimating similarities between publications to create the similarity matrix S and we present an algorithm named PURE-SIM for this task.

PURE-SIM is divided in two steps: HIN construction and publication similarity estimation. In the first step, PURE-SIM constructs a HIN by analysing the relations between different elements in the bibliographic database (e.g., publications, authors and keywords). PURE-SIM uses the star-schema and defines the node type of publications as the star type and all the metadata node types (e.g., keywords, authors and so on) as the attribute type. In these conditions, two publications are never directly connected to each other by an edge in the PURE-SIM HIN. Instead, publications are only linked the attribute-nodes that represent the metadata elements in the bibliographic database. In the second step, PURE-SIM uses a stochastic process to walk through the HIN and estimate the similarity between publications. The main idea is that the higher the number of metadata elements (represented as attribute-nodes) that connect two publications (represented as star-nodes), the higher the evidence that the publications are related to each other and the higher the estimated similarity.

In this thesis, we use community detection algorithms to cluster publications. This requires converting the similarity matrix into a similarity network in order to being able to identify the clusters. In general, the complete publications clustering process is illustrated in Figure 4.1. First, we use an approach to estimate publications similarities (e.g., PURE-SIM) and produce a similarity matrix. Second, we convert the similarity matrix into a similarity network. Finally, we use a community detection algorithm to obtain the clusters of publications.

[1] Modularity and the Constant Potts Model described in Section 2.1.2.2 are some examples of Λ functions available

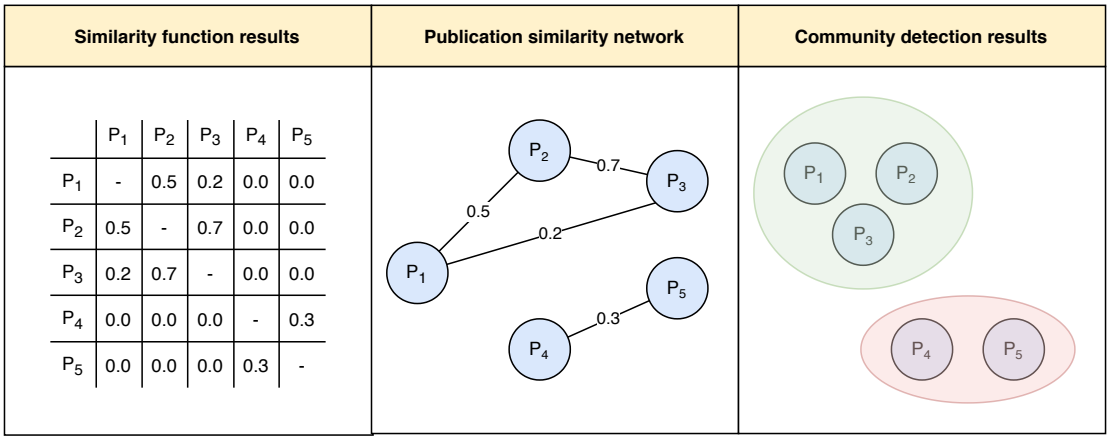


Figure 4.1: Different steps of the problem description in the publications clustering problem.

4.4 Methodology

We now describe PURE-SIM in more detail. PURE-SIM is a publication similarity estimator that aims to take advantage of the metadata relations in a bibliographic database to produce accurate similarity measures between publications. PURE-SIM is a flexible algorithm that estimates different similarities depending on a selected set of metadata elements, a weighting scheme and a variable that controls the computational cost of the process. We study the use of the similarities produced by PURE-SIM to create a publication-to-publication similarity network which is given as input to a clustering algorithm in order to identify clusters of related publications. In contrast to the approaches discussed in Section 2.3, PURE-SIM is capable of measuring the similarity between publications even in cases where data is incomplete in a bibliographic database (e.g., if for some publications an abstract or information on cited references is missing). Furthermore, by a simple change of its parameters, PURE-SIM is capable of providing the user with multiple perspectives on how the publications in a bibliographic database are related to each other and how they can be partitioned in clusters (e.g., it is possible to obtain similarities between publications that are identified based on authorship information only, citation information only, or a combination).

We present PURE-SIM in the following sections. First, we describe the process of constructing a star-schema HIN using the metadata elements that are typically available in a bibliographic database. Then, we detail how PURE-SIM utilises the HIN to estimate similarities between publications.

4.4.1 Constructing the HIN

Bibliographic databases have a rich structure where publications, $p_i \in \mathcal{P}$ are linked to metadata elements, $\mathcal{M}_{p_i} = \{m_1, m_2, ..., m_l\}$. Furthermore, each metadata element is associated with a type that identifies its source. Consider for instance the example in Table 4.1. In this example, publication p_1 is linked to 7 metadata elements, $\mathcal{M}_{p_1} = \{a_1, a_2, j_1, k_1, k_2, p_{10}, p_{11}\}$, and these metadata elements have 4 different types: 2 authors, 1 journal, 2 keywords and 2 references.

The most common publication metadata types that are found in bibliographic databases are: authors, affiliated organisations, keywords, journals and references. However, not necessarily all the publications have the same amount of information. It could be the case that for some publications information about keywords is available while it is not for others. This is for instance the case for publications p_2 and p_3 in the example in Table 4.1. When using PURE-SIM the user must define the set of metadata types, $\mathcal{M}$, that is used to estimate the similarity between publications (e.g., $\mathcal{M}$ = authors + keywords). The choice of $\mathcal{M}$ limits the amount of data used by PURE-SIM to construct the HIN and, consequently, it affects the computational cost of the process and the obtained publication similarities. We now describe the process of constructing a star-schema HIN used by PURE-SIM. We iterate over all the publications in a bibliographic database and we add each publication as a star-node in the HIN. Additionally, we add the metadata of each publication as attribute-nodes if the metadata type is selected by the user. For instance, if the user selects $\mathcal{M}$ = authors, then in the example of Table 4.1 only the authors are added as attribute-nodes in the HIN while the rest of the data is not considered. Note that metadata elements that are shared by multiple publications are represented by a single node. For example, in Table 4.1 author a_1 has authored both publications p_1 and p_2 and will therefore be represented by a single node with 2 edges in the HIN.

Table 4.1: A small example of the metadata information that is often found in bibliographic databases. Metadata elements that are shared by multiple publications are presented in bold.

Publication	Authors	Journal	Keywords	References
p_1	$\boldsymbol{a_1}; a_2$	j_1	$k_1; k_2$	$p_{10}; p_{11}$
p_2	$\boldsymbol{a_1}; a_4$	$\boldsymbol{j_2}$	-	$\boldsymbol{p_1}; p_{13}$
p_3	$a_5; a_6; a_7$	j_3	-	$\boldsymbol{p_1}; p_{14}; p_{15}$
p_4	a_8	j_4	$\boldsymbol{k_3}; k_4$	$\boldsymbol{p_1}; p_{16}; p_{17}$
p_5	$a_9; a_{10}$	$\boldsymbol{j_2}$	$\boldsymbol{k_3}; k_5; k_6; k_7$	-

In this thesis we cover the use of 5 different metadata types: authors, journals, keywords, direct citations (publication i cites publication j) and bibliographic coupling relations (publication i and j cite publication k). Authors, journals and keywords are added to the network by adding the metadata directly as attribute-nodes in the HIN (e.g., consider the previous example with author a_1). The reference-based metadata (i.e., direct citations and bibliographic coupling relations) is treated differently. This type of metadata connects publications to other publications. Implementing such connections directly in the HIN results in star-to-star edges. This type of edge is not allowed in the star-schema model. Therefore, in the case of reference-based metadata, we create fictional attribute-nodes that act as bridges between publications. In this way we maintain the star-schema property of the network. For direct citations we add a fictional attribute-node that represents the relation between the citing publication and the cited publication (e.g., a fictional node connecting the nodes of publications p_1 and p_3 will be used to represent the direct citation between these publications in Table 4.1). Note that a fictional attribute-node representing a direct citation has always two edges in the HIN. One edge to the star-node of the citing publication and another edge to the star-node of the cited publication. For bibliographic coupling relations we create a fictional attribute-node that represents the action of citing (or referencing) a certain publication p_i. Edges are created between the fictional attribute-node and the star-nodes of the publications that cite p_i (e.g, in Table 4.1, the nodes of publications p_2, p_3 and p_4 will be linked to a fictional attribute-node because they all cite p_1).

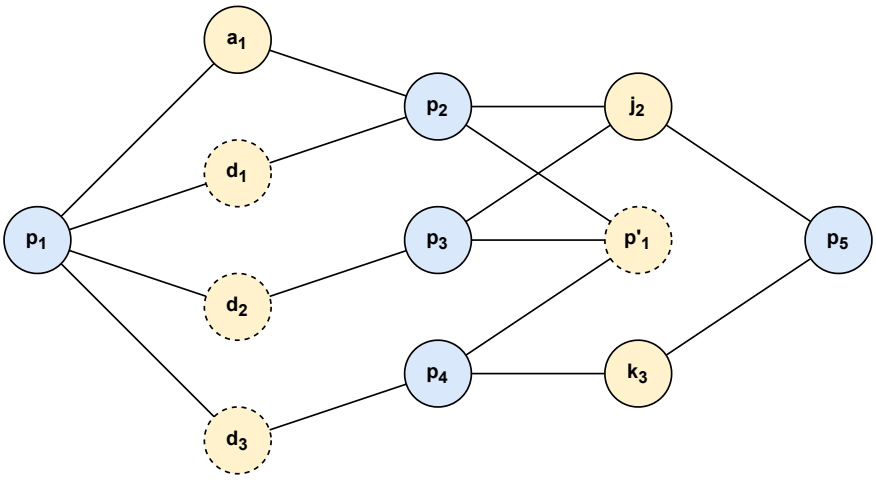


Figure 4.2: Star-schema HIN of the data presented in Table 4.1 using metadata types $\mathcal{M}$ = authors+journals+keywords+direct citations+bibliographic coupling relations. Blue circles represent star-nodes while yellow ones represent attribute-nodes. Furthermore, circles with a dashed border represent fictional attribute-nodes while the others represent metadata elements that are directly extracted from the dataset.

Figure 4.2 illustrates the nodes and edges created in the star-schema HIN using $\mathcal{M}$ = authors+journals+keywords+direct citations+bibliographic coupling relations and Table 4.1 as the dataset. In Figure 4.2, nodes d_1, d_2 and d_3 represent the direct citations between the publications in the dataset and node p'_1 represents the action of citing (referencing) publication p_1 (i.e., bibliographic coupling relations). Note that metadata elements that are connected to a single publication are not added to the HIN since they do not provide any valuable information for estimating the similarity between publications.

After defining the nodes and the edges in the HIN, the next step of PURE-SIM is to assign a weight to each edge. The idea of this step is to distinguish more important information from less important information in the process of estimating the similarity between publications. We define two weighting schemes for computing the weights of the edges in the HIN: publication normalisation and metadata normalisation. The two edge weighting schemes capture different properties of the dataset. Publication normalisation assigns higher weights to the edges of a publication node in cases where the publication node is linked to fewer attribute-nodes. For example, in Figure 4.2 the edge between nodes j_2 and p_5 has a higher weight than the edge between nodes j_2 and p_4 because p_5 is linked to two attribute-nodes while p_4 is linked to four attribute-nodes. Metadata normalisation assigns higher weights to the edges of an attribute-node in cases where the attribute-node is linked to fewer publication nodes. For example, in Figure 4.2 the edge between nodes k_3 and p_5 has a higher weight than the edge between nodes j_2 and p_5 because k_3 is linked to two publications while j_2 is linked to three publications.

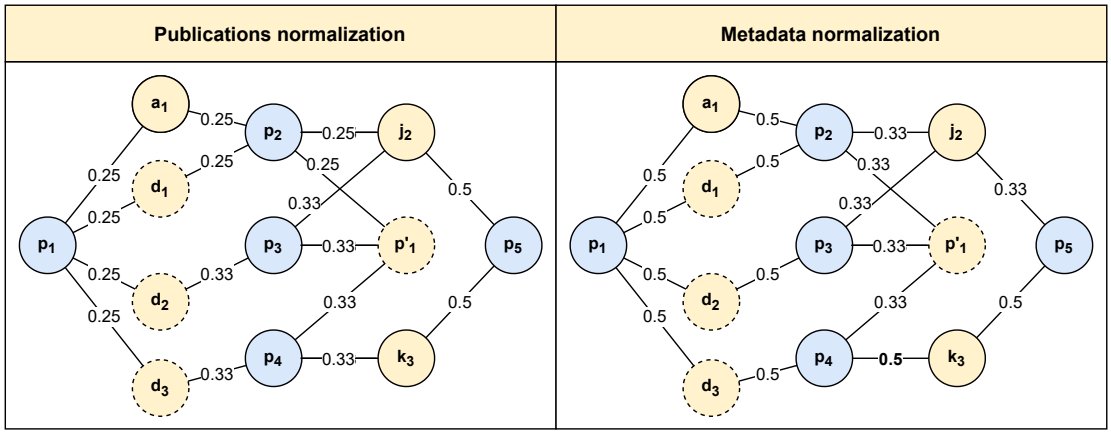


Figure 4.3: The different results obtained by using the publication normalisation and metadata normalisation weight strategies based on the data represented in Table 4.1.

In the case of the publication normalisation weighting scheme, the weight of the edge w_{im} that connects a publication i to a metadata attribute m is defined using the following equation:

$$w_{im} = \frac{1}{|\varepsilon(i)|} \tag{4.1}$$

where $|\varepsilon(i)|$ is the number of different metadata attributes connected to i. In the case of the metadata normalisation weighting scheme, the weight of the edge w_{im} is defined as follows:

$$w_{im} = \frac{1}{|\varepsilon(m)|} \tag{4.2}$$

where $|\varepsilon(m)|$ is the number of different publications connected to m. Figure 4.3 illustrates the difference between the two edge weighting schemes for the dataset represented in Table 4.1.

4.4.2 Estimating publication relatedness

PURE-SIM estimates the similarity between publications based on the principle that similar publications often share some metadata. PURE-SIM goal is to assign higher similarities to pairs of publications where (1) the publications have more metadata attributes in common and (2) the publications share more exclusive edges to the metadata attributes. To illustrate this, consider Figure 5.2. In this scenario and in line with the first goal, p_1 is more similar to p_2 than to p_3 because p_1 shares two metadata attributes with p_2 (nodes a_1 and d_1) and it shares only one metadata attributes with p_3 (node d_2). Furthermore, in line with our second goal, although p_2 and p_4 both share only one metadata attribute with p_5 (k_3 and j_2, respectively), p_5 is more similar to p_4 than to p_2 because the metadata relation $p_4 - k_3 - p_5$ is more exclusive than the metadata relation $p_2 - j_2 - p_5$. This is the case since p_4 is linked to only three metadata attributes while p_2 is linked to four metadata attributes and, in addition, k_3 is shared by only two publications and j_2 is shared by three publications. Note that the exclusiveness of relations is estimated in the process of constructing the HIN and assigning weights to the edges

We now describe the process of using the HIN to estimate the similarity between publications. For each publication i, we estimate its similarity with other publications

using walks of length two in the HIN. Due to the star-schema property of the HIN, a walk of length two that starts at a publication (i.e., a star-node) is guaranteed to end at another publication. The similarity of publications i and j is estimated using the following equation:

$$s_{ij} = \frac{\theta(i,j)}{\Theta(i)} \tag{4.3}$$

where $\theta(i,j)$ represents the sum of the weights of all the paths of length two that start at publication i and end at publication j. More concretely, $\theta(i,j)$ is estimated using the following equation:

$$\theta(i,j) = \sum_{m \in \mathcal{Z}} w_{im} \times w_{mj} \tag{4.4}$$

where $\mathcal{Z}$ is the set of metadata attributes shared by publications i and j, and w_{im} and w_{mj} are the weights associated with edges e_{im} and e_{mj} in the HIN. $\Theta(i)$ represents the sum of all $\theta(i,k)$ that start at publication i and end at any other publication k. Table 4.2 illustrates the process of calculating the similarities for publication p_2 using the metadata normalised HIN from Figure 4.3.

When considering all the possible paths of length two between publications, the number of pairs of publications that will have a similarity value greater than zero drastically increases with the number of metadata attributes that exist in the HIN.

Table 4.2: Illustration of calculating similarities between publication p_2 and all other publications in the dataset presented in Table 4.1 and using the HIN from Figure 4.3 that makes use of the metadata normalisation weighting scheme. Note that $\Theta(p_2) = 0.94$ corresponds to the sum of the multiplication of the weights of all paths of length two starting at publication p_2 and ending at another publication.

Similarity	Formula	Computation
$s_{p_2p_1}$	$\frac{w_{p_2a_1} \times w_{a_1p_1} + w_{p_2d_1} \times w_{d_1p_1}}{\Theta(p_2)}$	$\frac{0.5 \times 0.5 + 0.5 \times 0.5}{0.94} = 0.53$
$s_{p_2p_3}$	$\frac{w_{p_2j_2} \times w_{j_2p_3} + w_{p_2b_1} \times w_{b_1p_3}}{\Theta(p_2)}$	$\frac{0.33 \times 0.33 + 0.33 \times 0.33}{0.94} = 0.23$
$s_{p_2p_4}$	$\frac{w_{p_2b_1} \times w_{b_1p_4}}{\Theta(p_2)}$	$\frac{0.33 \times 0.33}{0.94} = 0.12$
$s_{p_2p_5}$	$\frac{w_{p_2j_2} \times w_{j_2p_5}}{\Theta(p_2)}$	$\frac{0.33 \times 0.33}{0.94} = 0.12$

For example, consider a case where a single publication has been cited by 1000 other publications. This bibliographic coupling metadata relation, by itself, produces a similarity value greater than zero for 499,500 pairs of publications. Thus, analysing all the possible paths between publications of length two results in a high number of publication pairs with a similarity value greater than zero. Furthermore, since the similarity values are used as edge weights in an undirected network that is provided as input to a clustering algorithm, it produces highly dense networks of publications which increase the computational cost of detecting clusters of similar publications. In order to control the number of publication pairs for which the similarity is estimated by PURE-SIM, and consequentially the computational cost of estimating publication similarities and the detection of clusters of similar publications, we added a stochastic process based on random walks to the algorithm. The idea consists of focusing on the most important metadata relations of a publication while calculating its similarity with other publications. As a result, only the stronger similarities are identified. For every publication i, instead of considering all the possible paths of length two, we perform N random walks. Each random walk starts at publication i and has length two. At each step of the random walk, an edge is randomly selected based on the weights of the edges (i.e., an edge with weight 0.5 is twice more likely to be selected than an edge with weight 0.25). After performing the N random walks, the similarity between publications i and j is estimated using the following equation:

$$s_{ij} = \begin{cases} \frac{\theta(i,j)}{\Theta(i)} & \text{if N} = 0 \\ \frac{\tau(i,j)}{N} & \text{if N} > 0 \end{cases} \tag{4.5}$$

where $\tau(i,j)$ is the number of times a random walk ended in publication j. The value of N is user defined. When $N = 0$, all the paths of size two are considered and the similarity of publications is estimated using Equation 4.3.

An important point to highlight is that, using any of the previous equations, the similarities between publications are normalised so that the total similarity of a publication with other publications equals 1. However, the resulting similarities are not symmetrical (i.e., s_{ij} is not necessarily equal to s_{ji}). In order to be able to use the similarities as edge weights in an undirected network that is provided as input to a clustering algorithm, we make the similarities symmetrical by applying the following equation to every publication pair:

$$s'_{ij} = s_{ij} + s_{ji} \tag{4.6}$$

4.5 Experimental setup

PURE-SIM is a flexible algorithm that allows the user to specify different parameters. It is for instance possible to specify which metadata elements should be taken into account in the HIN, the edge weighting scheme of the HIN and the number of random walks in order to limit the number of publication pairs for which a similarity is calculated. In this way it is easy to obtain different publication similarity results as well as to comply with database restrictions related to lack of information (e.g., several methods discussed in Section 2.3 could not be used if information on references is missing). The flexibility of PURE-SIM also raises several questions when one wants to use it in practice. Given that bibliographic databases often have different types of metadata available, which is the best combination of metadata elements to use for calculating publication similarities? Furthermore, how does limiting the number of publication pairs for which a similarity is calculated affects the overall performance of the clustering algorithm? Finally, which of the two weighting schemes for assigning weights to the edges in the HIN yields the best results? In this section we address these questions by performing a series of experiments. In this way we aim to provide a comprehensive and insightful view of the different parameter settings of PURE-SIM. Additionally, after establishing the best parameter settings for PURE-SIM, we compare its similarities against the similarities produced by other state-of-the-art similarity measures taken from the literature. We do this in an experiment that aims to determine which similarities lead to clusters that are best in line with the similarities that are defined partly based on human judgement.

4.5.1 Dataset description

In order to have a comprehensive dataset for our experiments, we used the same dataset that was used in a previous study about publication similarity estimators [37]. We replicated this dataset by using the PubMed database and extracting all the MEDLINE publications (the biggest subset of publications in PubMed) with a print year in the period of 2013–2017. This yielded a total of 4,191,763 publications. Since PubMed does not contain information about the citation relations between publications we also used data from the Web of Science database. By matching the publications from the PubMed database in our dataset with the publications included in an in-house version of Web of Science database that is available at the Centre for Science and Technology Studies at Leiden University we were able to obtain the citation relations for the

publications in our dataset. At this point we decided to keep only publications that (1) have a title and an abstract and (2) cite or are cited by at least one other publication in the dataset. While PURE-SIM is flexible enough to comply with missing information for some publications, the state-of-the-art approaches which we aim to compare our results to are not. As discussed in Section 2.3, some approaches rely exclusively on textual or citation data. By introducing restrictions (1) and (2) we guarantee that all approaches in our experiments are capable of estimating similarities for all the publications in the dataset and in this way we ensure that none of the approaches is at a disadvantage when it is compared to one of the other approaches.

In total our dataset contains 2,941,119 publications, each one representing a star-node in the HIN. In order to gather more data for the HIN construction process, for every publication in the dataset we obtained the following metadata: authors, journal, keywords (assigned by the authors to their publications) and references. We added the authors, journals and keywords as nodes in the HIN. In total, we have 2,083,579 author nodes, 4,386 journal nodes and 618,998 keyword nodes. Note that these numbers represent the number of unique authors, journals and keywords that exist in the dataset. We used the references to determine the direct citation and bibliographic coupling relations in the dataset. In the HIN, we have 16,805,179 direct citation nodes which represent the number of citations between publications in the dataset and 9,949,751 bibliographic coupling nodes which represent the number of times two or more publications cite (or reference) the same publication. With respect to the edges of the HIN, we have 18,591,037 author-publication edges, 2,941,119 journal-publication edges, 10,958,112 keyword-publication edges, 33,610,358 direct citation-publication edges and 111,047,143 bibliographic coupling-publication edges. For the bibliographic coupling relations we also considered citations to publications that are outside of the dataset. For example, if two publications i and j from 2015 (part of our dataset) cite a publication k from 2010 (not part of our dataset), there still is a bibliographic coupling relation between i and j despite k not being part of the set of publications considered.

Due to the large number of experiments we have to perform in order to establish the best parameter settings for PURE-SIM and to obtain the results in a viable amount of time, we divided the dataset in two smaller subsets: D_{10} and D_{20}. The two smaller subsets represent a sampling of 10% and 20% of the complete dataset, respectively. Table 4.3 details the number of nodes and edges for all the datasets used in the experiments.

Table 4.3: Describing the datasets used throughout the experiments section. DT10 and DT20 correspond to a 10% and 20% sampling of the full dataset respectively. Note that the number of nodes and edges are in millions (M). A - authors, K - keywords, J - journals, BC - bibliographic coupling and DC - direct citations.

	Number of nodes (1×10^6)						Number of edges (1×10^6)				
	Pubs.	**Authors**	**Keywords**	**Journals**	**BC**	**DC**	**A - P**	**K - P**	**J - P**	**BC - P**	**DC - P**
D_{10}	0.294	0.290	0.101	0.001	2.165	0.169	1.852	1.096	0.294	8.390	0.338
D_{20}	0.589	0.595	0.177	0.004	3.954	0.676	3.752	2.193	0.589	19.327	1.353
Full	2.941	2.084	0.619	0.004	9.950	16.805	18.591	10.958	2.941	111.047	33.617

4.5.2 Evaluation scenario

The outcome of publication similarity estimators such as PURE-SIM is a similarity matrix S that contains the similarity for pairs of publications. This information can be used in different applications. In this thesis, we focus on the publications clustering task and we use the similarity between pairs of publications to create a publication network which is then used to detect clusters of publications. As a result, we evaluate all the publication similarity measures studied in this section with respect to the quality of the clustering solutions that they produce.

We follow the evaluation framework for comparing similarity measures for clustering publications originally proposed in [46] and later also applied in [37]. This evaluation framework can be used to evaluate the accuracy of clustering solutions obtained using different similarity approaches, where an independent similarity measure is used as the evaluation criterion. The similarity measure that we use as the independent evaluation criterion is the same one that was also used in [37]. This similarity measure is calculated based on the Medical Subject Headings (MeSH) of MEDLINE publications which are a set of terms that are assigned by humans to categorise biomedical and health-related literature. A publication embedding scheme is defined based on the MeSH terms and the cosine similarity is used to estimate the similarity of publications. More details about the MeSH similarity measure that we use as the evaluation criteria are presented in Section A.1. The accuracy of a clustering solution is then calculated using the following equation:

$$A^X = \frac{1}{|P|} \sum_{i,j} I(C_i^X, C_j^X) s_{i,j}^{\text{MeSH}} \tag{4.7}$$

where,

$$I(C_i^X, C_j^X) = \begin{cases} 1, & \text{if } C_i^X = C_j^X \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

where i and j are two publications from the dataset, $|\mathcal{P}|$ is the total number of publications in the dataset, C_i^X and C_j^X denote the cluster to which publications i and j belong in clustering solution X, and s_{ij}^{MeSH} is the similarity between publications i and j that is obtained using the independent MeSH similarity measure.

We measure the accuracy of a clustering solution at different levels of granularity. For each publication similarity approach evaluated, we use the Leiden algorithm [62] with 11 different resolution parameters $\gamma = (0.000001, 0.000002, 0.000005, 0.00001, 0.00002, 0.00005, 0.0001, 0.0002, 0.0005, 0.001, 0.002)$ to produce 11 clustering solutions[2].

The accuracy of the different publication similarity approaches is then compared using Granularity-Accuracy (GA) plots [46] (which were introduced in more detail in Section 2.3.4). In a GA plot, the horizontal axis represents the granularity of a clustering solution and the vertical axis the accuracy which in this case is defined by Equation 4.7. With respect to granularity, as the granularity increases so does the number of clusters identified. In terms of research topics, lower values of granularity indicate more broad topics (i.e., clusters containing a larger number of publications on average) while higher values represent more specific topics (i.e., clusters containing a smaller number of publications on average). For each publication similarity approach, the GA line is drawn based on the accuracy and granularity of its 11 clustering solutions. A publication similarity approach whose GA line is consistently higher than the GA line of another one represents the more accurate approach.

To further complement the comparison of different publication similarity approaches with a numerical value, the Piecewise Cubic Hermite Interpolation function (P) is utilised to estimate the accuracy of approaches at a specific point of granularity (thus making the value of accuracy comparable). The gain of approach S_1 over approach S_2 is estimated using the following equation:

$$Gain(S_1, S_2) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \frac{\mathrm{P}_{S_1}(l)}{\mathrm{P}_{S_2}(l)} \tag{4.9}$$

[2]We use the Leiden algorithm and these different resolution parameters since they were already used in the previous studies.

where $\mathcal{L}$ is a set of granularity points, and $\mathrm{P}_{S_1}(l)$ and $\mathrm{P}_{S_2}(l)$ are the interpolated accuracy of approaches S_1 and S_2 at granularity l, respectively. The interpretation of Equation 4.9 is that on average approach S_1 presents a gain of g over approach S_2. If $g > 1$ then approach S_1 outperforms S_2. The greater g is the better is S_1 compared to S_2. Conversely, a value between 0 and 1 indicate that S_1 performs worse than S_2. The lower the value is the worse S_1 is compared to S_2.

In this thesis, we extend the evaluation framework applied in [37, 46] by measuring the gain of a group of approaches over another group. This comparison is useful for example to measure the gain of citation-based approaches over text-based approaches. For these cases, the accuracy of the group at a certain granularity level is the average accuracy of all the approaches in the group. More concretely, the accuracy of a group of approaches $\mathcal{Y}$ at granularity l is obtained using the following equation:

$$\mathrm{P}_{\mathcal{Y}}(l) = \frac{1}{|\mathcal{Y}|} \sum_{S_i \in \mathcal{Y}} \mathrm{P}_{S_i}(l) \tag{4.10}$$

where $|\mathcal{Y}|$ is the number of approaches in the group and $\mathrm{P}_{S_i}(l)$ is the interpolated accuracy of approach S_i at granularity l. Redefining the $\mathrm{P}_{S_1}(l)$ and $\mathrm{P}_{S_2}(l)$ values with $\mathrm{P}_{Y_1}(l)$ and $\mathrm{P}_{Y_1}(l)$ in Equation 4.9, allows us to estimate the gain of group of approaches Y_1 over group Y_2.

4.5.3 Parameter tuning

In this section we test several PURE-SIM configurations in order to provide insights on the best parameter settings to use. PURE-SIM has three user-defined parameters: metadata selection (M), weighting scheme (W) and number of random walks (N). M is used to define the metadata elements that should be included in the HIN. W is used to specify how the weights of the edges in the HIN are determined (i.e., based on publication or metadata normalisation). N is used to control the number of publication pairs for which a similarity is calculated by PURE-SIM. Throughout the experiments we use the nomenclature PURE-SIM$_{(M,W,N)}$ to denote the parameters that are used by a PURE-SIM variant. For example, PURE-SIM$_{(\text{BC+DC,metadata,100})}$ represents a PURE-SIM variant that uses bibliographic coupling and direct citation as metadata elements, uses metadata normalisation as weighting scheme and 100 random walks for each publication to calculate the similarity with other publications. For readability we abbreviate the bibliographic coupling and direct citation metadata types as BC and DC, respectively. In our experiments, for each parameter evaluated,

we test all the possible metadata combinations. Given that we take into account five metadata types, we have a total of $2^5 - 1 = 31$ possible combinations. Due to space limitations we only present the results for a limited number of combinations.[3] Our selection includes the following combinations: single metadata elements (i.e., Author, Keyword, DC and BC)[4], combination of citation-based metadata elements (i.e., BC + DC), combination of non-citation-based metadata elements (i.e., Author + Journal + Keyword), combination of all metadata elements (i.e., Author + Journal + Keyword + BC + DC) and the best performing combination of metadata elements according to our experiments (i.e., Keyword + BC + DC). Furthermore, we also include in our selection the combination Author + DC since this combination requires a low computational cost.

Number of random walks (parameter N)

We start our experiments by studying the impact of the number of random walks parameter N. The lower the value of N is the lower the number of publication pairs for which a similarity is identified by PURE-SIM. Similarity estimations and clustering solutions are obtained faster using lower values of N. However, by lowering the value of N we may also lose valuable information on the relatedness of publications and obtain clustering solutions of lower quality. Therefore, it is important to find a good balance between computational cost and clustering quality. Note that $N = 0$ is used to denote that all the possible paths of length two are considered and the 0 should therefore not be interpreted as a lower value for N compared to 50 for example. In this first experiment, we tested all 31 metadata combinations with W = metadata and for each variant we tested $N = (0, 10, 20, 50, 100, 200, 300, 500, 1000)$. In order to obtain the results in a reasonable amount of time for such a large number of variants, we performed our tests using dataset D_{10}. Figure 4.4 shows the GA plots obtained for 9 out of the 31 possible metadata combinations.

With respect to the granularity, results show that the number of random walks does not have a high impact on the size of clusters. The only noticeable difference is for $N = 10$ where we observe that for the five lower resolutions of the clustering algorithm there are less clusters and for the higher resolutions there are more clusters when compared to the other values of N. These results are expected due to the large amount of information that it is lost when considering only 10 random walks

[3] A full overview of all the results is available online at `https://github.com/JSilva90/PURE-SIM`

[4] Journals are not presented due to their low accuracy results when not combined with any other metadata element.

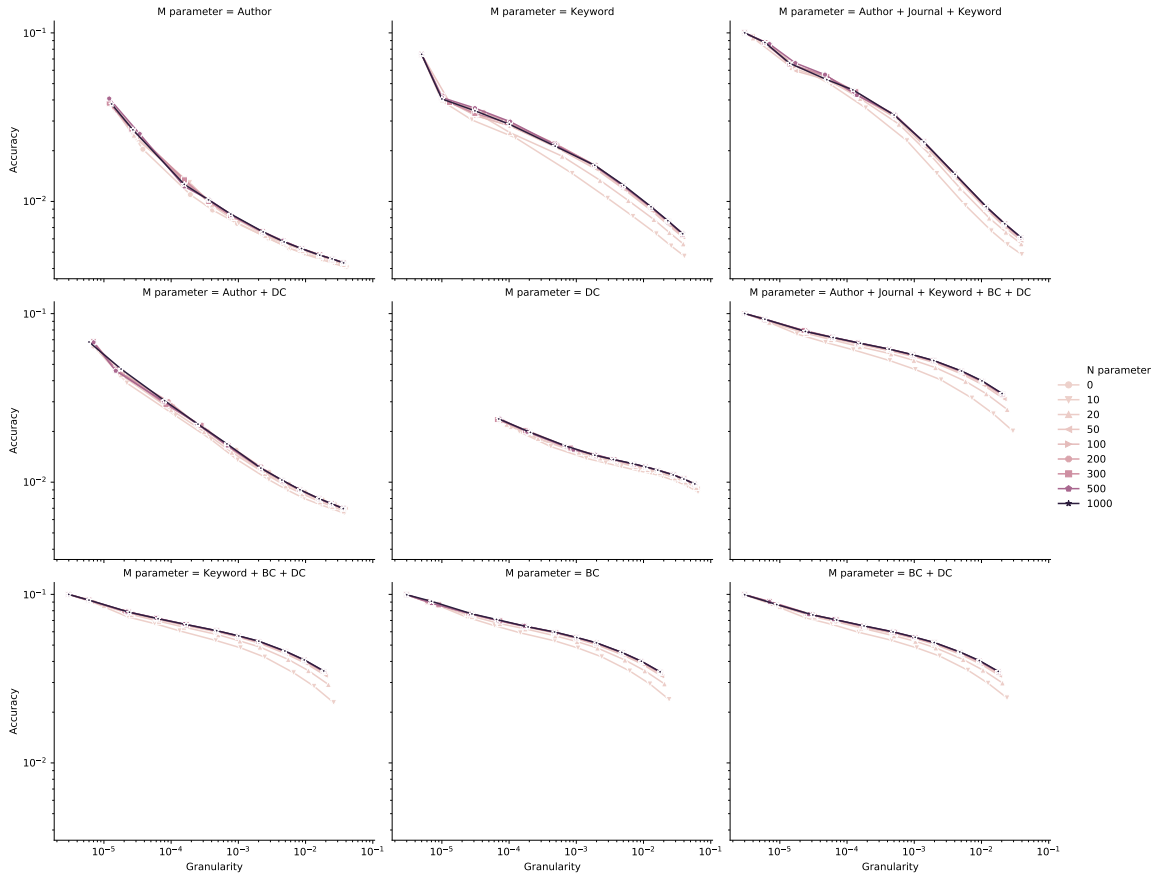


Figure 4.4: GA plots for the D_{10} dataset for variants PURE-SIM$_{(M,\text{metadata},N)}$ with 9 different values of M and $N = (10, 20, 50, 100, 200, 300, 500, 1000)$. For some cases we have $N = 0$ which represents the results of using all the possible paths of size two.

per publication. With respect to accuracy, results show that using $N = (10, 20, 50)$ significantly reduces accuracy for all metadata combinations. $N = (100, 200, 300)$ show little difference in accuracy with $N = 300$ always obtaining the best accuracy values. Furthermore, there is no benefit in adding a value of N higher than 300. All the tests with $N = (300, 500, 1000)$ always have the same accuracy, however the last two $N = (500, 1000)$ require more time to obtain the clusters. With respect to using $N = 0$, due to the computational cost of this variant we were only able to get the results for $M = (\text{Author}, \text{DC}, \text{Author} + \text{DC})$. The results show that considering all the possible paths of length two between publications results in clustering solutions with an accuracy that is similar to the accuracy obtained using $N = (300, 500, 1000)$ while it exponentially increases the computational power required. An important point to highlight is that although the number of edges of the HIN drastically changes with parameter M, the results from changing parameter N remain constant. Thus, revealing that parameter N is not sensitive to parameter M apparently.

Table 4.4: The total number of publication similarity pairs discovered by each metadata combination (parameter M) in dataset D_{10} and the respective percentage of pairs maintained while changing the number of random walks (parameter N). Metadata combinations are sorted according to the total number of similarity pairs. The last row represents the average reduction of each value of the parameter N on all the metadata combinations.

M parameter	Percentage of pairs maintained after changing parameter N								Total similarity pairs (1×10^6)
	10	20	50	100	200	300	500	1000	
DC	80.4%	91.5%	97.7%	99.0%	99.5%	99.7%	99.8%	99.9%	0.338
Author	38.5%	53.5%	73.0%	84.6%	92.5%	95.3%	97.5%	99.0%	2.288
Author + DC	36.3%	51.4%	71.1%	83.1%	91.4%	94.7%	97.2%	98.9%	2.560
Keyword	2.1%	3.8%	7.9%	13.2%	21.0%	27.0%	36.1%	50.8%	77.203
Keyword + DC	2.0%	3.5%	7.2%	11.8%	18.8%	24.2%	32.5%	46.3%	77.497
Author + Keyword	2.1%	3.7%	7.1%	11.3%	17.4%	22.1%	29.4%	41.7%	79.386
Author + Keyword + DC	2.1%	3.5%	6.8%	10.6%	16.3%	20.7%	27.5%	39.2%	79.628
BC	1.9%	3.5%	7.1%	11.6%	17.6%	21.8%	27.8%	36.9%	100.709
BC + DC	1.9%	3.4%	7.0%	11.4%	17.3%	21.5%	27.4%	36.4%	100.791
Author + BC	1.9%	3.4%	7.1%	11.5%	17.4%	21.6%	27.5%	36.5%	102.518
Author + BC + DC	1.8%	3.3%	7.0%	11.3%	17.2%	21.3%	27.2%	36.1%	102.594
Keyword + BC	1.2%	2.1%	4.6%	7.6%	12.0%	15.3%	20.1%	28.1%	174.530
Keyword + BC + DC	1.1%	2.1%	4.5%	7.5%	11.8%	15.0%	19.8%	27.7%	174.605
Author + Keyword + BC	1.1%	2.1%	4.5%	7.5%	11.8%	15.0%	19.7%	27.5%	176.289
Author + Keyword + BC + DC	1.1%	2.1%	4.4%	7.4%	11.6%	14.8%	19.5%	27.1%	176.358
Journal	1.3%	2.4%	5.3%	8.8%	13.4%	16.6%	20.9%	27.4%	210.612
Journal + DC	1.0%	1.8%	3.8%	6.3%	9.9%	12.5%	16.3%	22.1%	210.919
Author + Journal	0.9%	1.5%	3.0%	4.8%	7.6%	9.6%	12.8%	17.8%	212.772
Author + Journal + DC	0.8%	1.4%	2.6%	4.2%	6.6%	8.4%	11.2%	15.9%	213.023
Journal + Keyword	0.9%	1.7%	3.7%	6.5%	10.7%	14.1%	19.3%	28.2%	286.271
Journal + Keyword + DC	0.8%	1.4%	3.1%	5.3%	8.9%	11.8%	16.3%	24.2%	286.544
Author + Journal + Keyword	0.7%	1.3%	2.7%	4.6%	7.6%	10.0%	13.8%	20.5%	288.348
Author + Journal + Keyword + DC	0.7%	1.2%	2.5%	4.2%	6.9%	9.1%	12.6%	18.9%	288.574
Journal + BC	0.6%	1.2%	2.5%	4.2%	6.6%	8.3%	10.9%	15.3%	309.401
Journal + BC + DC	0.6%	1.2%	2.5%	4.1%	6.4%	8.2%	10.8%	15.0%	309.477
Author + Journal + BC	0.6%	1.2%	2.5%	4.1%	6.5%	8.2%	10.7%	14.9%	311.127
Author + Journal + BC + DC	0.6%	1.2%	2.5%	4.1%	6.4%	8.1%	10.6%	14.7%	311.198
Journal + Keyword + BC	0.5%	1.0%	2.2%	3.7%	6.0%	7.8%	10.4%	15.1%	381.875
Journal + Keyword + BC + DC	0.5%	1.0%	2.2%	3.7%	5.9%	7.6%	10.3%	14.9%	381.945
Author + Journal + Keyword + BC	0.5%	1.0%	2.2%	3.7%	5.9%	7.6%	10.2%	14.7%	383.560
Author + Journal + Keyword + BC + DC	0.5%	1.0%	2.1%	3.6%	5.8%	7.5%	10.0%	14.5%	383.626
Average % of pairs maintained	*6.0%*	*8.2%*	*11.7%*	*15.0%*	*19.2%*	*22.1%*	*26.3%*	*33.1%*	

Table 4.4 shows for which each metadata combination (parameter M) the resulting number of publication pairs with a similarity value when all the possible paths in the HIN are considered and the respective percentage of pairs that are maintained while changing the number of random walks (parameter N). If we look at the total number of similarity pairs identified by each single metadata combination (i.e., Author, Journal, Keyword, BC and DC), we see that DC and Author result in the lowest number of pairs (0.3 and 2.3 million, respectively) while Journal identifies most pairs (more than 210.6 million). This result is expect since journals are in general connected to a large number of publications while, for example, authors are connected to only a limited number of publications. Regarding the percentage of pairs that is maintained while changing parameter N, we observe that this value is negatively correlated with the total number of pairs that the metadata combination identifies. The higher the total number of pairs, the smaller the percentage of pairs maintained. Overall we observe that using $N = 10$ on average maintains 6.0% of the pairs while $N = 1000$ maintains

on average 33.1% of the pairs.

Considering both the GA plots and the percentage of pairs maintained, we conclude that $N = 300$ provides a good balance between computational cost and clusters quality. Therefore, in all the remaining experiments we use $N = 300$ for any PURE-SIM variant.

Weighting scheme (parameter W)

Next, we discuss the parameter W which defines the weighting scheme. Intuitively, the metadata normalisation weighting scheme is a better fit to estimate publication similarity since it seems more important to have higher weight paths between publications when they share more unique metadata attributes (in the sense that the metadata attribute is not frequent among other publications, e.g., a very specific keyword) than when publications are connected to fewer metadata attributes (e.g., two publications that have fewer authors and keywords than other publications). Nevertheless we tested PURE-SIM using $N = 300$ and again with all the possible metadata combinations M, and for each metadata combination we compare the two different weighting schemes W to investigate this claim. Similar to the previous experiment, we again use dataset D_{10} in order to obtain the results faster.

Figure 4.5 illustrates the obtained results. To ease visualisation we only show the GA plots for 9 different values of M. The results show that regarding granularity and accuracy there are no significant differences between the two weighting schemes. Still, in most cases the metadata normalisation weighting scheme provides a slightly better performance in terms of accuracy. Another important aspect to highlight is the different results obtained when changing parameters M and W. The two lines in each plot represent changing the W parameter while the different plots represent changing parameter M. M defines the relations that are considered in the HIN, while W changes the weights associated to these relations. The results show that changing M results in very different GA plots while there is no significant difference of changing parameter W. Therefore, it is clear that it is much more important to define the correct relations in the HIN than to calculate the weights of these relations using one of the two weighting schemes. In all the remaining experiments of PURE-SIM we use $W =$ metadata.

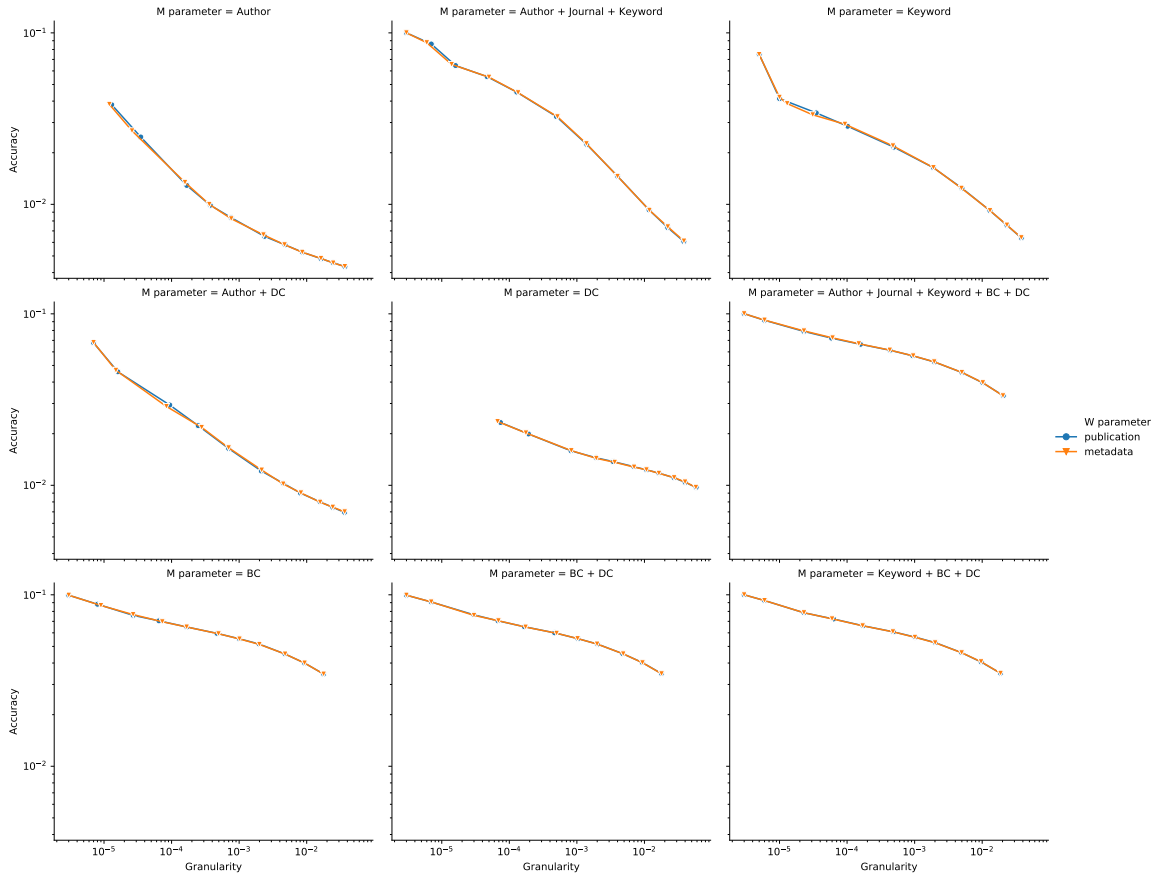


Figure 4.5: GA plots for the D_{10} dataset for variants PURE-SIM$_{(M,W,300)}$ with 9 different values of M and $W = (\text{metadata}, \text{publication})$.

Metadata combination (parameter M)

Finally, we evaluate the performance of PURE-SIM when changing parameter M with fixed parameters $W = \text{metadata}$ and $N = 300$. The selection of metadata types defines the amount of data that is available in the HIN. The more metadata types are added to the metadata selection the higher the number of edges in the HIN which subsequently results in an higher number of publication similarity pairs (see Table 4.4). Therefore, M also has a direct impact on the computational cost of estimating the similarities and obtaining a clustering solution. In this experiment we focus solely on evaluating the best metadata combination out of the 31 possible combinations when considering 5 metadata types (i.e., Author, Journal, Keyword, BC and DC). We already showed that parameter N can effectively control the computational power required by PURE-SIM and that having the correct relations in the HIN is the more important aspect of PURE-SIM. We therefore recommend that in cases where the user needs to lower the computational cost of the task it is better to lower parameter N while using the best

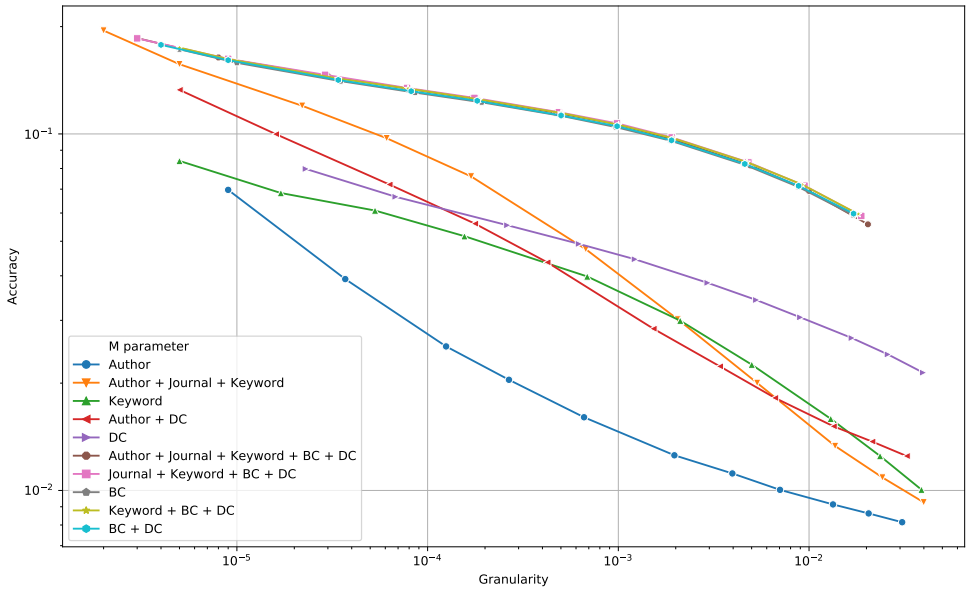


Figure 4.6: GA plot for the D_{20} dataset for variants PURE-SIM$_{(M,\text{metadata},300)}$ with 10 different values of M.

possible parameter M. For this experiment we used dataset D_{20} since there are fewer computations needed for this experiment compared to the previous ones.

Figure 4.6 illustrates a GA plot for 10 values of M. To ease visualisation we do not present the GA lines for all 31 possible values of M. We show the results for the best values of M and some other combinations that are good alternatives to handle cases where information for a certain metadata type is not available. For example, we show M = Author+Journal+Keyword that represents the best performance obtained when references are not available in the dataset (i.e., it is not possible to use bibliographic coupling and direct citations as metadata types). Furthermore we use Equations 4.9 and 4.10 to compare groups of approaches that use certain metadata types against other groups that do not. Consider for example the Author metadata type. For this metadata type, we compare all metadata combinations that include the Author metadata type (Author only or Author combined with other metadata types) against the metadata combinations that do not include the Author metadata type. We then use the indicated equations to provide insights into the gain of including the Author metadata type in the metadata selection. Figure 4.7 illustrates a heat map for the accuracy gains considering single metadata types (diagonal cells) and combinations of two metadata types (off-diagonal cells).

Figure 4.6 also shows that regarding metadata combinations that are based on a single metadata type that M = Author is the worst performing one, followed by M = Keyword, M = DC and M = BC. Again, we do not show the results for M = Journal since it results in clustering solutions with a very different granularity when compared to the others. The bibliographic coupling metadata relation, by itself, produces clusters with a similar accuracy as the best performing values for M. These results are further confirmed by Figure 4.7 which shows that adding the Author metadata type to a metadata combination results in a lower gain (0.97) than not adding this metadata type, while adding the BC metadata type results in an improvement of 2.97 times. In general, we also observe that combining metadata types together results in better performance. Most of the pairs considered in Figure 4.7 present positive gains with the exception being the ones that contain the BC metadata relation. Furthermore, we observe in the GA plot that the best metadata combination to use when there is no citation information in the dataset is M = Author + Journal + Keyword. The performance of this metadata combination is good to identify broad research areas (lower values of granularity) but its accuracy quickly decreases as research areas get more specific (higher values of granularity).

In conclusion, we observe that there are 5 combinations of M that produce the most accurate clustering solutions. The performance of these combinations is very similar. All of these combinations use the bibliographic coupling metadata type. Overall the best performance is obtained using the metadata combination M = Journal + Keyword + BC + DC.

4.5.4 Comparing PURE-SIM against other approaches

In this section we use the complete dataset obtained from PubMed in order to compare PURE-SIM$_{\text{(Journal+Keyword+BC+DC,metadata,300)}}$, the best PURE-SIM variant, and PURE-SIM$_{\text{(Author+Journal+Keyword,metadata,300)}}$, a PURE-SIM variant that does not use citation information nor textual evidence, against a set of state-of-the-art publication similarity approaches from the literature. For the sake of readability, we refer to the best PURE-SIM variant as PURE-SIM A and the variant that does not use citation information nor textual evidence as PURE-SIM B. We compare PURE-SIM against 2 text-based approaches (BM25 and TFIDF), 6 citation-based approaches (DC, Extended Direct Citations (EDC), BC, Co-Citation (CC) and the combinations DC-BC-CC and DC-BC), and 3 hybrid approaches (DC-BM25, EDC-BM25 and DC-TFIDF). The similarity of the BM25 and TFIDF approaches is obtained by estimating

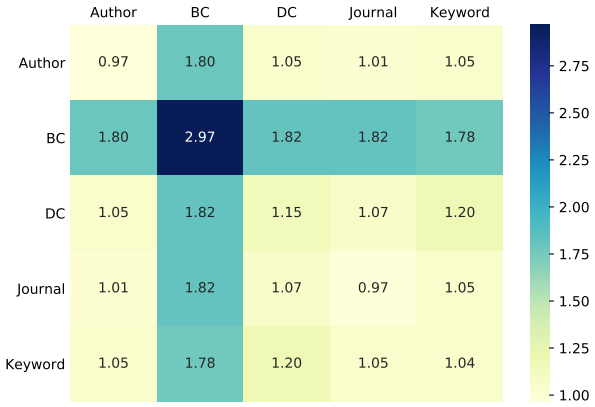


Figure 4.7: Performance gains estimated using Equations 4.9 and 4.10 when considering metadata combinations that contain certain metadata types against metadata combinations that do not contain these metadata types for dataset D_{20}. For example, the cell Author-Author represents the gain of considering all the metadata combinations that contain the Author metadata type versus the metadata combinations that do not contain the Author metadata type. Additionally, the cell Author-BC represents the gain of considering all the metadata combinations that consider both Author and BC against the ones that do not. The matrix is symmetrical.

the cosine similarity between the vectors of terms identified in the title and abstract of publications using the term weighting schemes BM25 and tf-idf, respectively. The similarity of the DC approach consists of 1 if a publication i cites publication j and 0 otherwise. EDC is an extension of the DC approach that also considers citations to publications that are not in the dataset [46]. The similarity of two publications i and j using the BC approach is the number of publications i and j cite in common, while in the case of the CC approach it is the number of publications that cited i and j. All the strategies that result from the combination of these previous approaches are obtained by summing all the individual similarities. We also analyse the clustering solution obtained using the MeSH similarity approach that was used as the independent evaluation criterion. The accuracy and granularity results of these clustering solutions are used as an indicator of the best results that are possible to obtain for the dataset [46]. The results for the state-of-the-art approaches and the MeSH approach were obtained from the study [37]. Following this previous study, for the state-of-the-art approaches we also only consider the top 20 similarities for each publication. This is necessary to decrease the number of publications with similarities greater than 0 and decrease the computational cost of estimating the clusters of publications. Furthermore, it is important to note that the relations considered by bibliographic coupling in PURE-

SIM are more similar to the ones used in the EDC approach than the ones using in the BC approach. The BC approach does not increase the similarity between publications i and j if both cite a publication k that is outside the dataset. The EDC approach, similarly to the bibliographic coupling relations used in PURE-SIM, also considers citations to publications that are not in the dataset.

Figure 4.8 shows the GA plot obtained for all the approaches on the full dataset. For the state-of-the-art approaches we observe that in general text-based approaches perform worse than citation-based ones. Overall, TFIDF is the worst performing similarity measure while CC is the second worst one. The performance of the CC approach is expected since we are considering a relatively small time window of publications (2013–2017) and as a result there are not many co-citations relations in the dataset. In fact, this is the reason why we did not include the co-citation relations as one of the possible metadata types for PURE-SIM in this thesis. With respect to the other citation-based approaches, BC outperforms DC by a significant margin. This result reinforces the results of Section 4.5.3 where we showed that the bibliographic coupling metadata relation is more important than the direct citation one. Approaches such as DC-BC and DC-BC-CC present the benefits of combining several metadata relations based on citations, with both approaches outperforming BC and DC. Still, the best citation-based measure is the EDC approach. This result indicates that the more citation information is considered to estimate publications similarity, the higher the accuracy of the clusters produced. Regarding hybrid approaches its very noticeable the benefits of combining textual evidence with citations. Methods such as BM25 and TFIDF present good improvements when combined with the DC and the BM25 method. The results of EDC-BM25 and EDC are quite similar, however, the EDC-BM25 approach is outperformed by the EDC approach for clustering solution with a larger granularity. Overall, the best result of the state-of-the-art approaches is obtained when using the EDC approach. This is in line with the study presented in [37].

With respect to the PURE-SIM method we observe two very distinct results for the two tested variants. The PURE-SIM A variant outperformed all the other approaches and obtained the highest accuracy for both low and high granularity values. The PURE-SIM B variant underperformed all the other approaches. The accuracy of this method is competitive with the accuracy of the other approaches for low granularity values. However, as the granularity increases, the accuracy of PURE-SIM B deteriorates faster than the accuracy of other approaches. We would like to highlight that low granularity values represent solutions with a small number of clusters and high granularity values

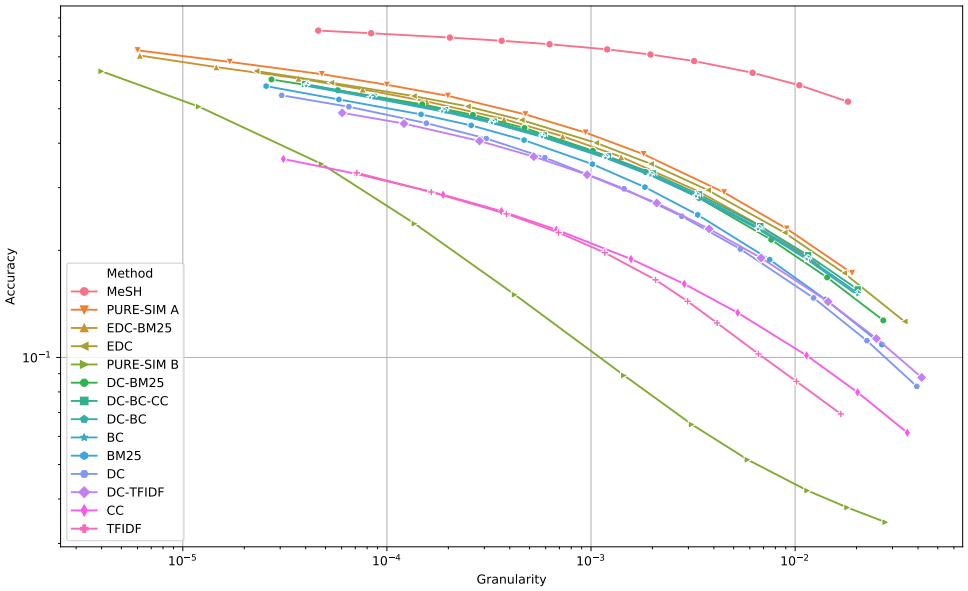


Figure 4.8: GA plot for the full dataset comparing 2 PURE-SIM variants against 11 state-of-the-art approaches and one based on MeSH which is used as the independent evaluation criterion. Approaches are sorted in descending order by the highest accuracy that they achieve.

represent solutions with a high number of clusters. Furthermore, the number of clusters in the solution is correlated with the number of publications per cluster. The more clusters exist, the fewer is the number of publications per cluster and vice-versa. Clusters with more publications usually represent broader research areas while clusters with a small number of publications refer to a more specific research area. For example, if there are 1000 publications in a cluster their only relation could be the broader topic of *"machine learning"* while if there are only 10 publications in a cluster then their relation could be a specific topic as for example, a particular algorithm in *"machine learning"*. In general, one may assume that the lower the granularity of the clustering solution the broader are the research areas identified, and the higher the granularity of the clustering solution the more specific are the research areas identified. The results show that PURE-SIM A is better than other approaches in identifying broader and more specific research areas. However, PURE-SIM B is considerably worse than other approaches in identifying more specific research areas. This result is expected since PURE-SIM B relies on the metadata relations of authors, journals and keywords to determine publication similarity. Intuitively, the authors and journals of publications are good candidates to identify broader research areas but are perhaps not adequate

enough to identify specific research areas since authors often do research on several topics and journals usually publish publications on a broad range of related topics. With respect to the keyword metadata, keywords are typically used to define both broad and more specific topics. However, keywords are usually not uniform. Multiple keywords can be used to refer the same topic (the keywords publication similarity and publication relatedness have for example the same meaning). In addition, the number of keywords provided per publication is very limited. Keywords therefore most likely often fail to connect similar publications in our HIN. For this reason, keywords, by themselves, are not adequate enough to identify more specific topics. We added the PURE-SIM B variant to this experiment to compare the accuracy obtained when no citation information or textual evidence is used while estimating publication similarity. Our results lead to two conclusions. First, they highlight the importance of having citation information and/or textual evidence to estimate the similarity between publications[5] since PURE-SIM B underperformed all the other approaches. Second, we observed that metadata information such as authors, journals and keywords is adequate to identify broader research areas but is not effective to identify more specific topics. However, this metadata in combination with citation information, as is the case in the PURE-SIM A variant, is capable of outperforming all the other approaches.

In order to numerical quantify the performance of PURE-SIM A over the other approaches and to provide further insights in the findings, we use Equation 4.9 to compare the average accuracy of the clustering solutions of each similarity approach against the average accuracy of the clustering solutions based on the MeSH approach (i.e., the similarity approach that was used as the independent evaluation criterion). The calculation of the average accuracy is based on 100 interpolation points. Figure 4.9 illustrates these results. We observe that the PURE-SIM A has a gain of 0.411 compared to MeSH while the best state-of-the-art approach tested has a gain of 0.398. These results indicate that on average, the accuracy of the clustering solutions obtained with PURE-SIM A is 41.1% of the accuracy of the clustering solutions obtained with MeSH, while the accuracy obtained with the best competitor is 39.8%. This means that PURE-SIM A performs 1.3% better than the best competitor.

We also estimated the improvement of PURE-SIM A variant over the other approaches

[5]In particular citation information is important since our method PURE-SIM A does not use textual evidence and is capable of outperforming all the other approaches

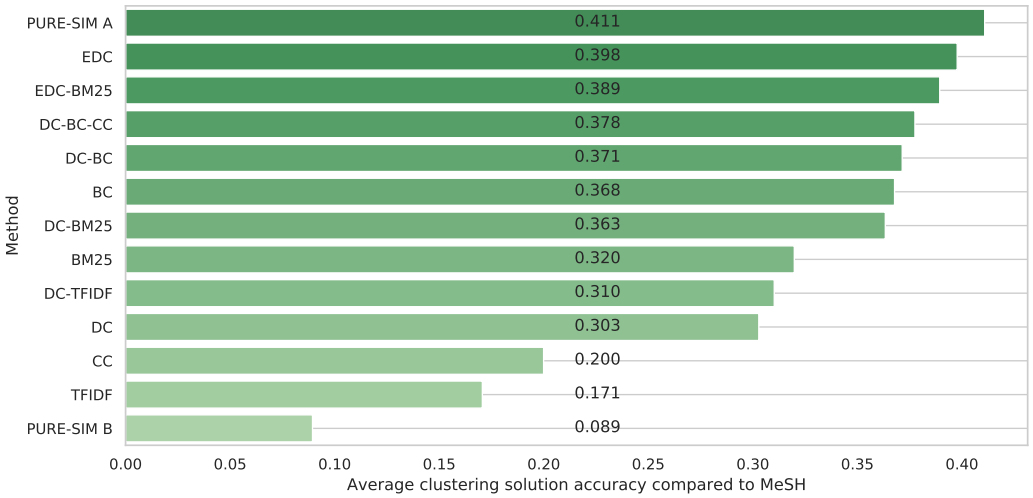


Figure 4.9: Average accuracy of the clustering solutions of the tested approaches compared to the average accuracy obtained using MeSH. The value 0.41 for PURE-SIM B represents that on average the accuracy of the clustering solutions obtained with PURE-SIM B is 41% of the accuracy obtained with MeSH.

tested using the following equation:

$$\frac{Gain(X_{\text{PURE-SIM A}}, X_{\text{MeSH}}) - Gain(X_M, X_{\text{MeSH}})}{MIN(Gain(X_{\text{PURE-SIM A}}, X_{\text{MeSH}}), Gain(X_M, X_{\text{MeSH}}))} \times 100 \tag{4.11}$$

where $Gain(X_{\text{PURE-SIM A}}, X_{\text{MeSH}})$ and $Gain(X_M, X_{\text{MeSH}})$ are the gains estimated using Equation 4.9 for PURE-SIM A and a method M compared to MeSH. We observe that PURE-SIM A presents an improvement of 3.3% compared to the best state-of-the-art approach (EDC) and on average it presents an improvement of 36% over all the other approaches tested.

Another important point to highlight in our experiments is that in order to create a fair evaluation scenario we had to exclude publications from our experiment that (1) do not have a title or an abstract, or (2) are not citing or cited by any other publication in the dataset. This resulted in the exclusion of 1,250,644 publications because some of the tested approaches would not have been able to compute similarities for these publications. For example, the DC approach is not capable of computing similarities for publications that do not have citation relations with other publications in the dataset, while the BM25 approach cannot be used to obtain the similarities for publications for which a title and abstract is not available. In PubMed, all publications are linked to a journal and for most of them keyword and author information is

available. This means that PURE-SIM A not only outperforms the other tested approaches but it is also able to estimate the similarities for the 1,250,644 publications that we excluded from the dataset in order to comply with the inability of the other approaches in dealing with missing information. Similarly, the variant PURE-SIM B is also able to compute the similarities for the excluded publications.

4.6 Summary

In this chapter we presented PURE-SIM an approach to estimate publications similarities. PURE-SIM uses an HIN to model the metadata relations that exist in bibliographic databases and use this model to compute the publications similarities. The metadata combination used by PURE-SIM to estimate publications similarity is user-defined. This aspect is particular important since (i) it allows the user to define a metadata combination that suits the information that is available in their bibliographic databases (e.g., using direct citations combined with authors and keywords in cases where some of the publications in the database do not have references information) and (ii) provides the user with easy access to different publications similarities results (for some applications it may be interesting to compare publications similarities using authors versus the ones obtained while using citations). Furthermore, PURE-SIM also presents other two user-defined parameter that let the user define the importance of metadata relations and control the computation cost of the algorithm.

We used an evaluation framework proposed in a previous study to compare PURE-SIM against other 11 approaches that estimate publications similarities in the context of the publications clustering problem. The evaluation scenario contained more than 2,9 millions of publications and a ground-truth similarity based on human judgement. First, we used the evaluation scenario to thoroughly study the impact of the user-defined parameters in PURE-SIM and define the best combination of parameters to use. Our results showed that the variant PURE-SIM$_{\text{(Journal+Keyword+BC+DC,metadata,300)}}$ is the one that produces the publications similarities that lead to the best clusters. This variant consists in using journals, keywords, bibliographic coupling and direct citations as metadata relations, the metadata normalisation weighting scheme and 300 random walks. We compared this PURE-SIM variant against a total of 11 textual-based, citation-based and hybrid approaches from the literature and results showed that PURE-SIM outperforms the other approaches in identifying more broader clusters (smaller values of granularity) as well as more specific ones (higher values of granularity). We observed that PURE-SIM slightly outperforms the best state-of-the-

art approach. An important point to highlight is that in our experiments we had to exclude 1,2 millions of publications since they did not have textual or reference information therefore the approaches from the literature are not able to compute the similarities of these publication. The excluded publications contain a link to a journal and most of them information about keywords and authors. This means that the best PURE-SIM variant not only outperforms the best approaches from the literature but would also is able to estimate the similarities of these publications thus solving an important problem in the publications clustering task.

Chapter 5

Expertise profiling

The number of scientific documents published every year, as well as the number of active researchers has been increasing every year for the past decades [115]. Due to this exponential growth, identifying the areas of expertise of researchers has become an important task. This task is known as expertise profiling and has important applications such as managing personal in institutions, finding possible collaborations for research and tracking the evolution of expertise for individuals and institutions. Furthermore, expertise profiles can be used to aid the problem of expertise finding. For example, one can use the expertise profile of a researcher to aid with peer-reviewing assignments.

Traditionally, the knowledge or interests of researchers is estimated through their published documents. The general assumption is that a document has a certain topic (i.e., a knowledge area) and since the researcher is (one of) the author(s) of the document, then he has knowledge or interest about that topic. Thus, expertise profiling strategies are divided in two steps. First identify the topics addressed in the publications (i.e., also known as the topic modelling problem). Second, transfer the topics from the publications to the experts in order to create their expertise profile. Due to the large number of available documents and experts in the community, this task cannot be achieved manually. Therefore, automatic tools are necessary to create expertise profiles.

In this chapter we present HEPHIN (Hierarchical Expertise Profiling using Heterogeneous Information Networks) which is a new expertise profiling approach that con-

structs a multi-typed topical hierarchy and maps the knowledge of the experts into this structure. Thus, generating hierarchical expertise profiles. We evaluate HEPHIN in a real-world bibliographic database. We create expertise profiles for 12 researchers, we group the most similar profiles and we use the Google Scholar interests of the researchers to evaluate these groups. The goal of our experiments is two-fold. First, we aim to show that our expertise profiling strategy is capable of creating accurate expertise profiles. Second, we aim to demonstrate that hierarchical expertise profiles provide more information than the tradition expertise profiles that characterise the knowledge of an expert in a flat line of topics. In the final part of our experiments, we also present a series of applications with the HEPHIN expertise profiles that show the benefits of having hierarchical expertise profiles and a multi-typed topical hierarchy.

Note that throughout this chapter we use the terms author, researcher and expert interchangeably in the sense that they both refer to a person whose knowledge we aim to profile. We also use the terms knowledge and interest interchangeably since we consider that if a person has knowledge on a certain topic he is also interested in that topic.

5.1 Motivation

In most cases, the expertise profiling problem does not have a pre-defined set of topics where the knowledge of the experts is mapped to. Instead, the knowledge areas are identified in a data-driven fashion by using a topic modelling approach over the expert's documents. The Latent Dirichlet Allocation (LDA) [44] model is the most widely used strategy to identify topics discussed in text. Due to its potential, the LDA algorithm was adapted to output the distribution of authors over the topics [4]. This discovery fostered the development of a group of algorithms named Author-Topic models that, not only identify topics in documents, but also profile the author's expertise. Since then, several other Author-Topic models have been proposed [5, 41, 103].

The core of the Author-Topic models is the LDA algorithm which despite being widely used, still have some known flaws such as [116]: lacking an intrinsic methodology to choose the number of topics, containing several hyper-parameters that can cause overfitting and incompatibility with properties of text such as Zipf's law for the frequency of words. As a result, strategies that avoid the disadvantages of LDA-based topic modelling are necessary in the expertise profiling domain. It is worth to highlight that there are several other strategies to topic modelling besides the LDA

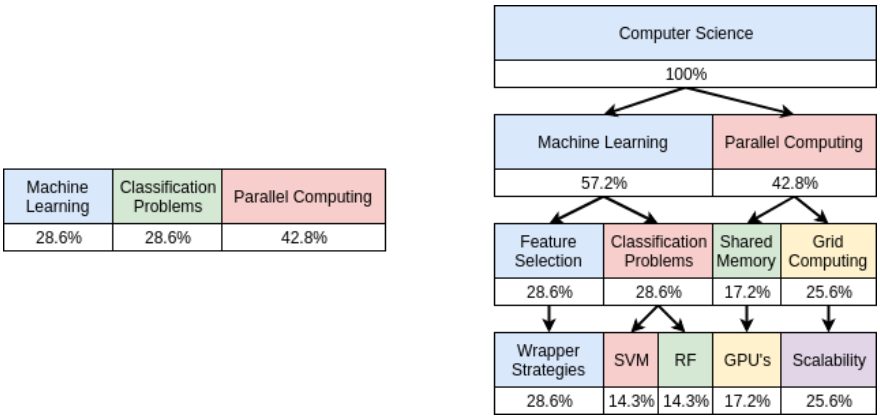


Figure 5.1: An expertise profile generated over a set of independent topics (profile on the left) compared to one generated over a topical hierarchy (profile on the right).

algorithm. However, these approaches do not address the task of mapping the experts into the discovered topics. Thus, they are not adequate to generate expertise profiles.

Another point that motivate the work presented in this chapter was the study of Berendsen et al. [16] which states that experts have reported that the expertise profiles created for them are often redundant, and either too general or too specific. This occurs because the expertise profiles are generated from a flat list of topics without any relation between them. A solution to this problem is to use (or create) a topical hierarchy where topics are related to each other over "sub-topic of" relations. Figure 5.1 illustrates the differences between expertise profiles generated over flat list of topics versus the ones generated over a topical hierarchy. The hierarchical expertise profile provides information of an expert's knowledge at different granularity levels, thus presenting a more detailed overview of an expert's knowledge.

In the literature, there are some strategies to generate topical hierarchies. However, similarly to other topic modelling approaches, they do not offer a solution to map experts into the hierarchy thus it is not possible to use them to generate expertise profiles. With respect to the expertise profiling task, there are a few works that created hierarchical expertise profiles. For example, Bin et al. [117] uses explicit feedback from persons and their bookmarks information to extract keywords that reflect their expertise. Afterwards, these keywords are mapped into a pre-defined topical hierarchy. Thus, constructing an hierarchical profile. Rybak et al. [48] uses publication's meta-data to maps authors into the ACM computation classification system. Since this is organised in hierarchies, the expertise profile is also hierarchical. There are two limitations in both strategies. First, both use a manually created topical hierarchy which requires a lot of human effort. Furthermore, these structures are dynamic which

indicates that this is not a one-time task. Second, not all the expert's information was considered as evidence of their knowledge. In [117] the authors restricted the keywords to the ones that are on the topical hierarchy. Similarly, Rybak [48] restricts the author's publication to the ones published in ACM conferences. Both strategies potentially leave out details that may be relevant to characterise the experts' knowledge. Thus, strategies that automatically create topical hierarchies and provide ways to map the expertise of persons into the structure are necessary for the expertise profiling task.

5.2 Overview of our contribution

In this chapter we present HEPHIN (Hierarchical Expertise Profiles using Heterogeneous Information Networks) which is a new expertise profile approach that is capable of generating hierarchical expertise profiles. HEPHIN analyses the metadata relations between the documents of the experts and creates a multi-typed topical hierarchy where each topic is represented by lists of attributes. Then, the experts are mapped into the multi-typed topical hierarchy according to their relations with the attributes and the hierarchical expertise profile is created. The hierarchical expertise profile provides a detailed overview of an experts knowledge. In more detail, the profile starts describing the knowledge at broad topics (e.g., computer science) and ends in very specific topics (e.g., support vector machines - an algorithm used in machine learning).

HEPHIN is also capable of creating the expertise profile of other entities besides persons. For example, we can create the expertise profile of a document, a thesis or a institution. This extends the applications of HEPHIN profiles compared to other expertise profiling strategies. For example, the expertise profile of an institution can be used to summarise the knowledge of an entire organisation. Additionally, the expertise profile of a thesis can be compared with the expertise profiles of experts in order to determine candidates to be part of the thesis jury.

We test HEPHIN in a real-world bibliographic database consisting of Portuguese researchers and we show the advantages of having hierarchical profiles compared to traditional flat profiles. Furthermore, we show multiple scenarios where we benefit from having a multi-typed topical hierarchy and a strategy to map knowledge in this structure through an attribute-entity relation.

5.3 Problem formalisation

We formalise the problem of expertise profiling as the task of receiving a set of authors and their publications and profiling their knowledge in an hierarchical profile. First, we create a star-schema Heterogeneous Information Network (HIN) G using the publications metadata. The HIN contains $|\mathcal{A}| + 1$ node types. There are $|\mathcal{A}|$ node types that represent the number of different metadata attributes considered from the publications (e.g, keywords and authors) and the remaining node type represents the publications. The publications node type is defined as the star type in the HIN. Second, we use a community detection strategy to create a multi-typed topical hierarchy T.

Definition 5.3.1 *We define a multi-typed topical hierarchy T as tree structure where each node t_i represents a topic. A topic $t_i \in T$ has a parent node $parent(t_i)$ and a set child nodes $children(t_i)$. The root is the only node where $parent(root) = \varnothing$. Each topic t_i is on a certain level $L(t_i)$ of the tree. The level of the root is 0 (i.e., $L(root) = 0$), the level of the topics $t_i \in children(root)$ is 1 since they are the children of the root node, and so on. Each topic t_i contains $|\mathcal{A}|$ lists of attributes. Each list $A_k \in t_i$ contains a subset of the attributes of type k of the network that are part of the topic t_i. Furthermore, these attributes are ranked according to their importance in the topic. More concretely, the top ranked attributes are more representative of the topic. The attributes are not exclusive to topics (i.e., an attribute a_k can be part of topics t_1 and t_2, for example).*

Finally, we map the experts into the topical hierarchy and create their hierarchical expertise profile K.

Definition 5.3.2 *An hierarchical expertise profile K is a sub-tree of T. Each node $k_i \in K$ represents a topic from T and contains a value q that represents the knowledge of the expert with respect to topic t_i. The values of q are normalised with respect to all the nodes in the same tree level. More concretely, $\forall l \in L, \sum_{t_i \in P_l} q = 1$, where L is the number of levels in the tree and P_l is the set of topics at level l.*

5.4 Methodology

HEPHIN is a graph-based algorithm that analyses the metadata relations among publications to create a multi-typed topical hierarchy and then, maps entities (e.g., experts, documents, thesis or institutions) into the topical hierarchy in order to obtain an expertise profile. HEPHIN is capable of using any metadata relations that are defined by the user. However, to ease understanding of the discussion of HEPHIN as well as the experiments in Section 5.5, we present the algorithm using a use-case on the Authenticus bibliographic database [118]. In the following sections we explain HEPHIN in more detail. The discussion is divided into network construction (how the HIN is constructed using the metadata from the publications), topic modelling (how the topics addressed by the publications are discovered through the analysis of the HIN), attributes ranking (how the attributes are ranked inside each topic of the topical hierarchy), topical hierarchy construction (how the topics relations are estimated) and knowledge mapping (how we map entities into the topical hierarchy in order to obtain expertise profiles).

5.4.1 Network construction

In the case of the Authenticus database we intuitively assessed that the most valuable metadata that is available for most publications and useful to estimate the topics is: authors, keywords and ISI fields. These are the metadata attributes that we use to construct the HIN. In more detail, $\mathcal{A} = \{\text{Author}, \text{ISI Field}, \text{Keyword}\}$ for this particular case. As a side note regarding the data in the Authenticus database, the authors are disambiguate using a manual process, the keywords are a joint set of the keywords manually associated by the authors and the ones extracted automatically using keyword summarisation tools, and the ISI fields are research areas defined by the Institute for Scientific Information.

The process of constructing the HIN starts with the definition of the set of publications $\mathcal{P}$. For every $p \in \mathcal{P}$ we query tje database for the authors, keywords and ISI fields associated with these publications. Then, we add every publication p into the HIN and create an edge to its metadata. For example, a publication p_1 with authors a_1 and a_2, keywords k_5, k_7 and k_8, and ISI field i_1 has a total of 6 edges, each one representing a connection to a different metadata attribute[1]. It is important to note

[1]Note that this process is similar to the one used in Section 4.4.1. The only difference is related to the metadata considered in both cases.

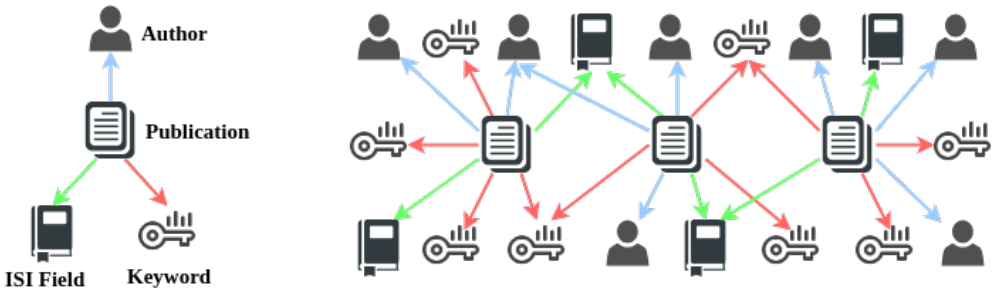


Figure 5.2: Network schema of the HIN constructed for the Authenticus database.

that since publications are the star-nodes of the HIN, two publications are never connected through an edge. Instead, the likelihood of publications being part of the same topic is measured through the analysis of the meta-paths star-attribute-star connecting them. For example, consider publications p_1 and p_2 both sharing the attributes a_5 and k_3. In this case, the HIN contains the meta-paths $p_1 - a_5 - p_2$ and $p_1 - k_3 - p_2$. Figure 5.2 illustrates the schema of the HIN constructed.

In the use-case of HEPHIN in the Authenticus database, there are three different metadata relations in the HIN: publication-author (P-A), publication-keyword (P-K) and publication-ISI field (P-I). HEPHIN allows the users to distinguish the most important metadata relations to estimate the topics. For this purpose, each different metadata relation has a weight W_{p-x} that represents the importance of attributes of type x in the HIN. The W_{p-x} values are used by HEPHIN to define the weights on the edges in the HIN. For each publication, the weight of its edges to attributes type x is normalised by the number of attributes of the type x and multiplied the weight of the relation W_{p_x}. For example, consider that publication p_1 has attributes a_1, a_2 and a_3, and that the weight of the publication-author relation is $W_{p-a} = 0.5$. Then, the 3 edges between p_1 and its authors (a_1, a_2 and a_3) would be $\frac{1}{3} \times 0.5 \approx 0.17$.

5.4.2 Topic modelling

HEPHIN identifies the topics addressed in the set of publications $\mathcal{P}$ by unveiling communities in the network structure of the HIN. We assume that each community represents a topic or a knowledge area for the expertise profiling task. HEPHIN can use any community detection algorithm to unveil the communities. In this particular use-case, we used the Louvain algorithm [61] which is a greedy optimization algorithm that maximises the modularity of the communities discovered[2]. We decided to use

[2]More details about the Louvain algorithm were provided in Section 2.1.2.2

this version of the Louvain algorithm due to its expected runtime $O(n\ \log(n))$, where n is the number of nodes in the network. Furthermore, modularity optimization is a widely used strategy to discover communities and, in our opinion, is an adequate measure for this specific problem.

The Louvain algorithm (similarly to many other community detection algorithms) does not produce overlapping communities and does not consider nodes and edges heterogeneity. As a result, using this algorithm in the HIN leads to a loss of information and produces the undesired effect of hard-clustering the attribute-nodes (e.g., a keyword is exclusive to a topic). To overcome these problems, HEPHIN converts the HIN into a publication similarity graph $G' = (\mathcal{V}', \mathcal{E}')$ where $\mathcal{V}'$ is the set of publications in the HIN and $\mathcal{E}'$ are the set of edges that represent the pairwise similarity between two publications. This value is estimated using the following equation:

$$E_{p1,p2} \in \mathcal{E}' = \sum_{n\ \in\ \mathcal{Y}} E_{p1,n} + E_{p2,n} \tag{5.1}$$

where $\mathcal{Y}$ is the set of attribute-nodes that are adjacent to p_1 and p_2 in the HIN and $E_{n1,n2}$ is the edge weight between nodes n_1 and n_2 in the HIN.

After obtaining the similarity graph G' HEPHIN uses the community detection algorithm (in this use-case the Louvain algorithm) to obtain the community partition C. Extrapolating the community partition C back to the HIN, HEPHIN obtains the community membership of all the star-nodes (i.e., the publications). On the next step, HEPHIN expands these communities to estimate the community membership for the attribute-nodes in the HIN. Due to the star-schema topology of the HIN, every attribute-node is linked to at least one star-node that belongs to a certain community $c_k \in C$. HEPHIN estimates the community membership of attribute-nodes as the fraction of their edge weights connecting to different communities. In more detail, if an attribute-node a_1 is linked to star-nodes p_1, p_2 and p_3, and p_1 and p_2 are members of community c_1 and p_3 is member of community c_2, then the community membership of a_1 is $\approx 67\%$ in c_1 and $\approx 33\%$ in c_2. Note that on this example, for simplicity, we are considering equal weight for all the edges between the attribute-node and the publications. In the end of the process, all the nodes in the HIN are assigned to one more more communities. Figure 5.3 illustrates the complete topic modelling process of HEPHIN when a community detection algorithm such as the Louvain algorithm (i.e. one that does not considered nodes and links heterogeneity) is utilised.

An important point to highlight is that the similarity graph conversion presented in this

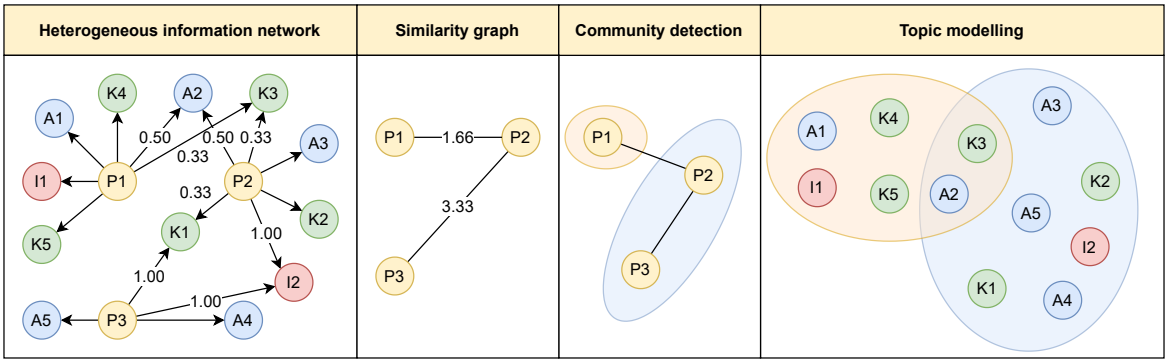


Figure 5.3: The different phases of the topic modelling approach used in HEPHIN.

section is only necessary when the community detection algorithm utilised presents the same disadvantages as the Louvain community detection algorithm. More concretely, the community detection algorithm is not adequate for the network structure of a HIN. In the case of using one algorithm for overlapping community detection in HINs, the similarity graph step is not necessary and communities can be unveiled directly from the HIN. Unfortunately, most of the community detection algorithms available in network science are not adequate for the HIN and for that reason we introduce this step in HEPHIN.

5.4.3 Attributes ranking

For every discovered topic using HEPHIN we aim to provide the users with some information about what does the topic represent. For that propose, we rank the attribute-nodes within each topic according to their importance (i.e., how well they represent the topic) and their type. Since each topic is a subgraph of the HIN, HEPHIN removes the publications from the subgraph, creates an edge between attribute-nodes that shared one or more publications (see the right most part of Figure 5.3 for an example) and uses centrality measures in order to determine the attribute-nodes importance. In our experiments we tested the node's degree, betweenness, closeness and PageRank centrality measures. Our results showed that PageRank seems to be the best metric for our propose. Therefore, HEPHIN uses the PageRank algorithm to rank the attribute-nodes within each topic.

5.4.4 Topical hierarchy

The topic modelling strategy presented in Section 5.4.2 generates a flat list of topics $\mathcal{C} = \{C1, C2, ..., Cn\}$ for a HIN. In order to construct the topical hierarchy HEPHIN considers each community that is discovered as a HIN $G_{Ci} \subseteq G$ and repeatedly applies the topic modelling strategy. Note that each community represents a topic, therefore the communities discovered using the topic modelling on the subgraph G_{Ci} are the children of the topic represented by community Ci. Similarly, the subgraph G_{Cj} that discovered the community represented by the subgraph G_{Ci} is the parent of the topic represented by community Ci. HEPHIN requires the user to define the value l which is the depth of the topical hierarchy constructed. In more detail, the topic modelling process stops when the new topics are at the l level of the topical hierarchy. Overall, the complete HEPHIN process to created the multi-typed topical hierarchy is described in the following steps:

1. Start with HIN $G = (\mathcal{V}, \mathcal{E})$
2. Convert the HIN into a similarity graph G' of star-nodes.
3. Apply the community detection algorithm such that CommDetect(G') = $\mathcal{C}'$ where $\mathcal{C}' = \{C'1, C'2, ..., C'n\}$ and every $C'i$ represents a community of star-nodes.
4. Expand the communities $\mathcal{C}'$ into $\mathcal{C}$ by estimating the membership of all the attribute-nodes in G
5. For each $Ci \in C$:
    - 5.1. Obtain subgraph $G_{Ci} = (\mathcal{V}_{Ci}, \mathcal{E}_{Ci})$ where $\mathcal{V}_{Ci}$ is the set of nodes in community Ci and $\mathcal{E}_{Ci}$ is the set of edges between those nodes, i.e., $(n1, n2) \in \mathcal{V}_{Ci} : n1 \in Ci, n2 \in Ci$ and $(n1, n2) \in \mathcal{E}$.
    - 5.2. Rank the attribute-nodes in Ci using PageRank.
    - 5.3. If the current level is smaller than l, set $G = G_{Ci}$ and go back to step 1.

We should highlight that in case of using HEPHIN with an overlapping community detection for HINs, steps 2 and 4 are not necessary. Figure 5.4 shows a sample of the multi-type topical hierarchy obtained with HEPHIN on the Authenticus database.

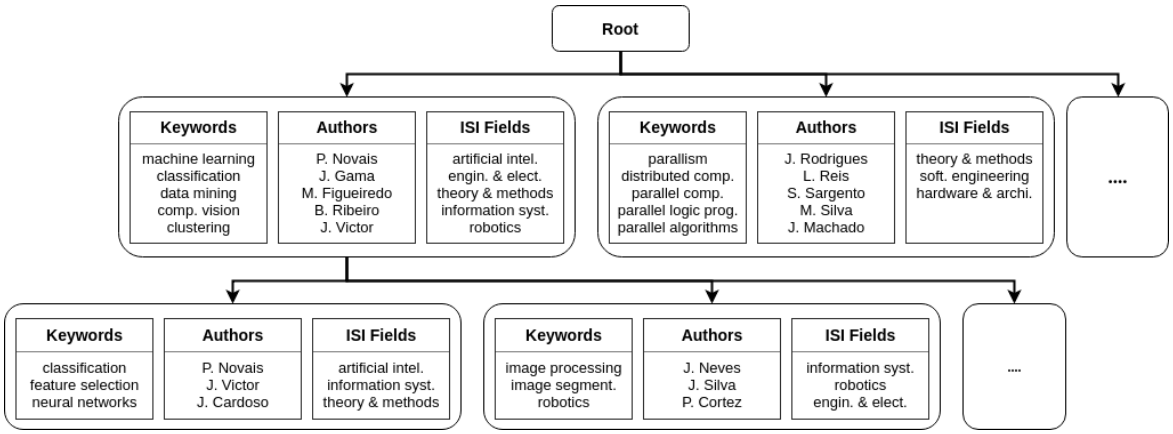


Figure 5.4: Sample of the multi-typed topical hierarchy constructed by HEPHIN.

5.4.5 Knowledge mapping

One of the problems of using a topical hierarchy on the expertise profiling task is that most of the times mapping experts into the hierarchy is either not trivial, or it requires discarding information (e.g., [117, 48]). HEPHIN creates a topical hierarchy where each topic consists of multiple attributes, therefore it is possible to describe the knowledge of a person in function of these attributes. For example, consider the use case of the Authenticus database where each topic consists of authors, keywords and ISI fields. In this scenario, it is possible to obtain the expertise profile of authors in the Authenticus database since they were used to create the topical hierarchy. Thus, they already have a topical distribution across the topical hierarchy that represents their knowledge (e.g., the author "P. Novais" on Figure 5.4). We name this approach *"direct mapping"*. In cases where we want to create the expertise profile of an expert that is not part of the topical hierarchy (i.e., the expert is not an author in Authenticus, therefore he is not represented by an attribute-node in the HIN) it is possible to describe the person knowledge in function of other attributes that are present in the topical hierarchy (e.g., the keywords that the expert uses in his publications). We name this approach *"indirect mapping"*. An interesting point to highlight is that the *"indirect mapping"* strategy can be used to create the expertise profile of other entities besides experts. For example, HEPHIN, in this specific case of Authenticus, can create the expertise profile profile of institutions (using authors as attributes), thesis and publications (using keywords or ISI fields for both cases).

For the sake of interpretability, we continue the discussion of knowledge mapping considering the example of creating an expertise profile for an expert p. HEPHIN creates the expertise profile of an expert p using a *"direct mapping"* when $p \in \mathcal{V}$. More

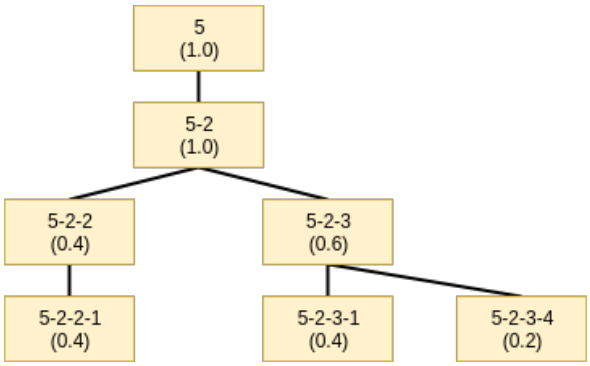


Figure 5.5: Example of an HEPHIN hierarchical expertise profile.

concretely, p is represented by an attribute-node in the HIN. The expertise profile is obtained by using a walk-through process in the multi-typed topical hierarchy T and gathering all the topics $G_Ci = (\mathcal{V}_{Ci}, \mathcal{E}_{Ci})$ where $p \in \mathcal{V}_{Ci}$, i.e., the attribute-node of the expert is one of the nodes in the community Ci. In the end, we obtain the hierarchical expertise profile $K \subseteq T$. For example, consider that p at the lowest level of T is 40% in topic*"5-2-2-1"*, 40% in *"5-2-3-1"*, and 20% in *"5-2-3-4"* (the percentage values are obtained considering the edge weight distribution of the attribute-node to the star-nodes). Then, his expertise profile considering the complete topical hierarchy is illustrated by Figure 5.5.

In the case of *"indirect mapping"* HEPHIN obtains the expertise profile using the *"direct mapping"* for all the attributes that characterise the knowledge of person. Then, the profiles are merged into a single one using weighted average. For this merged profile, HEPHIN normalises the membership per tree level (i.e., for any level of the profile the total membership in topics is 100%). Then, HEPHIN removes topics which membership is below 5% and recompute the normalisation step. HEPHIN discards these topics to remove topics that are not significant for the person. We estimate the value 5% through experimentation.

5.5 Experimental setup

In this section we test the performance of HEPHIN with respect to the quality of the topics in the multi-typed topical hierarchy and the hierarchical profiles constructed. In order to have some a priori knowledge about the dataset used in our experiments we used a subset of the Authenticus database. We manually selected 20 authors from the *Computer Science* area, then we added to the subset their co-authors and the co-

Table 5.1: Description of the eight topical hierarchies constructed. The left part of the table details the relation weights used in each hierarchy while the right part of the table details the number of topics discovered in total and at each level of the hierarchy. p-k: publication-keyword. p-a: publication-author and p-i: publication-ISI field.

	Relation weights				**N. of topics per level**				
	uniform?	W_{p-k}	W_{p-a}	W_{p-i}	0	1	2	3	**Total**
T_1	Yes	1.0	1.0	1.0	4	9	10	10	33
T_2	No	1.0	1.0	1.0	4	55	122	200	381
T_3	No	2.0	1.0	0.5	4	85	352	684	1125
T_4	No	2.0	0.5	1.0	4	72	253	479	808
T_5	No	1.0	2.0	0.5	4	51	235	563	853
T_6	No	0.5	2.0	1.0	4	22	54	94	174
T_7	No	1.0	0.5	2.0	4	14	30	49	97
T_8	No	0.5	1.0	2.0	4	9	19	21	53

authors of their co-authors. Furthermore, we obtained the publications of all the authors and its respective metadata. In the end, the subset contained 8587 publications, 2715 different authors, 19662 different keywords and 120 different ISI fields. These values represent the number of nodes from each type in the HIN. We built a total of 8 topical hierarchies with 4 levels each by changing the weight associated to the metadata relations in the HEPHIN algorithm. We determined through experimentation that topical hierarchies with 4 levels yield the most comprehensible topical hierarchy for our dataset. The topical hierarchies are numbered from T_1 to T_8. Table 5.1 presents the weights associated to each metadata relation in order to generate the topical hierarchy, and the number of topics discovered at each level of the hierarchy for all cases. For evaluation purposes, T_1 represents a topical hierarchy constructed when all the weights in the HIN are uniform ,i.e., $E_{x,y} = 1,\ \forall E_{x,y} \in \mathcal{E}$.

The results show that the metadata relations have a huge impact on the number of topics discovered. The publication-keyword metadata relation is the one that increases the most the number of topics discovered as the value of this relation increases. We obtained the fewest number of topics (excluding the topical hierarchy generated with uniform weights) when the weight of the publication-keyword metadata relation was 0.5 and the weight of the publication-ISI field metadata relation was 2.0. The topical hierarchy generated from a HIN with uniform weights presents the fewest number of topics discovered by a high margin. This number of topics seems extremely far from an adequate one, thus showing that it is important to have adequate weights for the different metadata relations in the HIN.

5.5.1 Topic evaluation

In the literature, there are several metrics to evaluate the quality of topics modelled. However, they assume that the topics consist only of words and that they were obtained using statistical inference on text. As a result, they are not adequate to evaluate the multi-typed topics discovered by the HEPHIN algorithm, i.e., topics that consist of lists of attributes. We evaluate our topics using the heterogeneous pointwise mutual information (HPMI) metric. We decide to use HPMI since it was already used to evaluate multi-typed topics in a previous study [50]. HPMI is an extension of the point mutual information metric (see Section 2.4.5) which is commonly used in topic modelling. For every modelled topic, HPMI estimates the average relatedness of each pair of attributes ranked at top-k:

$$HMPI(v^x, v^y) = \begin{cases} \frac{2}{k(k-1)} & \sum_{1 \leq i < j \leq k} log(\frac{p(v_i^x, v_j^y)}{p(v_i^x)p(v_j^y)}) \quad x = y \\ \frac{1}{k^2} & \sum_{1 \leq i,j \leq k} log(\frac{p(v_i^x, v_j^y)}{p(v_i^x)p(v_j^y)}) \quad x \neq y \end{cases} \tag{5.2}$$

where v^x is a node of type x, ranked among the top-k attributes of type x in a certain topic. The higher the HPMI is, the more coherent the topics are. We estimate the HPMI of the 8 topical hierarchies obtained using $k = 20$ and $k = 40$. Following the idea of [50], we defined $k = 5$ for ISI fields since this attribute only has 120 nodes in the HIN. In these specific cases, the part $\frac{1}{k^2}$ of the formula changes to $\frac{1}{5k}$. Note that the higher the values of HPMI the better (i.e., the more coherent) the topics discovered are.

Table 5.2 shows the scores obtained. Each column represents the average relatedness of a pair of object types (x, y) for all the topics discovered. The last column presents the average obtained for all the 6 possible relations. The results show that the scores are very similar for $k = 20$ and $k = 40$. HEPHIN obtained a positive HPMI for 5 out of the 8 topical hierarchies constructed. The best result (T_3) presents topics that are highly coherent with an average of 1.818 for $k = 20$ and 1.825 for $k = 40$. This topical hierarchy was obtained when the weight of publication-keyword relations was multiplied by 2, the weight of the publication-author relation multiplied by 1 and the weight of the publication-ISI field relation multiplied by $\frac{1}{2}$. Thus, showing that the publication-keyword relation, in this specific dataset, is the most important one. Conversely, the two topical hierarchies constructed using a weight of 2 for the publication-ISI field metadata relation (T_7 and T_8) are the only ones with a negative HPMI. Furthermore, the results further validate the necessity of using non-uniform weights in the HIN construction. The topical hierarchy T_1 that was constructed from

Table 5.2: HPMI results for the topical hierarchies constructed using $k = 20$ and $k = 40$. The highest values for each k are presented in bold. NT is the total number of topics modelled.

	NT	K-K	K-A	K-I	A-A	A-I	I-I	Average
k = 20								
T_1	33	-1.847	-0.960	-0.726	-1.910	-0.764	-1.056	-1.211
T_2	381	0.204	1.420	0.222	3.164	0.439	0.057	0.918
T_3	**1125**	**1.392**	**2.355**	**0.467**	**5.780**	**0.692**	**0.223**	**1.818**
T_4	808	0.855	1.932	0.347	4.807	0.559	0.144	1.441
T_5	853	1.025	1.425	0.263	2.735	0.425	0.032	0.984
T_6	174	0.557	0.479	-0.030	-0.382	0.009	-0.209	0.071
T_7	97	-1.040	0.492	-0.218	-0.955	-0.135	-0.270	-0.354
T_8	53	-1.816	-0.946	-0.645	-1.899	-0.671	-0.561	-1.090
k = 40								
T_1	33	-1.791	-0.966	-0.755	-1.912	-0.757	-1.056	-1.206
T_2	381	0.289	1.395	0.213	3.171	0.435	0.057	0.927
T_3	**1125**	**1.443**	**2.349**	**0.467**	**5.777**	**0.691**	**0.223**	**1.825**
T_4	808	0.902	1.938	0.345	4.808	0.559	0.144	1.449
T_5	853	1.082	1.423	0.269	2.739	0.422	0.032	0.995
T_6	174	0.588	0.479	-0.018	-0.394	0.003	-0.209	0.075
T_7	97	-0.972	0.472	-0.205	-0.969	-0.130	-0.270	-0.346
T_8	53	-1.730	-0.944	-0.636	-1.922	-0.645	-0.561	-1.073

a HIN with uniform weights presents the worst HPMI values for both $k = 20$ and $k = 40$.

5.5.2 Profiles evaluation

We evaluate the expertise profiles obtained with HEPHIN for 12 experts that are *Computer Science* professors at the University of Porto. We selected these experts due to our personal knowledge about their expertise which facilitates the process of evaluating expertise profiles. For each expert, we obtained the research interests that they manually added to their Google Scholar page (see Figure 2.11 for an example). In our experiments, we assume that the research interests of the experts reflect their topics of expertise. Table 5.3 presents the data obtained from the Google Scholar pages for the 12 authors.

For each expert, we use HEPHIN with the topical hierarchic T_3 to create their hierarchical expertise profile. We use the topical hierarchy T_3 since this is the one that generated the most coherent topics (i.e., the highest values of HPMI). After obtaining

Table 5.3: Expert's research interests obtained from their Google Scholar pages. NP refers to the number of publications the experts have in the Authenticus database.

Name	NP	Google Scholar research interests
Alipio Jorge	133	data mining; machine learning; text mining; recommender systems; artificial intelligence machine learning
Fernando Silva	91	parallel and distributed computing; logic programming; information mining; algorithms; complex networks
Luis Torgo	90	data mining; machine learning
Ricardo Rocha	90	logic programming; tabling; parallelism; language implementation
Nelma Moreira	89	automata theory; descriptional complexity; formal verification of software
Rogerio Reis	81	formal languages; automata theory; combinatorics
Veronica Orvalho	40	computer graphics
Pedro Ribeiro	37	complex networks; algorithms and data structures; parallel and distributed computing; computer science education; artificial int
Pedro Brandao	31	communication networks; body area networks; ehealth; distributed systems
Antonio Porto	30	logic programming; coordination; artificial intelligence
Rita Ribeiro	25	data mining; machine learning
Sergio Crisostomo	16	computer networks; communications; computer science

all the expertise profiles we compare their pairwise similarity per hierarchy level. More concretely, to estimate the similarity between two experts at a certain level l we obtain the topical distribution of each expert at l and measure the intersection (i.e., topics at that specific level that are common to both experts). The similarity values per level range from 0 to 1, where 1 indicates a perfect match between the experts expertise, while 0 describes no match in terms of expertise for the experts. We also estimate the total similarity for the profiles which is the sum of the similarities obtained for all the hierarchy levels. The topical hierarchy used in this experiment has 4 levels, therefore the total similarity value ranges between 0 and 4. The higher the similarity the more similar is the expertise of two experts.

In this test we aim to use the research areas as evidence to evaluate whether or not two experts should have similar profiles. In total we have 132 pairwise comparisons. We divided the results discussion into two groups considering the total similarity between experts in order to filter the number of cases discussed. On the first group the total similarity between the experts is greater or equal to 2, while in the second group the total similarity is lower than 1. In terms of expertise comparison, the first group contains pairs of experts whose expertise is similar while the second ones contains experts with expertise in different topics. Tables 5.4 and 5.5 present the pairwise similarity between the experts and the number of topics in common for both groups.

Only 7 out of 132 comparisons scored a total similarity greater or equal to 2. This number is expected since we have a broad range of research interests and the lower levels of the topical hierarchy (represented by the higher level numbers in the table) represent very specific topics. Thus, making it more difficult to find similar experts

Table 5.4: Pairwise similarity between experts with a total similarity greater or equal to 2. NT refers to the number of topics in which both experts have expertise.

		Level 0		Level 1		Level 2		Level 3		Total	
Author 1	**Author 2**	**Sim**	**NT**	**Sim**	**NT**	**Sim**	**NT**	**Sim**	**NT**	**Sim**	**NT**
Nelma Moreira	Rogerio Reis	1.00	2	1.00	3	1.00	4	1.00	4	4.00	13
Fernando Silva	Pedro Ribeiro	0.73	3	0.73	3	0.60	3	0.60	3	2.66	12
Pedro Brandao	Sergio Crisostomo	0.66	2	0.66	2	0.66	2	0.66	2	2.64	8
Alipio Jorge	Luis Torgo	0.85	3	0.59	4	0.56	4	0.56	4	2.56	15
Fernando Silva	Ricardo Rocha	0.77	3	0.62	4	0.56	4	0.28	3	2.23	14
Pedro Ribeiro	Ricardo Rocha	0.76	3	0.57	3	0.42	3	0.26	2	2.01	11
Luis Torgo	Rita Ribeiro	0.67	2	0.67	2	0.34	2	0.32	2	2.01	8

at those levels. The highest pairwise similarity score (Nelma Moreira and Rogerio Reis) represent a perfect profile match at all hierarchical levels. Although their Google scholar interests are very similar, we further looked into this case due to the fact that it represents a wide gap in terms of similarity to the other cases. A co-authorship analysis on the network revealed that the two experts are co-authors in 66 publications (81.5% of Rogerio Reis's publications). Therefore, the perfect match is expected. Regarding the other cases, we observe high similarity between expertise profiles pairs from experts that have knowledge in topics such as: machine learning (Alipio Jorge, Luis Torgo, and Rita Ribeiro), parallel programming (Fernando Silva, Pedro Ribeiro and Ricardo Rocha), and communication networks (Pedro Brandao and Sergio Crisostomo).

Another interesting point to highlight is that two experts, Veronica Orvalho and Antonio Porto, are not similar enough with any other expert. In the case of Veronica Orvalho, this is anticipated due to the fact that her interest on computer graphics is not shared by any other expert. However, in the case of Antonio Porto, since his interests refer to areas shared by other experts an higher comparison was expected. A further look into his expertise profile revealed that it is scattered by several topics. As a result, his intersections with other experts are not significant enough to be considered similar with other expertise profiles.

With respect to profiles with total similarity lower or equal to 1 we observe that in general the results complement the observations from the other group. More concretely, all the experts that have knowledge in a specific area such as machine learning have a low similarity profile with all the experts that have have knowledge in other areas such as parallel programming and communication networks, and vice-versa. An interesting point to highlight is that in most cases we observe that there is a similarity between the profiles in the level 0 of the hierarchy (i.e., on the broader topics), however as the topics get more specific the intersections between the authors knowledge decreases. Another interesting case to point out is the case of expert Sergio

Table 5.5: Pairwise similarity between experts with a total similarity lower or equal to 1. NT refers to the number of topics in which both experts have expertise.

		Level 0		Level 1		Level 2		Level 3		Total	
Author 1	**Author 2**	**Sim**	**NT**	**Sim**	**NT**	**Sim**	**NT**	**Sim**	**NT**	**Sim**	**NT**
Nelma Moreira	Veronica Orvalho	0.25	1	0.25	1	0.25	1	0.25	1	1.00	4
Rogerio Reis	Veronica Orvalho	0.25	1	0.25	1	0.25	1	0.25	1	1.00	4
Antonio Porto	Pedro Ribeiro	0.66	2	0.33	1	0.00	1	0.00	1	0.99	5
Antonio Porto	Pedro Brandao	0.66	2	0.33	1	0.00	1	0.00	1	0.99	5
Fernando Silva	Luis Torgo	0.37	2	0.37	2	0.17	1	0.00	1	0.91	6
Nelma Moreira	Pedro Ribeiro	0.33	1	0.33	1	0.25	1	0.00	1	0.91	4
Nelma Moreira	Pedro Brandao	0.58	2	0.33	1	0.00	1	0.00	1	0.91	5
Nelma Moreira	Sergio Crisostomo	0.58	2	0.33	1	0.00	1	0.00	1	0.91	5
Pedro Ribeiro	Rogerio Reis	0.33	1	0.33	1	0.25	1	0.00	1	0.91	4
Pedro Brandao	Rogerio Reis	0.58	2	0.33	1	0.00	1	0.00	1	0.91	5
Rogerio Reis	Sergio Crisostomo	0.58	2	0.33	1	0.00	1	0.00	1	0.91	5
Alipio Jorge	Pedro Brandao	0.61	3	0.28	2	0.00	1	0.00	1	0.89	7
Alipio Jorge	Antonio Porto	0.64	2	0.14	1	0.00	1	0.00	1	0.78	5
Fernando Silva	Sergio Crisostomo	0.33	1	0.33	1	0.00	1	0.00	1	0.66	4
Pedro Ribeiro	Sergio Crisostomo	0.33	1	0.33	1	0.00	1	0.00	1	0.66	4
Rita Ribeiro	Sergio Crisostomo	0.33	1	0.33	1	0.00	1	0.00	1	0.66	4
Ricardo Rocha	Sergio Crisostomo	0.47	2	0.14	1	0.00	1	0.00	1	0.61	5
Luis Torgo	Sergio Crisostomo	0.34	2	0.17	1	0.00	1	0.00	1	0.51	5
Alipio Jorge	Sergio Crisostomo	0.28	2	0.14	1	0.00	1	0.00	1	0.42	5
Antonio Porto	Sergio Crisostomo	0.33	1	0.00	1	0.00	1	0.00	1	0.33	4
Sergio Crisostomo	Veronica Orvalho	0.25	1	0.00	1	0.00	1	0.00	1	0.25	4

Crisostomo who matches on the first two levels with almost every other expert, but with none (exception to Pedro Brandao, who shares a high similar profile with him) on the last two levels of the hierarchy. This indicates that from the level 2 of the topical hierarchy, there is a clear distinction of the communication network topics (his most specific google scholar interests).

Another case worth to note is the fact that although Veronica's interests are further away in comparison to the others, she still has some pairwise similarities with a total value greater than 1. A further look into her profile revealed that her expertise profile is scattered through several topics and her node in the HIN is never a highly ranked attribute-node of the topics. In our dataset, the computer graphics area does not have as many publications as other areas such as machine learning and parallel programming for example. As a result, HEPHIN fails to model the topic correctly and scatters its publications (or metadata) among other more predominant topics.

5.5.3 Hierarchical expertise profile applications

In this section we show some other use-cases of HEPHIN hierarchical expertise profiles that are not possible using traditional expertise profiling strategies. First, we show how HEPHIN can be used to study the evolution of the hierarchical expertise profile over time. Second, we show an application of HEPHIN expertise profiles to recommend experts for a peer-review system (i.e., an expertise finding problem). Finally, we show how the HEPHIN expertise profile can be converted into an image that summarises the expertise of the experts. For these applications we used all the Authenticus dataset which contains 21555 publications (in contrast to the previous experiment, this dataset consists of publications from multiple areas such as: mathematics, computer science and biology), 7846 authors, 59265 keywords and 230 ISI fields. Furthermore, the weight assigned to each metadata relation is the same as the ones used for T_3, i.e., 2.0 for the publication-keyword relation, 1.0 for the publication-author relation and 0.5 for the publication-ISI field relation.

Temporal profiles

While assessing the expertise of experts it is often useful to consider how it changed over time. For example, it is common for researchers to change between projects that are not necessarily in the same knowledge areas. Furthermore, researchers usually expand their expertise as their careers develop. Being able to track these changes not only provides an overview of the evolution of a person's expertise but also identifies shifts in the topics that they are interested over time. More concretely, an expert may have knowledge in topic *"A"* and *"B"* but he may have been more interested in topic *"A"* at the beginning of his career and is now more interested in topic *"B"*.

Temporal expertise profiles provide information for this analysis and some approaches have been proposed to create them [40, 48]. Still, they present the same drawbacks as traditional expertise profiling approaches (some are LDA-based and the profiles are generated over a flat line of topics). HEPHIN allows the user to create an expertise profile by describing the knowledge of a person in function of a set of attributes (e.g., keywords). Thus, it is possible to create a temporal analysis of the expertise profile by dividing the set of attributes that characterise the knowledge of an individual over different time windows. The general idea consists in creating an hierarchical expertise profile for each time window. Comparing these profiles provides an overview of the evolution of the knowledge and interests of the researcher over time. Next we show a

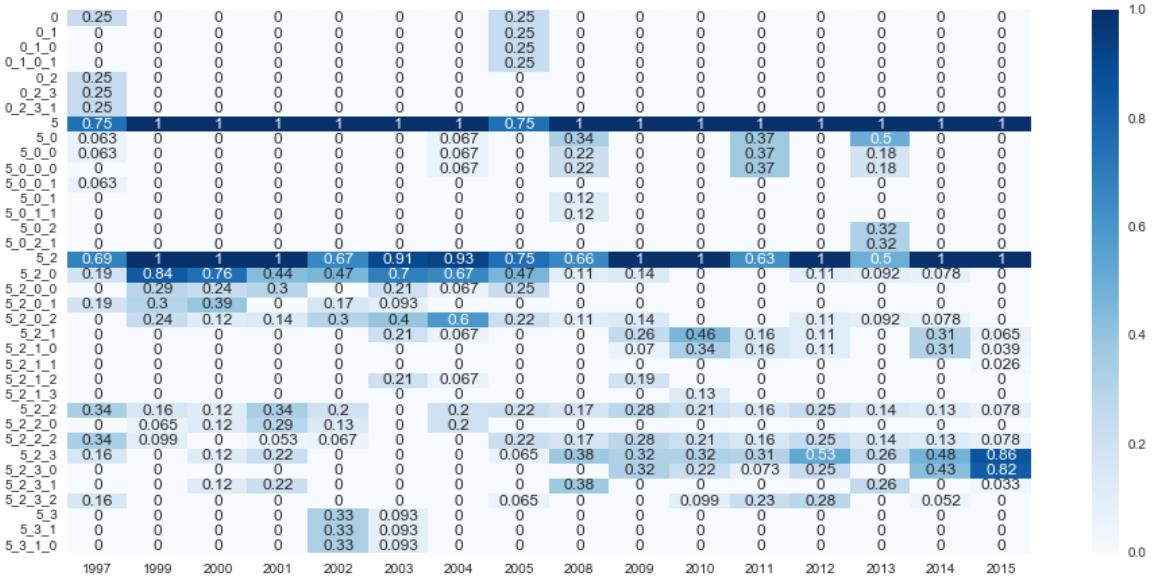


Figure 5.6: Temporal analysis of the expertise or interests of an expert using multiple hierarchical profiles obtained with HEPHIN.

concrete example.

For this example, consider that we want to analyse the evolution of knowledge and the change in interests of researcher A in the Authenticus database. A is an expert that is present in the topical hierarchy so we can create his expertise profile using the *"direct mapping"* strategy. However, this profile provide us with the snapshot of his expertise at the current time and does not provide information how it evolved over time. To analyse the temporal evolution of A's expertise, we consider all the keywords he was used over his authored publications. Then, we divide them according to the year of publication of the document. For each year that A was published some work, (i.e., A has keywords that characterise his expertise in that year) we create an hierarchical expertise profile using the keywords to describe his knowledge at that specific year. A has published documents in the following years: 1997, 1999 to 2005 and 2008 to 2015. As a result we have 16 hierarchical expertise profiles for these years. Figure 5.6 shows the expertise evolution and interests change of expert A over time. Each row in the figure represents a topic from the topical hierarchy (constructed using the same weights for the metadata relations as the topical hierarchy T_3 presented in Table 5.1) and each column represents the hierarchical expertise profile of A for that specific year. Furthermore, each cell represents the value of expertise of expert A in a specific topic in a certain year. The higher the value the more expertise or interest the expert had in that topic.

The topics shown in the left side of the figure use the *"_"* symbol to represent a level change in the hierarchy level. The number of *"_"* in a topic represent the hierarchy level of the topic. For example, the topic *"5"* is at the level 0 of the hierarchy (the most broad topics modelled), while *"5_2_1_1"* is topic at the level 3 of the hierarchy (the most specific topics). From the temporal analysis we observe that there is one broad topic (*"5"*) which remained constant throughout the career of the researcher *A*. This is expected since changes at a broad spectre of the topics are not expected in a researcher's career. More concretely, there are only a few cases where researchers change their knowledge or interests from a broad topic such as computer science to another one such as chemistry. Considering more specific topics, we observe that there were changes in the expertise or interest of the researcher over time. For example, in his early career the researcher had more interest in the topics *"5_2_0"*, *"5_2_0_0"*, *"5_2_0_1"* and *"5_2_0_2"*, but this interest has shifted to topics *"5_2_3"*, *"5_2_3_0"*, *"5_2_3_1"* and *"5_2_3_2"* after the middle of his career. The hierarchical expertise profiles compared to traditional expertise profiles offer the users the opportunity to observe changes in the interests of an expert at different granularity levels of the topics. For some applications it could be interesting to observe that at the most broad topics the expert's interest has remain constant, but for other applications it could be important to understand the reason for the change of interests from topic *"5_2_0"* to *"5_2_3"* at the middle point of this researcher's career.

Expertise finding

In the academia is it often necessary to nominate experts for peer-review tasks. The ideal scenario is that the expert nominated for the task has expertise about the topic discussed in the evaluation. For example, the reviewers of a publication about *"data mining"* should have expertise in this topic. In the literature, the problem is known as expertise finding and there are several approaches that have been proposed for the task (an extensive analysis of these approaches is behind the scope of this thesis). In this section, we show an application of the HEPHIN expertise profiles in order to tackle this problem. Note that our goal is not to show that expertise finding with HEPHIN is better than with other approaches. We are not able to test this hypothesis because we did not had access to ground-truth data for this experiment. Instead, our aim in this application is to show that the expertise profiles created with HEPHIN are useful in other expertise retrieval problems.

For the expertise finding application we manually obtained data from 10 Master's

thesis presented throughout 2015 and 2016 at the Computer Science department of the Faculty of Science University of Porto. We decided to use this data since most of the advisors are experts in the Authenticus database. The goal of our experiment is to create the expertise profile for the thesis and for all the experts in the Authenticus database and then, through profile comparison determine the experts whose expertise is the most adequate to be one of the juries of the thesis presentation (i.e., determine the experts with more expertise for the topics discussed in each thesis). To create the expertise profile for the thesis, we either obtained the keywords from the document or manually assigned keywords that, in our opinion, characterise the thesis topics. Table 5.6 presents the information obtained for each thesis with respect to the title, advisor, co-advisor, jury and keywords used of each thesis.

We obtain the hierarchical expertise profile for every thesis with the HEPHIN algorithm using the keywords shown in Table 5.6. We observed that all the thesis are exclusively assigned to the topics *"5"* at the broader level of the hierarchy (this is expected since they are all thesis from the computer science area). Using this information, we identified the experts that have expertise in topic *"5"* and defined them as the group of candidates to be jury of the thesis. In total, we have 1733 experts as candidates.

For each experts, we created their hierarchical expertise profile using the *direct mapping* strategy in HEPHIN. Furthermore, to determine the best candidate for each thesis we compare the thesis profiles against the ones from the 1733 experts. We based our comparison in three different criteria scores. Note that we are not going into too many details about these scores since this was an early experiment and the methods discussed here are preliminary in the sense that they need further testing. Again, our goal here is to unveil the promising possibilities with HEPHIN and its hierarchical expertise profiles. The three scores considered are:

- **Relevance.** The relevance score measures the divergence between the thesis expertise profile and the expertise profile of an expert. We use the Kullback-Leibler divergence metric to compare the two profiles at a certain hierarchical level. Then, we sum the values obtained for each level to obtain the final relevance score. The value of the relevance score in this experiment range between 0 and 4 with higher values indicating higher similarity between the profiles.

- **Trending.** The trending score measures whether the interest of the expert in the topic(s) of the thesis is increasing or decreasing. We use the temporal profile

Table 5.6: Master's thesis dataset used in the expertise finding application. Keywords are separated by ";".

	Title	Advisor	Co-Advisor	Jury	Keywords
1	Automatic Coherence Evaluation Applied to Topic Models	Alipio Jorge	NA	Rui Camacho	topic models; data mining; text mining; natural language processing; coherence
2	Implementacao e Avaliacao do Algoritmo MCTS-UCT para o jogo Chinese Checkers	Ines Dutra	NA	Pedro Mariano	Artificial Inteligence; game theory; Monte Carlo Tree Search (MCTS); Upper Confidence Bounds for Trees (UCT)
3	Controlo e Ocultacao de dados pessoais em dispositivos moveis	Manuel Correia	NA	Andre Zuquete	Android; privacy; SECURITY; Security and privacy
4	Recommender System for an e-learning platform	Alipio Jorge	José Leal	Carlos Soares	Technology Enhanced Learning; recommendation systems; e-learning; Collaborative filtering
5	Clustering de relacionamentos entre entidades nomeadas em textos com base no contexto	Alipio Jorge	Maria Rocha	Nuno Escudeiro	Named-Entity Recognition; Context Extraction; clustering; similarity measures; text mining
6	Domain Oriented Bi-clustering Validation	Luis Torgo	Catarina Magalhaes	Paulo Azevedo	biclustering; Clustering validation; clustering;
7	Biometrics on Mobile Devices Using the Heartbeat	Luis Antunes	Manuel Correia	Mario Antunes	supervised learning;Heart rate variability (HRV); Biometric system; Machine Learning
8	Lexicon Expansion System for Domain and Time Oriented Sentiment Analysis	Luis Torgo	Alvaro Figueira	Ricardo Campos	Sentiment Analysis; lexicon expansion; text mining
9	Large Scale Parallel Subgraph Search	Pedro Ribeiro	NA	Herve Paulino	Subgraphs;MapReduce; g-tries; parallel programming
10	A Parallel Feature Hybrid Approach for Feature Selection	Ana Aguiar	Fernando Silva	Herve Paulino	feature selection; parallel programming; scalability; supervised learning

analysis of experts to estimate a multiplier m for the expertise of the expert with respect to the topic(s) of the thesis. For this experiment we considered a 5 year time window and defined that $m_t = 2$ if the interest of the expert in topic t is increasing, $m_t = 1$ if it is constant and $m_t = 0.5$ if it is decreasing. Then, we measure the intersection between the profile of the thesis and the profile of an expert. For each topic that they have in common (i.e., both have expertise about that topic) we use the multipliers to increase or decrease the similarity between the profiles. The value of the trending score ranges between 0 (the author has no ares of expertise in common with the thesis profile) and 8 (the authors has exactly the same profile as the thesis and his interest in these topics is increasing). Note that obtaining a score of 8 in the trending score is highly unlikely.

- **Authority.** The authority score measures the importance of the expert in the topic(s) of the thesis. For the topics of the thesis, we measure the importance of an expert in a topic by applying the PageRank algorithm to the sub-graph that represents the topic. Then, we normalise the PageRank values for the range between 0 and 2, define them as multipliers and apply them in a similar strategy as the one presented for the trending score. More concretely, for each topic in common between the profile of the thesis and the profile of the expert, we multiple the intersection value by the normalised PageRank score (between 0 and 2) of the expert in the topic. The value of the authority score ranges between 0 and 8 with higher values indicating that the expert has more expertise in the topics of the thesis and he is also a more important expert in those topics.

We determinate the best candidate for the jury using the expertise score which is the sum of the relevance, trending and authority scores. Therefore, the values of the expertise score range between 0 and 20 (4 + 8 + 8). Again, higher values indicate that the expert is a better candidate for the thesis jury according to our scores.

Table 5.7 presents the candidates with the highest expertise score for each thesis. For results discussion we also include in the table the scores for relevance, trending and authority. A first look into the table reveals that some candidates (Paulo Novais, Joel Rodrigues and Luis Reis) were selected as the best candidate for more than one thesis. Although we have 1733 candidates for the thesis, in some cases selecting the same candidate for more than one thesis is expected. For example, Paulo Novais is the best candidate for thesis 1 and 7 which both address the topic of machine learning. Another example is Luis Reis who is selected for thesis 4, 5 and 8 which are strongly

related to the text-mining topic. However, the candidate Joel Rodrigues is selected for thesis number 2, 3, 6 and 9 which have weakly related topics such as: game-theory, security, clustering and graph-theory. A further analysis of this case revealed that some of these thesis topics were underrepresented in the number of publications in the dataset therefore they are not correctly modelled in the topical hierarchy. For this reason the expertise profile of these thesis have more topics in common than expected (i.e., the thesis profiles are more similar) and Joel Rodrigues is the best expert in those common topics.

With respect to the three defined scores (relevance, trending and authority), the authority score presented the highest variance between the 1733 candidates and it was often the highest score for all the thesis. We observed that the low variance in the candidate selection for the thesis was mostly caused by the higher impact of the authority score compared to the other ones. For this reason we repeated the candidate selection for each thesis but this time we removed the authority score. Table 5.8 shows the selected candidates considering three different scores: relevance, trending and relevance + trending. As expected the results show a large variety of candidates selected. In fact, the best candidates according to any of the three scores (relevance, trending and relevance + trending) are rarely the best ones in the expertise score. This results shows that adjusting the authority score may be necessary in order to improve the expertise finding results. Nevertheless, this experiment is a small one and further testing is necessary.

For the presented results it is difficult to estimate the quality of the candidates selected. With some case-by-case analysis we observe that the publications of the

Table 5.7: Candidates with the highest expertise score for each thesis. The thesis column identifies the thesis from Table 5.6.

Thesis	Name	Relevance	Trending	Authority	Expertise
1	Paulo Novais	2.370	2.614	3.224	8.208
2	Joel Rodrigues	1.388	2.286	2.456	6.130
3	Joel Rodrigues	1.735	2.603	4.188	8.525
4	Luis Reis	1.622	2.299	2.823	6.744
5	Luis Reis	1.889	2.643	2.708	7.240
6	Joel Rodrigues	1.714	2.351	3.957	8.022
7	Paulo Novais	2.502	2.678	3.208	8.388
8	Luis Reis	1.891	2.780	3.806	8.477
9	Joel Rodrigues	2.145	2.321	3.294	7.760
10	Joao Cardoso	3.826	2.887	1.589	8.302

Table 5.8: Candidates with the highest relevance, trending and relevance + trending scores for each thesis. The thesis column identifies the thesis from Table 5.6.

	Relevance		Trending		Relevance + Trending	
Thesis	**Candidate**	**Score**	**Candidate**	**Score**	**Candidate**	**Score**
1	Paulo Novais	2.370	Joao Cordeiro	5.402	Joao Codeiro	5.902
2	Janio Monteiro	2.711	Carlos Oliveira	3.712	Janio Monteiro	5.490
3	Jose Tribolet	2.830	Joao Moutinho	4.850	Jose Tribolet	6.957
4	Luis Reis	1.622	Sergio Nunes	3.846	Zafeiris Kokkinogenis	4.786
5	Joao Sobral	2.709	Maria Antunes	4.988	Ricardo Goncalves	6.025
6	Ricardo Goncalves	2.246	Eduardo Ferme	5.212	Ricardo Goncalves	6.351
7	Joao Gama	2.720	Jorge Silva	6.374	Jorge Silva	6.874
8	Pedro Furtado	2.182	Jorge Silva	6.236	Jorge Silva	6.736
9	Joao Cardoso	2.563	Sergio Nunes	4.465	Ricardo Goncalves	5.929
10	Joao Cardoso	3.826	Eduardo Ferme	4.832	Joao Cardoso	6.713

best candidates match the topics discussed in the thesis for most cases. However, we do not have enough validated data or metrics to convert this interpretation to a numerical value. To overcome this limitation and provide the reader with some more results we present Table 5.9 which shows the scores (relevance, trending, authority and expertise) for the jury members that were selected for each thesis as well as their rank with respect to the 1733 candidates. Two of the experts selected as jury did not have any publication in the Authenticus database. As a result we could not create their hierarchical expertise profile (even considering the *indirect mapping* strategy of HEPHIN we did not have enough evidence to characterise the expert's knowledge) and their score cannot be estimated. For the remaining 8 juries their average rank position is 127 for the expertise score. In 5 out the 8 cases the jury rank is lower than 50, and in the best case (Carlos Soares) the jury ranked at the 16th position. In the

Table 5.9: The scores and rankings for relevance, trending, authority and expertise for the jury members of each thesis. Two thesis from Table 5.6 were not considered due to the lack of information about their jury.

	Relevance		Trending		Authority		Expertise	
Jury	**Score**	**Rank**	**Score**	**Rank**	**Score**	**Rank**	**Score**	**Rank**
Carlos Soares	1.281	60	2.740	86	0.681	37	4.703	16
Ricardo Campos	0.500	65	4.575	13	0.229	181	5.304	23
Herve Paulino	1.461	37	2.992	93	0.325	158	4.778	33
Mario Antunes	1.410	20	2.846	138	0.357	105	4.612	46
Herve Paulino	1.457	40	2.697	190	0.177	298	4.331	80
Andre Zuquete	1.500	21	2.990	183	0.107	446	4.597	161
Rui Camacho	0.775	112	2.494	224	0.075	571	3.344	199
Paulo Azevedo	0.500	64	2.246	414	0.031	1080	2.777	460

worst case (Paulo Azevedo) ranked at position 460th mostly due to his low relevance and authority scores.

With respect to the individual scores (relevance, trending and authority) there are some interesting points to highlight. The results show that the relevance score is the one that on average the juries are better placed in terms of rank. This indicates that the most important fact for a jury selection is his knowledge about the thesis topic. Moreover, we observe that the juries are often not the authors with highest importance according to the authority score. This result is expected because the juries of Master's thesis are often young researchers whose career is not as long as some other researchers that rank higher according to the authority score (usually the longer the career of an author the more publication he was and consequently, the more important he is for the network). Regarding the trending score we observe that the juries present high scores however their rank is on average much lower compared to the relevance score. A further analysis revealed that the juries trending score on the topics of the thesis is constant mostly because of their young careers. According to the trending score definition, the multiplier is 2 when the interest of the author has increased in recent years. Since most of the juries showed interest in these topics since the beginning of their career their multiplier is 1 and they are at disadvantage compared to other authors with longer careers that started showing interest in other areas and are now shifting towards the topics discussed on these thesis. In general we observe that the relevance score seems adequate for the expertise finding application, while the trending and authority scores need some more tuning.

Profile visualisation

Being able to quickly describe the expertise and interests of an expert is one of the many goals of expertise profiling. Bibliographic databases often use this information to provide their users with valuable information about the experts they search. For example, the Google Scholar page of a researcher contains his research areas so users can quickly understand the researcher's background without having to look into his publications. Some bibliographic databases (e.g., AMiner [1]) already uses expertise profiles to describe an expert. However, these still are generated over flat topics thus they some times fail to provide the user with enough details about an expert's expertise. The hierarchical expertise profiles offer a full-detailed view of the expert's knowledge, starting at the most broad areas and ending at the most specific ones. As a result, hierarchical expertise profiles (more concretely, the ones generated with HEP-

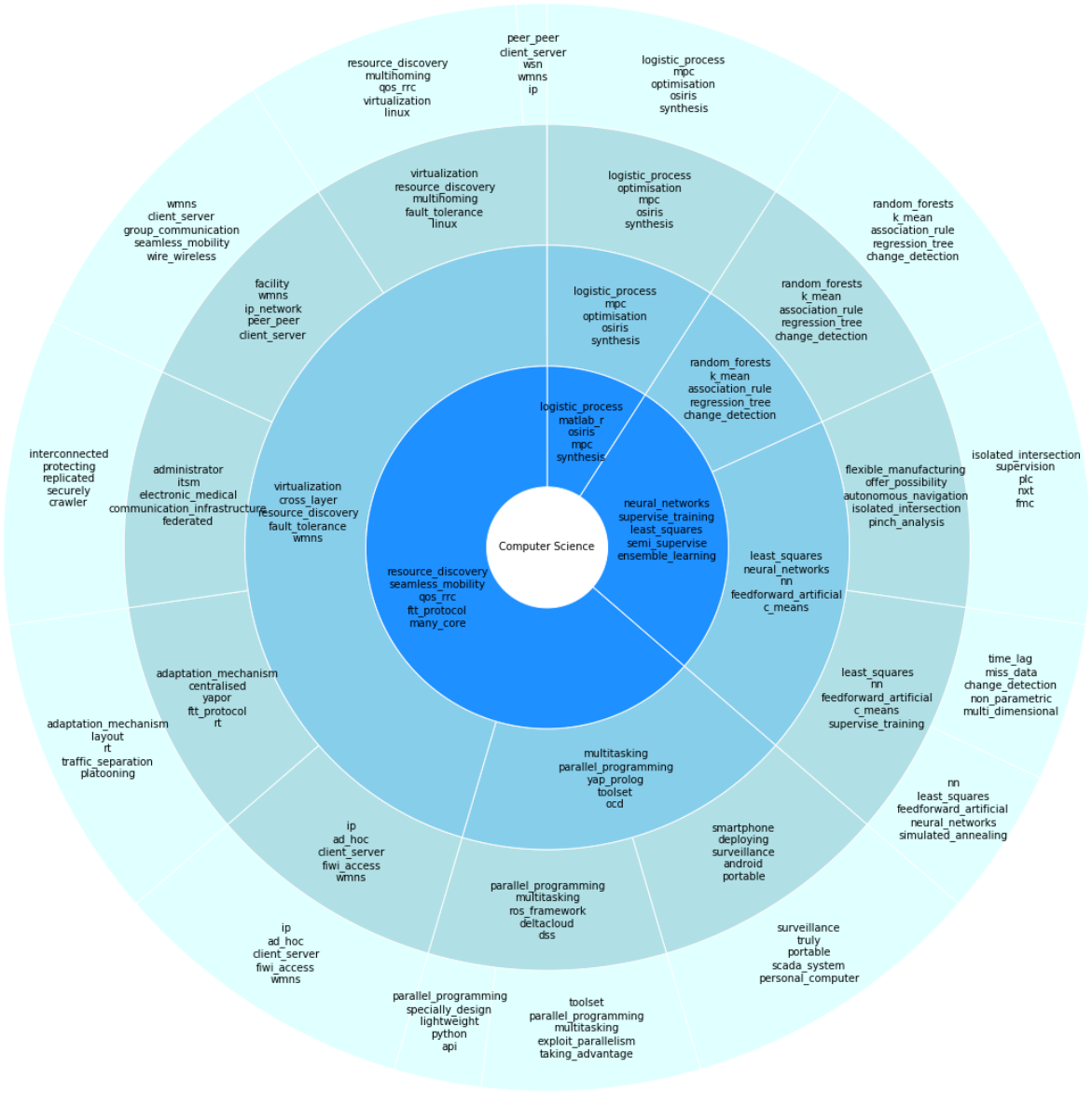


Figure 5.7: Profile visualisation of an hierarchical profile. Labels for each topic were obtained considering the top-5 PageRank keywords of each topic.

HIN) are an improvement over the current strategies used in bibliographic databases. Our goal in this section is two-fold. First, we present a straightforward strategy to label the topics discovered by HEPHIN. Second, we define an elegant strategy to visually represent the hierarchical expertise profiles.

The topics discovered by HEPHIN consist of multiple attributes. In the use-case presented with the Authenticus database, each topic contains a list of authors, keywords and ISI fields. Among these attributes, keywords are the ones that provide the most valuable information to describe a topic. Additionally, each topic is a sub-graph of the HIN. Therefore, we can use a centrality metric to estimate the importance of each attribute with respect to a topic. Using these two facts, we label each topic in the topical hierarchy as the 5 keywords with the highest PageRank value in the topic. With respect to profile visualisation we use a circular graphic where the inner circle represents the knowledge of an expert with respect to broad topics, and as the circles move away from the centre the specificity of the topics increases. Additionally, the areas within each circle are proportional to the knowledge of the expert with respect to that topic. For example, if at a certain hierarchy level, an expert has twice more knowledge about topic "A" than "B", then the area of topic "A" in the circular graphic is the double of the one from "B". Figure 5.7 presents an example of the profile visualisation using the top-5 keywords to label each topic.

The circular graphic presents an elegant solution to show the areas of expertise of an expert at multiple topic granularities. Therefore, solving the problem of not providing a detailed understanding of an expert's knowledge. Furthermore, the graphic shows how the interests of an expert are divided in quantity. Thus, if the expertise of an expert is more focused towards a specific topic we can visualise this information. With respect to the labelling strategy, we observe that the strategy is adequate to label specific topics (i.e., the outer circles). However, it fails to label the most broad ones with more broad terms. For example, the keyword "neural networks" is good to describe topics at the most specific level of the hierarchy but is not adequate to describe a broader topic. Therefore, the labelling step should use different strategies to label topics at different levels of the hierarchy.

5.6 Summary

In this chapter, we presented HEPHIN a new expertise profiling strategy that analysis metadata relations between publications to create a multi-typed topical hierarchy, and

then maps the knowledge of experts into this structure to obtain an hierarchical expertise profile. HEPHIN overcomes two limitations that are present in most expertise profiling algorithms. First, HEPHIN does not require the user to define the number of topics in the dataset. HEPHIN uses community detection algorithms to automatically estimate the correct number of topics from the data. Second, HEPHIN builds an hierarchical expertise profile that presents the knowledge of experts at multiple granularity levels. Thus, the problem of constructing profiles that are redundant and either too specific or too broad is solved. Furthermore, we also presented a *"indirect mapping"* strategy for HEPHIN which allow us to build expertise profiles for multiple entities in function of their attributes. This feature is important to understand the areas of knowledge of a thesis or an institution, for example.

In our experiments, we use data from the Authenticus database and we used the following metadata relations in the HIN: publication-author, publication-keyword and publication-ISI fields. We evaluated the coherency of the topics discovered by HEPHIN and we obtained the best results when HEPHIN weights are 2.0 for the publication-keyword relation, 1.0 for the publication-author relation and 0.5 for the publication-ISI fields relation. We also evaluated the expertise profiles constructed by HEPHIN in a task of comparing profiles. We gathered data from 12 experts from the Computer Science area from the Authenticus database, we grouped them according to their interests (these were manually added by the experts) and we compared the 12 profiles constructed using HEPHIN for these experts. We observed that HEPHIN profiles were more similar among experts with the same expertise. Furthermore, this experiment showed the importance of having hierarchical expertise profiles since we were able to observe that the profiles were similar at the broad topics (which is expected since all the experts are from the Computer Science field) but then at the more specific topics the HEPHIN profile clearly was able to distinguish their interests. Finally, we presented several applications for the HEPHIN profiles, namely temporal profiles, expertise finding and profile visualisation. These applications are not necessarily experiments since they lack a solid evaluation strategy. Instead, we present them to show the advantages and potential of using hierarchical expertise profiles (in particular the ones generated by HEPHIN) over traditional profiles that are generated from a flat line of topics.

Chapter 6

Conclusions and future work

The word bibliometrics refers to quantitative methods that use statistical analysis of scientific literature to track the output and impact of research. Bibliometrics allow users to better understand science and are often used to aid decision making in research and development funding. Therefore, bibliometrics are fundamental for the evolution of science. The goal of this thesis was to contribute to the creation of a complete tool to measure the scientific impact of researchers. To achieve this, we tackled three bibliometric problems, namely author ranking, publications clustering and expertise profiling. We implemented our algorithms in Python and the source code is available at [119].

This final chapter presents the main contributions of our work, discusses the limitations of our methods and proposes directors for future researcher. Furthermore, it also presents a conceptual design of a framework that uses all the developed methods to solve the problem of measuring the scientific impact of researchers. Finally, it presents concluding remarks.

6.1 Main contributions

This thesis aims to offer contributions towards the creation of a complete tool to measure scientific impact. Our envisioned solution required tackling three problems in the bibliometrics area. As a result, the work described in this thesis consists of the design, implementation and evaluation of bibliometric algorithms. In more detail, we propose methods for the author ranking and expertise profiling problems, and a publication similarity measure which we use in the context of publications clustering. The developed methods aim for flexibility with respect to (i) the data required by the algorithms (e.g., our publication similarity measure and expertise profiling strategies allow the users to define the data that is used as input) and (ii) the criteria used in the algorithms (e.g., the author ranking algorithms allow the user to define different criteria to consider while measuring the author's scientific impact). Furthermore, our methods aim to be more accurate than similar state-of-the-art approaches when evaluated in real-world datasets. Next, we provide a more detailed description of our contributions.

OTARIOS. We present a PageRank-based measure for author ranking named OTARIOS. Previous methods for author ranking assume that the citation network is complete and fail to efficiently combine the features in the network. OTARIOS divides the citation network in insiders (i.e., authors whose citation information is known) and outsiders (i.e., authors whose citation information is unknown) and proposes different strategies to measure the PageRank score of authors in each group. Thus, OTARIOS efficiently deals with the problem of missing information in citation networks. OTARIOS also allows the user to define the evaluation criteria through the combination of different features in citation networks. In total, 511 different criteria can be used in OTARIOS. We compare OTARIOS against 9 other state-of-the-art approaches and we observed that OTARIOS is consistently more accurate than the other approaches in predicting an author ranking that is more similar to one constructed based on human judgement.

FOCAS. We propose a penalty system for citation networks named FOCAS. Author ranking algorithms are not able to couple with the problem of citation boosting that leads to undeserved scientific impact. FOCAS aims to decrease the impact of citation boosting patterns in citation networks and promote fairer author ranking systems. FOCAS uses the co-citation and citation networks to infer penalties for *friendly* citations. FOCAS does not produce an author ranking by itself, instead it is integrated with any author-level author ranking algorithm. We compared the performance of 8

author-level author ranking algorithms with and without the FOCAS algorithm. Our results showed that FOCAS consistently improves the produced author rankings.

PURE-SIM. We present a publication similarity measure named PURE-SIM. Previous methods rely on references or textual evidence to measure publications similarities and disregard metadata attributes such as authors, keywords and journals. Considering these additional metadata attributes is important because it is often the case where publications in bibliographic databases do not contain references or textual information. Thus, current approaches are not able to estimate the similarities of these publications. PURE-SIM efficiently combines metadata attributes such as authors, references and journals. As a result, PURE-SIM is capable of estimating the publications similarities for all the publications in bibliographic database (since information such as authors and journals is available for all publications). We compared PURE-SIM against 11 state-of-the-art approaches that estimate publications similarities in the context of the publications clustering problems. Our results show that PURE-SIM similarities lead to more accurate clusters .

HEPHIN. We propose an expertise profiling strategy named HEPHIN. Previous methods for expertise profile create profiles over a flat line of topics which often generate profiles that are redundant or either too specific or too general. HEPHIN constructs a multi-typed topical hierarchy and maps the knowledge of a person into the hierarchy. As a result, HEPHIN creates hierarchical expertise profiles which describe the knowledge of experts at different granularity levels. HEPHIN is also capable of creating expertise profiles for other entities such as publications, thesis or institutions which increases the number of potential applications for HEPHIN. We evaluated HEPHIN in a real-world bibliographic database. Our results show that HEPHIN is capable of discovering coherent topics (in the process of creating the multi-typed topical hierarchy) and creating accurate expertise profiles. Furthermore, we presented some applications in which the HEPHIN expertise profiles can be used in the tasks of expert recommendation, temporal profile analysis and profile summarisation.

6.2 Future work

We hope that the work developed in this thesis can lead to future research in the bibliometrics area. We presented methods for three different bibliometric problems. In this section we give some possible directions for future research. These ideas are divided in two parts. First, we discuss limitations and present future work for the

developed methods. Then, we present our conceptual design to create a complete tool to measure scientific impact using the developed methods.

6.2.1 Research directions for the proposed methods

Here, we present the research directors for the methods proposed for the problems of author ranking, publications clustering and expertise profiling.

Automatically identify outsiders (OTARIOS). The OTARIOS algorithm presented in this thesis defines as an outsider any author whose received citations are unknown in the citation network. However, there are authors in the citation network whose received citations are only partially known and perhaps should also be considered outsiders. A possible direction for future work consists of using network science methods to identify outsiders (e.g., insiders with low density in the citation network, or insiders with low co-authorship ration to other insiders).

Penalise reciprocal citations (FOCAS). Several studies have shown that the abuse of reciprocated citation between groups of authors leads to undeserved scientific merit and the problem is not handled by author ranking algorithms. A possible direction for future work is to add penalties to reciprocated citations in FOCAS.

Penalties threshold (FOCAS). FOCAS penalises all the *friendly* citations in the citation network independent of the number of *friendly* citations that an author has. A possible direction for future research is to analyse the citation and co-authorship network to determine a *normal* number of *friendly* citations to receive and only penalise (or penalise more) authors that exceed that number.

Metadata weight definition (PURE-SIM). In our research, the current user-defined metadata parameter of PURE-SIM presents two possible values (0 or 1) which gives the user the choice of using or not a certain metadata relation. A possible research direction is to change this parameter to a continuous interval ($[0, 1]$) where the users can assign higher importance to certain metadata relations. For example, defining the author-publication relation with weight 0.5 and the keyword-publication relation with weight 1.0 results in doubling the importance of the latter.

Estimate the similarities of other metadata elements (PURE-SIM). PURE-SIM estimates publications similarities because the publications are defined as the star-nodes in the HIN. An interesting idea to explore is to evaluate if PURE-SIM is also suitable to measure the similarity between metadata elements such as authors,

keywords and journals. One straightforward application would be to define the journals as the star-nodes in the HIN and use PURE-SIM to identify the journals that are more similar to each other (i.e., the journals that address the same topics).

Labelling techniques for the topics (HEPHIN). The preliminary strategy used to automatic label the topics discovered by the HEPHIN algorithm fails to produce good labels for topics at the higher levels of the hierarchy. A possible research direction is to use the multi-typed topical hierarchy to consider different attributes and ranking strategies to label topics at different levels of the hierarchy. Thus, it would be possible to have broader terms for higher topics in the hierarchy and more specific terms for the lower topics in the hierarchy.

Expert recommendation with hierarchical expertise profiles (HEPHIN). We presented a preliminary strategy to use the expertise profiles created by HEPHIN to recommend experts to be part of a jury of several thesis (i.e., a peer-review process). We believe that the ability of creating expertise profiles for researchers, documents and journals is one of the most promising features of HEPHIN. In our preliminary experiments we lacked a good evaluation strategy to properly evaluate the results and make the necessary adjustments to being able to recommend experts using HEPHIN expertise profiles. An interesting research direction is to further test this application and compare it with other expert recommendation strategies.

Improved topic modelling (PURE-SIM + HEPHIN). The HEPHIN algorithm allows the users to easily change the clustering process that leads to the creating of the multi-typed topical hierarchy. The publications similarities obtained by PURE-SIM can be used to construct the similarity graph in the HEPHIN workflow and potentially improve the topics modelled.

Expertise finding (HEPHIN + OTARIOS). The expertise finding task consists in identifying persons that have knowledge about a certain topic. Ideally, the outcome of the expertise finding algorithm is a ranking of the persons that have the most knowledge about the topic. A future researcher direction is to use the HEPHIN expertise profiles to determine the experts about a topic (similarly to the expert recommendation application presented) and then use the OTARIOS algorithm to rank the authors within this topic in order to distinguish them with respect to their scientific impact (here we assume that authors with more scientific impact also have more knowledge about topic). Thus, presenting a solution to the expertise finding problem.

6.2.2 A framework to measure scientific impact

The goal of this thesis is to contribute to the creation of a tool to measure scientific impact. Here, we present a conceptual design of this tool that details how each one of the proposed algorithms in this thesis can be used to tackle the problem of measuring scientific impact. Furthermore, we specify the parts of this tool that need to be developed.

Figure 6.1 presents the workflow of our envisioned tool. The process starts with the PURE-SIM algorithm analysing the publications metadata to estimate the publications similarities. These similarities are then fed to the HEPHIN algorithm along with the author's publications which creates the authors expertise profiles. The publications similarities are also fed to a clustering algorithm that creates the publications clusters. The expertise profiles along with the publications clusters are fed to the *component 1* that uses this information to determine different author contributions for each publication. After this point the FOCAS algorithm analysis the co-author network to estimate the citations penalties. The citations penalties as well as the different author contributions are then fed to *component 2* which uses this information to create a weighted citation information. This citation information is fed to the OTARIOS algorithm which produces the author rankings. Finally, the author rankings as well as the publications clusters are fed to *component 3* which produces the final author scientific impact.

The *component 1*, *component 2* and *component 3* parts have not been developed in this thesis and are presented here as future work. Now, we describe in more detail these parts. The *component 1* is used in our tool to discriminate the contributions of the authors in the same publication. Our envisioned solution for this task consists in using temporal analysis of the HEPHIN expertise profiles and the publications clusters to determine the authors with more knowledge about the topics of a publication at the time of publishing. The general idea is to give more credit to the authors of a publication that have more expertise about its topics. Note that temporal analysis of the expertise profiles is essential for this task, since the expertise of an author about a topic changes over time. The *component 2* is used in our tool to combine the output of the different author contributions with the citation penalties. The goal of this part is to create a strategy that constructs an author-level citation network where authors that contribute with more expertise to their publication and that have fewer citations with penalties have in-coming citation links with higher weights in the network. Finally, the *component 3* is used in our tool to normalise the produced author rankings and

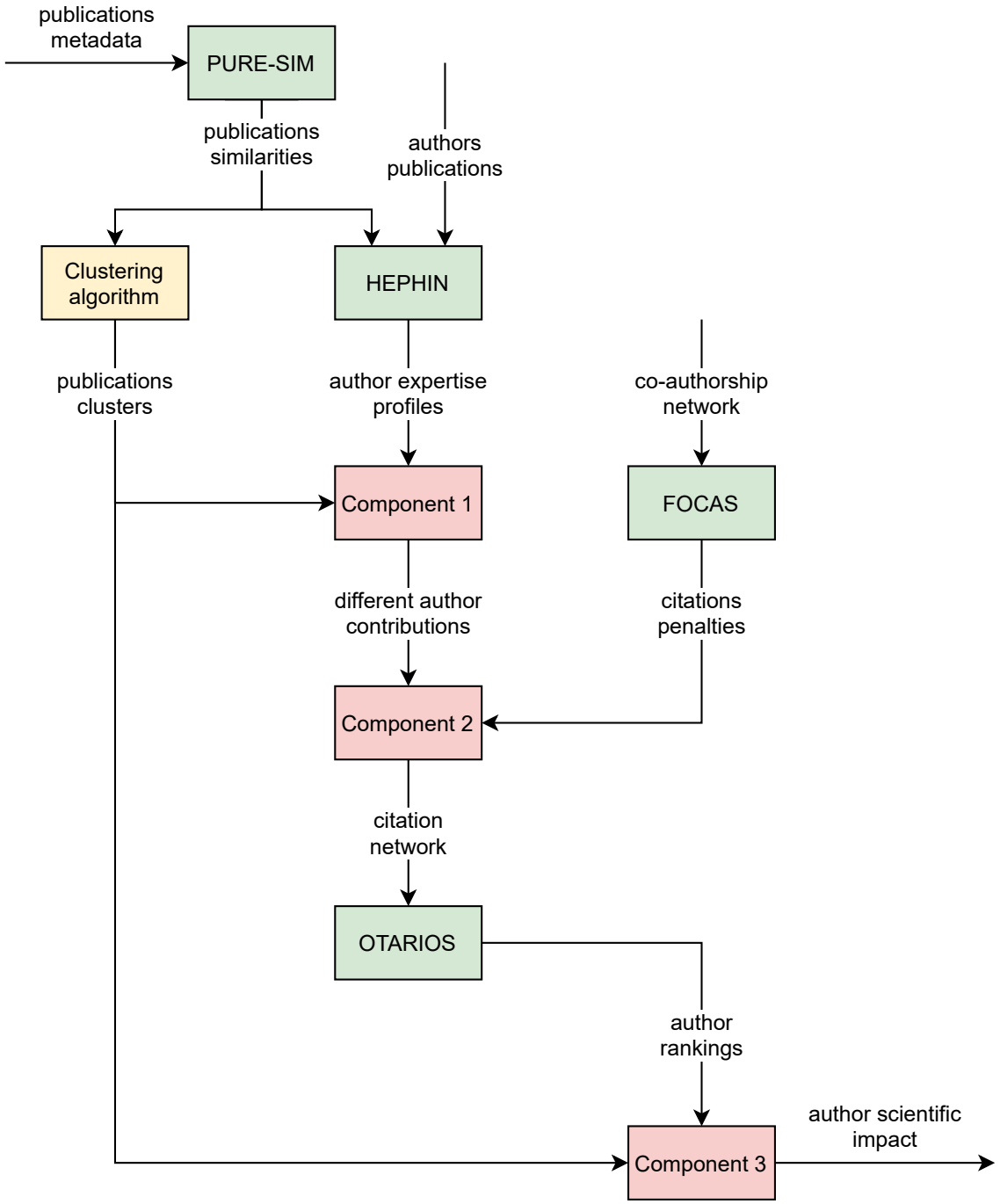


Figure 6.1: The conceptual design of a tool to measure scientific impact. The green rectangles represent the methods developed in this thesis, the red rectangles represent the components that still need to be implemented, and the yellow rectangle represents using an algorithm from the literature.

make them comparable across different research areas. Our envisioned solution for this task uses the produced rankings from the OTARIOS algorithm and the publications clusters to separate the authors rankings according to their research area. Then, a normalisation strategy is utilised to adjust the produced rankings for each research area and to generate a new author ranking that allows users to compare the scientific impact of authors in different research areas.

6.3 Closing remarks

When this dissertation started more than four years ago, its theme was the very broad topic of bibliometrics. In the early stages we focused on the problem of expertise profiling which is a very broad topic. We created what we believe to be a promising algorithm to create expertise profiles with several potential applications. Due to the lack of ground-truth data and the emergence of a new challenge proposed by a workshop invitation, we shifted our focus to author ranking which is a different problem but one which we envisioned with potential to be used along the expertise profiling strategy developed. We presented two novel methods with more accurate results compared to state-of-the-art approaches. At the final year of this thesis, a collaboration opportunity with the Centre for Science and Technology Studies emerged and we took the opportunity to develop some work in the area of measuring publications similarity. Although, again, this is a different problem we think that the developed studies are important for future work about our expertise profiling strategies. In the end of this thesis, we ended up working in three different problems of bibliometrics but our selection of problems was always made with the end goal of creating systems that integrate the methods developed with each other. Unfortunately, we did not have time to develop all the systems that we envisioned and the systems had to be presented in this thesis as future research directions.

At the beginning of this thesis our understanding of the bibliometrics field was very limited and researching on this topic was a very interesting adventure. We studied several problems of the current state-of-the-art, we presented methods to overcome these problems and tested our methods in real-world data. We hope that our tools are used in future bibliometric studies. One of the most interesting aspects of working in the bibliometrics field is that most of the researchers have different and strong opinions about topics such as author ranking. This lead to very interesting and productive discussions during conferences, workshops and presentations. Finally, we also hope that some of the opportunities for research are followed through by other people.

Appendix A

Appendix

A.1 MeSH similarities

In this section we describe the process used to obtain the ground-truth similarity for the MEDLINE publications. This methodology was originally proposed in the following study [37].

The MEDLINE publications contain the Medical Subject Headings (MeSH) tags which are assigned by experts to every publication to categorise them according to their topic. There MeSH tags are divided in more than 28,000 descriptors and almost 80 subheadings. The descriptors are terms often used in MEDLINE publications. The descriptors are organised hierarchical in a way that broader terms are at the top of the hierarchy. Consequently, descriptors may have descendants which capture the nature of sub-topic of relations. Regarding the association of descriptors to publications, every publication contains major descriptors (if the descriptor corresponds to a highly relevant topic of the publication) and minor descriptors (if the descriptor represents a topic that is marginally addressed in the publication). With respect to the subheadings, there are human defined words that represent a topic. Furthermore, subheadings are used to classify a descriptor with respect to a publication (i.e., the same descriptor may refer to different topics depending on the subheading associated to it). Thus, descriptors are usually indexed with one or more subheadings. For more information regarding the MeSH tags please refer to [120].

APPENDIX A. APPENDIX

In order to compute the MeSH similarities we follow the methodology presented in [37]. This methodology is divided in two parts: embedding scheme and similarities estimation. The embedding scheme represents the process to model the MeSH tags into a machine readable format. The similarities estimation is the task of using the resulting embedding to identify the similarities between MEDLINE publications. For the embedding scheme, we start by calculating the information content IC of each descriptor ($desc_i$) using the following formula:

$$IC(desc_i) = -log(P(desc_i)) \tag{A.1}$$

where

$$P(desc_i) = \frac{freq(desc_i) + \sum\limits_{d \in descendants(desc_i)} freq(d)}{\sum\limits_{k=1}^{s} \left(freq(desc_k) + \sum\limits_{d \in descendants(desc_k)} freq(d) \right)} \tag{A.2}$$

where $descendants(desc_i)$ is the set of descriptors that are children, direct or indirect, to descriptor i, $freq(desc_i)$ is the frequency of the descriptor i in the dataset, and s is the set of all unique descriptors. Then, we represent each publication as a vector of length $s + (s \times m)$ where s and m are the total number of unique MeSH descriptors and subheadings. The vector position for the ith descriptor is given by $(m+1) \times i - m$ and the corresponding weight for a publication p ($\omega_i(p)$) is defined as:

$$\omega_i(p) = \begin{cases} 0, & \text{if } desc_i \text{ is absent in } p \\ IC(desc_i), & \text{if } desc_i \text{ is a minor descriptor in } p \\ IC(desc_i) \times 2, & \text{if } desc_i \text{ is a major descriptor in } p \end{cases} \tag{A.3}$$

The vector position for the jth subheading with respect to the ith descriptor is given by $(m+1) \times i - m + j$ and the corresponding weight for publication p (ω_{ji}) is defined as:

$$\omega_{ji}(p) = \begin{cases} 1, & \text{if subheading } j \text{ and descriptor } i \text{ are present in } p \\ 0, & \text{otherwise} \end{cases} \tag{A.4}$$

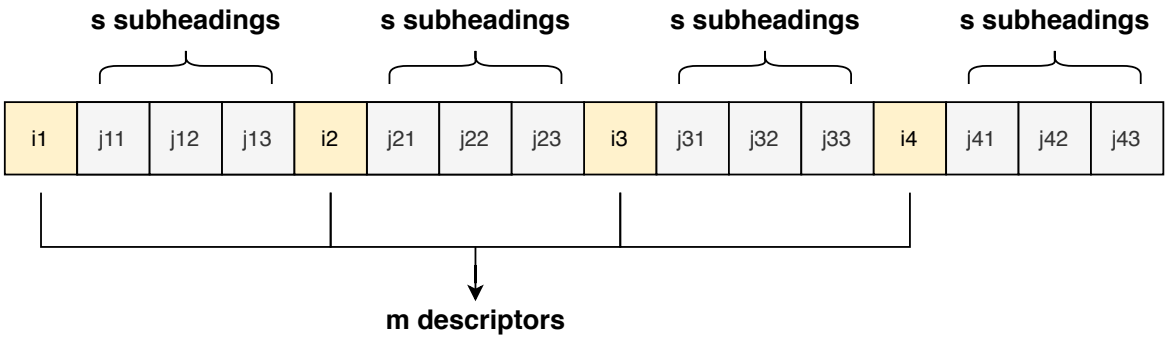


Figure A.1: Example of a vector for a publication when there are 4 descriptors (m=4) and 3 subheadings (s=3).

Figure A.1 illustrates the scheme of the resulting vector for each publication. Note that the scheme presents a scenario with a reduced number of MeSH descriptors and subheadings.

For the process of estimating similarities we apply the cosine similarity between the publications vectors obtained from the embedding scheme. Due to computational limitations, we apply the k-nearest neighbours technique with $k = 20$. This means that only the top 20 similarities for each publication are considered. Finally, we normalise the total similarity of the publications (i.e., the total similarities of a publication is 1) and guarantee that the similarities are symmetrical by using the following formula:

$$s'_{ij} = s_{ij} + s_{ji} \tag{A.5}$$

References

[1] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Citation Network Dataset." `https://aminer.org/citation`, 2017. Accessed: 14-09-2018.

[2] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *KDD'08*, pp. 990–998, 2008.

[3] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2016.

[4] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 487–494, AUAI Press, 2004.

[5] D. Duan, Y. Li, R. Li, Z. Lu, and A. Wen, "Mei: Mutual enhanced infinite community–topic model for analyzing text-augmented social networks," *The Computer Journal*, vol. 56, no. 3, pp. 336–354, 2012.

[6] A. Pritchard *et al.*, "Statistical bibliography or bibliometrics," *Journal of documentation*, vol. 25, no. 4, pp. 348–349, 1969.

[7] V. V. Nalimov and Z. M. Mulchenko, "Naukometriya. izuchenie nauki kak informatsionnogo protsessa (scientometrics. the study of science as an information process).," *Moscow, Russia*, 1969.

[8] A. J. Lotka, "The frequency distribution of scientific productivity," *Journal of the Washington academy of sciences*, vol. 16, no. 12, pp. 317–323, 1926.

[9] S. C. Bradford, "Sources of information on specific subjects," *Engineering*, vol. 137, pp. 85–86, 1934.

REFERENCES

[10] R. A. Fairthorne, "Empirical hyperbolic distributions (bradford-zipf-mandelbrot) for bibliometric description and prediction," *Journal of Documentation*, 1969.

[11] "The history of bibliometrics." https://www.ecoom.be/nodes/degeschiedenisvanbibliometrie/e Accessed: 22-11-2020.

[12] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National academy of Sciences*, vol. 102, no. 46, pp. 16569–16572, 2005.

[13] M. Gusenbauer, "Google scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases," *Scientometrics*, vol. 118, no. 1, pp. 177–214, 2019.

[14] K. Balog, Y. Fang, M. De Rijke, P. Serdyukov, and L. Si, "Expertise retrieval," *Foundations and Trends in Information Retrieval*, vol. 6, no. 2–3, pp. 127–256, 2012.

[15] L. Waltman and N. J. Van Eck, "A new methodology for constructing a publication-level classification system of science," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 12, pp. 2378–2392, 2012.

[16] R. Berendsen, M. De Rijke, K. Balog, T. Bogers, and A. Van Den Bosch, "On the assessment of expertise profiles," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 10, pp. 2024–2044, 2013.

[17] A.-L. Barabási *et al.*, *Network science*. Cambridge university press, 2016.

[18] D. J. D. S. Price, "Networks of scientific papers," *Science*, pp. 510–515, 1965.

[19] M. Dunaiski and W. Visser, "Comparing paper ranking algorithms," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pp. 21–30, 2012.

[20] F. Radicchi, S. Fortunato, B. Markines, and A. Vespignani, "Diffusion of scientific credits and the ranking of scientists," *Physical Review E*, vol. 80, no. 5, p. 056103, 2009.

[21] S. Kumar, "Co-authorship networks: a review of the literature," *Aslib Journal of Information Management*, 2015.

[22] G. Abramo, C. A. D'Angelo, and A. Caprasecca, "Allocative efficiency in public research funding: Can bibliometrics help?," *Research policy*, vol. 38, no. 1, pp. 206–215, 2009.

[23] E. S. Vieira, J. A. Cabral, and J. A. Gomes, "How good is a model based on bibliometric indicators in predicting the final decisions made by peers?," *Journal of Informetrics*, vol. 8, no. 2, pp. 390–405, 2014.

[24] H. Wang, H.-W. Shen, and X.-Q. Cheng, "Scientific credit diffusion: Researcher level or paper level?," *Scientometrics*, vol. 109, no. 2, pp. 827–837, 2016.

[25] Y. Ding, "Applying weighted pagerank to author citation networks," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 236–245, 2011.

[26] J. D. West, M. C. Jensen, R. J. Dandrea, G. J. Gordon, and C. T. Bergstrom, "Author-level eigenfactor metrics: Evaluating the influence of authors, institutions, and countries within the social science research network community," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 4, pp. 787–801, 2013.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[28] A. Sidiropoulos and Y. Manolopoulos, "Generalized comparison of graph-based ranking algorithms for publications and authors," *Journal of Systems and Software*, vol. 79, no. 12, pp. 1679–1700, 2006.

[29] W.-S. Hwang, S.-M. Chae, S.-W. Kim, and G. Woo, "Yet another paper ranking algorithm advocating recent publications," in *Proceedings of the 19th international conference on World wide web*, pp. 1117–1118, 2010.

[30] M. Seeber, M. Cattaneo, M. Meoli, and P. Malighetti, "Self-citations as strategic response to the use of metrics for career decisions," *Research Policy*, vol. 48, no. 2, pp. 478–491, 2019.

[31] J. Fowler and D. Aksnes, "Does self-citation pay?," *Scientometrics*, vol. 72, no. 3, pp. 427–437, 2007.

[32] M. Bras-Amorós, J. Domingo-Ferrer, and V. Torra, "A bibliometric index based on the collaboration distance between cited and citing authors," *Journal of Informetrics*, vol. 5, no. 2, pp. 248–264, 2011.

[33] W. Li, T. Aste, F. Caccioli, and G. Livan, "Reciprocity and impact in academic careers," *EPJ Data Science*, vol. 8, no. 1, p. 20, 2019.

[34] S. Yi and J. Choi, "The organization of scientific knowledge: the structural characteristics of keyword networks," *Scientometrics*, vol. 90, no. 3, pp. 1015–1026, 2012.

[35] W. Glänzel and A. Schubert, "A new classification scheme of science fields and subfields designed for scientometric evaluation purposes," *Scientometrics*, vol. 56, no. 3, pp. 357–367, 2003.

[36] R. Klavans and K. W. Boyack, "Which type of citation analysis generates the most accurate taxonomy of scientific and technical knowledge?," *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 984–998, 2017.

[37] P. Ahlgren, Y. Chen, C. Colliander, and N. J. Van Eck, "Enhancing direct citations: A comparison of relatedness measures for community detection in a large set of pubmed publications," *Quantitative Science Studies*, vol. 1, no. 2, pp. 714–729, 2020.

[38] M. J. Cobo, A. G. López-Herrera, E. Herrera-Viedma, and F. Herrera, "An approach for detecting, quantifying, and visualizing the evolution of a research field: A practical application to the fuzzy sets theory field," *Journal of informetrics*, vol. 5, no. 1, pp. 146–166, 2011.

[39] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," *arXiv preprint arXiv:1207.4169*, 2012.

[40] A. Daud, "Using time topic modeling for semantics-based dynamic research interest finding," *Knowledge-Based Systems*, vol. 26, pp. 154–163, 2012.

[41] Y.-S. Jeong, S.-H. Lee, and G. Gweon, "Discovery of research interests of authors over time using a topic model," in *2016 International Conference on Big Data and Smart Computing (BigComp)*, pp. 24–31, IEEE, 2016.

[42] D. Duan, Y. Li, R. Li, Z. Lu, and A. Wen, "Mei: Mutual enhanced infinite community–topic model for analyzing text-augmented social networks," *The Computer Journal*, vol. 56, no. 3, pp. 336–354, 2013.

[43] M. Gerlach, T. P. Peixoto, and E. G. Altmann, "A network approach to topic models," *Science advances*, vol. 4, no. 7, p. eaaq1360, 2018.

[44] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[45] L. M. De Campos, J. M. Fernández-Luna, and J. F. Huete, "Committee-based profiles for politician finding," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. Suppl. 2, pp. 21–36, 2017.

[46] L. Waltman, K. W. Boyack, G. Colavizza, and N. J. Van Eck, "A principled methodology for comparing relatedness measures for clustering publications," *Quantitative Science Studies*, vol. 1, no. 2, pp. 691–713, 2020.

[47] M. Karimzadehgan, R. W. White, and M. Richardson, "Enhancing expert finding using organizational hierarchies," in *European conference on information retrieval*, pp. 177–188, Springer, 2009.

[48] J. Rybak, K. Balog, and K. Nørvåg, "Temporal expertise profiling," in *European Conference on Information Retrieval*, pp. 540–546, Springer, 2014.

[49] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 797–806, ACM, 2009.

[50] C. Wang, J. Liu, N. Desai, M. Danilevsky, and J. Han, "Constructing topical hierarchies in heterogeneous information networks," *Knowledge and Information Systems*, vol. 44, no. 3, pp. 529–558, 2015.

[51] J. Silva, P. Ribeiro, and F. Silva, "Hierarchical expert profiling using heterogeneous information networks," in *International Conference on Discovery Science*, pp. 344–360, Springer, 2018.

[52] J. Silva, D. Aparício, and F. Silva, "Otarios: Optimizing author ranking with insiders/outsiders subnetworks," in *International Conference on Complex Networks and their Applications*, pp. 143–154, Springer, 2018.

[53] J. Silva, D. Aparício, and F. Silva, "Feature-enriched author ranking in incomplete networks," *Applied Network Science*, vol. 4, no. 1, pp. 1–15, 2019.

[54] J. Silva, D. Aparício, P. Ribeiro, and F. Silva, "Focas: penalising friendly citations to improve author ranking," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 1852–1860, 2020.

[55] J. Silva, N. J. van Eck, P. Ribeiro, and F. Silva, "Pure-sim: utilising metadata relations to estimate publication similarity," *Quantitative Science Studies (under revision)*, 2021.

[56] M. Tsvetovat and A. Kouznetsov, *Social Network Analysis for Startups: Finding connections on the social web.* " O'Reilly Media, Inc.", 2011.

[57] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *Journal of Network and Computer Applications*, vol. 108, pp. 87–111, 2018.

[58] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[59] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the national academy of sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[60] V. A. Traag, P. Van Dooren, and Y. Nesterov, "Narrow scope for resolution-limit-free community detection," *Physical Review E*, vol. 84, no. 1, p. 016114, 2011.

[61] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[62] V. A. Traag, L. Waltman, and N. J. Van Eck, "From Louvain to Leiden: Guaranteeing well-connected communities," *Scientific Reports*, vol. 9, p. 5233, 2019.

[63] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[64] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[65] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "Rankclus: integrating clustering with ranking for heterogeneous information network analysis," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 565–576, 2009.

[66] J. Chen, W. Dai, Y. Sun, and J. Dy, "Clustering and ranking in heterogeneous information networks via gamma-poisson model," in *Proceedings of the 2015 SIAM International Conference on Data Mining*, pp. 424–432, SIAM, 2015.

[67] C. Shi, R. Wang, Y. Li, P. S. Yu, and B. Wu, "Ranking-based clustering on general heterogeneous information networks by network projection," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 699–708, 2014.

[68] X. Liu, W. Liu, T. Murata, and K. Wakita, "A framework for community detection in heterogeneous multi-relational networks," *Advances in Complex Systems*, vol. 17, no. 06, p. 1450018, 2014.

[69] J. Yang, L. Chen, and J. Zhang, "Fctclus: a fast clustering algorithm for heterogeneous information networks," *PloS one*, vol. 10, no. 6, p. e0130086, 2015.

[70] S. Gupta and P. Kumar, "Community detection in heterogenous networks using incremental seed expansion," in *2016 International Conference on Data Science and Engineering (ICDSE)*, pp. 1–5, IEEE, 2016.

[71] J. Wu, Y. Wu, S. Deng, and H. Huang, "Multi-way clustering for heterogeneous information networks with general network schema," in *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pp. 339–346, IEEE, 2016.

[72] J. Priem, D. Taraborelli, P. Groth, and C. Neylon, "Altmetrics: A manifesto," 2010.

[73] S. Haustein, I. Peters, C. R. Sugimoto, M. Thelwall, and V. Larivière, "Tweeting biomedicine: An analysis of tweets and citations in the biomedical literature," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 656–669, 2014.

[74] Y. Ding, "Applying weighted pagerank to author citation networks," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 236–245, 2009.

[75] L. Page, S. Brin, R. Motwani, T. Winograd, *et al.*, "The pagerank citation ranking: Bringing order to the web," 1998.

[76] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[77] E. M. Voorhees *et al.*, "The trec-8 question answering track report," in *Trec*, vol. 99, pp. 77–82, Citeseer, 1999.

[78] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.

[79] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*, 2013.

[80] G. R. Miranda, R. Pasti, and L. N. Castro, "Detecting topics in documents by clustering word vectors," in *Distributed Computing and Artificial Intelligence, 16th International Conference*, vol. 1003 of *Advances in Intelligent Systems and Computing*, pp. 235–243, Springer, 2020.

[81] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, vol. 32 of *Proceedings of Machine Learning Research*, pp. 1188–1196, PMLR, 2014.

[82] B. Bigi, "Using Kullback-Leibler distance for text categorization," in *Advances in Information Retrieval*, vol. 2633 of *Lecture Notes in Computer Science*, pp. 305–319, Springer, 2003.

[83] K. Sparck Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: Development and comparative experiments: Part 1," *Information processing & Management*, vol. 36, no. 6, pp. 779–808, 2000.

[84] K. Sparck Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: Development and comparative experiments: Part 2," *Information processing & Management*, vol. 36, no. 6, pp. 809–840, 2000.

[85] K. W. Boyack, D. Newman, R. J. Duhon, R. Klavans, M. Patek, J. R. Biberstine, B. Schijvenaars, A. Skupin, N. Ma, and K. Börner, "Clustering more than two million biomedical publications: Comparing the accuracies of nine text-based similarity approaches," *PLOS ONE*, vol. 6, no. 3, p. e18029, 2011.

[86] W. Chen, F. Yin, and G. Wang, "Community discovery algorithm of citation semantic link network," in *2013 Sixth International Symposium on Computational Intelligence and Design*, vol. 2, pp. 289–292, IEEE, 2013.

[87] Y. Chen, X. Xiao, Y. Deng, and Z. Zhang, "A weighted method for citation network community detection.," in *Proceedings of the 16th International Conference on Scientometrics and Informetrics (ISSI 2017)*, pp. 58–67, 2017.

[88] K.-C. Chu and C.-C. Yeh, "Knowledge flow of biomedical informatics domain: Position-based co-citation analysis approach," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1119–1126, IEEE, 2016.

[89] O. Persson, "Identifying research themes with weighted direct citation links," *Journal of Informetrics*, vol. 4, no. 3, pp. 415–422, 2010.

[90] S. Liu and C. Chen, "The proximity of co-citation," *Scientometrics*, vol. 91, no. 2, pp. 495–511, 2012.

[91] K. Fujita, Y. Kajikawa, J. Mori, and I. Sakata, "Detecting research fronts using different types of weighted citation networks," *Journal of Engineering and Technology Management*, vol. 32, pp. 129–146, 2014.

[92] G. Colavizza, K. W. Boyack, N. J. Van Eck, and L. Waltman, "The closer the better: Similarity of publication pairs at different cocitation levels," *Journal of the Association for Information Science and Technology*, vol. 69, no. 4, pp. 600–609, 2018.

[93] M. R. Hamedani, S.-W. Kim, and D.-J. Kim, "SimCC: A novel method to consider both content and citations for computing similarity of scientific papers," *Information Sciences*, vol. 334, pp. 273–292, 2016.

[94] S. Zhang, Y. Xu, and W. Zhang, "Clustering scientific document based on an extended citation model," *IEEE Access*, vol. 7, pp. 57037–57046, 2019.

[95] J. Jin, Q. Geng, Q. Zhao, and L. Zhang, "Integrating the trend of research interest for reviewer assignment," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1233–1241, International World Wide Web Conferences Steering Committee, 2017.

[96] D. Mimno and A. McCallum, "Expertise modeling for matching papers with reviewers," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 500–509, ACM, 2007.

[97] J. Tang, R. Jin, and J. Zhang, "A topic modeling approach and its integration into the random walk framework for academic search," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 1055–1060, IEEE, 2008.

[98] J. Wang, X. Hu, X. Tu, and T. He, "Author-conference topic-connection model for academic network search," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 2179–2183, ACM, 2012.

[99] Y. Sun, J. Han, J. Gao, and Y. Yu, "itopicmodel: Information network-integrated topic modeling," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pp. 493–502, IEEE, 2009.

[100] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin, "Probabilistic topic models with biased propagation on heterogeneous information networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1271–1279, ACM, 2011.

[101] M. Neshati, S. H. Hashemi, and H. Beigy, "Expertise finding in bibliographic network: Topic dominance learning approach," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2646–2657, 2014.

[102] Z. Yin, L. Cao, Q. Gu, and J. Han, "Latent community topic analysis: Integration of community discovery with topic modeling," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, p. 63, 2012.

[103] C. Li, W. K. Cheung, Y. Ye, X. Zhang, D. Chu, and X. Li, "The author-topic-community model for author interest profiling and community discovery," *Knowledge and Information Systems*, vol. 44, no. 2, pp. 359–383, 2015.

[104] Y. Liu, A. Niculescu-Mizil, and W. Gryc, "Topic-link lda: joint models of topic and author community," in *proceedings of the 26th annual international conference on machine learning*, pp. 665–672, ACM, 2009.

[105] M. Revelle, C. Domeniconi, M. Sweeney, and A. Johri, "Finding community topics and membership in graphs," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 625–640, Springer, 2015.

[106] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Computing Surveys (CSUR)*, vol. 44, no. 4, p. 20, 2012.

[107] E. B.V., *Research Metrics Guidebook*. Elvesier, 2018.

[108] J. P. Ioannidis, "A generalized view of self-citation: Direct, co-author, collaborative, and coercive induced self-citation," *Journal of psychosomatic research*, vol. 78, no. 1, pp. 7–11, 2015.

[109] R. Sinatra, D. Wang, P. Deville, C. Song, and A.-L. Barabási, "Quantifying the evolution of individual scientific impact," *Science*, vol. 354, no. 6312, p. aaf5239, 2016.

[110] A. Sidiropoulos and Y. Manolopoulos, "Reciprocity and impact in academic careers," *EPJ Data Science*, vol. 8, no. 20, 2019.

[111] M. Seeber, M. Cattaneo, M. Meoli, and P. Malighetti, "Self-citations as strategic response to the use of metrics for career decisions," *Research Policy*, 2017.

[112] M. E. Newman, "The structure of scientific collaboration networks," *Proceedings of the national academy of sciences*, vol. 98, no. 2, pp. 404–409, 2001.

[113] E. Sarigöl, R. Pfitzner, I. Scholtes, A. Garas, and F. Schweitzer, "Predicting scientific success based on coauthorship networks," *EPJ Data Science*, vol. 3, no. 1, p. 9, 2014.

[114] Y. Xu, S. Zhang, W. Zhang, S. Yang, and Y. Shen, "Research front detection and topic evolution based on topological structure and the pagerank algorithm," *Symmetry*, vol. 11, no. 3, p. 310, 2019.

[115] A. E. Jinha, "Article 50 million: an estimate of the number of scholarly articles in existence," *Learned Publishing*, vol. 23, no. 3, pp. 258–263, 2010.

[116] M. Gerlach, T. P. Peixoto, and E. G. Altmann, "A network approach to topic models," *arXiv preprint arXiv:1708.01677*, 2017.

[117] N. A. bin Jamaludin, M. Annamalai, N. Jamil, and Z. A. Bakar, "A model for keyword profile creation using extracted keywords and terminological ontology," in *e-Learning, e-Management and e-Services (IC3e), 2013 IEEE Conference on*, pp. 136–141, IEEE, 2013.

[118] "Authenticus, a bibliographic database for portuguese researchers.." `https://www.authenticus.pt/pt`. Accessed: 20-06-2021.

[119] J. Silva, "JSilva90 github." `https://github.com/JSilva90`, 2021. Accessed: 30-07-2020.

[120] "Mesh browser overview." https://www.nlm.nih.gov/mesh/mbinfo.html. Accessed: 30-07-2020.