Università degli Studi di Udine
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Informatica

Ph.D. Thesis

Knowledge-Based Techniques for Scholarly Data Access: Towards Automatic Curation.

Candidate
Dario De Nart

Supervisor
Prof. Carlo Tasso

Cycle XXVIII — A.Y. 2015-2016

Institute Contacts
Dipartimento di Matematica e Informatica
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine — Italia
+39 0432 558400
http://www.dimi.uniud.it/

Author's Contacts
Via A. Boito, 16
32100 Belluno — Italia
+39 3289008907
http://www.example.com
dario.denart@uniud.it

He who was living is now dead
We who were living are now dying
With a little patience.

T.S. Eliot, The Waste Land

Acknowledgements

Thanks to my friends and colleagues Dante Degl'Innocenti and Marco Pavan, most of the work presented in this thesis began as random conversation at the coffee machine. Also thanks to my brave co-author Marco Peressotti for giving me theoretical support in my bash of RDF's reification that eventually became a paper, despite not knowing anything on both the subject and the aim of the dissertation.

Finally, huge thanks to whoever runs Sci-hub, without which the bibliography of this thesis would have been much shorter.

Abstract

Accessing up-to-date and quality scientific literature is a critical preliminary step in any research activity. Identifying relevant scholarly literature for the extents of a given task or application is, however a complex and time consuming activity. Despite the large number of tools developed over the years to support scholars in their literature surveying activity, such as Google Scholar, Microsoft Academic search, and others, the best way to access quality papers remains asking a domain expert who is actively involved in the field and knows research trends and directions. State of the art systems, in fact, either do not allow exploratory search activity, such as identifying the active research directions within a given topic, or do not offer proactive features, such as content recommendation, which are both critical to researchers. To overcome these limitations, we strongly advocate a paradigm shift in the development of scholarly data access tools: moving from traditional information retrieval and filtering tools towards automated agents able to make sense of the textual content of published papers and therefore monitor the state of the art. Building such a system is however a complex task that implies tackling non trivial problems in the fields of Natural Language Processing, Big Data Analysis, User Modelling, and Information Filtering. In this work, we introduce the concept of Automatic Curator System and present its fundamental components.

Contents

1 Introduction and Motivation **3**
- 1.1 Scholarly Data Access . . . 4
- 1.2 Towards an Automatic Curator System . . . 6
- 1.3 Thesis outline . . . 7

2 Personalised Information Access and Document Recommendation **9**
- 2.1 Adaptive Personalisation and Information Access . . . 9
  - 2.1.1 Collaborative Filtering . . . 10
  - 2.1.2 Scholarly Data Access . . . 11
- 2.2 Content-Based Filtering . . . 13
  - 2.2.1 TF-IDF . . . 13
  - 2.2.2 Content-based scholarly data access . . . 15
- 2.3 Towards Knowledge-Based Filtering . . . 16
  - 2.3.1 Semantics sell... But Who's Buying? . . . 17
- 2.4 Wrapping it up . . . 19

3 Understanding text **21**
- 3.1 Keyphrase Extraction . . . 21
- 3.2 Named Entity Recognition, Entity Linking, and Word Sense Disambiguation . . . 25
- 3.3 Domain Model Extraction . . . 28
  - 3.3.1 Semantic Similarity . . . 28
  - 3.3.2 Ontology Extraction . . . 29
- 3.4 Towards a unifying framework . . . 31

4 Understanding, Filtering, and Explaining **35**
- 4.1 Co-occurrence Networks ad Content Description . . . 35
- 4.2 Filtering Algorithm . . . 37
- 4.3 Overview of the RES system . . . 38
  - 4.3.1 Presentation and Explanation . . . 39
- 4.4 Evaluation . . . 41
  - 4.4.1 Benchmark Evaluation . . . 41
  - 4.4.2 User Evaluation . . . 43
- 4.5 Lessons Learnt . . . 45

5 Monitoring the Research Community 47
5.1 Scholarly Data Network Analysis in the Literature . . . 47
5.1.1 Social Network Analysis . . . 48
5.1.2 Content-based analysis . . . 48
5.2 Proposed Methodology . . . 49
5.3 Results . . . 51
5.3.1 Single year CEUR proceedings analysis . . . 51
5.3.2 Three-Years analysis . . . 54
5.3.3 IRCDL proceedings analysis . . . 58
5.4 Lessons Learnt . . . 60

6 Towards Domain-Aware Information Access 63
6.1 Introducing referential spaces . . . 63
6.2 Tasks . . . 65
6.2.1 Neighbour Entity Retrieval . . . 65
6.2.2 Non-Exact Matching based on Semantic Similarity . . . 66
6.3 Experimental Evaluation . . . 66
6.3.1 Crowdsourcing Evaluation of Similar Item Retreival . . . 67
6.3.2 Evaluation of Non-Exact Matching on a Dataset . . . 70
6.4 Lessons Learnt . . . 72

7 Conclusions 75
7.1 Proposed solution overview . . . 75
7.2 The many criticalities of evaluation . . . 77
7.3 Future work . . . 79

I Appendix 81

A Complete Publications List 83

B Remarks on the Resolution of Data Citation and Its Feasibility 85
B.1 Introduction . . . 85
B.2 Related Work . . . 87
B.3 Formalisation . . . 88
B.3.1 Families of Named Graphs . . . 88
B.3.2 A simple algebra for NG families . . . 90
B.3.3 Reasoners over NG families . . . 94
B.4 Coherent (meta)information . . . 96
B.4.1 Well-founded relations . . . 96
B.4.2 Well-stratified NG families . . . 97
B.4.3 Assessing well-stratification using types . . . 98
B.4.4 Assessing well-stratification using graphs . . . 99
B.5 Towards a Well-Stratified data language . . . 101
B.6 Conclusions . . . 104

1

Introduction and Motivation

It is a fact well known to anyone who has ever worked in the research that surveying previous results is a critical step before venturing in any task that could be deemed as innovative. Getting a complete, or good enough, picture of the state of the art even in a very narrow field is, however, no trivial task. Research gets published on journals, books, and conferences, moreover a lot of publishers provide open access to scientific publications, however thinking of going through *all* the potentially relevant literature to a given topic is simply delusional, due to the amount of published research. While there are no official statistics on how much research has been published, the authors of [13] estimate that 1.346 million articles were published in 23.750 journals within 2006. Such a number was evaluated upon the most comprehensive citation indexes, Web of Science and Ullrichsweb. That number, however can be regarded as a severe underestimation of the actual number of published peer-reviewed research papers and the American National Science Board estimates the average annual growth of the indexes within Scopus to be at 7.0%[1]. Surveying such a large volume is made considerably easier by the usage of bibliometric indexes such as the citation count, the journal's impact factor, or the author's h-index that allow to identify, respectively, works that influenced subsequent research, high-profile journals, and influential researchers. Those indexes, however, take a lot of time to build up and even the most relevant works will need time to gain enough citations to be deemed truly influential. Finally, while the most cited and well-known works are likely to be found with search engines that consider bibliometrics or by word of mouth, a lot of works containing extremely relevant results slip into the long tail of unknown papers [170], preventing scholars to find them. As a matter of fact, no matter which is the field of interest, some extremely relevant results might have already been published unknown to the scientific community. A clear example of such a situation is the case of Jack Thomas Andraka, who in 2012, aged 15, designed a new type of sensor for early-stage pancreatic cancer screening leveraging already published results that he came across after a thorough review of the literature published at the time. At the time Andraka had little prior knowledge over pancreatic cancer and, by using Google, Facebook, Youtube and other means available to anyone who has an

[1] https://www.nsf.gov/statistics/indicators/

internet connection, was able to retrieve an information the scientific community was oblivious of [34].

How many critical results are there in the literature, hiding in plain sight? How many obscure works carry extremely relevant conclusions? Probably a lot, and getting them with present-day tools, like Andraka did, is an extremely time consuming activity that most researchers cannot afford. It is therefore vital to provide researchers and practitioners alike with better tools to improve their access to potentially relevant results.

1.1 Scholarly Data Access

Several scholarly data access tool already exist: scholarly search services such as *Google Scholar*[2], *Semantic Scholar*[3], or *CiteSeerX*[4] are well known and widely used within the research community, and there also exist plenty of large digital libraries with internal search tools such as *ScienceDirect*, *Springerlink*, *Scopus*, *PubMed*, and *CEUR* only to name a few[5].

These tools, however, present several shortcomings in the information access they provide. The authors of [143] identify three critical issues typical of current scholarly data access tools:

- No semantic characterization of research areas: research is a knowledge intensive domain, however most systems do not ground the terms found in the documents or manually annotated into a structured domain representation. This syntactic, rather than semantic, treatment of research topics and areas causes two well known problems: systems often misinterpret key terms and do not take into account important semantic relations between research areas, such as an area being a subarea of another one, or two labels referring to the same research area.

- Lack of granular analysis: most available systems provide little or no research trend analysis, and where such feature is provided, like in Microsoft Academic Search, it is typically limited to very broad topics such as "World Wide Web" and "Databases", while researchers and practitioners are likely to be more interested in narrower ones such as "Recommender Systems" or "Machine Learning".

- Digital library bias: scholarly data access tools typically put the emphasis on classic digital library functionalities, such as supporting search for publications and providing citation services. While these features are undoubtedly useful, they cannot really support tasks such as exploratory analysis of a specific topic.

Furthermore, most state of the art systems rely on algorithms and policies designed for more general purpose applications like search engines and do not take into account some specific user requirements that are typical of researchers:

[2] scholar.google.com
[3] www.semanticscholar.org
[4] http://citeseerx.ist.psu.edu
[5] for a more comprehensive list of scientific publications repositories we address the curious reader to en.wikipedia.org/wiki/List_of_academic_databases_and_search_engines

- Adaptive Personalisation: researchers, depending on their goals, background, and prior knowledge can look for dramatically different things when using the same search key, it is therefore vital to provide some degree of personalisation to address this issue.

- Timing: while accessing classic and well established works is important, the real value for researches is being able to access new result as they get published, to be updated with the latest advancements in their field. In recent times Google Scholar began to provide a personalised alert service to scholars based on their bibliography, however this kind of personalisation, depending on the published bibliography is slow to adapt to changes in the user's interests.

- Explanation: reading a paper is a time consuming activity, therefore it is extremely desirable to be able to figure out why a paper was retrieved and what is its connection with the topic of interest before reading it. A common solution is to show the abstract of retrieved papers as description but sometimes even abstracts can contain lots of information and thus take time to be truly understood, or on the other hand they may contain just too little information to decide whether or not the paper is relevant.

All these features are highly desirable for researchers and those who are to be introduced in the research practice as well, but are seldom provided by state of the art systems and to the best of our knowledge no currently available system fulfils them all. The third one in particular represents an open issue, because very few information access techniques proposed so far which allow the user to scrutinise their results and understand why an item was retrieved. This kind of interaction between the user and the system is helpful to obtain an overview of the retrieved items and to refine future searches or to fine tune the user profile if available. To fulfil these requirements a series of non trivial problems related to very nature of the scholarly data domain must be overcome. We can pinpoint the following challenges in providing an ideal access to scholarly data:

- Access and management of raw data: as mentioned before the number of published research works is huge and keeps growing at a steady pace, with countries like China increasing their scientific output by 18.9% each year. Research topics and trends change over time, with new venues, e.g. new journals and conferences, gaining momentum and getting more and more relevant. Moreover there is lot of heterogeneity in the academic publication world: different editors have different formats, different communities have different venues, and there exist a plethora of licensing policies and reviewing processes to take into account. On top of that, there exist predatory editors who claim to publish peer-reviewed literature, but do not [10].

- Effective content representation: the ultimate goal of accessing scholarly data is providing the right contents to the right scholar and to accomplish this goal the system must have some mean to assess the content of the items it can retrieve. Such a representation should be informative, easy to understand, and allow the system to provide means to produce explanations. Several solutions could be viable, from sets of relevant concepts to more structured representations including

relevant features for scholars such as the claim, the methodology, and the result described in a paper, in the likes of *Micropublications* [31]. To achieve it manual annotation would be too demanding and Natural Language Processing should be taken into consideration.

- Need for extensive background knowledge: research is not easy to understand, mostly due to its rich and ever-increasing vocabulary. Some fields, like medicine use standardised vocabularies formalised in thesauri, but in general this is not the case. Researchers coming from different areas are likely to describe the same topic using a different jargon and it is not uncommon for them to be unaware of the vocabulary used by their colleagues active in a different community. Moreover some concepts, e.g. Hidden Markov Models and Conditional Random Fields, though not being identical may be tightly connected, synergistic, or complementary.

Overcoming these challenges implies going beyond traditional information retrieval and recommender systems: a lot of knowledge must be embedded in the system to cope with these problems, namely knowledge over the research community, over the texts, and over the various domains of research we intend to run the system upon.

1.2 Towards an Automatic Curator System

The amount of background knowledge required to fulfil the requirements presented in the previous section implies moving from systems such as recommenders or search engines to something more similar to an intelligent agent that builds and maintains a detailed user profile, gathers items that might be of interest from a number of selected sources, and filters the most relevant items in such a way that its decisions are somehow understandable to the user. To achieve this goal we believe that the very paradigm of interaction between the user and the system should change, allowing a richer information exchange between the user and the system. First of all, since researchers tend to have multiple topics of interest, the system should allow the creation of thematical collections of item, possibly initialised with a single query in likes of a search engine, for instance "Conditional Random Fields". The system will then start building a relevant bibliography for that topic. The user, aside from specifying one or more topics of interest, should be able to tell the system his or her prior knowledge, for instance by specifying some works already read or some concepts or topics already mastered. On the other hand the system should be able to retrieve a set of items that satisfy the user's information need. Let us assume, for instance, that a user wished to gain information on Conditional Random Fields: depending on the user's prior knowledge the system should infer his or her field of interest and therefore suggest a consistent set of papers. If the user were interested in Natural Language Processing it would be of little interest for him/her to be prompted with papers dealing with usage of such a technique in Image Processing; vice-versa a user whose interest is Image Processing probably wouldn't find relevant a series of papers about Named Entity Recognition, which commonly involves Condition Random Fields as well. Moreover the system should be able to modify the retrieved document set over the time, as new relevant items are made available. Like the curator of a museum or a library, such a system, given enough user information, should be able to build and maintain a collection of relevant items in the most autonomous

way possible. We will herein refer to such a system as an *Automatic Curator System*. Such a system could also use its background knowledge to generate mashups of relevant authors, venues, and related topics to the current interest of the user. We can identify three main components of an Automatic Curator System:

- An Information Filtering module that selects documents according to the system's knowledge over the user. Due to the large volume of contents available, the performance of this module is critical. Its main responsibilities are to build and maintain the user profile, receive user queries and filter accordingly the contents included in the system's catalogue.

- A Data Gathering module holding the responsibility of selecting data sources to monitor and enriching the catalogue of items that the system can push to its users. This module also provides metadata such as topics, keywords, and, bibliographic information for the items in the catalogue.

- A Background Knowledge manager responsible for building and maintaining a background knowledge base that can be exploited by the other two components to estimate how similar or related are concepts and topics. Such a knowledge base could be built either by exploiting external knowledge sources such as ontologies publicly available, like *DBpedia*, or in unsupervised way by aggregating textual information gathered by the Data Gathering module with clustering or automatic ontology acquisition techniques, or an hybrid solution leveraging both informations.

The proposed architecture of the Automatic Curator System is shown in Figure 1.1. With respect to current information access systems, the most notable difference is the highly proactive behaviour of the Curator System: while traditional recommender systems tend to operate within a closed catalogue, the ACS actively seeks new items, similarly to a search engine that exploits Web spiders to enlarge the amount of indexed pages, but while search engines require continuous user stimulation, expecting the user to refine his or her queries, an ACS keeps pushing potentially relevant content and updates the collection of potentially relevant items as the user interests and the state of the art evolve over time. Finally, an ACS may look similar to a news recommender: they both operate at a content level, they both require an initial setup after which they behave autonomously, and they both are meant to give priority to novel items. However, the ACS goes one extra mile thanks to its background knowledge: it allows relevant, but old, items to remain in the recommended set. While news recommenders provide a stream of content wherein the new items will always come first, an ACS provides a set of items that ideally represents the state of the art at that time, which can include very recent papers, as well as classic ones as long as they are still actual and relevant in the investigated field.

1.3 Thesis outline

In this work, we will describe the fundamental components of an Automatic Curator System that we investigated over the previous four years, exploiting different Natural

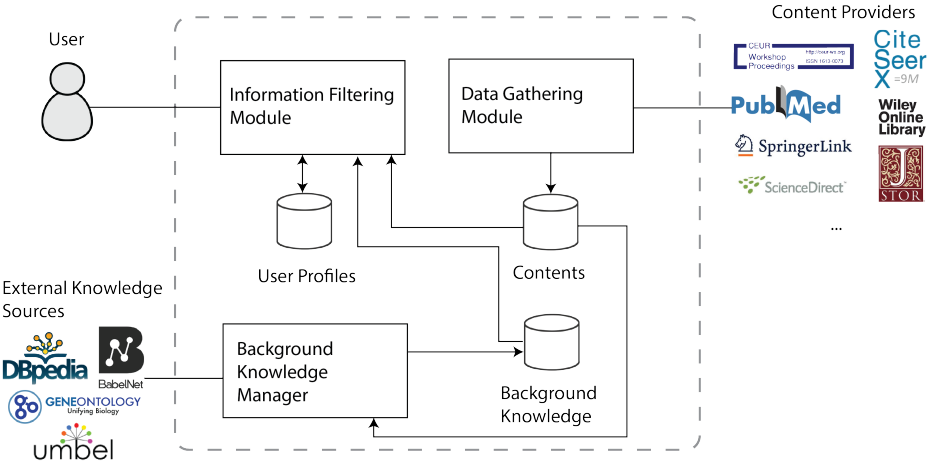

Figure 1.1: Architecture of an Automatic Curator System.

Language Processing, Data Mining, and Semantic Web technologies to overcome the many challenges we encountered in the development of such modules. Their integration allows to build a system which fulfils the requirements presented in the previous sections.

The rest of the thesis is organised as follows: in Chapter 2 we survey the domain of Personalised Information Access and introduce the many challenges of providing an effective personalised information access in the domain of scholarly data, in Chapter 3 we introduce some Natural Language Processing techniques we leveraged in chapter 4, 5, and 6 that describe, respectively, the aforementioned recommendation, data source, and domain knowledge models. Finally, in Chapter 7 we conclude this work with some final remarks and future outcomes.

2

Personalised Information Access and Document Recommendation

In this chapter we introduce the most critical aspects of personalised information access, present the domain of scholarly data access, and provide a brief description of some state of the art solutions. We will mostly refer to the extensive Recommender System survey work presented in [78] and integrate such information with some recent findings and a literature survey of the solutions presented in the scholarly data domain.

2.1 Adaptive Personalisation and Information Access

With the exponential growth of the Web and digital libraries we are experiencing nowadays, traditional information retrieval techniques may result in information overload, i.e. retrieving way too much information for the user to browse. Scientific digital libraries, in particular, host huge catalogues of publications browsed every day by thousands of researchers who seek relevant results. Due to the large availability of data, this activity is extremely time consuming and therefore tools to provide a better information access are desirable. Over the last few years Adaptive Personalisation (herein AP) technologies, such as personalised search engines, recommender systems, and other information filtering tools, proved to be extremely effective in providing a better information access, hence easing the problem of information overload.

The ultimate goal of all AP technologies is to bring the right information to the right user at the right time. To do so a *user model* is needed to allow the system to know something about each user and therefore select items to show him or her accordingly. A broad span of approaches to address this problem have been discussed in the literature over the years and *collaborative filtering* emerged as the leading solution for most AP applications.

Table 2.1: An example of user-item matrix

User	$item_1$	$item_2$	$item_3$	...	$item_n$
$user_1$	1	3	2		5
$user_2$	5	2	4		3
$user_3$	1	4	?		4
...					
$user_m$	4	3	4		2

2.1.1 Collaborative Filtering

The main idea of collaborative filtering approaches is to exploit information about the past behaviour or the opinions of an existing user community to predict which items the current user of the system will most probably like or be interested in [59]. These types of systems are in widespread industrial use today, in particular as a tool in online retail sites to customize the content to the needs of each customer [78, 158]. Pure collaborative approaches rely on a matrix of given $user - item$ ratings where each row includes all the ratings expressed by its corresponding user, as shown in Table 2.1. User ratings can be collected either in an explicit way, as judgements, or implicitly, based upon user behaviour, observing the clickstream, view time, or performing other log analysis. Due to the large amount of items considered, such a matrix is likely to be extremely sparse with missing values corresponding to items never rated, which are likely to be unknown to the user. The missing values of each user can be estimated using various techniques generating user rating predictions. With respect to the example depicted in Table 2.1, $user_2$ did not rate $item_3$, thus leaving a missing value in the ratings matrix. The missing value can be estimated with a nearest neighbour approach: it can be easily observed how $user_3$ and $user_1$ consistently expressed similar judgements over the same items, therefore, since $user_1$ rated $item_3$ we can use his rating as a prediction of how much $user_3$ will like $item_3$. This basic idea can be expanded by considering larger user neighbourhoods, using different neighbour selection techniques, using dimensionality reduction approaches, like singular value decomposition, or machine learning techniques like random forests, support vector machines, and others [45]. Regardless of the way missing values are estimated, collaborative filtering presents the major advantage of being *content agnostic*, which means that the considered items could be anything. The main advantage of this is that the costly task of providing detailed and up-to-date item descriptions to the system is avoided and, in theory, allows collaborative filtering to scale to any domain since there is no need for the system to maintain a content model or any form of domain knowledge. Moreover, explicit ratings are generally easy to obtain from users and where they cannot be obtained, implicit user feedback can be leveraged.

Even though these characteristics make it extremely effective in most real-world scenarios, collaborative filtering cannot be applied to every domain: for instance news recommendation cannot be done in a collaborative way, since it would take too long to gather enough ratings to recommend a news item with sufficient precision and in the meantime it would become obsolete. In fact, collaborative filtering strategies present three issues that must be considered before applying such techniques to a new domain:

- *Need for a large and active user base*: collaborative filtering assumes that the num-

ber of users who rate products is much larger than the number of items, with the average user expressing a large enough number of judgements over different items. This assumption holds in the domain of e-commerce, where there is a specialized catalogue, focused on a single domain, for instance books. In this scenario users are clients who are very likely to interact with the system several times, often consuming similar items, and therefore ratings are quite easy to obtain.

- *Cold start problem and long tail*: new items or old items that received very few ratings are unlikely to be recommended in a collaborative system. In the domain of e-commerce these problems are not a critical issue since new products are already pushed by advertising and most of the profits are driven by a few blockbuster items, with the so-called long tail of the catalogue generating a relatively small fraction of the income.

- *Assumption on user interest*: the underlying assumption of collaborative filtering is that a user's interests will remain stable over the time, thus users who showed similar interests in the past will keep having similar interests. This is a sound assumption in domains such as movies, books, or other entertainment products, where the general taste of a user is unlikely to change abruptly.

When a large and active enough user base is available, cold start and long tail problems are mitigated by alternative information access techniques, and it is likely to assume that user interests will not change abruptly over the time, collaborative filtering is effective. However not all domains fulfil these assumptions, in fact several of them lack either a large enough user base, tolerance towards cold start/long tail problem, or coherence in user interest evolution.

2.1.2 Scholarly Data Access

Scholarly data access represents a perfect example of a domain wherein collaborative filtering strategies are unlikely to produce satisfactory results, since none of the assumptions described in the previous paragraph holds. More specifically, the scholarly data domain presents the following characteristics:

- Relatively little user base: research communities con be quite large, but their numbers simply pale when compared the number of customers popular e-commerce sites in the likes of Amazon or eBay can count. Moreover, even though some large multidisciplinary archives exist, such as Scopus[1], ScienceDirect[2], JSTOR[3], and Springerlink[4], most digital libraries are domain specific and attract only a small fraction of the research community. Moreover the typical user of such digital libraries is likely to have contributed with some publications, thus raising the number of items to be rated.

- Scarce user activity: the primary feedback offered by the research community is citations, if a scholar finds an item relevant, he will cite it; however to express such

[1] https://www.elsevier.com/solutions/scopus
[2] http://www.sciencedirect.com/
[3] http://www.jstor.org/
[4] http://link.springer.com/

a feedback a paper must be published, making expressing feedback extremely slow and hard. Some Web sites such as Researchgate[5] tried to provide scholars with alternative and more agile ways of expressing feedback, however they received mixed reception from the research community [165].

- Criticality of novelty: similarly to the news recommendation domain, in scholarly data access the novelty of the proposed contents is a fundamental quality aspect and new items should be given precedence, therefore the cold start problem is a critical issue.
- Frequent changes in user interests: it is not unusual for a scholar to move from one topic of interest to another during his career, especially when assigned to different projects. These changes are hard to predict and rather unlikely to affect a large community of users simultaneously, thus making user interest evolution rather incoherent.

Several authors over the last 15 years addressed the problem of finding relevant scientific literature, making this field an interesting case study. Some authors tried leverage collaborative information, namely citations between papers gathered from online citation repositories like CiteUlike[6]. The seminal work presented in [107] suggests that collaborative filtering systems could effectively support scholarly data access. However, it soon emerged the well known bias of these techniques towards "blockbuster" items, which is particularly evident in domains where the ratings matrix is very sparse, such as scientific literature where it can be estimated that roughly the 90% of published works do not get feedback in the form of citations [109]. In the scientific literature blockbuster are typically highly cited papers, likely to be included in the bibliography of most works in their same field of application. While being biased towards such items might make sense when building introductory reading lists to help novices investigate a research area [44], in the general case scholars wouldn't be interested in items they already know. It is well known in fact that scientific literature is full of so called sleeping beauties [170], i.e. works with very relevant results that however go unnoticed for years, and that is the kind of item that most scholars desire to be pushed to them; these items, however, will be in the long tail of scarcely rated and viewed items. More recent work, building on this observation, couples collaborative information with some kind of content modelling in the likes of search engines: the authors of [175] propose an hybrid recommender system including a collaborative module and a content-based one. The authors of [76] consider instead co-citations networks, i.e. overlapping references between academic papers. Citations are automatically extracted from the bibliography of the considered papers. The underlying assumption is that including a reference in a paper can be considered as a peculiar form of tagging and papers citing the same works can be considered similar; building on such an assumption the authors manage to achieve a significant precision in recommendation. The presented evaluation, however, does not take into account other key factors of recommendation quality, such as novelty and diversity of the recommended content. Finally, some authors proposed to broaden the spectrum of leveraged information considering the relations involving users, publications, tags, and other metadata: in [43] this information is used to produce a graph

[5] http://www.researchgate.net
[6] http://www.citeulike.org/

Table 2.2: An example of user and content model

	$feature_1$	$feature_2$	$feature_3$	...	$feature_n$
$user_1$	5	1	1		3
	$feature_1$	$feature_2$	$feature_3$	...	$feature_n$
$item_1$	1	3	2		1
$item_2$	5	2	1		3
$item_3$	3	4	2		2
...					
$item_m$	4	5	5		3

according to which personalized suggestions are computed by means of the FolkRank algorithm [81]. All these techniques, though using user generated information that does not strictly model document content, lack the truly content agnostic nature of pure collaborative filtering approaches, leaning towards content-based filtering.

2.2 Content-Based Filtering

The key principle of *content-based filtering* is to build a content model to be matched with a user model. The former is a description of the contents of each item the system can provide to the user, the latter a description of the interests of each user. In Table 2.2 an example of a typical content-based filtering situation is shown: user needs and item contents are represented with the same set of features. With this model, finding the best item for $user_1$ reduces to finding the item which feature vector correlates the best with $user_1$'s own feature vector. When compared to collaborative techniques, content-based approaches require additional knowledge over the items to be filtered that might be hard to obtain. Typically quantitative aspects of the content to be recommended, e.g. the number of pages of a book or the lens width of a camera are fairly easy to obtain, but more qualitative features, such as the narration pace of a book or the claim of a paper, are much more subtle and hard to get. When the items to filter are cameras, laptops, or power tools relying on purely quantitative features may be enough, however this is not the case of scholarly data access, where qualitative aspects such as the research topic, claim, and method are relevant to the interests and the needs of the typical user. Even though qualitative features such as the main topic of an article are often easy to get for a human annotator, their gathering may not be cost and/or time effective, especially in the scholarly data domain where quality annotation needs expert knowledge to be produced. A common practice in the document filtering domain is to adopt a vectorial representation of documents wherein each position models the presence and possibly the relevance of a term inside the considered document. This representation is often referred to as the *Bag of Words* model.

2.2.1 TF-IDF

Bag of Words models used in document filtering tasks are typically built using the TF-IDF encoding format. TF-IDF was originally developed for document retrieval [156] but often used for related tasks such as text categorisation [38]. The core idea of such a

representation is that two aspects of the words within a text are to be considered: frequency and specifcity. The frequency of a term is simply the number of times it appears in the considered document: if a term is often used, it is likely to be somewhat relevant, for instance a document containing several repetitions of the term "engineering" it is likely to be about engineering. Some words, however, are likely to be extremely common, e.g. conjunctions, prepositions, auxiliary verbs and other function words, and therefore their presence is likely to carry very little information even if they have a high frequency. The specificity of a term on the other hand represents how characteristic of a document a term is, in other words how much a given term helps in discriminating a document from the rest of a corpus. Specificity is usually quantified as inverse document frequency that is, given a document corpus, the size of the corpus divided by the number of its documents including the considered term. Frequency and specificity are brought together in the TF-IDF weighting scheme, where for each term in each document, its term frequency (TF) is multiplied by its inverse document frequency (IDF). The most widely adopted formalisation of TF-IDF in Information Filtering applications, and the one we will refer to in the rest of this thesis is the following [24]:

$$TF - IDF(i,j) = TF(i,j)IDF(i) \tag{2.1}$$

Where i is a term and j is a document. The term frequency part is evaluated as follows:

$$TF(i,j) = \frac{freq(i,j)}{maxOthers(i,j)} \tag{2.2}$$

where $freq(i,j)$ is the number of occurrences of i in j and $maxOthers(i,j)$ is the number of occurrences of the most frequent term in j distinct from i. Which means that if i is the most frequent term in j, $TF(i,j)$ will be greater than 1. The inverse document frequency part is evaluated as follows:

$$IDF(i) = log\frac{N}{n(i)} \tag{2.3}$$

where N is the number of documents included in the considered corpus and $n(i)$ is the number of documents wherein i appears. The logarithm is introduced to assign to very common terms, such as function words, near-zero values and to damper the effect of using very large document corpora. Text documents can be encoded with TF-IDF as vectors in a multidimensional Euclidean space. Each dimension corresponds to the a keyword (also called term or token) appearing in the documents.

Document representations built this way suffer, however, from great sparsity issues due to the fact that every term included in the considered document set will be regarded as a dimension. Several techniques can be used to counter this well-known issue, the most popular ones being:

- Stopword removal: some words clearly do not characterise a document, nor are useful to represent its content. Aside from function words these word may include extremely common, non-interesting names and adjectives in the considered domain of application, and are generally referred to as stopwords. Stopwords are generally matched against an ad-hoc built dictionary.

- Stemming: some terms may appear in different forms throughout a document or a corpus, for instance the word "tool" might also appear in its plural form "tools", normally these two variants of the word would generate distinct dimensions, however removing the suffix "-s" when parsing the documents can make them collapse into a single feature. Stemming procedures are commonly implemented as a combination of morphological analysis using, for instance, Porters suffix-stripping algorithm [148] and word lookup in dictionaries such as WordNet.

- Size cut-offs: another common solution to reduce the number of considered dimensions is to use only the top n most informative words, however estimating n is no trivial matter: a small number might not include enough relevant features, while if too many are selected the data might be still noisy or the dimensionality of the problem still too high.

Regardless of the devices used to counter the issue of vector sparsity, TF-IDF remains a purely statistical technique that does not take into account several linguistic and domain aspects that are vital in assessing the relevance of a term within a document. More advanced techniques consider Information Extraction strategies to extract document features from their textual content. Such strategies are extensively surveyed in the next chapter and they all serve the same purpose of TF-IDF: generating a document content representation that is easily processable by information access applications.

2.2.2 Content-based scholarly data access

Due to the peculiar characteristics of the scholarly data access domain introduced in Section 2.1.2, it comes with little surprise that the content-based filtering approach has been largely favoured by researchers and practitioners in this field. Several Scientific paper recommender systems, therefore, analyse the textual contents of papers in order to provide recommendations. Aside from the works presented in [175, 76, 43], already introduced in Section 2.1.2, that introduce content-based components into collaborative strategies, other works describe other approaches wherein the filtering process is primarily guided by the items content representation. In [60], the authors propose a content-based filtering system based on a simple, unsupervised, term extraction technique to identify relevant concepts and entities by considering their frequency in the document. To limit the number of considered terms, the authors proceed to cluster them according to their semantic similarity, e.g. the fact that "America" is tightly related to "United States"[7], measured by means of Google distance [28] and then a vector of related terms is used as document model. Other works aim at extracting from the documents, instead of key terms, the very concepts described in a paper to achieve a more semantics-oriented modelling of both content and user interest. The authors of [50] to achieve this goal extract Keyprhases from the document textual contents employing a combination of statistics and linguistic knowledge, then the obtained information is processed with vector similarity metrics to generate recommendations.

Closing the loop with collaborative information, some authors in the literature suggested to use collaborative annotation as a source of content information to feed content-

[7] We will provide a broader discussion of semantic similarity and of the techniques commonly employed to estimate it in Chapter 3

based filtering algorithms. Several systems such as CiteUlike, BibSonomy, and Researchgate allow researchers to express feedback, including explicitly stating their interests, annotating papers with keywords, and making bookmarks. All this information can be leveraged to model the content of scientific papers without processing their textual content. This choice has the obvious advantage of leaving to human annotators the burden of determining which terms are relevant and which are not. The authors of [79] adopt this strategy and use as document representation tags provided by the users of CiteUlike to generate a dictionary that is used to identify relevant concepts present in the abstract of scientific publications. Also the authors of [52] propose to leverage user-annotated tags to build content models.

2.3 Towards Knowledge-Based Filtering

Textual content alone, however, may not be informative enough, especially in knowledge-intensive domains such as scholarly document access. Consider for instance the case of a paper containing the following key terms: "Information Filtering", "Collaborative Filtering", and "Adaptive Personalisation". Consider now a second document described by the following key terms: "Recommender System", "User Modelling", and "User-user Filtering". Although it is clear to a human expert that the two documents deal with similar topics a content-based algorithm sees no similarity between them because none of the key terms describing the first document is to be found in the second. This scenario, due to the rich vocabulary commonly used in research, is likely to happen over scholarly data. To overcome this limitation domain knowledge has to be introduced in the filtering strategy. There exist several ways of introducing such a knowledge into an Information Access system, and the line between content-based and knowledge based techniques is often blurry [78], however we are primarily interested in techniques that involve a background knowledge base such as a taxonomy, an ontology, or some other knowledge representation.

Taxonomies are a simple, yet powerful, form of background knowledge representation that has been widely explored in the literature. The authors of [25] train a classifier to associate documents key terms to the concepts included in the ACM Computing Classification System (CCS). The hierarchical structure of the CCS allows the system to represent both user interests and document contents with trees of concepts. A user profile and a paper representation are then compared by a tree edit-distance which computes a similarity measure among trees. By doing this, even if the user description and the document's one do not share key terms, it is still possible to asses how close they are leveraging the structure of the CCS.

The idea of exploiting an authoritative knowledge representation to provide better scholarly information access is pushed further by the authors of [143, 139, 141], who introduce Rexplore, an information access tool built on top of several academic knowledge bases. The considered knoweldge bases include Microsoft Academic Search and DBLP++ for scholarly documents and authors metadata, DBpedia and GeoNames for background knowledge, and Wikipedia, Google Scholar, and Conference Alerts to gather full-text data. Aside from authoritative structured knowledge gathered from the Internet, Rexplore includes also its own OWL ontology, automatically built with the Klink algorithm [142] that infers relationships between key terms contained in the

text of documents. All these kinds of information are then integrated into a single knowledge base. The goals of Rexplore are understanding the dynamics of research areas, unveiling shared interests, goals, and expertise between researchers, and allowing fine-grained academic expert search along multiple dimensions. Rexplore provides an integrated exploratory search and analysis environment that allows a domain expert to accomplish several non-trivial data exploration tasks, such as finding trending topics in a given research area, in a more efficient way than other commonly used tools such as Microsoft Academic Search. Rexplore, however, lacks proactivity: the user is expected to perform the data exploration tasks with the support of the extensive knowledge base the system provides, but such a knowledge is not used to predict user behaviours and provide personalisation.

Graph-based recommenders over the last year have emerged as a recommendation paradigm that can greatly benefit from the integration of semantic information, such as the one available in the Linked Open Data cloud [144]. In graph-based recommenders items and users are represented as node and predictions are calculated by walking the graph to nearby nodes and combining the opinions of the nearby users. Being Linked Data graphs as well, a recommender's graph model can be easily extend with nodes and edges taken from the LOD cloud, however the amount of nodes and edges that can potentially be imported is huge and a selection step is needed to prevent overfitting, avoid excessive noise in the model, and allowing acceptable computation times. The authors of [119] provide a detailed survey of graph-based techniques using Linked Data, focusing on the various selection strategies that can be adopted to filter the most relevant LOD features, and introduce a novel technique based on the Personalised Page Rank algorithm. The Personalised Page Rank technique coupled with a Principal Component Analysis based feature selection step appears to substantially outperform the other configurations on two public evaluation data sets. Though not designed for scholarly data, these techniques appear promising for such a domain, wherein the background information provided by the LOD cloud could be extremely useful to associate users to potentially relevant documents.

2.3.1 Semantics sell... But Who's Buying?

Semantic technologies, as shown in the previous section, have showcased a huge potential and therefore in the last years attracted a growing number of scholars and practitioners alike. However semantic technologies present three severe limits to their extensive usage, namely they are:

- Subjectivity
- Computational complexity
- The need for ontology alignments

Being ontologies and linked data the result of a knowledge-intensive process aimed at providing a formal description of a domain, subjectivity is intrinsic in their very nature: different domain experts may in fact have different visions and therefore not agree on the formalisation of their domain. Even when several experts and large, authoritative organisations are involved, the resulting domain formalisation ay still be questionable or

simplistic to some extent. It is the case of the aforementioned ACM classification that, as noted by the authors of [143] includes only four sub-topics for the entry Intelligent Web Services and Semantic Web, failing to reflect the variety of topics being tackled by the Semantic Web research community, which happens to be among the largest ones in ICT. Another example of a questionable design choice in the ACM classification is that Ontology Languages is classified under Intelligent Web Services and Semantic Web; however one could argue that the former is a sub-topic of the latter is rather strange, given that ontology languages were being designed well before the Semantic Web was recognized as a research area. We could go on with other examples such as these, not to mention naming decisions that often ignore the rich span of synonyms a domain offers. These limitations are, however, intrinsic in the process of expert knowledge representation. Collaborative efforts involving a very large number of users such as social tagging, on the other hand, rely on the wisdom of the crowds to provide the most agreeable description. Assuming that a crowd of non-experts who work independently and with different goals in mind will provide a good description on their own may be however wishful thinking if not controlled. For instance DBpedia, arguably the most well-known collaboratively built knowledge base, is the outcome of an highly controlled process: the Wikipedia articles are constantly reviewed by the Wikipedia community and the information to be included in DBpedia is further selected in an automatic way leveraging community-designed article templates [6]. The authors of [42] pinpoint with practical examples the most insidious shortcomings of uncontrolled collaborative document annotation.

Letting aside subjectivity, human errors, and openly malicious behaviour, semantic technologies still present a major problem: impractical computational times. Providing semantic tools that can efficiently scale up to the large volumes of data involved in nowadays information access applications is in our opinion a critical step towards fully accomplishing the potential of the Web of data. However the very technologies on which Linked Data are built are, as a matter of fact, designed to be expressive and conceptually elegant, rather than efficient. For instance SPARQL, the W3C recommendation query language for RDF, consists in graph pattern matching, which is a notoriously complex operation. It can be proved that the evaluation of SPARQL patterns is PSPACE-complete in the general case [146], meaning that solving complex queries, aside from having an high complexity, cannot be effectively parallelised. Even if we limit the usage of SPARQL to the AND and FILTER operators[8], the complexity of solving a query is still $O(|P||D|)$ where $|P|$ is the number of triples in the graph pattern, and $|D|$ is the number of triples. With the current size of Linked Data datasets, counting millions if not billions of triples[9], that kind of complexity can seriously undermine the field usage of these technologies. Moreover OWL, the Ontology Web Language, is notoriously undecidable if all of its primitives are used, and even limiting the domain modelling to the DL subset of OWL, the complexity of reasoning is still NP-hard [10]. The OWL EL profiles introduced in the last W3C recommendation[11] allow to reduce

[8] thus not using the more advanced UNION and OPT operators, that lift the complexity respectively to NP-Hard and PSPACE-complete

[9] at the time of writing DBpedia included over 3 billion triples

[10] for a more detailed discussion of the complexity of reasoning operations over Linked Data we refer the curios reader to [75]

[11] `https://www.w3.org/TR/owl2-profiles/`

the complexity of reasoning to a more manageable polynomial complexity, however this implies renouncing to a lot of expressive power[12]. Moreover the reasoning operation can hardly be parallelised and when dealing with billions of triples even linear algorithms could still be too complex to solve in acceptable time. Another intrinsic limitation of the Linked Data format is that it is hard to keep track of the provenance of the used data. Even describing in a non ambiguous way the exact portion of a data set used to perform a certain task is a non trivial problem. In general, the problem of Data Citation on Linked Data is still an open issue and it can be proved that present day Linked Data technologies do not even guarantee data citations to be computable [123, 35][13].

The third and final issue of semantic technologies, the need for ontology alignments, is a direct consequence of the fist one: since domain formalisations are to a certain extent subjective, having multiple formalisations for the same domain is a common situation. With the rise of the Semantic Web a huge number of ontologies has been proposed, each one of them representing a unique conceptualization of a given task or domain. Semantic datasets are usually conformed to an ontology that is deemed as convenient depending on the purpose the data set was built for, resulting in a huge range of different ontologies being actively used on the Web. To integrate data conform to two distinct ontologies, an *ontology alignment* must be specified. In its most general definition, an ontology alignment is any formal representation of a set of relations between two ontologies [48, 15]; at a more practical level, alignments are ontologies themselves, consisting in a series of bridge axioms between two ontologies. The creation of ontology alignments is a complex and knowledge intensive task, involving domain experts and usually done mostly manually. With the rapid growth of the Web of Data and the equally rapid growth of the number of ontologies, ontology alignments have become critical components of any data integration activity and therefore extremely valuable assets. Over the last few years the Semantic Web community has proposed a wide range of techniques to automate the process of ontology alignment [27, 161]. Most of the proposed approaches rely on a large natural language text corpus to identify shared concepts between ontologies [135, 22, 98], often relying on statistic techniques to spot entity references and, therefore, detect synonyms. Regardless of the employed methodology, ontology alignment is a non-trivial task that must be performed to integrate heterogeneous knowledge bases.

2.4 Wrapping it up

Each one of the approaches to Scientific Literature recommendation described so far in this chapter has its own strengths, but also suffers some clear limitations. The authors of [56] pinpoint the most well-known limitations of the most widespread techniques. Citation analysis presents the following major biases:

- Homograph resolution: typically homographs, i.e. authors with same names cannot be resolved by citation analysis, this can sometimes lead to the impossibility of assigning a research paper to its correct author.

[12]Most notably class disjunction, disjoint properties, symmetrical and asymmetrical properties, and universal quantifiers are not allowed

[13]We address the curious reader to Appendix B for a more detailed discussion of this issue

- not all research papers are listed in citation databases, thus some references will not be resolved and cannot be exploited to filter documents.

- irrelevant citations: some citations just do not provide information, for instance self-citations [164], citation circles [55], ceremonial citations [109], and other irrelevant citations caused by the Matthew Effect, i.e. the fact that very famous works are often cited disregarding their actual relevance to the work presented [111].

Other problems pop up with text-based analysis, which has to cope with unclear nomenclatures, synonyms or context depending on the meanings of words. Accordingly, text-based systems are likely to mis-represent the contents of the documents they are meant to filter.

Knowledge-based systems can overcome the limitations of pure content-base analysis and showcased a great potential, especially when leveraging the vast amount of structured knowledge available in the form of Linked Data. Given the computational issues implied by the usage of ontologies and Linked Data presented in the previous section, it comes with little surprise that most applications do not use such a knowledge directly, but rather build a model upon it. The typical Linked Data usage pattern showed by graph-based recommenders involves extracting a small number of features that appear to have a strong influence over user decisions. This operation, however implies extensive graph traversing, which implies high computational costs. For instance, it took over 2,400 minutes to the authors of [119] to train a state of the art semantics-based recommender system on a data set such as *DBbook*[14] which includes around 70000 ratings from 6181 users on 6733 items and is relatively small when compared to real-world scenarios that may include millions of items and users. Though training is typically a batch-time operation, an excessive complexity may discourage its field usage since in Adaptive Personalisation applications re-training is common because existing users update their preferences, new items are included, and new users sign up as time passes. A more stable model, that does not need to be updated frequently due to user activity is therefore desirable. Regardless of the technique employed to promote access to the documents, the examples surveyed so far in the literature present modelling and filtering solutions that are opaque to the user. In fact, the user is never presented a summary of the reasons why the system considers an item relevant to his or her interests, in other words, the considered systems do not provide explanation. We claim that a more semantic approach could overcome these limitation and in Chapter 4 we will discuss a system that allows users to explore item contents, their own user models, and provides explanation of recommendations in the form of matching key terms lists.

A final substantial limitation of the research work done so far is the lack of extensive and reproducible evaluation on scholarly information access systems [11]. As a matter of fact, most works in the field of scholarly data access either present insufficient evaluation work with respect to their claims or lack reproducibility.

[14] `http://challenges.2014.eswc-conferences.org/index.php/RecSys#DATASET`

3

Understanding text

As introduced in the previous chapter, the key to achieve a satisfactory content-based information access is being able to represent both user information needs and item content in a satisfactory way to be able to provide good user-content matchings. In Section 2.2 we showed how several Information Access applications leverage the textual content of the documents they are intended to provide access to. Such applications use as content representation keywords or n-grams ranked with statistical metrics such as TF-IDF. Being the dimensionality reduction of the considered feature space a key challenge in the development of effective content-based information filtering systems[78], the usage of Information Extraction techniques to limit the content representation to n-grams that are relevant from a semantic point of view is an appealing solution. In this chapter we will describe some *Information Extraction* techniques that can be used to extract useful information directly from the text items to be filtered, thus adding a semantic layer to Information Access applications. We will investigate three complementary aspects of Information Extraction:

- Identifying words and phrases that are relevant in the text.
- Identifying words and phrases that represent entities.
- Investigating relationships among entities and building domain models in an automatic or semi-automatic way.

In this chapter we are providing a brief literature survey of these topics as well as an overview of the results published in [37, 39, 36, 70].

3.1 Keyphrase Extraction

It can be easily argued that in a document not all n-grams have the same relevance, but formalising this idea is a non trivial matter. The TF-IDF metric described in [156] while offering a generally reliable solution can be safely considered a huge simplification of reality since it does not take into account the internal structure and logical flow of

a document, coreference, synonymy, and other critical discourse aspects that are key factors in determining what can be considered as truly representative of a document's content. Moreover TF-IDF provides a ranking of all n-grams, but does not identify a threshold of relevance underneath which n-grams can be safely discarded, in other words it does not discriminate between n-grams and Keyphrases. A Keyphrase (herein KP) is commonly defined as a short phrase typically consisting in one to three words representing an entity or a concept that is somehow representative of the content of a given text. Keyphrases can be either extracted or assigned, in the former case they are selected among all n-grams included in the text, in the latter they are picked from a closed dictionary, or a set of dictionaries, such dictionaries can be built manually by experts or automatically inferred from a document corpus with techniques such as Probabilistic Topic Modelling [14]. Due to the everexpanding scientific vocabulary, KP assignment over scientific literature is likely to require frequent updates to the reference dictionaries and subsequent frequent re-evaluations of the assigned KPs, therefore in the rest of this work we will prefer extraction over assignment. Keyphrases can be therefore seen as n-grams that carry a high relevance inside a document, and their extraction from unstructured text is a trivial, but time consuming task for humans[1], hence the need for automatic techniques to cope with the large volume of textual data available on the Web and stored in digital libraries. The field of automatic Keyphrase Extraction (herein KPE) has been widely explored in the literature and several authors have addressed the problem of filtering document information by identifying KPs. A wide range of approaches has been proposed, the authors of [187] identify four types of KP extraction strategies:

- *Simple Statistical Approaches*: mostly unsupervised techniques, considering word frequency, TF-IDF or word co-occurency [104, 154, 88].
- *Linguistic Approaches*: techniques relying on linguistic knowledge to identify KPs. Proposed methods include lexical analysis, syntactic analysis, and discourse analysis [86, 89].
- *Machine Learning Approaches*: techniques that leverage an annotated text corpus as training data and handle KPE as a classification problem. Machine Learning algorithms such as Support Vector machines are commonly used to this purpose. Systems such as KEA [183] and all its subsequent evolutions [108] belong to this category.
- *Other Approaches*: other strategies exist which do not fit into one of the above categories, mostly hybrid approaches combining two or more of the above techniques [37, 61]. Among others, heuristic approaches based on knowledge-based criteria [95] have been proposed.

All these techniques can be broken down into two steps: a *Candidate Keyphrase Generation* phase wherein the text is processed to extract n-grams that may form meaningful KPs and a *Candidate Selection* step wherein the relevance of candidates is evaluated and the most relevant ones are returned as KPs.

[1] different human annotators however tend to extract different sets of KPs from the same document, mostly due to the ambiguous and highly subjective notion of "relevance within the text".

In the former step the text is commonly split into sequences, words are stemmed or lemmatized [70], and meaningful n-grams that can potentially be KPs are collected. The most widespread approach to generate such candidates includes Part Of Speech (herein POS) tagging of the whole text and, subsequently, the matching of well-known linguistic patterns such as noun phrases, i.e. sequences consisting in nouns only [8, 187, 37]. More refined techniques include considering also non-adjacent words to match the said patterns [62], thus abstracting over stylistic nuances, or exploiting sequence tagging techniques to identify potentially relevant strings of words [33]. Recent work has put a greater emphasis on this first step, providing evidence that considering some linguistic insights during candidate KP generation can significantly increase the recall of KPE [89] and reduce the number of nonsensical n-grams flagged as KPs by the following selection step [70].

The second step of candidate selection can be performed either by computing some metric, such as TF-IDF, and returning only the highest-scoring candidates or by using a classifier. The latter solution implies viewing the problem of assessing the relevance of KPs as a classification one and has been widely investigated in the literature. The most widespread solution consists in evaluating a set of features for each candidate KP, this building a vectorial representation of each candidate KP, and feed with them a classifier, typically a Logistic Regression model [185] or a Support Vector Machine [188]. A broad span of features to identify potentially relevant candidates has been surveyed in the literature and can roughly be divided into four categories [39]:

- *Distributional*: features based upon word distribution in the document and/or in a corpus of documents;
- *Linguistic*: features based upon lexical, morphological, and structural aspects of the considered language;
- *External*: features evaluated with the support of external structured data such as Linked Open Data, domain ontologies, gazetteers, or other knowledge bases;
- *Heuristic*: features built on top of prior knowledge on text structure such as knowing that scientific papers have an abstract where all relevant concepts are likely to be introduced or at least named.

In Table 3.1 a detailed overview of the features used in the literature so far is presented along with the most notable works introducing or surveying them.

Due to KPE being a critical component of several Information Retrieval and Filtering applications, the large number of available tools to perform this task comes with little surprise. However only a small number of these solutions is provided as open source code or with some licence allowing re-training or other forms of customisation needed to fit a specific application domain such as scholarly data. Furthermore, the typical out of the box, open source solution requires major refactoring to be extended or integrated with other systems and components. An example of such a solution is RAKE [154], that though being distributed as an open source package[2], it's a single purpose application with very limited configuration options and rather closed to extension. There also exists

[2]https://github.com/aneesha/RAKE

Table 3.1: Features used in literature to perform candidate KP selection overview.

Feature	Description	Type	Notably used in
Number of words	Number of words in the candidate keyphrase	Distributional	[168, 64, 96, 82]
Number of characters	Number of characters in the candidate keyphrase	Distributional	[103]
Candidate first occurrence (or depth)	First occurrence of the stemmed phrase in the document, counting with words	Distributional	[168, 183, 53, 74, 64, 137, 82, 96, 150]
Candidate last occurrence	Last occurrence of the stemmed phrase in the document, counting with words	Distributional	[82, 150]
Candidate stem first occurrence	First occurrence of a stemmed word of the candidate, counting with words	Distributional	[168]
Normalized phrase frequency (TF)	Frequency of the stemmed phrase in the document (TF)	Distributional	[168, 74, 64, 150]
Relative length	Number of characters of the candidate	Distributional	[168]
Proper noun flag	Candidate is a proper noun	Linguistic	[168]
Final adjective flag	Candidate ends with an adjective	Linguistic	[168]
Verb flag	Candidate contains a known verb	Linguistic	[168]
Acronym flag	Candidate is an acronym	Linguistic	[82, 137]
TF-IDF over corpus	TF-IDF of the candidate in the training text corpus	Distributional	[183, 53, 74, 96, 137, 82]
keyphrase frequency	frequency of the candidate as a keyphrase in a corpus	Distributional	[183, 53, 167, 96]
candidate frequency	frequency of the candidate in the corpus	Distributional	[74]
POS sequence	sequence of the POS tags of candidate	Linguistic	[74], [137], [103]
Distribution of the POS sequence	distribution of the POS tag sequence of candidate in the corpus	Distributional	[82]
number of named entities	number of named entities in the candidate	Linguistic	[103]
number of capital letters	used to identitify acronyms	Distributional	[103]
IDF over document	inverse document frequency	Distributional	[64]
Average TF-IDF	Average TF-IDF score of the words included in the considered n-gram	Distributional	[88, 89]
Variant of TF-IDF - 1	$\log(TF - IDF)$ - see [64]	Distributional	[64]
First sentence	First occurrence of the phrase in the document, counting with sentences	Heuristic	[64]
Head frequency	Number of occurrences of the candidate in the first quarter of the document	Heuristic	[64]
Average sentence length	average length of the sentences that contain a term of the candidate	Distributional	[64]
Substring frequencies sum	sum of the term frequency of all the words that compose the candidate	Distributional	[64]
Generalized Dice coefficient	see [64] or [96]	Distributional	[64], [96], [82]
Maximum likelihood estimate	estimation of the probability of finding the candidate in the document	Distributional	[64]
Kullback-Leibler divergence	see [64, 173]	Distributional	[64]
Document Phrase Maximality index (DPM)	see [64]	Distributional	[64]
DPM X TF-IDF	Document Phrase Maximality index multiplied by TF-IDF score	Distributional	[64]
Variant of TF-IDF - 2	TF-IDF of the candidate normalised on the TF-IDF of its most weighted word (see [64])	Distributional	[64]
k-means of the position	see [64]	Distributional	[64]
GRISP presence	presence in the GRISP database (see [96])	External	[96]
Wikipedia keyphraseness	probability of the candidate to be an anchor in Wikipedia	External	[96]
Title presence	Presence of the candidate in the title	Heuristic	[96]
Abstract presence	Presence of the candidate in the abstract	Heuristic	[96]
Introduction presence	Presence of the candidate in the introduction	Heuristic	[96]
Section title presence	Presence of the candidate in a title of a section	Heuristic	[96]
Conclusion presence	Presence of the candidate in the conclusions	Heuristic	[96]
Reference or book title presence	Presence of the candidate in at least one reference or book title	Heuristic	[96]
Variant of TF-IDF - 3	TF includes the TF of substrings of the candidate	Distributional	[82]
Variant of TF-IDF - 4	TF of substrings of the candidate without the TF of the candidate	Distributional	[82]
Variant of TF-IDF - 5	TF normalized by candidate types (noun phrases vs simplex words vs others)	Distributional	[82]
Variant of TF-IDF - 6	TF normalized by candidate types as a separate feature (not clear)	Distributional	[82]
Variant of TF-IDF - 7	IF-IDF using Google n-grams	External	[82]
Section information	Weight the candidate based on its location (abstract, title, ...)	Heuristic	[137], [82]
Section TF	TF of the candidate in key sections	Heuristic	[82]
Candidate section co-occurrence	Number of sections in which the candidates co-occur	Distributional	[82]
TF Occurrence in titles	Occurrence in the CiteSeer title collection as substring of a title	External	[82]
Occurence in titles	TF of the candidate in the CiteSeer title collection as substring of a title	External	[82]
Semantic similarity - 1	contextual similarity among candidates	Distributional	[82]
Semantic similarity - 2	semantic similarity among candidates using external knowledge	External	[167] (using a search engine)
Variant of Dice coefficient - 1	normalized TF by candidate types (noun phrases vs simplex words...)	Distributional	[82]
Variant of Dice coefficient - 2	weighting by candidate types (noun phrases vs simplex words...)	Distributional	[82]
Variant of Dice coefficient - 3	normalized TF and weighting by candidate types (noun phrases vs simplex words...)	Distributional	[82]
Suffix sequence	Sequence of the suffixes of the words that from the candidate	Linguistic	[137], [82]
Semantic similarity - 3	Co-occurrence based similarity	Distributional	[104]
Variant of TF-IDF - 8	Probability-based (see 3.4 of [167])	External	[167] (using a search engine)
First sentence	First occurrence of the phrase in the document, counting with sentences	Heuristic	[9]
Last sentence	Last occurrence of the phrase in the document, counting with sentences	Heuristic	[9]
Lifespan on words	Difference between the last and first appearance in the document	Heuristic	[150]
Lifespan on sentences	Difference between the last and first appearance in the document	Heuristic	[9]
Wikiflag	Presence of the candidate as a Wikipedia page title or surface (e.g. Big Blue vs IBM)	External	[37]
Noun value	Number of nouns in the candidate	Linguistic	[150, 37, 9]
Pointwise Mutual Infromation	Pointwise Mutual Information of candidate KPs in the document	Distributional	[61]

an open source implementation of the KEA algorithm [183] available online[3], but again, this software is a single-purpose solution with very little customization options, more akin to a proof of concept demonstrator than an actual lybrary or software component. The KEA algorithm is the basis for MAUI[4], which offers an open source implementation of an improved version of the KEA algorithm plus other tools for other common KE tasks such as Entity Recognition or Automatic Tagging [108]. Unfortunately the bulk of such useful features is part of a closed-source commercial product. Moreover, such an application is not meant to be extended, therefore both the implementation of new modules and the integration with existent systems require a deep refactoring of the provided solution, impractical in most real world scenarios. Finally, JATE[5] is a library that offers a set of KP extraction algorithms but its nature of mere collection of algorithms fits more a didactic purpose than a field usage one.

3.2 Named Entity Recognition, Entity Linking, and Word Sense Disambiguation

Keyphrases, though valuable for document representation and summarisation purposes, carry little semantics on their own. To introduce a semantic layer inside applications such as Personalised Information Access systems it is necessary to introduce a semantic layer in the information extraction process as well, enabling the system to have a better understanding of the textual content of the documents by classifying fragments of text or matching them against dictionaries or knowledge bases. Such a semantic layer can be provided by three different NLP tasks: Named Entity Recognition, Word Sense Disambiguation, and Named Entity Linking. *Named Entity Recognition* (NER) can be defined as the task of finding text strings representing entities and concepts in a text [120], which implies classifying words and phrases according to the kind of entity they refer to. The most typical application of NER is the detection of a restricted number of entity classes within a text corpus, like in *SemEval 2017*'s tenth task[6] where three classes, Task, Process, and Material, were to be detected inside a corpus of abstracts extracted from scientific papers. NER tasks are usually seen as sequential prediction problems and are performed leveraging distributional semantics and exploiting sequence tagging techniques like *Hidden Markov Models* (HMM), sequential applications of *Perceptron*, or *Conditional Random Fields* (CRF), with the latter emerging as a de facto standard over the last years, especially in the biomedical domain [90, 136, 94]. More formally, let $x = (x_1, ..., x_n)$ be an input sequence and $y = (y_1, ..., y_n)$ be the corresponding output sequence, the sequential prediction problem is to estimate the probabilities $P(y_i|x_{i-k}...x_{i+l}, y_{i-m}...y_{i-1})$ where k, l, and m are small numbers to achieve tractable inference and avoid overfitting [151]. Such a conditional probability is evaluated in NER using typically large sets of features. According to the authors of [166], three kinds of features commonly used in NER can be identified:

- *Local knowledge features*: features that can be extracted from the word they rep-

[3]http://www.nzdl.org/Kea/download.html
[4]https://github.com/zelandiya/maui
[5]https://github.com/ziqizhang/jate
[6]https://scienceie.github.io/

resent. They include capitalisation, the presence of certain suffixes, prefixes, or special characters, and the presence of sub-tokens such as the ones identified by hyphenation (e.g. the word "non-local" can be broken down into the two sub-tokens "non" and "local").

- *External knowledge features*: features including information that is not directly extracted from the text, but requires some background knowledge, such as linguistic or encyclopaedic one. They include POS tagging, word or phrase clustering analysis over a reference text corpus, and any information gathered by matching the considered word against gazetteers, thesauri, or ontologies.

- *Non-local dependency features*: features that build up on the hypothesis that the meaning a word yields depends on the context wherein it is found, and therefore aim at representing the surrounding phrase, sentence, or discourse. They include the number of times the considered word appears in a certain window, the presence of other significant words within a certain window, context aggregation [26], and preliminary classification provided by another sequence tagging algorithm [87].

The authors of [151] provide significant evidence that all these three kinds of features contribute to the quality of NER and should therefore be considered when designing the feature set to be used to train the sequence tagging algorithm. The classes considered by typical NER applications, however, tend to be few in number and very broad in nature (e.g. the class of people, organisations, or methodologies), however it is desirable for several applications to extend NER to a potentially much larger and fine grained number of classes. *Word Sense Disambiguation* (WSD) can be defined as the task of choosing the right sense for a word within a given context [117]. In this context the matched string is often referred to as the *surface form* of its corresponding sense. While NER matches a text fragment with a class, WSD matches it with a precise dictionary entry, such as a Wordnet[7] synset. This idea can be pushed further by grounding entity mentions to their corresponding node in a Knowledge Base such as Wikipedia or its Linked Data equivalent DBpedia. This task is commonly referred to as *Named Entity Linking* (NEL) [63] in general or *Wikification* when the considered knowledge base is Wikipedia [33, 21]. Both WSD and NEL are usually performed in two steps:

- *Candidate anchor search*: the text is parsed and all the words and phrases that may represent an entity are spotted. This step usually exploits heuristic approaches or very large dictionaries and gazetteers.

- *Entity selection*: among all possible candidates the ones actually referring to an entity are selected and, possibly, linked to the referenced entity. A wide range of techniques is used to evaluate the likeness of a string of referring to an entity and to disambiguate polysemous words according to their context.

The candidate anchor search phase can be further broken down in two passages: the tokenization of text, consisting in the detection of sentence boundaries and of potentially misspelled or mis-capitalized words, and the detection of surface forms within the tokenized text. The latter step can be implemented in several ways, the most widespread

[7] http://wordnet.princeton.edu

approaches include matching against dictionaries or gazetteers of surface forms [21], rule based matching guided by linguistic assumptions, or the usage of NER systems to detect particular classes of entities. More advanced techniques include the combination of the aforementioned strategies into a search pipeline [33], the usage of coreference resolution to map short surface forms, such as acronyms, to longer surface forms with the same dominant label [65], and fuzzy matching techniques [92, 171].

The second step can be performed exploiting either distribution semantics or the ontological structure of a background knowledge base. Distributional semantics approaches consider a reference corpus of annotated text and are trained to recognize surface-sense matches according to the context wherein the surface form is found. In the case of Wikification the collection of articles in Wikipedia is treated as an annotated text corpus where entities are characterized by the presence of hypertext links to other articles. For each surface-sense pair a context representation is evaluated based on word co-occurrences in the training text corpus. When an unannotated surface needs to be assigned a sense or an entity, its current context is evaluated, all possible matches are taken into consideration and the one with the most similar context is assigned [33]. Most Wikification systems use this approach such as DBpedia Soptlight [110] and TagMe [49]. Network based approaches, on the other hand, exploit the internal structure of a knowledge base or, in the case of Wikipedia, the fact that it is possible to build an network of concepts from its textual contents and semi-structured data. With a large and detailed enough network it is possible to exploit such a structure to perform entity selection and disambiguation and, intuitively, the most suitable surface-sense association can be found by searching the one that minimizes the distance with the already grounded terms. This approach relies upon heavy usage of graph search algorithms and clustering techniques: the authors of [169] disambiguate concepts exploiting the degrees of separation between candidate concepts in the reference ontology. The authors of [117, 116], instead, merge Wikipedia with Wordnet, thus using a much larger dictionary than pure Wikification strategies, and perform WSD using random walk with restart on minimum support graphs.

It is important, however to underline how, though sharing a similar methodology, WSD and NEL are driven by different assumptions, more precisely, we can pinpoint the following three differences [63]:

- External knowledge source nature: WSD applications rely on purely linguistic resources such as lexicons and dictionaries, while NEL application feed on domain knowledge provided by domain ontologies.
- Knowledge Source Completeness: WSD applications assume their knowledge base to be complete, i.e. the absence of a potential match for a candidate anchor implies the absence of a sense for that word, while NEL applications assume their knowledge to be incomplete, meaning that all candidate anchors are to be considered entities, even if a match with the entries of the knowledge base is not found [21, 106]. The latter assumption is, however, often ignored by Wikification applications that, in this sense, stand in between WSD and pure NEL[8].
- Candidate search: named entity mentions vary more than lexical mentions in WSD, due to the variety of synonyms, abbreviations, and paraphrases we expect

[8]In fact Wikification can be seen as a bridge between WSD and NEL [117]

to encounter when dealing with domain-specific terminology and the fact that entities that are defined by long and complex names typically tend to be referred in various ways in the same text [174]. The task of candidate search is therefore noisier in the design of NEL systems and there exist evidence in the literature that advanced candidate search techniques such as query expansion based on coreference resolution have a large impact on system performance [63].

Aside from these notable differences, the tasks of NER, WSD, and NEL share a significant conceptual overlap: they often build on similar assumptions and technologies, moreover, they can support similar tasks and applications as well. The NERD framework proposed by [152] addresses the possible overlaps between these three tasks and provides a developer-oriented environment to build said applications.

3.3 Domain Model Extraction

Up to this point in this chapter we discussed techniques to extract Keyphrases from documents and to associate the said KPs to a set of labels representing categories or entries of an authoritative knowledge base, thus representing entities. However, how introduced in the previous chapter, in Section 2.3 some advanced Information Access techniques require some form of domain modelling to populate their knowledge base. Authoritative domain ontologies and data sets should be taken into consideration to act as knowledge base, however they may present serious limitations due to the intrinsic subjectivity and goal-dependency of the domain modelling activity. Automatically extracting a domain model from a text corpus or by merging meta-information provided by different source such as distinct domain ontologies, therefore appears an appealing solution since it allows to abstract over subjective factors and merge the vocabularies used by different authors and communities. This operation, however, implies moving from detecting Keyphrases and key entities to identify relationships among them. In the rest of this section we will introduce two domain modelling techniques relevant to the extents of this work and to scholarly data analysis and access in general:

- *Semantic Similarity Evaluation*: evaluation of an index of semantic similarity between considered terms, this technique allows to quantify the distance between entities included in a text corpus or various kind of meta-information to be found as document annotation. Relationships among the considered entries are not formalised, but rather embedded in the similarity metric.

- *Domain Ontology Extraction*: extraction of a structured model, such as a taxonomy or a more detailed model with multiple relationships among its entries, from a set of text documents or by merging multiple models.

3.3.1 Semantic Similarity

Measuring semantic likeness between items such as words, texts, or DBpedia entities is a vital component of several Artificial Intelligence applications, supporting tasks such as question answering, ontology alignment, Word Sense Disambiguation, and exploratory search. The concept of semantic likeness over the years has attracted the interest of the

Natural Language Processing (NLP), Semantic Web, and Information Retrieval (IR) communities [66]. Two variants have been thoroughly discussed: *Semantic Similarity* which can be defined as the likeness of the meaning of two items, for instance "king" and "president" though not being synonyms have an high semantic similarity because they share the same function, and *Semantic Relatedness* which can be considered as a looser version of semantic similarity since it takes into account any kind of relationship, for instance "king" is semantically related to "Nation" because a king rules over a nation. Due to the high ambiguity of the very definition of these semantic relationships it is not uncommon to evaluate similarity and relatedness metrics upon their performance in a specific, well-defined and reproducible task [18]. A broad range of measures for assessing similarity and relatedness between entities has been proposed in the literature; such measures are grounded into set theory [153], statistics [181], and graph theory [147]. One of the best known semantic relatedness measures is the *Google Distance* [28] which exploits a search engine to estimate pairwise similarity between words or phrases. Such a metric has proven to be effective for a number of knowledge intensive tasks such as evaluating approximate ontology matching [58]. However the implied intensive usage of the underlying search engine makes this metric impractical or too expensive for most applications. Other strategies rely on structured knowledge bases such as taxonomies and ontologies. Wordnet is among the first and still most used resources to estimate semantic similarity with a variety of techniques including graph search algorithms and machine learning. An extensive survey of semantic similarity metrics built upon Wordnet is presented in [18, 19]. More recently, The LOD cloud has also been widely exploited and several authors proposed strategies to evaluate similarity and relatedness among entities included in such a cloud. Most LOD-based techniques rely on the selection of a limited number of features among the multitude of properties present in the cloud, to perform this task techniques such as Personalized Page Rank are commonly used in the literature [147]. These techniques, however involve graph-traversing, thus are particularly demanding from a computational point of view and their usage is typically limited to domain specific application such as the graph-based recommender systems introduced in the previous chapter, in Section 2.3. Wikipedia and its dense link structure have been used as well to compute semantic relatedness metrics: the authors of [54] introduce Explicit Semantic Analysis (ESA), a technique using machine learning to build vectorial representations of Wikipedia items based upon the textual contents of their corresponding articles. The authors of [182] propose an alternative to ESA which leverages the links included in Wikipedia articles to achieve similar performance but at a sensibly lower cost both in terms of computational complexity and of required data. The similarity metric therein presented is the combination of two metrics, one for incoming links and one for outgoing ones, the former one being closely related to the aforementioned Google Distance.

3.3.2 Ontology Extraction

Semantic similarity, though being a powerful mean of identifying related concepts, does not qualify the relationship between the considered entities. Consider for instance the concept of "Information Extraction", it is intuitively tightly semantically related to "Artifical Intelligence" and "Keyphrase Extraction", however the former is a broader concept which includes "Information Extraction", while the latter is a narrower concept,

being as a matter of fact KPE a particular task of Information Extraction. Qualifying the kind of relationships occurring among semantically similar or related entities provides a more detailed domain modelling and Taxonomical relationships (is-a, and equivalence in particular) can be instrumental to several purposes. Roughly three kinds of approach can be employed to detect relationships between terms:

- Natural Language Processing techniques: relationships are inferred from the structure of text wherein entities are mentioned [29]. A notable example of NLP technique is Lexico-Syntactic Pattern Extraction: relationships are inferred exploiting linguistic patterns to be found in the text such as "and other...", "in the likes of...", and so on [68]. These techniques require an extensive text corpus and, since they needs a textual context, cannot be applied to meta-information.

- Clustering techniques: entities can be clustered according to the various contexts wherein they can be found to identify some relationships such as synonymity or taxonomical ones [99]. Several tools such as TaxGen rely primarily on hierarchical agglomerative clustering over large text corpora to identify taxonomical relationships [118].

- Conditional Probability-based techniques: for each considered entity a conditional probability of being associated with the ones included in the considered data set is evaluated, and taxonomical relationships are inferred from such probabilities. The most widespread method of estimating these relationships is using the subsumption method [157], however several alternatives have been discussed, including considering second order co-occurrences computed with a variant of the Page-Rank algorithm [41].

- Graph-based techniques: starting from a simple starting ontology, a complex network is generated by aggregating other ontologies as well as entities extracted from other metadata or text analysis; entities and relationships to be included in the final ontology are then identified by means of spreading activation [184]. These techniques are typically used to extract relevant subsets of larger ontologies and knowledge bases [140].

A notable example of domain ontology extraction tool designed for the scholarly data access domain is the Kink-2 algorithm presented in [142]. Klink-2 identifies three relationships defined by the BIBO ontology[9]: *skos:broaderGeneric*, *contributesTo*, and *relatedEquivalent*, with the latter two being subproperties of *skos:related*. Various techniques are used to identify these relationships: *relatedEquivalent* is inferred by means of hierarchical clustering, while the hierarchical relationships *skos:broaderGeneric* and *contributesTo* are inferred with a variation of the subsumption method, coupled with domain knowledge that leverages temporal information to identify broader and narrower topics. Klink-2 operates over a large variety of data, including scientific publications metadata, textual data, and semantic information gathered from the Linked Open Data cloud. Merging these various kinds of knowledge, Klink-2 can build accurate topic taxonomies given enough data.

[9] `http://purl.org/ontology/bibo/`

3.4 Towards a unifying framework

In this chapter we surveyed several approaches ant techniques that allow to extract from unstructured text information that can be used to build a domain knowldege model. Wrapping it up, we introduced Keyphrase Extraction that allows to identify key the terms that can describe a text, Named Entity Recognition that allows to classify said terms, Word Sense Disambiguation and Named Entity Linking that allow to ground these terms into a pre-existing knowledge base, Semantic Similarity metrics that allow to estimate to what extent key terms or entities are related to each other, and Ontology Extraction that allows to qualify semantic relatedness by identifying specific relationships among entities. As mentioned in the previous sections, for each of these tasks several tools are already available, however they are usually very specific implementations of a single technique or of an ensemble of techniques, tailored for a specific target application, such as the Rexplore system for the Klink-2 ontology extraction algorithm. We can identify three typical pitfalls in state-of-the-art Information Extraction systems that limit their adoption in knowledge intensive domains:

- *Lack of multilinguality*: most of systems surveyed so far work on a single language and that language is English; however digital libraries and the Internet have a lot of content in other languages as well, indeed roughly half of the available Web pages include non-English text [10]. The large majority of Web users are non-English native speakers, and multilingual search, adaptation, and personalization are likely to be key features of future information access, however IE tools are tailored to work on the English language [120] and show, with a few notable exceptions [46, 49], a general lack of multilingual support. This implies, from a knowledge discovering point of view, giving up a priori on roughly half of the knowledge available on the Web.

- *Knowledge Source Completeness*: typically IE systems rely on a specific knowledge source (such as DBpedia, or BabelNet, or a larger knowledge base built aggregating pre-existing ones) acting in a closed-world fashion and assuming that such knowledge source is complete. This assumption is in contrast with the open-world assumption of semantic Web technologies and shows off its limitations when applied to texts such as scientific papers, where new concepts are often introduced. Therefore a more flexible approach open to more than one knowledge source and compliant to the open-world assumption seems more appropriate. Again, some notable exceptions exist in the literature: the Klink-2 algorithm is in principle open to other knowledge sources [142], but the vast majority of concurrent systems are not.

- *Knowledge Overload*: long texts, such as scientific papers, may include a lot of potentially relevant terms, named entities, and relationships among them, but not all are equally relevant. State-of-the-art IE systems currently provide detailed annotation, but do not filter non-relevant entities nor include relevance measures.

At least one of these limitations can be found in any tool surveyed so far, and the primary reason behind it is that none of the presented applications aims at being an

[10] http://w3techs.com/technologies/overview/content_language/all

integrated solution for information extraction. However such a tool would be extremely valuable for knowledge management experts and Information Access systems researchers and practitioners, allowing them to design and deploy custom IE pipelines with minimal effort. On the other hand, available state-of-the-art systems tend to provide a "one-size-fits-all" solution, that is either a very vertical solution tailored for one target application or is a generally domain independent application that does not allow domain-specific business logic to be introduced. To the best of our knowledge, none of the currently available solutions can be easily tailored by non-IE-experts to fit specific domain requirements, assumptions, or constraints of different target applications. Even most of the applications regarded as frameworks are very vertical and far from being friendly for those who do not have an extensive NLP and IE background: for instance LingPipe[11] offers a comprehensive set of Machine Learning algorithms commonly used for IE tasks, however its lack of abstraction over the techniques to be used makes it extremely hard to integrate into other applications, on the other hand The Stanford NLP pipeline[12] is a monolithic application and is a set of tools rather than a framework. The authors of [32] propose *CURATOR*, a NLP framework that allows to annotate text in various ways. However such a framework is far from being an integrated solution and its primary focus is to organise low-level text processing tasks such as sentence splitting, tokenisation, and Part of Speech (herein POS) tagging, lacking support for higher-level tasks such as KP Extraction, NER, or NEL. Leveraging on such a work, the idea of having a sequence of annotators treated as building blocks to achieve complex IE tasks is further pushed in [36] and [9] where the *Distiller* framework is presented. Distiller is an high-level IE framework that can handle several of the high-level IE tasks presented so far in this chapter.

Building on the work described in [39] and briefly introduced in Section 3.1, the design of Distiller is guided by the key principle that several different types of knowledge are involved in the process of IE and should be clearly separated in order to design systems able to cope with multilinguality and multi-domain issues. For example, by now we consider four types of knowledge:

- *Statistical*: word distribution in the document and/or in a corpus of documents;
- *Linguistic*: Lexical and morphological knowledge;
- *Social-Semantic*: Knowledge derived from external sources such as Wikipedia, or more specific domain ontologies, possibly cooperatively developed;
- *Meta-Structural*: heuristics based on prior knowledge on text structure (e.g.: knowing that scientific papers have an abstract).

Linguistic knowledge is language dependant Meta-Structural knowledge is domain dependent, and Social-Semantic knowledge is both domain and language dependant. At a more practical level this principle implies that different types of knowledge must reside in distinct modules, for instance, statistical and linguistic analysis must be handled by different modules.

[11] `http://alias-i.com/lingpipe/`
[12] `http://nlp.stanford.edu/software/corenlp.shtm`

Distiller is organized in a series of single-knowledge oriented modules, where any module is designed to encapsulate a single sub-task, e.g. POS tagging, statistical analysis, knowledge inference, and so on. This allows a highly modular design with the possibility of implementing different pipelines (i.e. sequences of modules) for different tasks. All these modules are required to insert the knowledge they extract on a shared blackboard so that a module can use the knowledge produced by another module. For example an n-gram generator module can generate n-grams according to the POS tags produced by a POS tagger module.

Implementing Information Extraction tasks with Distiller ultimately is reduced to specifying a pipeline including the right annotators. Consider for instance the task of KP Extraction introduced in Section 3.1. Usually such task is divided in the following steps: text pre-processing, candidate KP generation, and candidate KP selection and/or ranking. Distiller allows a quick deployment of such an application with the following annotators: a Sentence Splitter and a word Tokenizer to handle the pre-processing phase, a Stemmer, a POS Tagger and an optional Entity Linker to annotate the text, an N-Gram Generator to generate candidates, and Scoring a Filtering modules to filter the most relevant candidates according to the annotations produced in the previous steps. The resulting pipeline is shown in Figure 3.1. Since each Annotator provides only a specific kind of knowledge, tailoring the pipeline to specific needs requires little effort. For instance, switching to another language requires to replace only the language dependant annotators, namely the POS Tagger, the Stemmer, and the Word Tokenizer. Other pipelines can be specified to implement different Information Extraction and text mining tasks such as NER, NEL, Sentiment Analysis, Summarization, or Authorship Identification, and they can benefit from the same annotators that implement commonly performed sub-tasks. Due to its high flexibility, the Distiller framework was used to perform the bulk of text processing and Information Extraction activities presented in the following chapters of the present work.

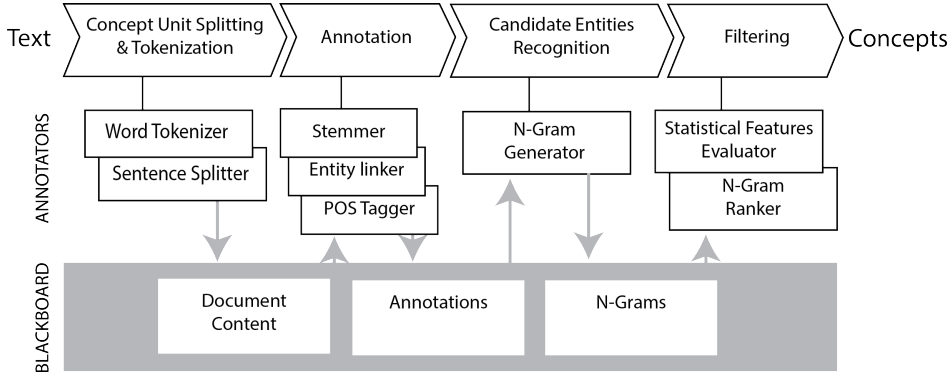

Figure 3.1: A Keyphrase Extraction pipeline in the Distiller framework.

4

Understanding, Filtering, and Explaining

As introduced in Chapter 2, the typical user of a scholarly data access system is interested in discovering new content, possibly recent and not yet well established, thus a content-based approach based on the matching of documents contents with the user interests seems an appropriate solution. State of the art systems, however, behave in a rather opaque way, and do not offer a justification of why a given document is suggested by the system. To address this issue and explore the benefits of explanation, we developed RES, a testbed system that exploits Keyphrase Extraction techniques to extract the main topics of documents and filters CiteSeerX queries according to user research interests, thus offering Personalised Information Retrieval and Filtering. In this chapter we sum up the results presented in [124, 125, 131, 127] and provide a more detailed description of the system's user interface and of the user evaluation process as well.

4.1 Co-occurrence Networks ad Content Description

The core idea of RES is to extract meaningful concepts and topics from the full text of scientific papers by means of Keyphrases gathered with automatic techniques, than use such a content description to build networks of co-occurrence to describe user interests and item contents. Predictions over user-document associations are then evaluated by assessing local similarity between their respective co-occurrence networks. In the RES system, these co-occurrence networks are called *Context Graph* (CG). For each document considered, a CG is built by processing a weighted list of KPs, as shown in Figure 4.1. In the testbed implementation discussed in the next section such a list is weighted using *Keyphraseness*, an index computed by the KP Extraction technique presented in [50]. Other metrics could be used as well as long as they are able to capture the relevance of a certain phrase within the document. We favoured Keyphraseness because it combines statistical, linguistic, and heuristic aspects. User profiles are represented by CGs built from a pool of KPs belonging to one or more documents marked by the user as interesting

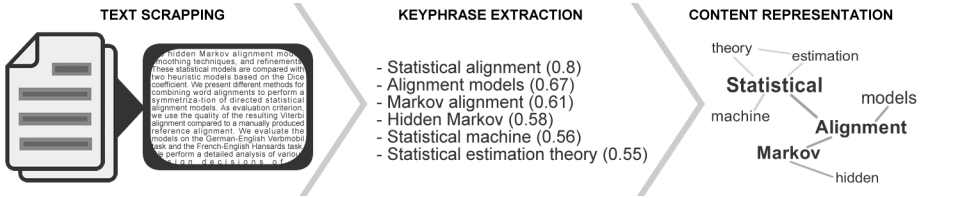

Figure 4.1: Document Keyphrase Extraction and Co-Occurrence Network construction overview.

and, possibly, enriched with other KPs gathered via relevance feedback. CGs are built by taking into account each single term belonging to each KP; each term is stemmed and then represented as a node of the graph; if two terms belong to the same KP, their corresponding nodes are connected by an arc. Both nodes and arcs are assigned a weight which is computed as the sum of the weights of the KPs that generated them and then normalizing such sum. In Figure 4.2 is shown a small CG formed by five KPs.

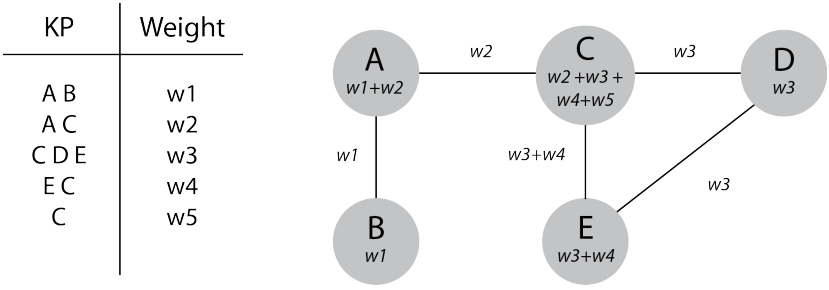

Figure 4.2: A simple Context Graph.

As new KPs are added to the CG, either by direct article insertion or relevance feedback, related concepts tend to link together, creating, in such a way, extensive networks of terms. Consider for example the profile CGs shown in Figure 4.3, the one on the left has been built from four articles dealing with 'Content-based Recommender Systems' and 'Information Extraction'; on the right-hand side two unrelated articles (the first dealing with Machine Learning, the second with Mechanical Engineering) are fed into a profile showing how unrelated concepts form different, non-connected groups. If a user expresses multiple domains of interest in his profile, they will form different groups in the corresponding CG. This fact makes CGs, albeit their simplicity, expressive tools, able to model both short term and long term interests.

CGs allow to create, for each term, a meaningful context of interest by simply checking its adjacency list. If, in two different documents, the same term is used in similar contexts (i.e. in the two respective CGs the same nodes are connected in the same or similar way), it reasonably refers to the same concept, proving a certain degree of similarity between the two items. This mechanism also represents our solution to the problem of disambiguating polysemic terms. When, as result of a user-specified query, a set of documents is to be ranked according to the user's interests, RES extracts a list

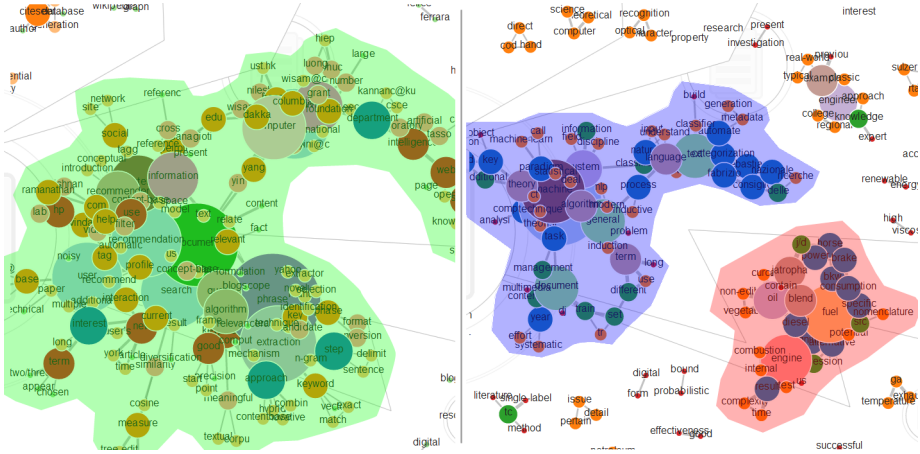

Figure 4.3: A comparison of a CG built from 4 articles dealing with related topics and one built with 2 unrelated articles.

of KPs from each one of the retrieved articles, builds a CG for each KP list and then generates a recommendation list.

4.2 Filtering Algorithm

Recommendation lists are generated in three steps: Matching/Scoring, Ranking, and Presentation, as summarised in Figure 4.4. In the first step every document (D) in a candidate set is matched against the user profile (U) by calculating the following parameters: *Coverage (C)*, *Relevance (R)* and *Similarity (S)*.

C represents the fraction of all the concepts present in D (referred as *totalTerms(D)*) which are also of interest for the user, since they are already included in the profile U (referred as *sharedTerms(D, U)*).

$$C(D,U) := \frac{|sharedTerms(D,U)|}{|totalTerms(D)|} \tag{4.1}$$

R estimates the importance of the concepts shared by the user profile (U) and the document (D). It is computed as the average tf-idf measure of the terms corresponding to the shared nodes between the user and the document CG with reference to the retrieved document set.

$$R(D,U) := \frac{\sum_{i \in |terms(D)| \bigcap |terms(U)|} tf\text{-}idf(i,D)}{|sharedTerms(D,U)|} \tag{4.2}$$

Finally, S is intended to assess the local overlap between the two CGs and to measure how relevant are the shared arcs, i.e. determine how similar are the contexts in which shared terms are used, the stronger the shared association, the higher the score. S is computed by considering the sub-graph of U (ΠU) constituted by nodes shared with D;

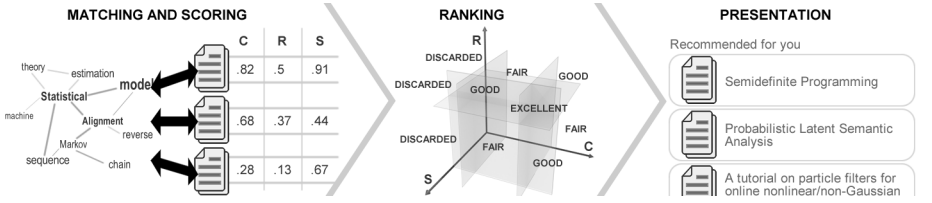

Figure 4.4: Document filtering workflow overview.

the parameter is evaluated as the sum of the weights (w) of the arcs in ΠU (E(ΠU)) which are also included in D (indicated as E(D)) divided by the overall weight of the arcs in ΠU.

$$S(D,U) := \begin{cases} 0 & \text{if } E(\Pi U) = \varnothing \\ \dfrac{\sum_{i \in E(\Pi U) \bigcap E(D)} w(i)}{\sum_{j \in E(\Pi U)} w(j)} & \text{otherwise} \end{cases} \tag{4.3}$$

S spans between 0 and 1. In this way, each document is considered a point in a 3-dimensional space where each dimension corresponds to one of the three above parameters. In the Ranking phase, the 3-dimensional space is subdivided into several subspaces according to the value ranges of the three parameters, identifying in such a way different regions in terms of potential interest for the user. For each dimension, low and high value ranges are identified. High values for all three parameters identify an excellent potential interest, while values lower than specific thresholds decrease the potential interest. According to the combination of the different ranges of the three dimensions, five subspaces have been identified from *excellent* to *not recommended*, and each document is ranked according to where its three-dimensional representation is located. In the current experimental prototype, the interest threshold for each parameter can be adjusted at runtime, for fine tuning of the matching algorithm

4.3 Overview of the RES system

To test our approach, we developed a testbed system called *Recommendation and Explanation System* (*RES*) [124], described in the following. The main goal of RES is providing personalized access to documents retrieved from CiteSeerX. The overall architecture of the system, showed in Figure 4.5, includes a database called *Scientific Paper Collection (SPC)*, a repository for user profiles, and the following three main modules:

- A *Web User Interface* devoted to (i) let the user create and manage profiles, (ii) specify one or more documents of interest to be used as positive relevance feedback, either by browsing a list of articles within the SPC or uploading new ones, (iii) query CiteSeerX, and (iv) request recommendations. The WUI also provides tools to visualise the user model and the recommended contents.

- A *Collection Manager Module*, devoted to: (i) execute queries on CiteSeer and crawl results, (ii) pre-process articles by extracting KPs from full text, and (iii)

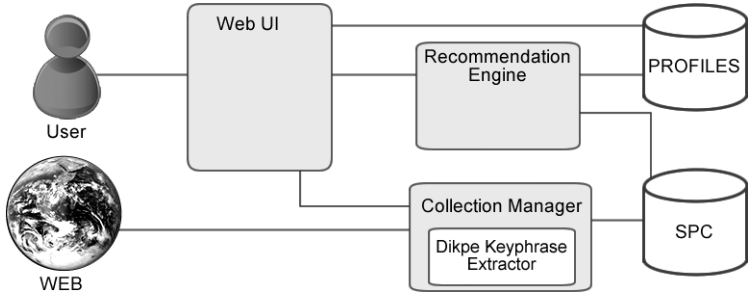

Figure 4.5: System Architecture Overview.

store their representations, as a list of KPs, into the SPC. This module has been developed using the Distiller KP extraction pipeline described in [51, 9], which was favoured for its good compromise between accuracy and ease of use. The KP extractor provides, as output, a list of KPs extracted from the document where each KP has a weight called Keyphraseness. The higher the Keyphraseness, the more relevant is the KP in the document.

- A *Recommendation Engine Module* devoted to: build and maintain individual user profiles; retrieve from the SPC the set of documents returned by a query, and then recommend the most promising papers.

The SPC, though not being strictly necessary to compute recommendation lists, is a critical part of the system since Keyphrase Extraction is, computationally speaking, a demanding task and a set of hundreds of query results cannot be processed in an interactive way. In particular, being the Keyphraseness index a combination of several statistical, linguistic, and heuristic features [1], KP extraction from a long text such as a journal paper may take several seconds, making the real-time processing of retrieved documents impractical. In order to address this issue, we decided to let RES process retrieved documents only once, in an asynchronous way, and cache their representation for later use. On the other hand, once the document KPs are known, the recommendation algorithm proposed is very efficient and it is able to rank large sets of documents in real time.

4.3.1 Presentation and Explanation

Being RES an highly interactive system, where the user is expected to visualise different kinds of information and to provide relevance feedback, Web user Interface is a crucial component and herein we provide a brief description of its key features as well as some screenshots.

The RES WUI basically offers two notable features: user model management, and recommendation lists explanation. The former includes user model construction by specifying a list of relevant papers, model visualisation, and model tuning and customisation. As relevant papers are chosen to be included in the user model, their KPs are

[1] namely, with reference to Table 3.1 Candidate First Occurrence, Candidate last Occurrence, POS Sequence, Lifespan on Words, Wikiflag, Noun Value, and Normalized Phrase Frequency.

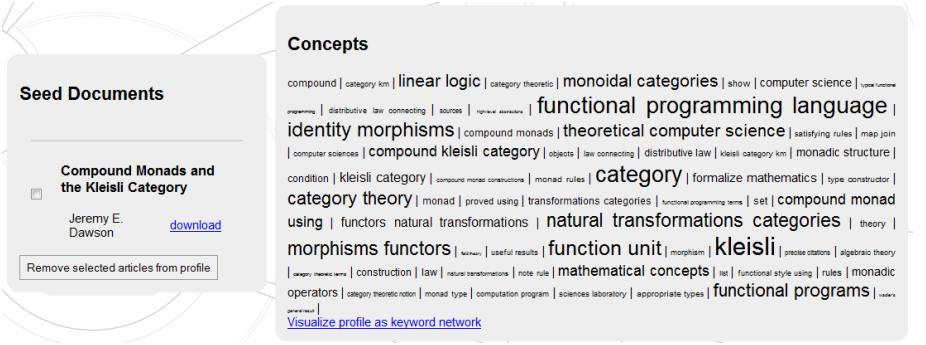

Figure 4.6: A user model built from a single document represented as a tag cloud.

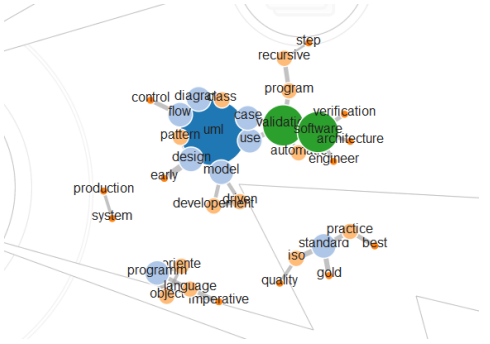

Figure 4.7: A user model represented as a graph.

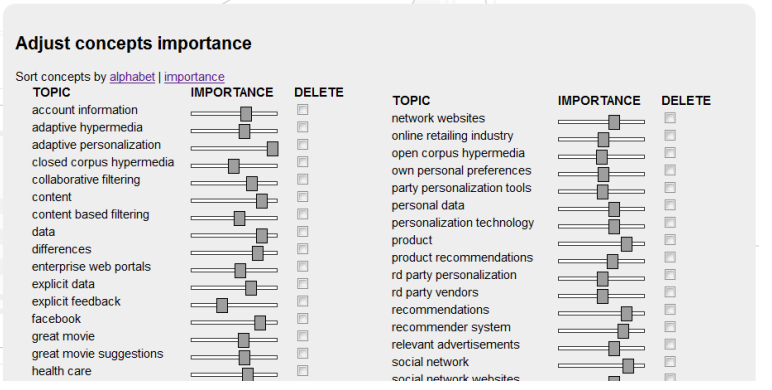

Figure 4.8: User model tuning interface.

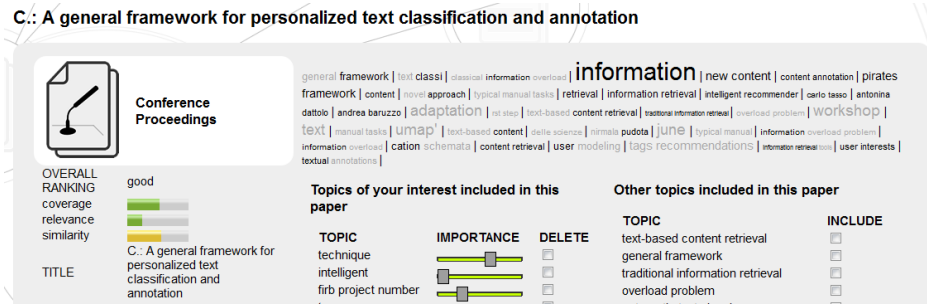

Figure 4.9: Recommendation screenshot.

shown to the user as a tag cloud where the font size is proportional to their importance, as shown in Figure 4.6. Alternatively, the underlying CG built by the system can be shown to highlight the various connected components of such a representation, as shown in Figure4.7. Finally, it is possible to fine tune the relevance of the KPs included in the user model by adjusting a set of sliders, as shown in Figure 4.8.

The latter feature is shown when re-ranked search results are presented to the user. Retrieved documents are presented as a ranked list where each one is represented by descriptive card and the top items are those that better match the user profile. Each document descriptive card includes some metadata, a tag cloud of concepts contained in the document, and an interface to provide relevance feedback, as shown in Figure 4.9. The tag cloud serves two goals: it provides an overview of the document's content and explains why a document was recommended by highlighting concepts that match the user interests. The lower part of the card, instead offers the user a way to provide relevance feedback. Users can explicitly adjust the weight of concepts already included in their model, deleting them or adding new ones.

4.4 Evaluation

Different evaluation activities have been performed in order to assess both system performance and user satisfaction. In this section we are presenting the benchmark evaluation with other more established information filtering techniques, and a user evaluation aimed at understanding the benefits of explanation and user model scrutiny.

4.4.1 Benchmark Evaluation

In order to evaluate the algorithm and compare it with state of the art recommenders, we needed a data set meeting all the assumptions under which all the considered algorithms may work correctly. Unfortunately, there is no public available dataset of scientific papers including enough user ratings to make a collaborative filtering strategy able of recommending a significant part of the data set. Taking into account the lack of an adequate data set in the field of scientific literature, we decided to exploit a data set focused on a different domain, e.g. a movie data set. However, we believe that this choice is acceptable, since the proposed content based approach is independent

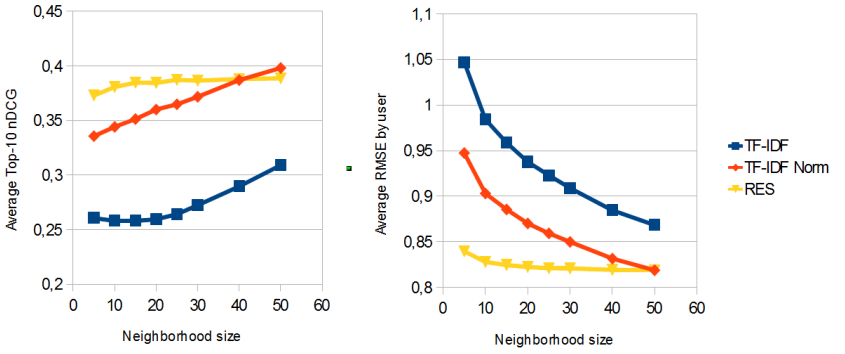

Figure 4.10: Comparison of accuracy (RMSE) and information gain (top-10 NDCG) between the RES algorithm and two TF-IDF based techniques.

from the specific domain of the natural language texts considered. Following this line of reasoning, a subset of the Movielens dataset containing 100 items and 1113 users was considered. Each item had a set of user-generated keyphrases and each user a non-empty set of expressed ratings. At first, the RES algorithm was benchmarked against the most widespread content-based technique: TF-IDF, considered both in its simple, naive implementation (simply labelled as TF-IDF) and in a more sophisticated form, taking into account user rating normalization. These techniques were used to compute, for each item, a set of neighbour items ordered by descending similarity value. Different content-filtering techniques provided different neighbourhoods. Such neighbourhoods were then used, in an item-item fashion, to predict a personalized score for the target item for any user. Intuitively, the larger the neighbourhood, the higher the chances that all the items actually similar to the target one are included, moreover false positive neighbours may introduce noise, reducing the accuracy of the prediction. Items with an high predicted score were then recommended to users. These predictions and recommendations were then compared to the ones generated, on the same data set, by three collaborative techniques: a knn user-user filtering, an item-item filtering and an SVD collaborative recommender. All the collaborative-filtering algorithms were tuned to work with an optimal number of neighbours or latent semantic features. The implementation of the baseline systems was provided by the LensKit framework [45], and the TF-IDF implementation was provided by Apache Lucene. Hidden-data analysis was performed taking into account accuracy and information gain metrics such as root mean squared error (RMSE), evaluated on rating and user basis, and Normalized Discounted Cumulative Gain (nDCG) evaluated on the whole set of recommended items and on the top 10 items of the list (the ones more likely to be consumed by the user). The RES algorithm proved to be able to perform well even with a very small neighbourhood, converging quickly towards accuracy and information gain values that other content-based algorithms reach only when considering very large neighbourhoods (forty or more) as shown in Figure 4.10. Such large numbers of neighbours imply that, to generate meaningful recommendations, a large quantity of data is needed, which may be not always available.

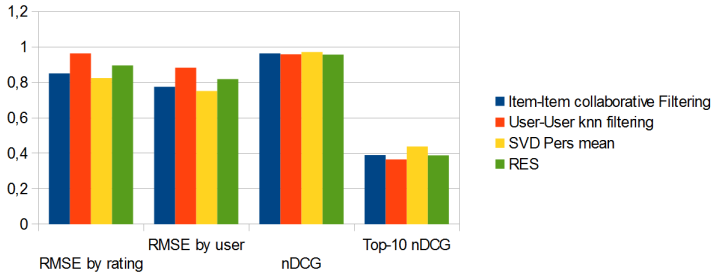

Figure 4.11: Comparison of accuracy and information gain between the RES algorithm and some of the most widespread CF techniques.

Benchmark against collaborative filtering algorithms proved RES to be on par with the most widely used techniques, performing slightly worse than SVD-based techniques, almost on par with item-item collaborative filtering, and slightly better than user-user filtering as shown in Figure 4.11. However, while showing similar levels of performance, RES has the advantage of using an understandable user model, allowing explanation of recommendations, whose lack is one of the major drawbacks of collaborative techniques and SVD-based techniques in particular. However It is important to point out how our system, being content-based, does not need rating data to provide recommendations.

4.4.2 User Evaluation

User tests have been performed with 30 volunteer master degree and PhD students who were asked to use the system in their ordinary research and study activity for a period of two weeks and to fill three anonymous questionnaires and, after the end of the test period, to undergo an informal interview. The first questionnaire was proposed at the beginning of the evaluation period, right after a demonstration of the system, the second after the first week of free usage, and the last at the very end of the evaluation period. All of them were designed according to the ResQue evaluation framework [149]. The questionnaires presented to the users included 15 questions from the framework[2], aimed at assessing the perceived quality of recommendation and users' behavioural intentions towards the system. Four perceived recommendation quality aspects were considered:

- Accuracy: how much accurate and relevant the proposed results are perceived. This aspect does not take into consideration if the recommended items are novel or diverse, but only if they can be regarded as related to the user's interests.
- Diversity: how much diverse each other the proposed results are perceived. This aspect is orthogonal with the Accuracy, since it does not consider if the proposed items are actually relevant for the user.
- Familiarity:how much familiar the proposed results are perceived, an high famil-

[2]Namely, the questions we asked to our users are the ones identified in the Resque paper by the following codes: A70, A11, A20, A17, A50, A13, A16, A17, A60, A70, A42, A30, A41, A82, and A83.

Table 4.1: Average values for user perceived quality and behavioural intentions at the end of the test period.

	Perceived quality				Behavioral intentions	
	Accuracy	Diversity	Familiarity	Novelty	Continuance and frequency	Recommendation to others
Explanation	2.65	3.6	2.1	3.1	2.75	2.7
No Explanation	2.55	3.25	1.85	3.3	2.55	2.7

iarity may indicate either the recurring presence of some items in the recommendations or the inclusion of already consumed or known items.

- Novelty: the perceived ability of the system to provide users with new content.

Also two behavioural intention aspects were considered:

- Continuance and frequency: how often and consistently the users are willing to use the system in the future.

- Recommendation to others: to which extent users would recommend using the system to friends and colleagues.

All of the questions allowed answers on a 1 to 5 likert scale and the test subjects were asked to compile the questionnaire online, with no interaction with the experimenters to avoid expectancy bias.

The test subjects were split into two groups: the first was provided with the full functionalities presented in Section 4.3.1, while the second was given access to a version of the system that to did not provide explanation[3], but still allowed to explore the user model. Though after the first two questionnaires there appeared to be no significant differences in the answers of the two groups, by the end of the test period the final questionnaire showed a significant discrepancy, with the users who were given access to explanation features expressing an higher perceived recommendation quality, though the underlying algorithm being the very same. The average values at the end of the test period for the four considered perceived quality aspects and the two behavioural intentions considered are shown in Table 4.1. It can be observed that though a presentation of the item contents and the highlighting of the concepts matching the user model makes them appear less novel, it raises all the other three perceived quality aspects. Behavioural intentions also appear to benefit from the presence of explanation, in particular the continuance and frequency of use dimension appears to improve significantly, while the average attitude towards suggesting the system to other people remains unchanged. Looking at the distribution of the 1-5 scores given by the users who used the explanation-enabled version of the system (shown in Figure 4.12), test results highlight an overall good user satisfaction and the ability of the system to differentiate recommendations enough to let users discover many novel items. Particular attention was also put in assessing the quality of explanations and questionnaires results showed how most of user perceived them as sound and satisfying, while only a very small fraction found them annoying, useless, or confusing.

[3]Document descriptive cards did not highlight matching concepts

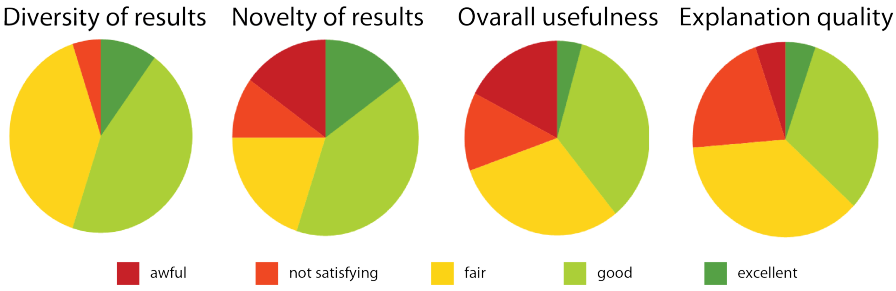

Figure 4.12: Summary of user perceived recommendation quality.

Finally, interviews highlighted how most of the participants who were given the system without the explanatory features lost interest in the system in a matter of a few days and shifted back to more traditional tools to accomplish their everyday tasks. Most notably, explanation appears to motivate users into spending more time managing and tuning their profile to obtain better results. On the other hand subjects who were given access to explanatory features learned how to interpret them and were more engaged with the system.

4.5 Lessons Learnt

Wrapping up, the RES system by employing Information Extraction techniques manages to interpret the textual content of papers, and even a single document can provide enough information to fuel a content-based filtering algorithm. According to our experiments, the extraction of concepts coupled with a matching technique that does not strictly require exact KP match can produce good recommendations even when users provide little information about their preferences and the explanation appears to be an highly appreciated feature. These results allow us to withdraw the following conclusions:

- Importance of Information Extraction: Natural Language Processing proved to be instrumental in representing document content in a detailed, unbiased, and cost effective way.

- Expecting an exact match may be too much: partial matches allowed by our graph-based content modelling can substantially ease the cold start problem, thus allowing the generation of meaningful predictions even when little information is available.

- Relevance of explanation: allowing the users to scrutiny their models and offering them insights on the documents contents and how they relate to their interest appears to be a relevant factor in establishing user satisfaction.

Therefore we can state that in the general framework of a Curator System for scholarly data access it is instrumental to include modules that can handle Information Extraction from unstructured text, non-exact matching, and explanation of results. Though the

single components presented in this chapter are rather simple on their own, their integration allows to achieve both a good information access and a satisfying user experience. In Chapter 6 we will introduce a more sophisticated non-exact matching technique that leverages a background domain model and that can, in principle, seamlessly replace the one presented in Section 4.2.

RES, however presents one major drawback: it is bound to a single scholarly data repository, namely CiteSeerX, while we would like a curator system to access multiple sources, possibly in an informed way and supporting the user in identifying research trends and directions that emerge from the literature. The next chapter is devoted to the discussion of this problem.

5

Monitoring the Research Community

In the previous chapter we discussed a content-based technique to provide personalised scholarly data access and offer them a concise explanation of why the proposed documents might be relevant to their interests. However, personalised document access is not the only feature we would expect from a comprehensive scholarly data access solution. As discussed in [143], making sense of large volumes of research data, thus providing the user with a bigger picture of his or her domain of interest. We would like our Curator System to "know" the various venues available such as journals and conferences, and address the user to the ones that suit his or her interest the most. To achieve this goal we propose in this chapter a content-based strategy, built exploiting the Distiller Information Extraction framework, to perform the following tasks:

- Identification of clusters of related venues and communities.
- Description of the evolution of the topics of interest of a venue and its relations with other venues over time.
- Identification of complementary venues and communities.

The solutions described in this chapter were introduced in [121, 122].

5.1 Scholarly Data Network Analysis in the Literature

Aside from the aforementioned Rexplore system [143, 139, 141] that leverages scholarly meta-data to offer an integrated solution wherein users are supported in a large variety of sense-making tasks, several other authors in the literature addressed the problem of extracting insights from the activity of a community. To fulfil such a task two kinds of information can be taken into consideration: social interactions and connections among

the people forming the observed community and the contents of the various outputs produced by the community.

5.1.1 Social Network Analysis

The study of the connections between people and groups has a long research tradition of at least 50 years [7] [178] [160] [179]. Social Network Analysis, herein SNA, is a highly interdisciplinary field involving sociology, psychology, mathematics, computer science, epidemiology, etc. [145] Traditional social networks studies have been performed in many fields. The traditional approach towards SNA consists in selecting a small sample of the community and to interview the members of such sample. This approach has proved to work well in self contained communities such as business communities, academic communities, ethnic and religious communities and so forth [132]. However the increasing digital availability of big data allows to use all the community data and the relations among them. A notable example is the network of movie actors [180] [1], that contains nearly half a million professionals and their co-working relationship [134].

Academic communities are a particularly interesting case due to the presence of *co-authorship* relations between their members. Several authors in literature have analysed the connections between scholars by means of co-authorship: in [132] [133] [134] a collection of papers coming from Physics, Biomedical Research, and Computer Science communities are taken into account in order to investigate cooperation among authors; in [7] a data set consisting of papers published on relevant journals in Mathematics and Neuroscience in an eight-year period are considered to identify the dynamic and the structural mechanisms underlying the evolution of those communities. Finally, the authors of [145] consider in their analysis the specific case of the SNA research community. Several authors proposed systems and frameworks to provide a better description of the social interactions occurring among members of the research community, the Semantic Web community in particular has provided several interesting solutions over the past years. *VIVO* [85] is a project of Cornell University that exploits a Semantic Web-based network of institutional databases to enable cooperation between researchers and their activities. The system however is quite "ad-hoc", since it relies on a specific ontology and there is no automatic way to annotate the products of research with semantic information, requiring in such a way a huge preliminary effort to prepare the data. Another SNA tool that is used in the academic field is *Flink* [114]. The system performs the extraction, aggregation, and visualization of on-line social networks and it has been exploited to generate a Web-based representation of the Semantic Web community.

5.1.2 Content-based analysis

Other authors in the literature have tackled the problem of discovering relationships by looking at the textual content of documents produced or consumed by the academic community. In [80] the problem of content-based social network discovery among people who appear in *Google News* is studied: probabilistic Latent Semantic Analysis [72] and clustering techniques have been exploited to obtain a topic-based representation. Another example of content-base analysis is presented in [105] where is proposed a system that exploits the full text of email messages between scholars is to infer connections between them. The authors claim that the relevant topic discussed by the community

can be discovered as well as the roles and the authorities within the community. The authors of [155], instead, perform text analysis over the Usenet corpus to achieve the same goals, however their tool is an exploratory system that serves for visualization purposes only. Finally the authors of [172] introduce a complex system for content-based social analysis involving NLP techniques which bears strong similarities with our work. The deep linguistic analysis is performed in three steps: (i) concept extraction (ii) topic detection using semantic similarity between concepts, and (iii) SNA to detect the evolution of cooperation content over time. However the approach relies on a domain ontology and therefore cannot be applied to other cases without an extensive knowledge engineering work.

5.2 Proposed Methodology

In order to support our analysis a testbed system was developed to access documents, integrate the keyphrase extraction system presented in [39], and aggregate and visualize data with purposes of inspection and analysis. Our approach is twofold: we take into ac-

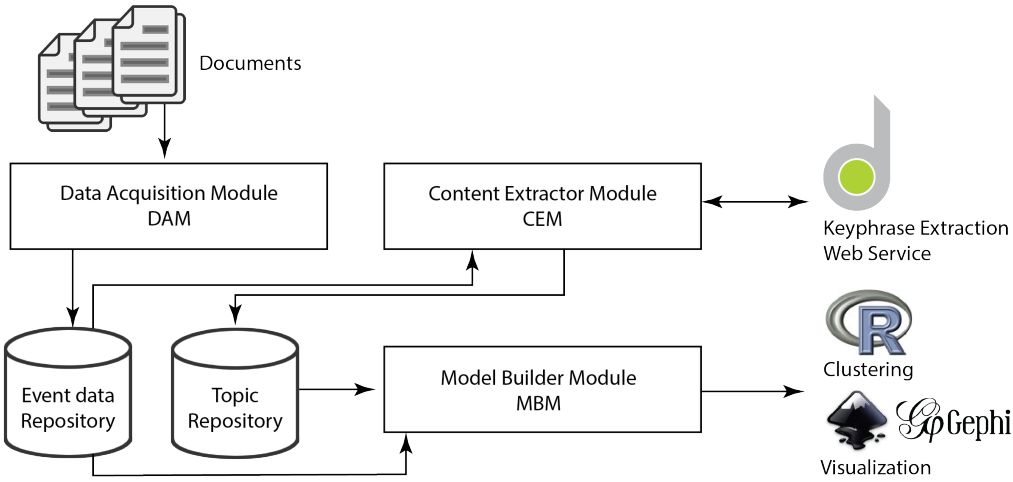

Figure 5.1: System architecture overview.

count social connections between events, considering the authors who contributed, and the semantic connections, analysing the topics discussed. These two different perspectives are then used to get a better overall picture of the considered research community.

The testbed system is constituted by three modules: the Data Acquisition (DA) module, the Content Extraction (CE) module, and the Graph Builder (GB) module, as shown in Figure 5.1. The DA module reads the considered documents and populates the Event Data repository, that contains the list of considered events and their related data including contributing authors, venue, date, and links to full text papers. The CE module retrieves the full text of each considered paper and acts as interface for a Keyphrase Extraction system built with the Distiller framework. Such a system extracts

a set of meaningful KPs from each article's full text using the algorithm described in [39]. Keyphrases identify relevant concepts in the document and each of them is associated with an estimated relevance score called *Keyphraseness* already introduced in the previous chapter and described in [50]. Associations between KPs and papers are then stored in the Keyphrase Repository. The GB module, finally, handles the creation of the network models: the SNA-based one and the Content-based one. Clustering and Visualization are handled by external tools such as R and Gephi.

The SNA part of our study is performed by exploiting established and well known methods: an *Author Graph* (AG) is built where events are nodes and the fact that two events share some authors is represented by an undirected link between the corresponding nodes. Nodes are weighted according to the number of authors involved in the corresponding event, links are weighted proportionally to the number of authors shared. Communities of similar events in the graph are then identified applying the Girvan-Newman clustering algorithm [57] which allows to cluster events corresponding to well connected communities.

The Content-based part of the study, instead, is performed in a novel way: the usage of automatic KP extraction allows us to model the topics actually discussed in a conference and to group events according to semantic similarities. For each considered event, all the accepted papers are processed creating a pool of *event keyphrases*, where each keyphrase is associated to the *Cumulative Keyphraseness* (CK) i.e. sum of the related keyphraseness values in the considered documents, as shown in Formula (5.1).

$$CK(k, event) = \sum_{paper \in event} Keyphraseness(k, paper) \tag{5.1}$$

By doing so a topic mentioned in few papers, but with an high estimated relevance, may achieve an higher CK than another one mentioned many times but with a low average estimated relevance. For each keyphrase an *Inverse Document Frequency* (IDF) index is then computed on event basis, namely we compute the logarithm of the number of events considered divided by the events in which the considered keyphrase appears, as shown in Formula (5.2).

$$IDF(k) = \log \frac{|AllEvents|}{|EventsContainingKPk|} \tag{5.2}$$

Intuitively, the larger the IDF, the least events are characterized by the considered keyphrase. When a keyphrase is relevant in all the considered events, its IDF is zero. Such value is then combined with the CK, as shown in Formula (5.3) to create, for each KP in each event a $CK - IDF$ score.

$$CK - IDF(k) = CK(k) * IDF(k) \tag{5.3}$$

The $CK - IDF$ score promotes keyphrases that are relevant within an event and, at the same time, not widely used throughout the whole set of considered events. This measure behaves in a manner that closely resembles the well known $TF - IDF$ measure.

Subsequently, a *Topic Graph* (TG) is built, where events are represented by nodes and the fact that two events share some keyphrases is represented by an undirected link

between such nodes. Nodes are weighted according to the number of different keyphrases extracted from their papers, and links according to the sum of $CK - IDF$ values of the keyphrases shared between two events. Communities of similar events in the graph are then identified, as in the previous scenario, with the Girvan-Newman clustering algorithm.

Finally, complementary communities analysis is performed by comparing the AG and the TG: events that are connected in the TG and have no direct connections in the AG are potentially complementary communities. To detect such situations a simple metric called *complementarity*, shown in Formula (5.4) is proposed.

$$\text{Complementarity(e1, e2)} = \text{TopicSimilarity(e1,e2)} - \text{AuthorSimilarity(e1,e2)} \quad (5.4)$$

Positive values suggest that the considered events bear a string topic similarity and a low author similarity, meaning that, even though the topics discussed are similar, the contributing authors have little or no overlap.

5.3 Results

In this section we present three case studies on research community analysis. In the first part of the section we present the analysis performed on the CEUR events published in 2014. The goal of such an analysis is to detect clusters of events that represent the meeting points of a specific research community (e.g. the Semantic Web community, the Recommender System one, the Digital Libraries one, and so on) and to identify groups of events dealing with similar or complementary topics. Once research communities are identified it is possible to further investigate their activities by analysing the evolution of the topics dealt with in the published papers. In the second case study we are considering, aside from 2014, also the CEUR volumes published during 2013 and 2012, and we are comparing the observable venue clusters. In the third case study we are limiting our focus to a single venue, the Italian Research Conference on Digital Libraries by considered a data set which includes the proceedings of ten IRCDL editions.

5.3.1 Single year CEUR proceedings analysis

The first case study is based upon 2014 CEUR volumes, upon which both social and semantic analysis are performed, thus generating both an AG and a TG. The considered data set contains all CEUR volumes published before December the 1st 2014 that are proceedings of events held during 2014; it consists in 135 events with over 8400 contributing authors and over 2000 accepted papers.

To get an overview of both the AG and the TG, we are considering five features: the number of edges, the average degree, that is the average number of outgoing edges for each node, the network diameter, that is the longest path in the graph, the graph density, that is a measure of how well connected the graph is, spanning between 0 (all isolated nodes) and 1 (perfectly connected graph), and the average path length, that is the average length of a path connecting two distinct nodes. The number of nodes is omitted because we are assuming that each event is represented by a node and therefore their count is 135 in both cases.

At first glance the AG presents a sparse network structure, with a very low density as shown in Table 5.1, with a few isolated nodes, meaning that relatively few authors contribute to more than one conference and some events do not share authors with the others.

# of edges	Average degree	Network diameter	Graph density	Average Path length
405	6	8	0.045	3.078

Table 5.1: Author Graph global statistics.

Figure 5.2 shows a visualization of the AG in which the size of the nodes is proportional to the number of authors who contributed to the event, and the colour depends on the *betweenness centrality* of the node (namely the number of shortest paths containing that node); edge size is proportional to the number of authors who contributed to both the events connected by the edge and edge colour depends on the betweenness centrality. Nodes and edges with a high centrality are red, while low centrality ones are blue. The centrality value allows to identify the events that serve as hubs for different communities: events with a high centrality, in fact, might be interdisciplinary meetings where members of otherwise distinct communities get together. On the other hand, events with a low centrality might be more focused and therefore interested only for the members of a single community.

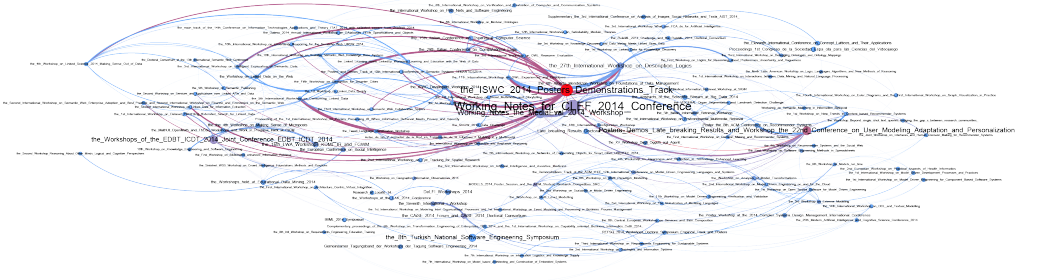

Figure 5.2: Overview of the Author Graph.

It can be noticed how the largest event in term of contributing authors (CLEF 2014) is not the most central one from a network perspective (which is the ISWC 2014 Poster and Demo Session), few events have a high centrality and some of them are relatively small in terms of number of contributing authors (such as the Workshops, Poster, and Demo Session of UMAP 2014), and, finally some large events in terms of contributing authors have an extremely low centrality (such as the Turkish Software Engineering Symposium or the International Workshop on Description Logics), meaning that they serve as the meeting point of a relatively closed community rather than a point of aggregation for diverse research areas.

In order to identify groups of events representing meeting points of wide research communities, a clustering step is performed, removing edges with a high betweenness centrality value. By doing so only groups of strongly interconnected events remain connected. The result of the clustering step is shown in Figure 5.3, where all the

isolated nodes are omitted.

Figure 5.3: The three main clusters in the Author Graph.

Three main clusters can be observed: the first and largest one groups, with little surprise, the ISWC 2014 Poster and Demo Session which is clearly a massively aggregating event, with all its co-located events and other Semantic Web related events as well; the other two clusters are much smaller and revolve around CLEF 2014 and the Workshops, Poster, and Demo Session of UMAP 2014. However, due to the sparsity of the graph, most of the events cannot be clearly clustered and therefore other kinds of correlations between events should be considered to get a better picture.

The TG, on the other hand is, as shown in Table 5.2 much more dense with a graph density of 0.94 and a diameter of 2. These data highlight how the papers presented at the considered events share a common lexicon, which is an expected result, since CEUR publishes only computer science proceedings. The generated TG is therefore extremely

# of edges	Average degree	Network diameter	Graph density	Average Path length
8543	126.56	2	0.94	1.041

Table 5.2: Topic Graph global statistics.

well connected and, considered as-is, it does not provide useful insights.

After pruning low-weight edges, representing the sharing of low $CK - IDF$ terms between two events, and application of the Girvan-Newman clustering technique we obtain the clusters shown in Figure 5.4 which are significantly different from the ones obtained by analysing the AG. There is a higher number of clusters and, even though many events remain isolated, more events are grouped in a cluster. The largest cluster includes two of the most central events, namely CLEF and UMAP, meaning that, although merging different communities, they deal with similar or tightly related topics. ISWC, the most central event in the AG, however, in the TG is included in a relatively small cluster in which only few of its co-located events appear. The majority of the events that are included in the ISWC cluster in the AG are, indeed, in the TG included in the UMAP/CLEF cluster or form a cluster on their own, like the ISWC Developers' Workshop and the LinkedUp Challenge. Several other small clusters are present, representing topics discussed only by a handful of events.

One final interesting insight about what research communities actually debate can be obtained by looking at the extracted concepts with the lowest IDF, which means the most widely used in the considered data set. They are listed in Table 5.3. Since we used

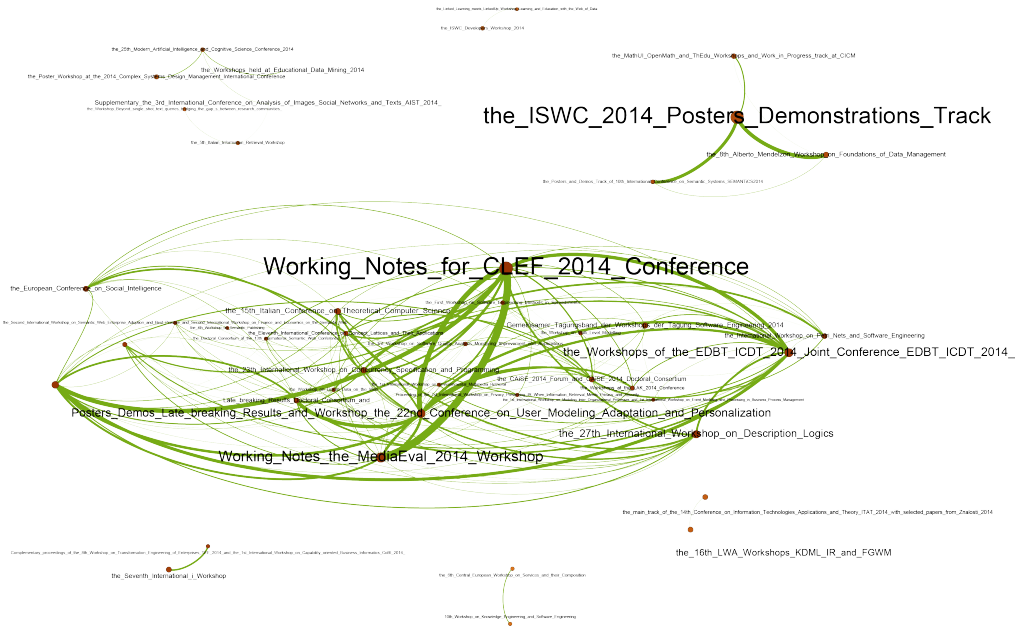

Figure 5.4: Clusters obtained from the Topic Graph.

the logarithm to the base 2, an IDF of 1 means that the considered concept is relevant in half of the considered conferences, and with an IDF of 0.5 in about 2/3. Even though all these concepts are relevant in most of the analyzed papers, their extremely broad adoption makes them nearly irrelevant when considered for differentiating and grouping events according to the discussed topics.

Most of these concepts are, as expected, very generic (such as “System” or “Model”) in the field of Computer Science and Information Technology (to which all the considered events belong), however some of them are very specific and usually associated with a precise research community, such as Semantic Web, Machine Learning, and Natural Language Processing. Semantic Web, in particular, appears in almost half of the considered events, even if the Semantic Web research community identified by cluster analysis is far from including half of the considered events.

5.3.2 Three-Years analysis

In this section we present and compare the results of our two analysis on the last three years of CEUR volumes. Only proceedings available on CEUR were considered, therefore, for conferences such as UMAP, ISWC, and CAiSE we are considering only the part of their proceedings published on such site (usually poster and demo session, workshops, and doctoral consortia). For each year, the AG and the TG are presented, picturing the evolution of the various communities active in the computer science field. Evaluated clusters as well as some events relevant for their size and/or centrality in the network are highlighted.

Figure 5.5 and Figure 5.6 represent our model of the 2012 events which proceedings

Topic	IDF
system	0.427
model	0.474
data	0.601
information	0.671
computer science	0.700
semantic web	1.076
language	1.144
web	1.144
semantics	1.191
software engineering	1.241
natural language processing	1.267
machine learning	1.267

Table 5.3: Most commonly extracted keyphrases ranked by their IDF.

were published on CEUR. The clusters obtained in the AG and in the TG are notably different: in the AG we can observe several clusters, while in the TG most of the events belong to two very large clusters with one of them clearly including all Semantic-Web related events. Several events, like ISWC and SEMANTiCS, in the AG belong to different clusters, while in the TG are found in the same one, meaning that they are potentially complementary communities. The UMAP conference is an outlier (i.e. is not included in any cluster) both in the AG and in teh TG, meaning that in 2012 UMAP hosted contributing authors who did not contribute extensively to other CEUR events and dealt with different topics from the vast majority of the other CEUR events.

Figure 5.7 and Figure 5.8 represent our model of the 2013 events. Again, the AG presents more clusters than the TG, however the four clusters found identify different groups of events: with respect to Figure 5.8 the upper one includes Data Science related events, the lower one Software Engineering related events, the eastern one E-Learning related events, and the western one theoretical Computer Science related events. In this case the UMAP conference is included, in the AG, in the Semantic Web related cluster, together with ISWC and SEMANTiCS, and in the TG in the large cluster of Data Science related events including ISWC and CLEF.

Finally, Figure 5.9 and Figure 5.10 represent our model of the 2014 events. Due to the large number of 2014 events published on CEUR it was possible to identify more clusters, however, there are still more clusters in the AG than in the TG. Both in teh AG and in the TG is clearly recognizable a large Semantic-Web related cluster, which in the TG includes also theoretic Computer Science events, such as the International Workshop on Description Logics. in the AG UMAP is included in a Personalization-related cluster together with RecSys, however in the TG is included in a broader Data Science related cluster, together with CLEF, in which RecSys is not features. In fact, RecSys forms along with other Recommender Systems related events a cluster on its own in the lower part of Figure 5.10. This fact suggests that, even though UMAP and RecSys receive contribution from the same community of authors, their focuses are distinct.

The compared analysis of the 2012, 2013, and 2014 models of CEUR events can

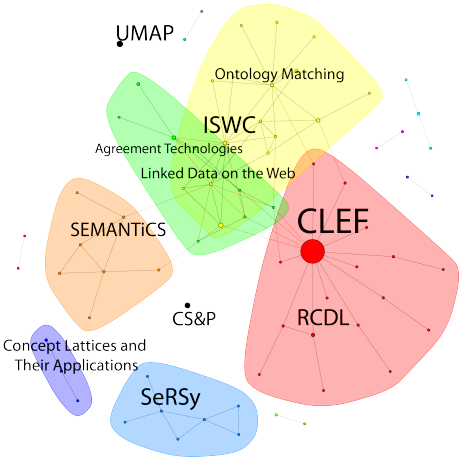

Figure 5.5: 2012 Author Graph

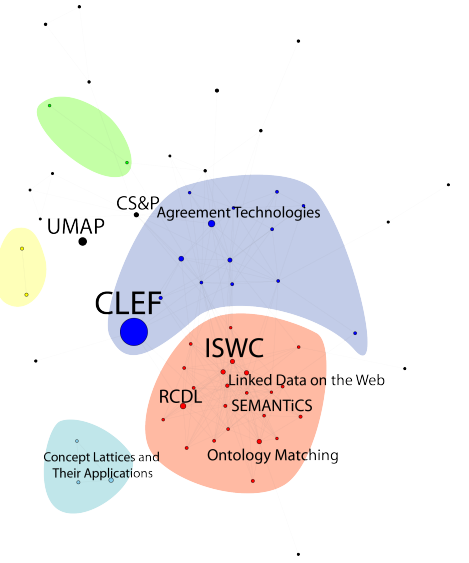

Figure 5.6: 2012 Topic Graph

provide insights on the evolution of the involved research communities. For instance, we can observe how CLEF and ISWC always attract different authors communities, even though there has been some topic overlap in the history of such events; moreover, we can observe how UMAP, in the considered period, was thematically closer to CLEF than ISWC, but it attracted part of the ISWC community as well. It can be also noted how in 2013 and 2014 a large number of Software Engineering related events with little or no overlap with the rest of the events both from an author and a topic perspective were held.

2012		2013		2014	
buzzword	frequency	buzzword	frequency	buzzword	frequency
system	0.30	system	0.521	system	0.662
data	0.291	data	0.416	model	0.607
computer science	0.261	model	0.385	data	0.576
model	0.246	computer science	0.378	information	0.533
information	0.231	information	0.335	computer science	0.478
ontology	0.201	Semantic Web	0.248	Semantic Web	0.355
knowledge	0.201	ontology	0.242	research	0.294
Semantic Web	0.201	Natural Language Processing	0.192	language	0.294
Natural Language Processing	0.134	Software	0.186	Natural Language Processing	0.282

Table 5.4: Most widespread scientific buzzwords and their frequency in the document corpus

Another interesting insight about what research communities actually debate can be obtained by looking at the extracted concepts with the lowest IDF, which means the most widely used in the considered data set. They are listed in Table 5.4. Most of these concepts are, as expected, very generic (such as “System” or “Model”) in the field of Computer Science and Information Technology (to which all the considered events

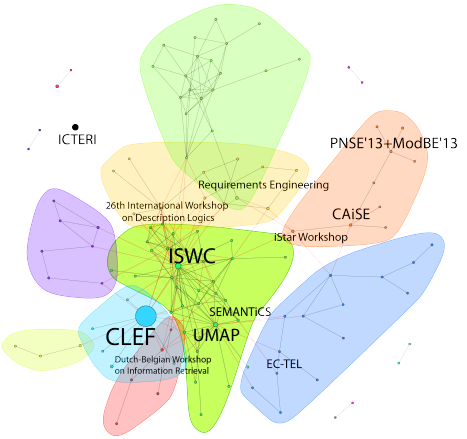

Figure 5.7: 2013 Author Graph

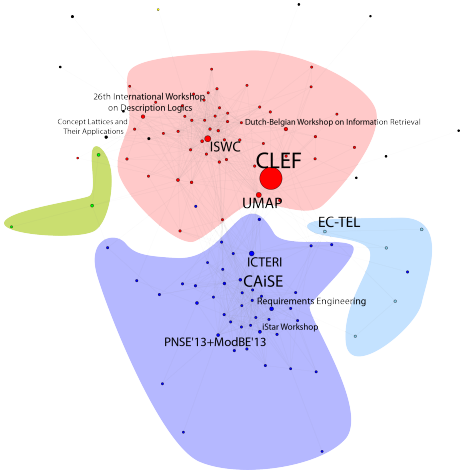

Figure 5.8: 2013 Topic Graph

belong), however some of them are very specific and usually associated with a precise research community, such as Semantic Web, Machine Learning, and Natural Language Processing. Semantic Web, in particular, appears in all the three considered years of CEUR proceedings on a broad fraction of the published papers (spanning from 0.2 to 0.35 of the considered proceedings) which is far larger than the one covered by the identified Semantic Web event cluster.

Event	**Complementarity score**
the Workshops held at Educational Data Mining 2014	0.267
the Workshops of the EDBT ICDT 2014 Joint Conference	0.238
the 16th LWA Workshops KDML IR and FGWM	0.235
Workshop on Semantic Matching in Information Retrieval	0.223
the Poster Workshop at the 2014 Complex Systems Design Management International Conference	0.216
the 8th International Workshop on Verification and Evaluation of Computer and Communication Systems	0.209
the Workshop on Multi Level Modelling	0.208

Table 5.5: Most complementary events to UMAP

Finally, the complementary communities analysis highlighted how every considered event has at least a potentially complementary event, that is an event focused on very similar topics, but attended by a different community with little or no overlap at all. Since listing all the pairs of potentially complementary events would require too much space, we are only reporting, in Table 5.5 the potentially complementary events for the UMAP community held in 2014.

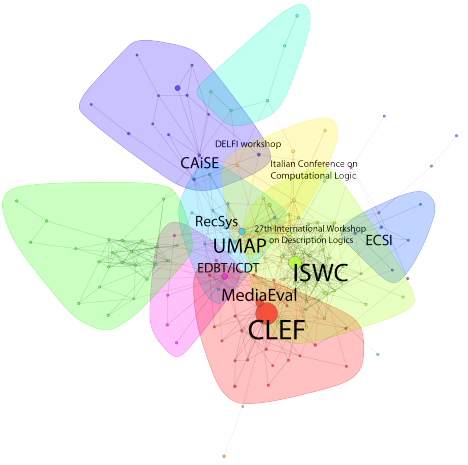

Figure 5.9: 2014 Author Graph

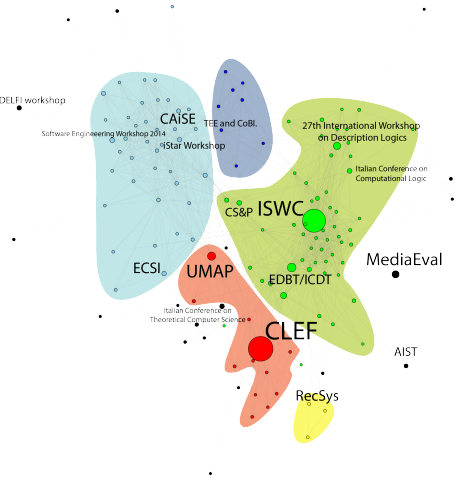

Figure 5.10: 2014 Topic Graph

5.3.3 IRCDL proceedings analysis

The third and last case study is focused on the evolution over time of the academic debate within a single community, namely the one participating to the Italian Research Conference on Digital Libraries (IRCDL)[1]. Since our interest is focused on topics, only the TG of these events is generated.

To achieve temporal modelling, papers are grouped by year, then using the approach described in Section 5.2 to model the TG, every group of papers is represented as a node in a network. The first relevant insight about how the scientific debate evolved over time is given by the mere distribution of extracted topics among the considered years: buzzwords come and go and their presence inside the full text of published papers reflects the trends in the research community. The fraction of papers including a specific term is a significant measure of how much widespread such term is at a specific time. In Figure 5.11 we show the result of this kind of analysis over the 10-years-wise most relevant buzzwords found in the IRCDL proceedings. It can be noticed how "Digital Libraries", which is the focus of the conference, is by far the most widespread term and consistently appeared in accepted papers over the ten years. On the other hand, some growing and diminishing trends can be easily spotted: "Cultural Heritage" has encountered a growing popularity in recent editions while it was somehow less relevant in first ones; on the contrary "Information Retrieval" was a widespread topic in the first editions, however in the more recent ones its presence diminished significantly, and this is related to the fact that the IRCDL steering committee in 2010 was promoting the launch of the "Italian Information Retrieval (IIR) Workshop" series, that started in 2010 with its first edition which was supported by IRCDL and organised in Padua side by side with the edition of 2010 of IRCDL.

This analysis, however, does not provide actual insights on the topics that characterized a specific year or a given time frame in research. In other words it does not answers

[1] the data used in this study is courtesy of the IRCDL steering committee

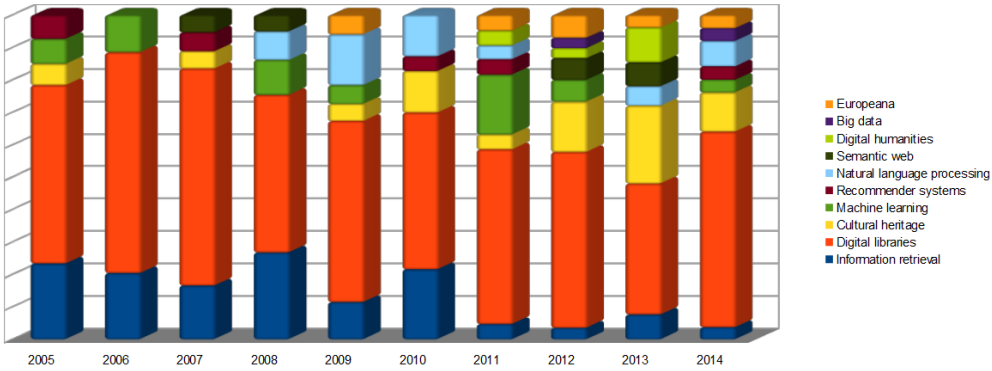

Figure 5.11: Most frequent topics over ten editions of the IRCDL conference.

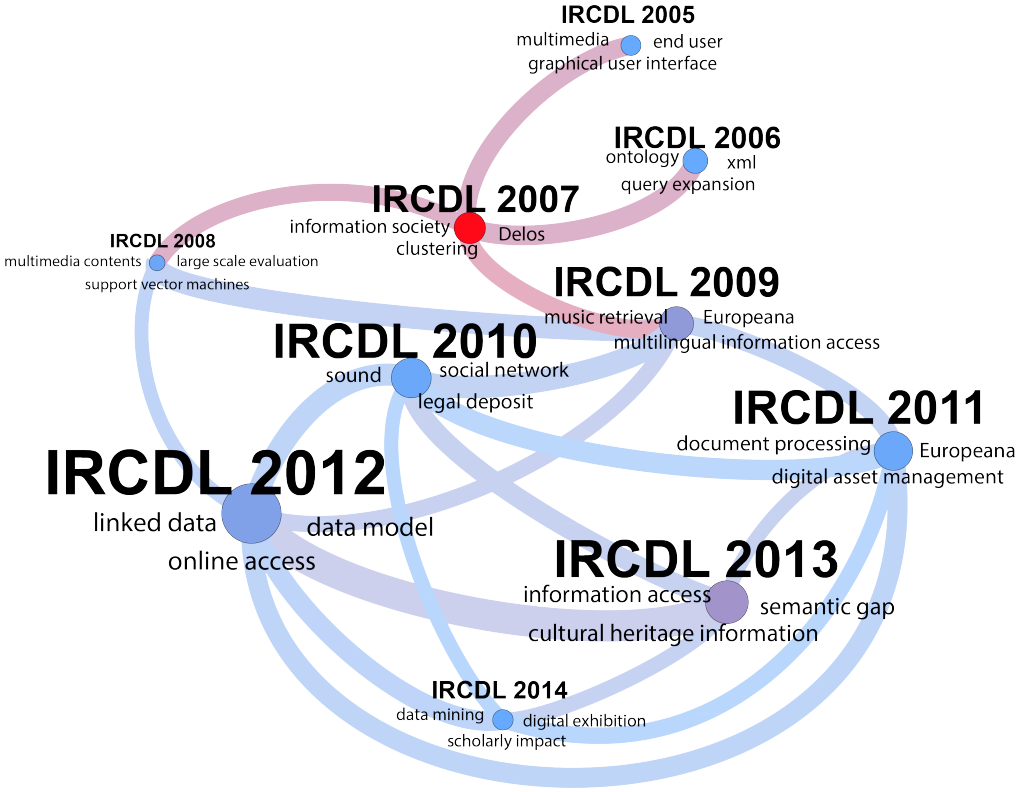

Figure 5.12: Characterizing topics of each IRCDL edition.

the question "what was that year about?". To achieve this goal we must evaluate a time-wise IDF that allows us to set apart buzzwords consistently present in the domain and concepts that surfaced only in a certain time frame. Again, creating a Topic Graph where papers are grouped by year to form nodes allows this kind of analysis.

Figure 5.12 shows the Topic Graph built upon IRCDL accepted papers grouped by year annotated with the most significant topics for each node (i.e. the ones with the highest $CK - IDF$). In this graph nodes represent editions of the conference, the larger the node is drawn, the more distinct topics were extracted from its associated papers; the presence of an arc between two nodes implies a significant overlap in the associated topics. Nodes in the figure are coloured according to their centrality with a "hot" colour (tending to red) meaning a high centrality in the network. Highly central nodes, such as the 2007 edition are to be considered turning points in the history of the conference, since they represent a bridge between distinct groups of topic-wise similar editions. It is interesting to note how, in this example, the part of the graph representing the first editions of the conference is relatively sparse indicating little overlap aside from buzzwords, while more recent editions are much more connected, indicating a great deal of shared topics which implies that the conference has found a consistent core of topics.

5.4 Lessons Learnt

In this chapter we presented a new approach towards scientific communities modelling based on a twofold view with the aim of discovering shared interests, spotting research communities and, hopefully, help scientist addressing the problem of finding the right venue for their work. The ability of identifying potentially complementary communities is , in our opinion, the most notable feature of the approach herein proposed: traditional social network based analysis can detect existing communities, but is unlikely to identify communities that should talk each other, meet or join. On the other hand, our approach exploits state of the art Information Extraction techniques to investigate the topics actually dealt by a community and, comparing the topology of the SNA based model and the content-based model can easily identify communities that deal with the same topics, but have little or no social connections at all. Identifying such communities, in our opinion, can help scholars to find relevant literature and, hopefully, to foster knowledge transfer from one community to another, improving the quality of research.

The work herein presented, however presents some limitations as well: firstly no knowledge base was considered, we relied on the large volume of considered data to abstract over stylistic nuances and synonymity. Secondly, this kind of exploratory analysis can hardly be evaluated: assessing the quality of the obtained insights is up to the expert who observes them and therefore it is nearly impossible to estimate in a quantitative way their adequacy. Nevertheless, the fact that these analysis spawned two peer reviewed publications somehow reassure us on the soundness of the proposed work. The final limitation of the work presented in this chapter it the practical difficulty in accessing the textual contents produced by the research community. We limited the largest part of our study to a single, open access repository, to purposely avoid this problem and though the number of considered papers is high, over 7000, they still represent a tiny fraction of the volume of research published during the considered years in the domain of Computer science. However a more complete survey would have had to face

serious licensing issues before being performed. In fact, shortly after the publication of [122] our research group was asked to replicate this very same study on a list of 200 hi-impact journals in the field of agricultural and environmental sciences and at the time of this writing we are still awaiting permission of conducting text mining activities from roughly two thirds of the considered editors and digital libraries who own the rights over those journals.

6

Towards Domain-Aware Information Access

In this chapter we present the final component of the Automatic Curator system, the domain knowledge base, and provide an example of its usage to enhance exploratory search and content-based recommendation. Our proposed solution relies on the concept of semantic relatedness introduced in Chapter 3, Section 3.3.1. While other works primarily rely upon Linked Data as a background knowledge base, we are opting for a more compact, less formal representation that can allow more efficient search operations. The work presented in this chapter is partly described in [40] and includes unpublished work as well.

6.1 Introducing referential spaces

As shown by the authors of [182] hypertextual connections between Web pages alone can carry a great deal of semantics at a reasonable computational cost. However their proposed method involving the combination of two distinct metrics for incoming and outgoing links can be still too demanding when a very large content base must be scouted to find related items. Wikipedia, which includes over 8 million items is a perfect example of such a situation. To overcome this limitations and to set up a minimal theoretical framework, we introduce a new hypothesis: the *Reference Hypothesis*. We assume that entities that are referenced in a similar set of documents might yield strong semantic affinity. For instance, in Figure 6.1 two entities (A and B) are referenced by three different documents: this implies a semantic affinity between A and B.

This assumption is motivated by the fact that intuitively referencing something in a document implies the referenced item to be relevant in the context of the document, therefore entities that get constantly referenced together are relevant in the same contexts, hence they might be semantically related. This hypothesis can be seen as a generalised version of the aforementioned distributional hypothesis, however we would like to stress how even though words can be seen as entities, entities can be intended as

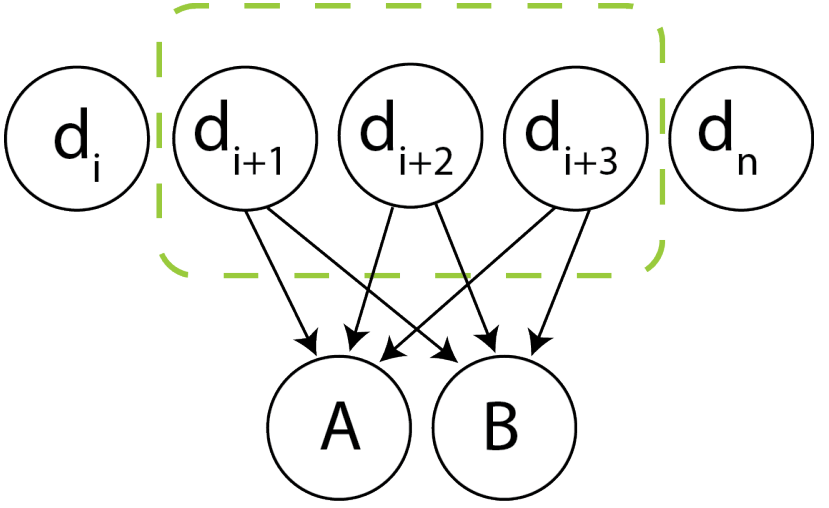

Figure 6.1: Two entities referenced by the same set of documents.

way more abstract items, for instance other documents or ontology entries. For instance, the reference hypothesis applies to the scientific literature since articles citing similar sources are very likely to deal with similar topics. Other works in literature embrace this assumption though not formalising it, such as [76] wherein a scientific paper recommender system exploiting co-citation networks is presented. Building a vector space exploiting the Reference Hypothesis is straightforward once a large enough corpus of documents annotated with hyperlinks is provided. Within the corpus, two sets must be identified: the *entity set* E and the *document set* D; the first includes all the referenced entities, while the latter is the considered annotated documents. The vector space is represented with an $E \times D$ matrix that initially is a zero matrix. Iteratively, for each $d \in D$ all the references to elements in E are considered, and for each $e \in E$ referenced in d, the (e, d) cell of the matrix is set to 1. Since referencing a given entity only once in a document is a typical best practice in several domains[1] we are not considering how many times e is referenced in d. Once all documents are processed we obtain a matrix where each row represents all the references to a given entity: we call such matrix *Reference Matrix* and the vector space it generates *referential space*.

Evaluating the similarity of two entities in such a vector space reduces to computing the distance between their vectors. The semantic similarity evaluated in this way can be seen as the likeness of two concepts to co-exist in the same paragraph of text. This notion allows us to introduce some non-trivial background knowledge that current state of the art systems often fail to capture. Consider for instance the highly polysemic phrase "US President" that can refer to 45 different people who had over the time similar powers and could all be found in sentences including keyphrases such as "law" or "foreign policy". However when that phrase can be associated with information such as an year or a major event, it is possible to associate it with one of the 45 candidate

[1] For instance in Wikipedia only the first time an entity is referenced it is annotated with an hyperlink, and in literature bibliographies have no duplicate entries.

entities.

Countless distance metrics exist in the literature such as norms, cosine similarity, hamming distance, and many others surveyed in [176]. All these metrics can be used in the Reference Matrix, however we prefer the Jaccard similarity coefficient (also known as Jaccard index [77]), defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{6.1}$$

where A and B are sets of items. Since each entity $e_i \in E$ can be considered a binary vector, it can also be expressed as the set that contains all the document $d_j \in D$ such that $(e_i, d_j) = 1$ in the Reference Matrix. This choice is motivated by the intimate simplicity of such metric that scales well to large sparse matrices since it ignores zero values, and easily translates into computational efficiency. A critical factor that influenced our decision of adopting the well-established Jaccard index in spite of more recent and elaborate metrics is the fact that its computation can be reduced to constant time using the MinHash optimisation [16]. Such a technique allows to efficiently compute the similarity between sets without explicitly computing their intersection and union. Its most common form consists in using an hash function to map each element of the set to an integer number and then selecting the minimum as a representative of the whole set. The probability that two different sets share the same minimum with respect to the hash function tends to the Jaccard similarity coefficient between the two sets [93]. The more hash functions are used, the closer the estimate gets to the real Jaccard similarity coefficient value. In this work, we used 256 distinct hash functions to achieve a fine enough approximation of the Jaccard similarity coefficient. The similarity of two equal sets is one, whereas the similarity between two sets that have no elements in common is zero. Finally, it is known in the literature that Jaccard index performs better than other methods for finding word similarities in Vector Space Models approaches [91, 100].

6.2 Tasks

Building on the insights presented in [18], we decided to test the applicability of the semantic similarity metric described in the previous section on well-defined tasks. More specifically, we identified two tasks: the retrieval of a set of neighbour entities for exploratory search purposes, and the recommendation of a set of items. To perform both tasks we built a referential space with the full set of English Wikipedia articles.

6.2.1 Neighbour Entity Retrieval

The first task is easily defined: given an entry of the knowledge base, we intend to retrieve the top n entities present in the knowledge base that can be considered the most similar. Though being a simple task in principle, it implies evaluating pairwise relatedness between the considered entity and any other entry of the considered knowledge base. With the high dimensionality of the Wikipedia data set, counting over 8 million entries, however, finding the sweet spot between reasonable performance and accuracy is no trivial task. The MinHash optimisation, in this sense is instrumental in guaranteeing

Table 6.1: An example of similarity vector evaluation between an item and a user profile.

	item feature 1	item feature 2	item feature 3	item feature 4
user interest 1	0.01	0.05	**0.51**	**0.62**
user interest 2	**0.88**	0.2	0.24	0.56
user interest 3	0.25	**1**	0.09	0.49
user interest 4	0.34	0.43	0.31	0.04

acceptable computation times when millions of tests are required to fulfil a single query.

6.2.2 Non-Exact Matching based on Semantic Similarity

As introduced in Chapter 4, allowing non-exact or fuzzy matchings to be taken into consideration can greatly ease the problem of sparsity, which is critical when building a recommender system. In the Collaborative Filtering domain it is not unusual to cluster users into neighbourhoods [71] or to adopt dimensionality-reduction techniques such as Singular-Value Decomposition of the rating matrix to counter sparsity. Similarly, some authors have proposed to cluster terms in content-based filtering, a solution that can provide satisfactory results in the scholarly data access domain [60]. With this in mind, we can use the semantic similarity metric provided by the referential space to estimate the similarity between non-matching key terms describing user interests and item contents. As long as both item and user interests are represented with concepts included in the referential knowledge base, it is possible to estimate for each pair a similarity index that can range between 0, total unrelatedness, and 1, perfect matching. This allows us to move from a situation where the match can be either 0 or 1 to another one that admits multiple intermediate values, representing different degrees of relatedness. The solution we propose in this work consists in estimating, for each document key term, the distance between it and its most related term in the user model, as shown in Table 6.1, where the rows represent the key terms of the user model, and the columns the ones of the document content model. The resulting vector indicates the overall similarity between the user and the item and can be used to generate predictions. Several metrics could be used to estimate the length of the calculated similarity vector, but in this work, we favoured estimating the the average value of its positions to avoid excessive bias towards longer vectors, i.e. items described by a lot of terms. Once a similarity is evaluated for all documents, the ones that match the best the user profile are suggested as recommendation.

In this work, we are using a referential space built upon the set of articles included in Wikipedia, which is a convenient situation for two reasons: Wikipedia has a lot of concepts and covers multiple domains, and there exist several tools, surveyed in Chapter 3, that can provide Wikification, such as TagME, DBpedia Spotlight, and Agdistis.

6.3 Experimental Evaluation

In this section we describe the experiments performed so far to assess how well a knowledge base built on the referential hypothesis can support the tasks introduced in the

previous section.

6.3.1 Crowdsourcing Evaluation of Similar Item Retreival

Using the technique described in the previous section, a testbed system, herein named Referential Space Model (RSM), was developed and trained on Wikipedia, associating to each of its 8 million items a representative vector. Building on the results of [182] that provides evidence of the importance of both incoming and outgoing links, we also developed an alternative model relaxing the distributional hypothesis and considering outgoing links, i.e. the items mentioned in the article corresponding to a give item. We refer to this second testbed system as *RSM.outnode*. To assess the quality of the two alternative approaches we constructed a dataset of the top visited Wikipedia pages. As a reliable source of data we used the list of Wikipedia Popular Pages[2] that maintains a set of the most accessed 5000 voices on the English Wikipedia and it is updated weekly. For our data set we focused on the most *stable* voices during the last year (2015). We define the stable voices as the Wikipedia pages that constantly appear in every weekly version of that list throughout the year, and so receiving constant interest from the visitors of Wikipedia. A set of 1583 stable items were identified. The evaluation dataset was built by randomly selecting 100 items (experiment 1) and 25 items (experiment 2) upon which all of the four systems are able to retrieve related items. As a comparison we choose two of the most popular search engines on the market[3]: *Google* and *Bing*. One of the most prominent features of said search engines is the ability to leverage the LOD cloud to improve search results, more specifically they can retrieve a neighborhood of items closely related to the search query given by the user. To obtain fair and generic search results i.e. not influenced by the recorded browsing history, preferences, and location, Google and Bing search process was de-personalized to prevent the search engines from customizing the final result.

The goal of our first experiment was to assess which one of the four systems produces the overall best set of related items given one search key. To this extent, we considered the 100 items dataset. The crowdsourcing experiment was designed as follows: for each of the considered items a page was generated including the name of the item, a brief description, a picture, and a box including the results produced by the four systems i.e. four lists of five semantically related items. We decided to show only five results for two reasons: firstly both Bing and Google show at least five related items, which means that for some search queries no more than five items will be shown, secondly it is a known fact that users typically pay attention only to the top spots of search results lists, with the top five items attracting most of the attention[4]. To avoid cognitive bias, the names of the systems were not shown and the presentation order was randomized, so that the worker had no means of identifying the source of the presented item lists and couldn't be biased by personal preference or previous evaluations. The workers were then asked to rate the four item lists according to their perceived quality in terms of relatedness on a discrete scale from 1 to 5 where 1 meant total randomness and 5 that all presented items where perceived as strongly related. Each one of the 100 items in the data set was shown with the same related items lists to 5 distinct users and their judgements

[2] `https://en.wikipedia.org/wiki/User:West.andrew.g/Popular_pages`
[3] http://www.alexa.com/
[4] https://chitika.com/google-positioning-value

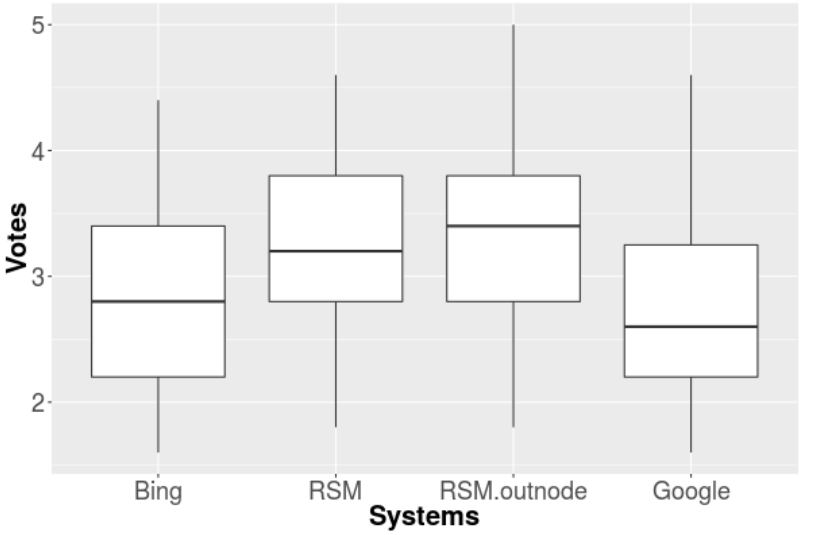

Figure 6.2: Experiment 1 distribution of worker's judgement.

were averaged per system to mitigate subjectivity of judgement. The experiment was performed using the popular crowd sourcing platform *Crowdflower*[5] and 32 users from 18 different countries were involved, with an average of 15.62 judgements per user. The distribution of the worker's judgement is shown in Figure 6.2.

The results of experiment one showed how our testbed systems RSM and RSM.outnode achieve, on a scale from 1 to 5, respectively a 3.20 and 3.33 average perceived quality, while Google and Bing respectively 2.79 and 2.82. The statistical significance of the judgment distributions shown in Figure 6.2 was evaluated as well showing how while there is a substantial difference between the perceived quality of our systems and the baseline ones (Bing and Google), between RSM and RSM.outnode there is no statistically significant difference. More specifically the Welch Two Sample t-test was used and produced the results shown in Table 6.2, where in the upper right half of the matrix are shown the p-values produced by the test, and in the lower left half the same values recalculated with the Benjamini & Hochberg correction for multiple hypotesis testing [12]. According to these results, Google's and Bing's related items lists are perceived almost as identical in terms of quality, while our testbed systems' outputs receive a significantly higher likely by the crowdsourced workers. Moreover, while RSM.outnode appears to achieve an higher perceived quality than RSM on average, the statistical significance analysis shows that such a difference is unlikely to be significant in the current experimental setting. Wrapping it up, in terms of overall perceived quality the neighborhoods of related items to a given search key produced by RSM and RSM.outnode do not differ significantly in terms of perceived quality, but there is evidence that consistently outperform the benchmark systems offered by Google and Bing.

The goal of our second experiment was to assess the perceived quality of each item

[5]http://www.crowdflower.com/

Table 6.2: Statistical significance of the difference between the considered systems. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction

	RSM	RSM.outnode	Google	Bing
RSM	-	0.1896	<0.0001	0.0001
RSM.outnode	0.2275	-	<0.0001	<0.0001
Google	<0.0001	<0.0001	-	0.6838
Bing	0.0003	<0.0001	0.6838	-

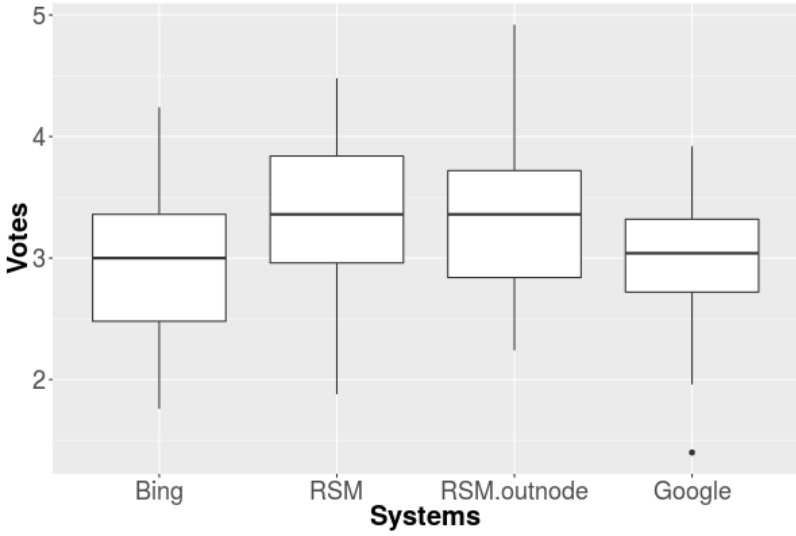

Figure 6.3: Experiment 2 distribution of worker's judgement.

included in the related items list. To this extent we considered the 25 items dataset. The experimental setup was similar to the previous experiment, using the same platform and displaying the same information about the target entity (i.e. title, description, and picture). Instead of four lists, this time the workers were shown a single list generated by one system only and were asked to rate each item in the list on a scale from 1 to 5 where 1 implied complete unrelatedness and 5 a very high perceived relatedness. The name of the system that generated the list was not shown to avoid bias. A hundred related items lists where therefore generated and human-rated item by item. Again, each item was judged by five distinct users to mitigate subjectivity of judgement. This second experiment involved by design substantially more workers to further abstract over subjective experience and thus obtain a more impartial judgement; in the end 146 workers from 38 countries were involved with an average of 3.42 judgement per user. In Figure 6.3 the distribution of workers' judgments is shown.

The results of experiment two support the evidence provided by the previous one. More specifically items retrieved by RSM and RSM.outnode on average score a 3.41 out of 5 on perceived quality while Bing and Google stop at 2.93 out of 5. These numbers,

Table 6.3: Distribution statistics on NDCG evaluation

	Bing	RSM	RSM.outnode	Google
Minimum	0.4629	0.6009	0.6006	0.4250
1st Quartile	0.6423	0.7829	0.7601	0.6631
Median	0.7376	0.8232	0.8293	0.7186
Mean	0.7247	0.8113	0.8226	0.7196
3rd Quartile	0.8475	0.8678	0.9066	0.7855
Maximum	0.9010	0.9771	0.9910	0.9102

however, provide little information being average values of perceived quality of item ranked in different positions. Looking at the whole distribution of judgments shown in Figure 6.3, the high variance of the four distributions can be easily noticed. Such a variance can be justified by the fact that all items included in the generated lists are considered and rated. However, not all positions of a result list are equal to the extents of exploratory search. To address this issue we evaluated the Normalized Discounted Cumulative Gain (NDCG) of the four considered systems. NDCG is a metric commonly used in IR to assess a search engine's performance basing on the comparison between an ideal list of the most relevant retrievable items and the actual list produced by the evaluated system. Its core idea is that the higher the position of an item in the result list the more important the quality of that item should be in the quality evaluation of the system, therefore the presence of scarcely relevant items in the top spots tends to "punish" the evaluated system. The ideal list was computed by considering, for each of the 25 search keys, all the items retrieved by the four systems, picking the five ones that on average received the highest user ratings and ordering them in descending average rating order. The distribution of the NDCG values scored by the four considered systems over the 25 search queries included in the data set is shown in Figure 6.4 and its detailed statistics are presented in Table 6.3. These results support the evidence brought by the first experiment as well, with RSM and RSM.outnode providing consistently results perceived as more relevant than the ones brought by Google's and Bing's tools by the crowdsourced workers. Again, there is no statistically significant difference in the average perceived quality between RSM and RSM.outnode (p-value = 0.68) and between Google and Bing as well (p-value = 0.88). On the other hand, the statistical significance between RSM and Google, RSM and Bing, RSM.outnode and Google, and RSM.outnode and Bing is high with p-values below 0.0001. Finally, the NDCG analysis shows how, despite scoring being on average on par with its RSM.outnode counterpart, the RSM system has the smallest variance in the perceived relevance of its results, implying that it is less likely to produce results perceived as poor on a single-try basis.

6.3.2 Evaluation of Non-Exact Matching on a Dataset

We tested our non-exact match recommendation approach on the DBbooks data set provided for the ESWC 2014 RecSys challenge, that contains user ratings over a set of over 8000 books. The books available in the dataset have been mapped to their corresponding DBpedia URIs. The mapping between DBpedia and DBbooks allowed us to extract key terms grounded into our referential knowledge base with no need

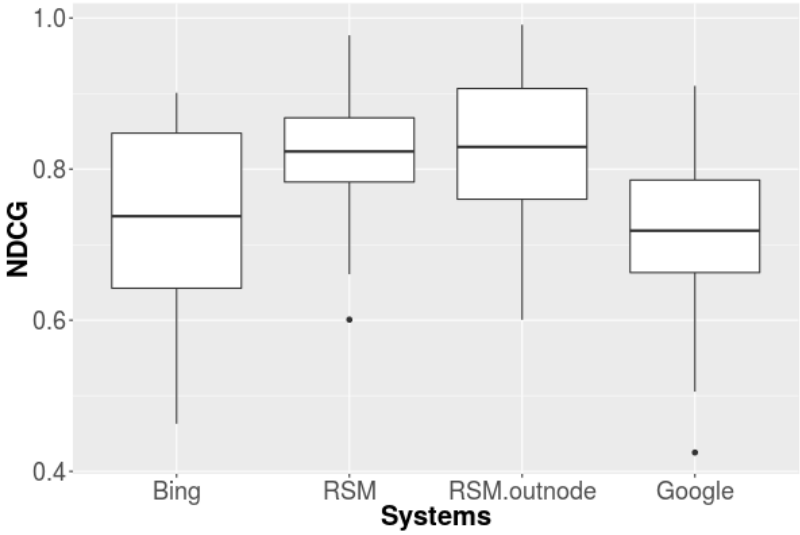

Figure 6.4: NDCG values distribution evaluated on the results of experiment 2.

for complex Information Extraction techniques. The mapping contains 8170 DBpedia URIs. These mappings can be used to extract semantic features from DBpedia or other LOD repositories. The dataset is split in a training set and an evaluation set. In the former, user ratings are provided to train a system while in the latter, ratings have been removed, and they are used as ground truth in the eventual evaluation step. We accessed the ground truth values after they were made available after the end of challenge[6]. Among the three tasks proposed by the challenge, we decided to perform the second one as a preliminary evaluation of our approach. Citing the challenge description, the task deals with the top-N recommendation problem, in which a system is requested to find and recommend a limited set of N items that best match a user profile, instead of correctly predict the ratings for all available items. Similarly to Task 1, in order to favor the proposal of content-based, LOD-enabled recommendation approaches, and limit the use of collaborative filtering approaches, this task aims at generating ranked lists of items for which no graded ratings are available, but only binary ones. Also in this case, the DBbook dataset is used. In this task, the accuracy of recommendation approaches will be evaluated on an evaluation set using the F-measure. Since the official online evaluation tool was not available any more at the time of our experimentation, we had to deploy locally our copy of the evaluation tool suggested by the challenge website[7], which calculated a F-Measure of 0.5497 for our system.

We compare our results with the extensive evaluation activity on the same data and task of state-of-the-art approaches described in [159]. The considered approaches include:

- A *Linked-Data-based recommender* (LDR) that makes prominent use of external

[6] `http://www.di.uniba.it/%7Eswap/datasets/dbbooks_data.zip`

[7] such a tool can be downloaded from `http://sisinflab.poliba.it/semanticweb/lod/recsys/2014challenge/ESWC2014ChallengeEvaluationTool_v1.3.zip`

Table 6.4: ESWC 2014 RecSys Challenge performance benchmark.

Technique	F-Measure
LD-based Recommender (LDR)	0.5607
Global Classifier	0.5596
Popularity-Based	0.5585
Semantic Similarity-Based	**0.5497**
Neighborhood Classifier	0.5462
User Classifier	0.5302

information, namely item features obtained from DBpedia. Its key components are the item representation model employing features from DBpedia data, the naive Bayes classifier for rating prediction, and the user-neighborhood-based collaborative filtering for reducing data sparsity.

- A series of classifiers trained on purely collaborative filtering information; the considered classifiers are all Naive Bayes ones trained in three different ways: training one classifier per user (referred as User Classifier), a User-neighborhood-based classifier [71], and a single classifier foe all users (referred as Global Classifier).

- A naive predictor that favours the items that on average received the most positive feedback from the user community, herein the Popularity-Based technique.

The authors claim that there is empirical evidence that Naive Bayes classifiers perform better than Support Vector Machines, Linear Regression, and ADTrees on this problem, however they do not provide data to uphold this claim. Nevertheless, their results are summarised in Table 6.4 where also our score is included for better comparison.

It can be noted how the naive approach is able to achieve on this dataset a surprisingly good result, implying that the provided test set is strongly biased towards blockbuster items receiving generally positive feedback from the user base. With respect to the considered baseline techniques, our approach, in spite of not considering collaborative information and not using advanced machine learning techniques, performs relatively well, outperforming some collaborative techniques. One major advantage of our technique, in an hypothetical field usage, is that it does not need to be re-trained as new users and items are added to system, but rather at a much slower pace to keep the knowledge base up to date.

This evaluation is, however, a preliminary assessment and additional tests will be performed including user evaluation, as already done for the RES system presented in Chapter 4.

6.4 Lessons Learnt

Wrapping it up, the referential hypothesis allowed us to build a sound VSM that captures semantic relatedness among the considered items, and the usage of the Jaccard index and the MinHash optimisation allowed us to handle the over 8 million items included in Wikipedia with ease. Providing semantic tools that can efficiently scale up to the

large volumes of data involved in nowadays information access applications such as personalised information retrieval and personalised recommendation, is in our opinion a critical step towards fully accomplishing the potential of the Web of data. It is well known that the graph nature of the LOD cloud implies high computational costs when exploration and reasoning tasks must be performed and many current state of the art algorithms involve extensive graph traversing. A less formal description knowledge, on the other hand, can support more efficiently several tasks such as exploratory search and recommendation, plus if such a representation can be grounded into the Web of data with a direct entry mapping, it is possible to integrate both techniques to provide a better data access. For instance, while a similarity based domain model can efficiently retrieve a set of related concepts, if these concepts are grounded into a Linked Data, a visit of its graph can provide an explanation of why the retrieved concepts are related. In the work presented in this chapter the referential space built on the top of Wikipedia articles is firmly grounded into the Web of data, since the identifiers of its entries are mapped to DBpedia as well. In the general design of an automatic curator system, we believe that the presence of a semantic similarity based model should be considered, since it can greatly support both sense-making tasks and information filtering, scaling up well as the amount of concepts introduced as background knowledge grows.

7

Conclusions

In this work, we tackled the complex and highly interdisciplinary problem of providing an integrated solution for scholarly data access domain. Though we did not manage to build a complete Automatic Curator system yet, we explored a number of practical challenges researchers are facing on daily basis, namely:

- Finding relevant scientific literature, browsing among the huge volume of scholarly data produced by the ever-increasing activity of the research community.
- Identifying emerging trends and communities, locating in the research community complementary skills and expertise, and finding appropriate venues where relevant new works can be found or published.
- Making sense of an ever-increasing amount of concepts, notions, and domain terminology, discovering new areas tightly related to a researcher's current interests.

Though we did not provide an integrated solution to address all these issues, we showcased a number of potentially good solutions to tackle them.

7.1 Proposed solution overview

Wrapping up the insights gathered from the literature and the results presented in Chapter 4, 5, and 6 we can now put together the gathered expertise and outline a more detailed scheme of an Automatic Curator system. With respect to the initial design presented in Chapter 1, we can present a more accurate breakdown of the key components of the system. We can outline the following main modules:

- a Web User Interface: following the experiments presented in Chapter 4, and the insights gathered from works such as [143], it emerges that a carefully designed user interface is instrumental to achieve user satisfaction. Such an interface should be a Web one, allowing easy access to the system from anywhere. Moreover, the user interface should provide an overview of the contents to be accessed and allow some

understanding of the reasons why the presented documents should be of interest to the user. Such information, as shown in Chapter 4 enhances user experience.

- a Recommender module: to achieve proactivity the curator system must include an information filtering component. The results gathered in Chapter 4 and 6 highlighted how such a recommender module should allow non-exact or fuzzy matches, given the typical sparsity of the scholarly data access problem, therefore it should have access, aside from the documents contents, to some form of knowledge base to allow such a partial matching. Our considerations on the domain of scholarly data access presented in Chapter 2 and the experiments described in Chapter 4 and 6 lead us to believe that such a component should be primarily content-based, possibly exploiting some background knowledge base.
- an Explanation module: the results of the user tests described in Chapter 4 provided us evidence that explanation is a relevant feature in a scholarly data access system and therefore should be included in the curator system, possibly encapsulated in a dedicated module processing the results produced by the recommeder before prompting them to the user.
- a User Profile Manager module: modelling the user preferences is a critical task in providing personalised data access, and the forms of feedback to be leveraged to extend the model are multiple. A module of the system should be devoted to such a task, allowing th user to access different views of his or her preferences, as proposed in Chapter 4.
- an Information Extraction module: results provided in Chapter 4 and 5 have shown as processing the textual content of scientific papers can lead to great benefits in providing access to them, therefore an Information Extraction module has to be included in the design of a curator system. Several solutions have been described in the literature, as discussed in Chapter 3, and our experiments described in 4 and 5 provided evidence that Keyphrase extraction and Named Entity Linking can be particularly useful to support scholarly data access.
- an Exploratory Search module: as shown by the authors of [143], exploratory search activities can substantially help users in making sense of the scholarly data, therefore an integrated solution to scholarly data access should include such a module, possibly backed by the research venues representations presented in Chapter 5 and by the Referential Space presented in Chapter 6.
- a Data Gathering module: to support content mining and exploratory search it is instrumental to provide the system access to scholarly data repositories. An ad-hoc module should be devoted to this task, allowing the rest of the system to abstract over the actual provenance of the contents.
- a Background Knowledge Manager module: as introduced in Chapter 6 and also discussed in [143], knowledge can come in many forms and in heterogeneous forms, thus the various kinds of background knowledge considered must be integrated into a coherent knowledge base. A module of the system has to be designed to fulfil this delicate task, allowing the other components to abstract over the nature and provenance of the information stored in the knowledge base.

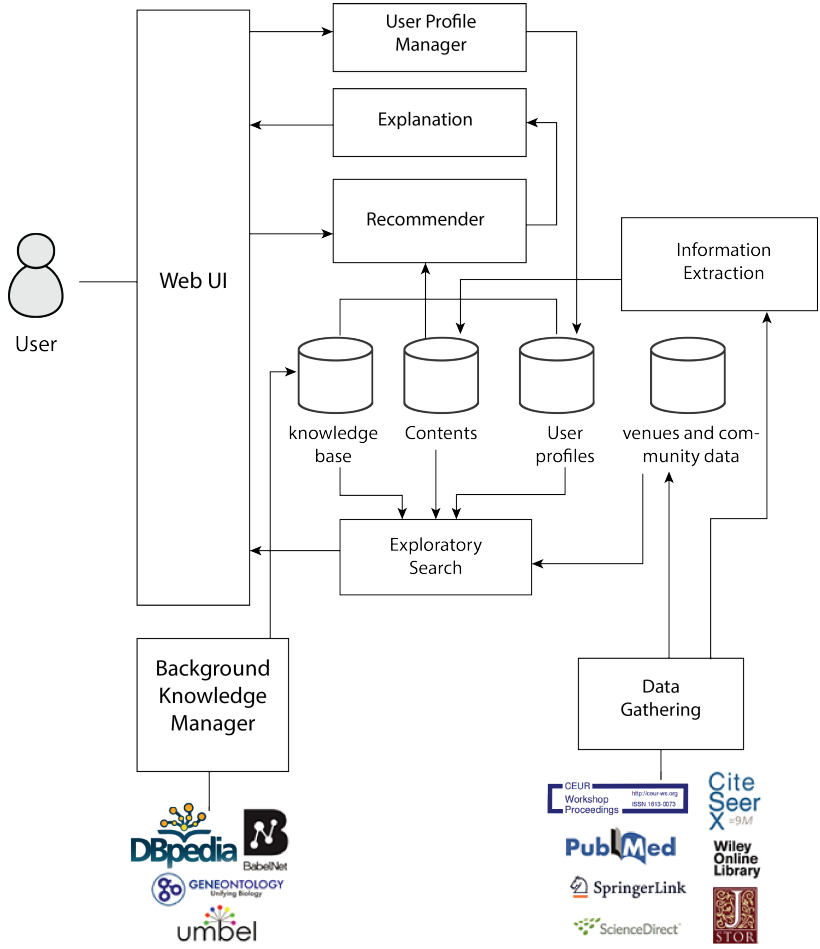

Figure 7.1: Proposed architecture of a Curator System with its main modules.

The proposed architecture of the system is illustrated in Figure 7.1 wherein the fundamental modules and repositories of the curator system are shown. The proposed architecture represents an integrated solution to the problem of scholarly data access and allows new advanced Artificial Intelligence technologies to be seamlessly plugged in as the state of the art in the fields of Information Extraction and Knowledge Representations progresses. One issue, however remains largely unresolved: designing and implementing a meaningful evaluation protocol for such a large and complex system.

7.2 The many criticalities of evaluation

Designing and implementing effective evaluation strategies is a constant issue we have faced throughout the development of the experimental systems described in this thesis. This is a widely shared problem within the research community, and as observed by

the authors of [11], no consensus exists on how to evaluate and compare research paper access approaches. This leads to the unsatisfying situation that despite the many evaluations, the individual strengths and weaknesses of the proposed approaches remain largely unknown. In fact, out of 89 approaches the authors of the survey reviewed, 21% were not evaluated. Of the evaluated approaches, 19% were not evaluated against a baseline. Almost all evaluations that compared against a baseline, compared against trivial baselines and only a meagre 10% compared against some kind of state of the art recommender system. Even when it comes to user evaluation, most presented work considers very few user, sometimes less than six, thus making the gathered feedback barely relevant.

The reasons behind this general lack of solid evaluation practices are many, and the experience gathered during the last years allows us to pinpoint the following:

- Lack of a shared evaluation framework: while there exist evaluation frameworks for more traditional information retrieval and filtering applications, such as the one presented in [149], no such thing exists for the scholarly data access domain. More notably there is also no such thing as a shared, and somehow authoritative, set of use cases for scholarly data access, which makes the design of experimental activities quite subjective.

- Lack of publicly available benchmark datasets: though several repositories of scholarly data exist, there is no such thing a shared, established, data set allowing to test in an offline way scholarly data access systems. Tough some authors do gather sets of ratings for scientific papers, allowing benchmark evaluation, such ratings don't get published, making impossible replicating the presented experiments.

- High cost of user-testing: testing a scholarly data access system requires a very specific kind of test subjects, researchers. Researchers, as a matter of fact, are highly qualified knowledge workers and their time is precious, thus gathering a sufficient number of them requires a lot of effort.

- Lack of reproducibility of existing work: as a matter of fact most existing works in the literature do not provide accessible implementations of their solution or provide insufficient information to replicate their approach. Finding a non-trivial benchmark system is therefore a hard task that most research groups do not have the means to carry out.

In this work, we partially addressed some of these issues by basing our experimental protocols on best practices established in other domains, and adopting evaluation frameworks and well-established data sets where possible. The obvious limitation of such an approach is that these frameworks and data sets are not meant for scholarly data access, but for different applications such as product recommendation systems, like the Movielens and the DBbooks data sets which contain data, respectively, about movies and books. Nevertheless, this experimental design choice allowed us to test our approaches against more general and well-established ones and to support their plausibility in a field application scenario. Evaluation of scholarly access systems, however, remains an open issue we sincerely hope the research community will address over the next years.

7.3 Future work

The insights gathered in this work allowed us to outline an integrated solution for scholarly data access, however such a solution has yet to be implemented. Though each single module is feasible on its own, the integration of so many components into a single solution is far from being trivial. The future work of our research group will be therefore aimed at addressing the many engineering challenges connected to the practical development of such an application. In particular we will investigate issues implied by the integration of heterogeneous knowledge sources, and the scalability of the proposed solution to real-world scenarios.

Aside from these technical challenges, we will also try to improve the evaluation design process, possibly addressing some of the issues presented in the previous section. The topic of scholarly data access has met a growing interest over the last years and it is encouraging to notice how initiatives aimed at promoting the research in this area, such as the *SAVE-SD* Workshop, have received good feedback, moreover, the big players of this market appear to be interested in new Information Access technologies. However, in our opinion, as long as shared frameworks including use-cases and evaluation protocols are not established, the research over scholarly data access will continue to be a somewhat grey area with no clear way of assessing the performance of various solutions, the user satisfaction and even the fulfilment of basic requirements.

I

Appendix

A

Complete Publications List

During the exploration of the scholarly data access domain we carried on over the last four years, an ever-increasing number of issues and challenges tightly related to our field of interest came to our attention. Most of the work we published over the past years has already been described to some extent in the chapters of this thesis. Namely the published works most tightly related to scholarly data access are the following:

- Personalized Access to Scientific Publications: from Recommendation to Explanation [124].
- RES: A Personalized Filtering Tool for CiteSeerX Queries Based on Keyphrase Extraction [125].
- Personalized Recommendation and Explanation by using Keyphrases Automatically extracted from Scientific Literature [131].
- A Personalized Concept-driven Recommender System for Scientific Libraries [127].
- A Content-Based Approach to Social Network Analysis: A Case Study on Research Communities [121].
- Modelling the User Modelling Community (and Other Communities as Well) [122].

The Information Extraction techniques used to perform the presented scholarly data filtering and exploration techniques presented are discussed in the following papers:

- A Domain Independent Double Layered Approach to Keyphrase Generation [37].
- A Keyphrase Generation Technique Based upon Keyphrase Extraction and Reasoning on Loosely Structured Ontologies [126].
- A New Multi-lingual Knowledge-base Approach to Keyphrase Extraction for the Italian Language [39].
- A Semantic Metadata Generator for Web Pages Based on Keyphrase Extraction [128].

- Introducing Distiller: a Lightweight Framework for Knowledge Extraction and Filtering [36].
- Introducing Distiller: a unifying framework for Knowledge Extraction [9].
- The Importance of Being Referenced: Introducing Referential Semantic Spaces [40].
- Leveraging Arabic Morphology and Syntax for Achieving Better Keyphrase Extraction [70].

A certain deal of effort was also put in solving some tangential problems and exploring some potentially relevant areas. A lot of ideas were left on the floor, but some of them eventually managed to make their way into the literature. In the following list we are presenting additional work that, though not being strictly instrumental to the discussion of Automatic Curator Systems, happened to touch some key topics in Information Extraction, knowledge management, and human computation.

- Users as Crawlers: Exploiting Metadata Embedded in Web Pages for User Profiling [130].
- A Thin-Server Approach to Ephemeral Web Personalization Exploiting RDF Data Embedded in Web Pages [129].
- Well-Stratified Linked Data for Well-Behaved Data Citation [123].
- Stratifying Semantic Data for Citation and Trust: an Introduction to RDFDF [35].
- Crowdsourcing Relevance Assessments: The Unexpected Benefits of Limiting the Time to Judge [97].

Among these last works, we consider particularly relevant for the scholarly data access domain, though not instrumental for our main discussion, the one presented in [123], since data citation is a hot topic in knowledge management and tracking the provenance of Linked Data and other information used to populate knowledge bases it is likely to become a critical task in the management of knowledge-base systems. The next Appendix is devoted to the presentation of that work.

B

Remarks on the Resolution of Data Citation and Its Feasibility

In this appendix we are presenting our study on the problem of resolving data citations in Linked Data. Though being critical for knowledge management purposes, such as tracking the provenance of data and supporting reproducibility of results, data citation is still far from being well established.

B.1 Introduction

Over the last years data has become a more and more critical asset both in research and in application. While there is a general agreement on the need for data citation to ensure research reproducibility and to facilitate data reuse, the research community is still debating how to concretely realise it. Citing data is not a trivial task since it has a few notable differences from citing literature: data evolve over time, data availability might change over time, only a subset of data might be relevant, and on top of that the authorship of data is not always clear since it may be the result of an automated process (e.g. sensor data), involve a large number of contributors (e.g. crowdsourcing), or even be built on the top of other data (e.g. inferring a taxonomy from a document corpus). Levering on the insights provided by [3], [4], [162], and [186] we outline the following Data Citation functional requirements:

- *Identification and Access*: Data Citation should provide a persistent, machine readable, and globally unique identifier for data; Moreover a reference to a persistent repository should also be provided to facilitate data access.
- *Credit and Attribution*: Data citation should facilitate giving credit and legal attribution to all contributors to the data. Such contributors might be humans as well as automated processes such as reasoners;
- *Evolution*: Data Citation should provide a reference to the exact version of the

cited data, since data might change over time. This is a fundamental requirement for research reproducibility purposes.

An additional, non functional requirement, is efficiency: the data citation should lead to the data in practical time, which means fast enough for the purposes of data consumer applications. For instance a database query allows to access the data in practical time, while solving a complex set of logical clauses probably does not.

In the last years Linked Data has rapidly emerged as the preferred format for publishing and sharing structured data, creating a vast network of interlinked datasets [69]. However the open nature of the format makes data provenance hard to track, moreover, the RDF Recommendation does not provide a clear mechanism for expressing meta-information about RDF documents. Semantic Web technologies such as OWL, RDF, and RDFS leverage upon description logic and first order logic and it is well known that an incautious usage of their primitives may lead to non decidable sets of conditions [73].

With respect to the requirements of a good data citation expressed above, the Semantic Web community has proposed a number of solutions to the data provenance problem which addresses the problem of assessing the authorship of data. Methods for partitioning RDF graphs have been proposed as well and also version identification and storage of RDF data have already been discussed. However most of those solutions imply the embedding of meta-information inside RDF data. This practice tends to make data cumbersome and the usage of reification [67] to realise tasks such as generating data subsets may lead to a combinatorial explosion of triples.

In this paper we discuss a simple framework to satisfy data citation requirements levering on the stratification of linked data, which basically means providing a separation between proper data and meta-information. Such separation can be effectively guaranteed with the usage of a simple type system allowing programs such as reasoners to discriminate in an efficient way. We would also like to show that the fact that Linked Data technologies such as RDF and OWL are powerful enough to let you seamlessly represent and embed meta-information inside the data does not mean that you really *should*.

Synopsis B.2 complements the introduction with a concise survey of related work and pointers for the interested reader. B.3 recalls an established formalisation of meta-information over linked data that abstracts from implementation dictated details and conventions. This formalisation is completed with a simple yet expressive algebra and an abstract notion of reasoner which offer the formal context for developing the paper main contribution. Said contribution is introduced in B.4 and is a notion of *coherence for (meta)information* capable of characterising properties of information organisation. This result is augmented with the notion of *coherent reasoner* i.e. an abstract characterisation of reasoners that can operate on coherent data while preserving such cornerstone property. Finally, we investigate the cost of verifying and preserving coherency:

- In B.4.3 we introduce a type system that allows us to reduce coherence checking to type checking; type inference can be applied to derive type-annotations in order to speed-up future checks.
- In B.4.4 we reduce coherence checking to suitable graph problems thus deriving an algorithm that can asses whether a given data store is coherent during a single

read i.e. in linear time.

- This verification algorithm proceeds incrementally; we show how, given a coherent data store, the algorithm can check whether an operation preserves coherency. Remarkably, this on-the-fly check has negligible costs: linear to the number of entries created, deleted or updated by the operation.

Each subsection is completed with a short *takeaway message* paragraph containing general remarks and intuitions aimed at rendering the technical results more accessible. In B.5 we discuss how these results and notions may be translated into practice: in particular, we envision a new modelling layer with richer language support on top of existing technologies such as RDF. Final remarks are provided in B.6.

B.2 Related Work

Data citation has already been explored by the Semantic Web community and it significantly overlaps with the problem of assessing data provenance since determining the authorship of data is vital for citation purposes and both tasks need meta-information over data. Provenance has already been widely discussed by the Semantic Web community leveraging on the insights provided by the Database community [20]. Provenance information can be represented exploiting two approaches: the annotation approach and the inversion approach [138]. In the first approach all meta-information is explicitly stated, while in the latter is computed when needed in a lazy fashion which requires external resources containing the information upon which provenance is entailed to be constantly available. The annotation approach is favoured since it provides richer information and allows data to be self-contained; several vocabularies have been proposed to describe meta-information over linked data such as *VoID* (Vocabulary of Interlinked Datasets) [2], that offers a rich language for describing Semantic Web resources built on top of well known and widely used ontologies such as foaf[1] and Dublincore[2], and *PROV Ontology* (PROV-O)[3], which is the lightweight ontology for provenance data standardized by the W3C Provenance Working Group. Regardless of the vocabulary used, adopting the annotation approach will result in producing a lot of meta-information which might exceed the actual data in size: provenance data in particular increases exponentially with respect to its depth [163]. For more information about the problem of data provenance, we reference the curious reader to [5]. The state of the art technique for embedding meta-information in RDF, is reification [186] which consists in assigning a URI to a RDF triple by expressing it as an *rdf:Statement* object. Recently the RDF 1.1 Recommendation [84] introduced the so called "RDF Quad Semantic" which consists in adding a fourth element to RDF statements which should refer to the name of the graph which the triple belongs to. The actual semantic of the fourth element however is only hinted, leaving room for interpretation and therefore allowing semantics tailored to fit application needs. In [162] is presented a methodology for citing linked data exploiting the quad semantics: the fourth element is used as identifier for RDF predicates allowing the definition of data subsets. Other usages of the fourth element include specification

[1] `http://xmlns.com/foaf/spec/`
[2] `http://dublincore.org/documents/dcmi-terms/`
[3] `http://www.w3.org/TR/prov-o/`

of a time frame, uncertainty marker, and provenance information container [23]. Finally, the idea of using a type system to ease the fruition of semantic resources is not new to the Semantic Web community: the authors of[30] propose a type system to facilitate programmatic access to RDF resources.

B.3 Formalisation

In this section we provide a uniform formalisation of meta-information over linked data and of the related operations. Our formalisation abstracts over implementation details (such as how triples are extended with a fourth element) retaining all relevant information.

B.3.1 Families of Named Graphs

A *Named Graph* (herein NG) is every labelled set of triples and may consist in a single labelled statement or in a larger set including multiple statements. Usually in RDF data triples are labelled using *reification*, however to the extents of our discussion how the triples are labelled is irrelevant. Following the formalisation proposed in [23] we define a family of NGs as a 5-tuple $\langle N, V, U, B, L\rangle$ where:

- U is a set of IRIs;
- L is a set of literals;
- B is a set of blank nodes;
- V is the union of the pairwise disjoint U, L, and B;
- N is a set of assignments $u \mapsto (v, u', v')$ mapping each $u \in U$ to at most one triple $(v, u', v') \in V \times U \times V$.

Equivalently, the set N can be read as a partial function $n\colon U \rightharpoonup V \times U \times V$ hence called *naming function* of the NG family. The set V is called *vocabulary* of the NG family for it defines all the IRIs, literals etc. appearing in the family.

Note how every pair in N consists in a label (the first element) and a non void RDF graph (the second element), thus is an NG. It is important to stress how the above formalisation holds regardless of the actual technique employed to associate an identifier to a named graph.

Intuitively, assigning an IRI to a triple puts that IRI in the rôle of *meta-information* with respect to that triple whence thought as *information*. Note that the separation between information and meta-information is not absolute but *relative* to the context i.e. the level at which the reasoning happens. For a concrete example, consider the RDF snipped:

```
x type      statement
x subject   y
x predicate b
x object    c
y type      statement
```

```
y subject   a
y predicate b
y object    c
```

Accordingly to the reification semantics, here x is assigned to triple (y, b, c) and y to the triple (a, b, c) hence, can be seen as an NG family whose N is corresponds to the assignments:

$$x \mapsto (y, b, c) \qquad \text{and} \qquad y \mapsto (a, b, c).$$

Clearly, y plays the rôle of meta-information with respect to the triple (a, b, c) and x plays the rôle of meta-information about (y, b, c) whence (a, b, c).

Remark B.3.1 (Blank nodes). Since the presence of blank nodes arises several non trivial problems for the purposes of merging and comparing linked data, the last W3C recommendation suggests the replacements of blank nodes with IRIs[4] when data references are expected. Linked data including blank nodes are still common on the Web, but one can always assume the existence a renaming function $r\colon B \rightharpoonup U$ that assigns new and unique IRIs to blank nodes.

Above we recalled NG families as introduced in [23] however, because of Remark B.3.1, we could equivalently characterise them by their associated naming function alone. Clearly, every 5-tuple $\langle N, V, U, B, L\rangle$ describing a family of NGs uniquely induces a naming function n but the opposite is not true in general. This can be traced down to the later lacking information the separation of $V \setminus U$ into blank nodes and literals. As a consequence of Remark B.3.1 we can safely assume $B = \emptyset$ and hence recover L as $V \setminus U$. Formally, for a naming function $n\colon U \rightharpoonup V \times U \times V$, define the associated NG family as follows:

1. N is the function graph of n i.e.

$$N \triangleq \{(u, (v, u', v')) \mid n(u) = (v, u', v')\};$$

2. V is the first or third projection of n codomain i.e.

$$V \triangleq \pi_1(\mathrm{cod}(n)) = \pi_3(\mathrm{cod}(n));$$

3. U is the second projection of n codomain i.e.

$$U \triangleq \pi_2(\mathrm{cod}(n));$$

4. L is the difference $V \setminus U$;
5. B is empty.

Henceforth, we use the two presentations interchangeably.

Takeaway message Along the lines of [23], we described an abstract formalisation encompassing RDF data triples labelled using reification, among other equivalent representations. The main reason behind this effort is to have a disciplined representation that:

[4] `http://www.w3.org/TR/rdf11-concepts/#section-blank-nodes`

- avoids the use of reification while retaining its expressive power;
- hides conventions and idiosyncrasies due to implementation details;
- clearly separates information from meta-information.

B.3.2 A simple algebra for NG families

Families of named graphs can be organised into the partial order[5] defined as:

$$n \sqsubseteq n' \overset{\triangle}{\iff} n(u) = (a, b, c) \implies n'(u) = (a, b, c).$$

Intuitively, $n \sqsubseteq n'$ means that n' has more data than n but does not carry any implication on the semantic of such information since, for instance, the former may contain semantically inconsistent information not held by the latter. We say that n' *is an extension of* n whenever $n \sqsubseteq n'$. Note that n and n' are not required to have the very same set of IRIs and literals (hence vocabulary). In fact, for any pair of NG families $n\colon U \rightharpoonup V \times U \times V$ and $n'\colon U' \rightharpoonup V' \times U' \times V'$, the only information about U, U', V, and V' we can infer from $n \sqsubseteq n'$ is that U' and V' overlap with U and V where n is defined. Formally:

$$n(u) = (a, b, c) \implies u, b \in U' \wedge a, c \in V'.$$

Two families are equivalent whenever they extend each other:

$$n \equiv n' \overset{\triangle}{\iff} n \sqsubseteq n' \wedge n \sqsupseteq n'.$$

Clearly, $\equiv$ is an equivalence relation and all of its equivalence classes have a canonical representative family i.e. the unique and minimal element in the class. Intuitively, to obtain such family we only need to start with any family in the equivalence class and trim its vocabulary and set of IRIs by removing any element that does not appear in the naming function range or image i.e. remove everything but u, a, b, c such that $n(u) = (a, b, c)$. Empty families form an equivalence class $\varnothing$ represented by the unique naming function for the empty vocabulary. Hereafter, we shall not distinguish families in the same equivalence class, unless otherwise stated.

A NG family is called *atomic* whenever it assigns exactly one name i.e. a naming function n that defined exactly on one element of its domain. We shall denote atomic families by their only assignment:

$$(x \mapsto (a, b, c))(u) \triangleq \begin{cases} (a, b, c) & \text{if } x = u \\ \bot & \text{otherwise} \end{cases}$$

which corresponds to the RDF triple a, b, c plus the fourth element x that identifies the named graph.

From the order-theoretic perspective, atomic NG families are atoms for the order $\sqsubseteq$.

[5] In general, even if we fix the vocabulary, NG families may form a proper class. Because of the scope of this work we shall avoid the technicality of partially ordered classes.

In fact, n is atomic whenever:

$$m \sqsubseteq n \implies m \equiv n \vee m \equiv \varnothing.$$

For the sake of simplicity, we shall abbreviate $x \mapsto (a, b, c)$ as (a, b, c) when the particular choice of x is irrelevant and confusion seems unlikely; we still assume x to be implicitly unique in the context of its use.

In general, any non-empty set S of NG families admits a minimal element $\sqcap S$ given by the intersection of all the families in S or, using the naming function presentation, to:

$$(\sqcap S)(u) = \begin{cases} (a, b, c) & \text{if } \forall n \in S \ n(u) = (a, b, c) \\ \bot & \text{otherwise} \end{cases}$$

from which it is immediate to see that $\sqcap S \sqsubseteq n$ for any $n \in S$. We may denote $\sqcap\{n_1, \ldots, n_k\}$ by $n_1 \sqcap n_2 \sqcap \cdots \sqcap n_k$ when confusion seems unlikely.

It could be tempting to dualise $\sqcap$ defining maximal elements as follows:

$$(\sqcup S)(u) = \begin{cases} (a, b, c) & \text{if } \exists n \in S \ n(u) = (a, b, c) \\ \bot & \text{otherwise} \end{cases}$$

however, note that this represents the merging of heterogeneous data and may not yield an NG-family, hence is not a total operation. For instance, consider any set containing some n and n' assigning some u to different triples e.g.: $n(u) = (a, b, c)$ and $n(u) = (a', b, c)$ where $a \neq a'$. The operation can be turned into a total one without altering the definition of NG family by adopting some resolution policy for conflicting overlaps: discarding both, either, introducing suitable renaming, or more sophisticated techniques developed under the name of *ontology alignment*[6] a complex task whose formalisation exceeds the scope of this work. In the remaining of this subsection we show how the aforementioned simple conflict resolution policies yield "surrogate" operators for $\sqcup$ corresponding to specific and well-known operations.

Before defining these operations let us introduce some auxiliary definitions and notation. For an NG family $n\colon U \rightharpoonup V \times U \times V$, its *support* is the subset $\lVert n \rVert$ of U whose elements are assigned to some graph by n:

$$\lVert n \rVert \triangleq \{u \mid n(u) \text{ is defined}\}.$$

For a pair of families n_1 and n_2, their *conflict set* $n_1 \not\downarrow n_2$ is the set of IRIs being assigned to different graphs by n_1 and n_2, i.e.:

$$n_1 \not\downarrow n_2 \triangleq \{u \mid \lVert n_1 \rVert \cap \lVert n_2 \rVert \wedge n_1(u) \neq n_2(u)\}.$$

For a family n, an injective map σ from its vocabulary defines a renamed family $n[\sigma]$:

$$n[\sigma](u) = \begin{cases} (\sigma(a), \sigma(b), \sigma(c)) & \text{if } n(u') = (a, b, c) \wedge \sigma(u') = u \\ \bot & \text{otherwise} \end{cases}$$

[6] Ontology alignment is a vast and debated topic that has been formalised in several ways, we address the curious reader to [189].

We adopt the convention that any renaming $\sigma\colon V \to V'$ implicitly extends to a renaming on any superset of V that acts as σ on V and as the identity elsewhere.

The first "surrogate" for $n_1 \sqcup n_2$ resolves conflicts by ignoring any conflicting information from n_2. Formally:

$$(n_1 \triangleright n_2)(u) \triangleq \begin{cases} n_1(u) & \text{if } u \in \llbracket n_1 \rrbracket \\ n_2(u) & \text{if } u \in \llbracket n_2 \rrbracket \setminus \llbracket n_1 \rrbracket \\ \bot & \text{otherwise} \end{cases}$$

We call this operation a surrogate for binary joins since it is total and agrees with joins whenever they are defined, i.e.

$$n_1 \triangleright n_2 = n_1 \sqcup n_2.$$

The operation $\triangleright$ is associative:

$$n_1 \triangleright (n_2 \triangleright n_3) = (n_1 \triangleright n_2) \triangleright n_3,$$

has the empty family as a left and right unit:

$$n \triangleright \varnothing = n = \varnothing \triangleright n,$$

is idempotent:

$$n \triangleright n = n,$$

and hence defines an idempotent monoid of NG families. In general, $\triangleright$ is not commutative: it is easy to check that commutativity does not hold unless n_1 and n_2 agree wherever both are defined *i.e.*, the set of conflicts $n_1 \not\downarrow n_2$ is empty. The operation $\triangleright$ is monotonic in both components:

$$\begin{aligned} n_1 \sqsubseteq n_2 &\implies n_1 \triangleright n_3 \sqsubseteq n_2 \triangleright n_3 \\ n_1 \sqsubseteq n_2 &\implies n_3 \triangleright n_1 \sqsubseteq n_3 \triangleright n_2 \end{aligned}$$

and distributes over meets:

$$n_1 \sqcap (n_2 \triangleright n_3) = (n_1 \sqcap n_2) \triangleright (n_1 \sqcap n_3).$$

The second "surrogate" for $n_1 \sqcup n_2$ uses an approach symmetric to $\triangleright$ by discarding conflicting data from its first operand. Because of the symmetry, it will be denoted as $\triangleleft$ and presented simply mirroring $\triangleright$:

$$n_1 \triangleleft n_2 \triangleq n_2 \triangleright n_1.$$

Like $\triangleright$, this operation yields an idempotent monoid over NG families; moreover, both operations share the same unit and distribute over each other.

A third option is to resolve conflicts by discarding every data assigned to IRIs in the

conflict

$$(n_1 \diamond n_2)(u) = \begin{cases} n_1(u) & \text{if } u \in [\![n_1]\!] \setminus (n_1 \not\downarrow n_2) \\ n_2(u) & \text{if } u \in [\![n_2]\!] \setminus (n_1 \not\downarrow n_2) \\ \bot & \text{otherwise} \end{cases}$$

Although this operation may seem more drastic than $\triangleright$ and $\triangleleft$, $\diamond$ treat its operands equally and still coincides with $\sqcup$ whenever the later is defined. As a consequence, $\diamond$ is commutative, associative, idempotent and has $\varnothing$ as its unit.

Note that the conflict set between $(n_1 \diamond n_2)$ and n_1 or n_2 is always empty and hence the following equations are well-defined and hold true:

$$n_1 \triangleright n_2 = (n_1 \diamond n_2) \sqcup n_1$$
$$n_1 \triangleleft n_2 = (n_1 \diamond n_2) \sqcup n_2.$$

Vice versa, $\diamond$ can be derived from $\triangleright$ and $\triangleleft$:

$$n_1 \diamond n_2 = (n_1 \triangleleft n_2) \sqcap (n_1 \triangleright n_2).$$

Finally, we describe a surrogate for $n_1 \sqcup n_2$ that handles conflicts without discarding any information from its operands by renaming all conflicting assignments made by n_1 and n_2[7]. Intuitively, this may be though to be implemented by "doubling" the conflict set as:

$$(n_1 \not\downarrow n_2) \times \{1, 2\} = \{\langle u, i\rangle \mid u \in n_1 \not\downarrow n_2 \wedge i \in \{1, 2\}\}$$

In general, we abstract from specific renaming policies by assuming a pair of injective maps σ_1 and σ_2 such that the conflict set $(n_1[\sigma_1] \not\downarrow n_2[\sigma_2])$ is empty. Referring to the intuitive implementation described above, each renaming σ_i, for i in $\{1, 2\}$, is defined as follows:

$$\sigma_i(u) = \begin{cases} \langle u, i\rangle & \text{if } u \in (n_1 \not\downarrow n_2) \\ u & \text{otherwise} \end{cases}$$

Then, the last operator is defined as

$$(n_1 \boxtimes n_2) \triangleq n_1[\sigma_1] \sqcup n_2[\sigma_2].$$

The binary join $n_1[\sigma_1] \sqcup n_2[\sigma_2]$ is always well-defined since $(n_1[\sigma_1] \not\downarrow n_2[\sigma_2])$ is empty by assumption on the renaming maps σ_1 and σ_2 and $n_1[\sigma_1] \sqcup n_2[\sigma_2] = n_1 \sqcup n_2$ whenever $n_1 \not\downarrow n_2 = \emptyset$.

We extend the conventions introduced for $\sqcap$ to the operations $\triangleleft$, $\triangleright$, $\diamond$, and $\boxtimes$: we shall use $\boxtimes\{n_1, \ldots, n_k\}$ for $n_1 \boxtimes n_2 \boxtimes \cdots \boxtimes n_k$ and vice-versa.

Every NG family can be defined using the constants and operations described in this subsection. More complex operations such as inferences are described in the remaining part of this section.

[7] For those familiar with OWL2, this strategy bears significant similarity with *punning* which is an implicit renaming of conflicting entities, e.g. a class and an individual or an object property and a data-type property sharing the same IRI.

Takeaway message In this subsection we introduced a simple yet expressive algebra for describing NG families. Although, more convenient constructs, operations or sophisticated techniques are not part of this algebra, we believe they can be easily implemented on top of it hence suggesting the use of this algebra as a *core* language for targeting (via compilation) any chosen implementation of NG families such as reified RDF [67] or graphical models like [17, 115, 47].

B.3.3 Reasoners over NG families

In this subsection we formalise the problems of provenance, subsetting, and versioning in the setting of NG families as formalised above. To this end, we introduce an abstract and general notion of *abstract reasoner* subsuming any process (automatic or not) transforming NG families. At this level of abstraction we formalise the problem of understanding whether

"x in n has been generated by γ"

for an IRI x, a named graph family n and reasoner γ and then show how provenance, subsetting, and versioning are covered as instances of the above.

Definition B.3.2. *An* (abstract) reasoner *is a (partial) function over NG families.*

For a simple example, consider a reasoner that expands a family n by computing the transitive and reflexive closure of any predicate b that n describes as being transitive or reflexive e.g. by means of some graph $(b, \texttt{predicate}, \texttt{transitive})$. Such reasoner can be described as assigning to any family n the least family closed under the derivation rules:

$$\frac{n(x) = (a, b, c)}{n \vdash (a, b, c)}$$

$$\frac{n \vdash (a, b, c) \quad n \vdash (c, b, d) \quad n \vdash (b, \texttt{predicate}, \texttt{transitive})}{n \vdash (a, b, d)}$$

$$\frac{n \vdash (b, \texttt{predicate}, \texttt{reflexive})}{n \vdash (a, b, a)}$$

Hence $\gamma(n) = \{(a, b, c) \mid n \vdash (a, b, c)\}$.

Likewise, symmetry can be computed considering the rule

$$\frac{n \vdash (b, \texttt{predicate}, \texttt{symmetric}) \quad n \vdash (a, b, c)}{n \vdash (c, b, a)}$$

and reversible predicates by means of:

$$\frac{n \vdash (b, \texttt{reverse}, \overline{b}) \quad n \vdash (a, b, c)}{n \vdash (c, \overline{b}, a)}$$

$$\frac{}{n \vdash (\texttt{reverse}, \texttt{predicate}, \texttt{symmetric})}$$

These examples describe *monotonic* reasoners since $n \sqsubseteq \gamma(n)$ for any family n; however, an abstract reasoner may be non-monotonic as well: consider for instance a human annotator performing a revision of an ontology, such an annotator is likely to both add

and withdraw triples from the knowledge base and still fits the definition of an abstract reasoner. A simple example of situation where the withdrawal of a triple is needed to keep data consistent is offered by the derivation rule:

$$\frac{n \not\vdash (a, \texttt{after}, b)}{n \vdash (a, \texttt{first}, \texttt{event})}$$

All examples of reasoners described so far essentially work at the triple level for they never really follow any IRI. This kind of reasoner never crosses the boundary between information and meta-information but this is not true in general. Actually, crossing such boundary is often necessary when reasoning about the (meta)information stored in an NG family as we introduce reasoners that handle operations over meta-information such as tracking data authorship, extracting data subsets, and managing versions. Before we delve into this topic, let us introduce some auxiliary reasoners and definitions that allow us to describe such reasoners as well.

For an abstract reasoner γ and an NG family n, we define the sets of created, updated, and deleted assignments as:

$$\begin{aligned} C_\gamma(n) &\triangleq [\![\gamma(n)]\!] \setminus [\![n]\!] \\ U_\gamma(n) &\triangleq \{x \mid x \in [\![\gamma(n)]\!] \cap [\![n]\!] \wedge n(x) \neq \gamma(n)(x)\} \\ D_\gamma(n) &\triangleq [\![n]\!] \setminus [\![\gamma(n)]\!] \end{aligned}$$

Given n and γ, computing the above sets could be prohibitively demanding even under the assumption of NG families being finite. In practice, changes are recorded by explicitly tagging all affected assignments. Since this good practice is not imposed by Definition B.3.2 we introduce new reasoners that extend the output of any given reasoner γ with this tagging information.

$$\begin{aligned} \Delta^C_\gamma(n) &\triangleq \gamma(n) \boxtimes \{(\gamma, \texttt{new}, x) \mid x \in C_\gamma(n)\} \\ \Delta^U_\gamma(n) &\triangleq \gamma(n) \boxtimes \{(\gamma, \texttt{upd}, x) \mid x \in U_\gamma(n)\} \\ \Delta^D_\gamma(n) &\triangleq \gamma(n) \boxtimes \{(\gamma, \texttt{del}, x) \mid x \in D_\gamma(n)\} \end{aligned}$$

Everything added by one of the above reasoners to the output of γ is meta-information with respect to said output. As expected, this data enables reasoning on the evolution of the family itself. For instance, consider the following simple inference that reconstructs which reasoner used other reasoners by looking whether they recorder their activity:

$$\frac{n \vdash (\gamma, \texttt{new}, y) \quad n(y) = (\delta, q, z) \quad p, q \in \{\texttt{new}, \texttt{upd}\}}{n \vdash (\gamma, \texttt{uses}, \delta)}$$

Clearly, any reasoning on the evolution of a named family due to the action of reasoners (via the oversight of Δ, for exposition convenience) inherently require that some references stored as named graphs are followed hence that the boundary between information and meta-information is crossed at some point. Levering on these definitions, one can easily describe reasoners that realise authorship attribution over data with different granularity, reasoners to label and extract data subsets and versions.

Takeaway message In this section, we unearthed the core issues arising from reasoning about information and meta-information as well and provided a framework for describing all operations needed to perform data citation. Since our formal treatment has been carried at the abstract level of named graph families, we have shown how these issues are inherent to the problem and independent from implementation details and specific techniques such as reification.

B.4 Coherent (meta)information

In this section we characterise a class of NG families called *well-stratified* with the fundamental property of stratifying meta-information over information in a way that prevents any infinite chain of "downward" references where the direction is interpreted as crossing the boundary between meta-information and information. Since practical NG families (hence triple stores) contain only a finite amount of explicit information, absence of such chains corresponds to the absence of reference cycles like, for instance, in the NG family:

$$x \mapsto (y, b, c) \sqcup y \mapsto (x, b, c).$$

We characterise a class of abstract reasoners, called *coherent*, that preserve the well-stratification property of named graph families they operate on. Finally, we introduce a decidable and efficient procedures for assessing the well-stratification of an NG family and operations on them.

B.4.1 Well-founded relations

Before we define well-stratified NG families, let us recall some auxiliary notions and notation. A binary relation R on a (non necessary finite) set X is called *well-founded* whenever every non-empty subset S of X has a minimal element i.e. there exists $m \in S$ that is not related by $s\, R\, m$ for $s \in S$. This means that we can intuitively walk along R going from right to left for finitely many steps i.e. we have to stop, eventually. In fact, well-foundedness can be reformulated say that R contains no (countable) infinite descending chain (i.e. an infinite sequence $x_0, x_1, x_2, \ldots$ such that $x_{n+1}\, R\, x_n$).

Example B.4.1. The predecessor relation $\{(x, x+1) \mid x \in \mathbb{N}\}$ on the set of natural numbers is well-founded. The prefix and suffix relations on the set Σ^* of finite words over the alphabet Σ is well-founded. Any acyclic relation on a finite set is (trivially) well-founded. Point-wise and lexicographic extensions of well-founded relations are well-founded.

This kind of relations are common-place in mathematics and computer science since they provide the structure for several inductive and recursive principles and, with regard to this work aim, approaches for proving termination. Intuitively, the idea is to equip the state space of an algorithm with a well-founded relation and then show that each step of the algorithm travels such relation right to left (descends). By well-foundedness hypothesis, all descent paths are bound to terminate in a finite number of steps.

Takeaway message Well-founded relations such as the successor relation over natural numbers are at the core of several techniques used to prove termination. Such techniques

revolve around the idea of reducing the problem under scrutiny to walks along said well-founded relation and hence termination follows by the fact that any such (descending) walk cannot be infinite.

B.4.2 Well-stratified NG families

The intuitive desiderata of an NG family being free from infinite paths descending along the chain of (meta-)information is formally captured by the following definition.

Definition B.4.2. *A family of named graph i.e.*

$$n(u) = (a, b, c) \implies u \succ a \wedge u \succ b \wedge u \succ c.$$

The relation $\prec$ is called witness *for n.*

Following any chain of assignments $x \mapsto a, b, c$ described by a well-stratified NG family has to eventually terminate since each step corresponds to a step along $\prec$ which is well-founded by hypotheses. Thus, any reasoner based on such visits is bound to terminate as long as it descends along $\prec$ and each internal step in its chain is decidable. Moreover, for a given NG family the length of these chains is known and bounded.

Operations described in Section B.3.2 preserve well-stratification under the assumption that all operands can share their witness $\prec$.

Abstract reasoners may easily break well-stratification. Intuitively most reasoning tasks and well-engineered human annotation processes should be coherent, however breaking the well-stratification of data is subtle and can be achieved even with monotonic reasoning. For instance, consider a set of triples where there exists a triple $(y, \texttt{type}, \texttt{statement})$ labelled with some IRI x and an abstract reasoner γ that adds a new triple $(x, \texttt{type}, \texttt{statement})$ labelled as y. This insertion is totally legit if we are using reification but introduces a circularity in the chain of meta data since the family now contains the following assignments:

$$x \mapsto (y, \texttt{type}, \texttt{statement}) \qquad y \mapsto (x, \texttt{type}, \texttt{statement})$$

and hence is no more well-stratified.

Definition B.4.3. *An abstract reasoner is called* well-behaved *whenever it preserves well-stratification.*

Reasoners for provenance, subsetting, and versioning are well-behaved since they cross the boundary between information and meta-information only in one direction: descent.

In general, assessing whether a reasoner is well-behaved may not be immediate especially since Definition B.3.2 describes them as "black boxes transforming NG families". There are several ways for describing classes of abstract reasoners with different degrees of expressivity. Covering all of them is out of the scope of this work and indeed impossible[8], still derivation rules are a presentation fit for many reasoners (like the ones described so far) and well known across the computer science community. This approach

[8] We leave to the reader the exercise of formalising an human reasoner and prove it well-behaved.

allows to quickly inspect the "internals of the box" and statically prove a reasoner well-behaved. Moreover, it is possible to define reasoners that are "well-behaved by design" by imposing suitable syntactic constraints on these derivation rules akin to rule formats developed in the field of concurrency theory ([113, 83]). Albeit interesting, this topic cannot be fully developed in the context of this work.

Takeaway message In this subsection we characterised a class of NG families that stratify meta-information over information without creating incoherences such as loops. In general, reasoners may easily break this cornerstone property and treating them as black boxes prevents any practical attempt to statically check whether they really break well-stratification. However, with access to enough information about the internal working of a reasoner (e.g. its description in terms of derivation rules), established formal methods can be applied to prove it well-behaved; even develop languages for creating reasoners guaranteed to be well-behaved. Remarkably, reasoners for provenance, subsetting, and versioning admit well-behaved implementations.

B.4.3 Assessing well-stratification using types

NG families share some similarities formal graphical languages like bigraphs and hierarchical graphs. This observation suggest to introduce a simple type system, along the line of [47], with a special type whose inhabitants are exactly well-stratified NG families. Then, to verify if a given family is well-stratified it would suffice to check if it is well-typed.

For the aims of this work, we introduce a simple type system whose only type ✓ is inhabited by exactly well-stratified families. Judgements are of the form

$$\Gamma \vdash n \colon \checkmark$$

where $n \colon U \rightharpoonup V \times U \times V$ is a family of named graphs and the stage Γ is a partial function from the vocabulary V to a well-founded structure. For instance, could map V to the set of natural numbers under the successor relation: $\Gamma \colon V \rightharpoonup \mathbb{N}$.

The proposed type system is composed by three typing rules:

$$\frac{}{\Gamma \vdash \varnothing \colon \checkmark}$$

$$\frac{\Gamma(x) > \Gamma(a) \quad \Gamma(x) > \Gamma(b) \quad \Gamma(x) > \Gamma(c)}{\Gamma \vdash x \mapsto (a,b,c) \colon \checkmark}$$

$$\frac{\Gamma_1 \vdash n_1 \colon \checkmark \quad \Gamma_2 \vdash n_2 \colon \checkmark \quad \Gamma = \Gamma_1 \sqcup \Gamma_2 \quad n = n_1 \sqcup n_2}{\Gamma \vdash n \colon \checkmark}$$

The first captures the fact that the empty family is always well-stratified. The second ensures that Γ describes relations on V such that the assignment $x \mapsto (a,b,c)$ is well-stratified. Finally, the third rule allows to break n and Γ reducing the problem to smaller objects which can then be checked separately (clearly, applying this rule with either n_1 or n_2 being $\varnothing$ is pointless).

Regardless of the structure used as codomain, Γ characterises a class of relations on the vocabulary V that are well-founded where Γ is defined: for a relation $\prec$ on V such

that

$$x \prec y \implies \Gamma(x) < \Gamma(y)$$

the restriction of $\prec$ to $\llbracket n \rrbracket$ is clearly well-founded. This property is enough to guarantee that any family n such that $\Gamma \vdash n\colon \checkmark$ (i.e. is well-typed) is well-founded. In fact, because of the above typing rules, Γ must be defined on every IRI and literal occurring in n and hence $\prec$ as above is a witness for n being well-stratified.

We do not need to "guess" Γ. This function can be obtained by applying the above typing judgements while considering Γ as an unknown collecting all the hypotheses on it (e.g. $\Gamma(x) > \Gamma(a)$ from the second rule) in a set of constraints. Any partial function satisfying these constraints can be used as Γ to derive $\Gamma \vdash n\colon \checkmark$. Although computing such solutions can be done pretty efficiently, at this point it suffices to prove solution existence to prove n well-stratified.

In practice, type checkers may be helped by providing typing annotations as separate meta-data, as primitives of a specialised language for NG families, or just as comments like in the following RDF snipped:

```
// $x$: 4; $y$: 2; $b$, $c$: 0
x type      statement
x subject   $y$
x predicate $b$
x object    $c$
```

This statically computed information can be used to optimise reasoners since $\Gamma(x)$ provides an upped bound to the length of meta-information/information steps starting from x. As we show in Subsection B.4.4, the very same information can be used to efficiently reject any operation that breaks well-stratification.

Takeaway message In this subsection we characterised well-stratified NG families by means of a simple type system and showed the type inference problem to be decidable. This approach suggests to explore the use of more expressive type systems and sortings in order to express/enforce richer properties about NG families. Moreover, the connection between NG families and formal graphical models suggests the possibility to extend compositionality results such us those offered by *monoidal sortings* [112] to this settings.

B.4.4 Assessing well-stratification using graphs

For a family n whose support $\llbracket n \rrbracket$ is finite, the only way to not be well-stratified is to contain cycles of dependencies between information and meta-information: the only way for a relation on a finite set to not be well-founded is to contain cycles. Therefore, if we read assignments described by n as arcs in a directed graph we can reduce the problem of checking if n is well-stratified to checking if this "graph of dependencies" is free from directed cycles.

Definition B.4.4. *For an NG family n its* dependency graph *G_n is a graph with $\llbracket n \rrbracket$ and $\{(x, y_i) \mid n(x) = (y_1, y_2, y_3) \wedge y_i \in \llbracket n \rrbracket\}$ as its set of nodes and edges, respectively.*

For a family n containing a finite amount of *explicit* meta-information (as in any real-world scenario) well-stratification can be checked in time linear to the number of assignments described by n (i.e. the cardinality of the set $\llbracket n \rrbracket$).

Proposition B.4.5. *For a family of named graphs n such that $\llbracket n \rrbracket$ is finite, n is well-stratified if and only if its dependency graph G_n is a directed acyclic graph.*

Proof. By hypothesis on n, G_n has finitely many edges and nodes hence the only way for it to contain an infinite path is to have a directed cycle. □

Absence of directed cycles reduces to the existence of a topological sorting which can be easily computed in polynomial time with Tarjan's algorithm. Intuitively, this amounts to a depth first visit of the dependency graph G_n: a graph whose nodes have at most three outgoing edges, hence the time complexity actually is linear.

Corollary B.4.6. *Well-stratification can be checked with a time cost linear in the size of $\llbracket n \rrbracket$.*

Because of the size that real-world triple stores can reach, computing a topological sorting of G_n from scratch every time an operation on n is performed can be a daunting task. In the remaining of this section we describe how to efficiently and precisely reject all changes that will break well-stratification.

In Section B.3.2 we described a core algebra for NG families highlighting that complex transformations basically reduce to insertions and deletions of name assignments (updates are modelled as atomic pairs of deletions and insertions, as usual). Clearly, deleting a named assignment preserves well stratification and hence only insertions need to be checked before being carried out.

A way to curb this cost is to cache the information about on the topological sorting in a map from $\llbracket n \rrbracket$ to some linear order relation on a dense but limited set such us the rational part of the interval $[0, 1)$. This order relation is not well-founded but it is acyclic and hence any restriction to a finite subset of $[0, 1)$ is well-founded. Moreover, being dense, we can always "make room" for newly inserted (meta)information.

In the following let n be a well-stratified NG family and let m be a partial map from U to the subset of rational numbers:

$$\{l \cdot 2^{-k} \mid k \in \mathbb{N} \wedge 0 \leq l < k\}.$$

Under the assumption that we start from an empty family n and an empty map m, the algorithm ensures that after an arbitrary sequence of insertions

- m is defined exactly on every element occurring in n as information or meta-information;
- the natural order on $\mathbb{Q}$ defines a well-founded relation on all elements where m is defined;
- an operation is rejected if, and only if, it does not preserve well-stratification of n.

Note that the first two points imply well-stratification.

Consider the insertion of $x \mapsto (a, b, c)$ in n; Since $x \notin \llbracket n \rrbracket$ we have to consider two main scenarios: in the first x does not occur in n (i.e. $m(x) = \perp$) whereas in the second x occur in n as information only ($m(x) \neq \perp$ but $n(x) = \perp$).

Assume $m(x) = \bot$, we need to assign to x a value above those assigned to a, b, and c but below everything that we already put above these three piece of data. Formally:

$$m(x) \triangleq y + \frac{|y - z|}{2} \tag{B.1}$$

$$y = \max\{m(a), m(b), m(c)\}$$

$$z = \inf\{w \mid y < w \wedge m(u) = w\} \tag{B.2}$$

Note that although the definition of z may seem a bit convoluted, it can be readily implemented by means of an ordered set of values from m: (B.2) is exactly the first successor to y in such structure and (B.1) corresponds to an insertion right between y and z.

Assume $m(x) \neq \bot$, we have three sub-cases:

1. If $m(x) < \max\{m(a), m(b), m(c)\}$ then at least one of a, b, or c occurs in n in the rôle of meta-information w.r.t. x and thus $x \mapsto (a, b, c)$ is rejected and the algorithm terminates.

2. If $m(x) = \max\{m(a), m(b), m(c)\}$ we need to promote x "pushing" everything above it up and everything else down as in the first scenario; therefore, $m(x)$ is redefined using (B.1).

3. If $m(x) > \max\{m(a), m(b), m(c)\}$ then no further action is required.

If the algorithm did not reject the operation, then it can be safely performed. The only step left is to assign the value 0 to any $d \in \{a, b, c\}$ for which m is undefined.

Carrying out the procedure sketched above requires a constant numbers of reads and writes of the map m whose efficiency depends on the implementation of choice but can be safely assumed as negligible.

Takeaway message As shown in the previous subsections, well-stratification is an useful property for NG families thus, being able to efficiently check

- if a family is well-stratified or
- if an operation preserves well-stratification

is of vital importance. In this subsection we described how these two questions can be answered with a cost that is linear in the number of assignments (i.e. the size of meta-information) and constant, respectively. These results are strictly related to those described in Subsection B.4.3 since the dependency graph G_n induced by a family n is a graph representation of the constraints on Γ derived by applying type inference to n. Therefore, we can read the above results as complementing the type system we introduced with an implementation.

B.5 Towards a Well-Stratified data language

In the previous sections we introduced the concept of well-stratified data, that is crucial to the practical realisation of data citation: as long as data is well-stratified resolv-

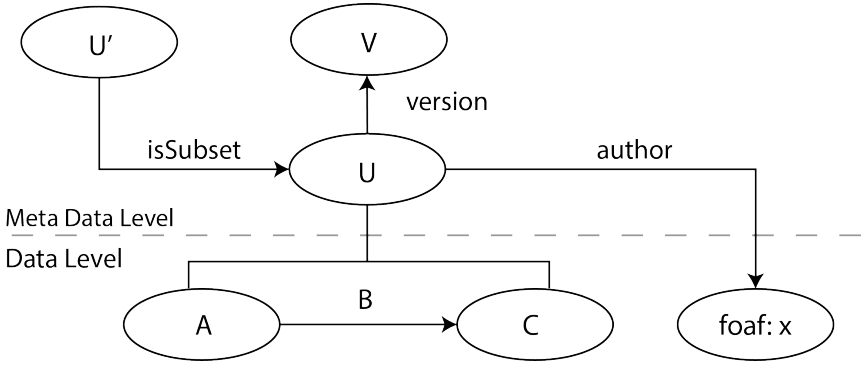

Figure B.1: An example of well stratified data.

ing a data citation is always possible. Moreover we showed how assessing the well-stratification of a data set is linear with respect to the size of the data and can be done incrementally as new statements are added to the data set. In this section we discuss how these notions may translate into practice.

Intuitively, for a data set to be well-stratified it means that it is always possible to draw a line separating information from meta-information while baring in mind that such separation is bound to be inherently relative to current datum. In order to illustrate this idea consider the simple example depicted in Figure B.1: the triple (A, B, C) is the information, while its identifier U and its related predicates represent the meta-information with respect to the datum considered i.e. (A, B, C). Intuitively U is the IRI of the reified *statement* that has as subject, object, and predicate respectively A, C, and B. Each IRI in the data set should belong to exactly one of these levels. Theoretically, there should be no upper bound to the number of meta-information levels since one might be interested in expressing statements over meta-information. An hypothetical reification, for instance, of $(U, \texttt{version}, V)$ having as identifier I would imply a further meta-information level containing only I.

This intuitive stratification can be thought as assigning levels to (meta-)information that decreases as we move from meta-information to information with the lowest level containing all the data that the resolution of a data citation should ultimately reach. Well-stratification not only ensures the existence of a lowest level but it also guarantees that it wherever a reasoner starts unravelling the chain of (meta-)information it will eventually reach said level.

Indeed this is the approach described in B.4 and embodied by the function $\Gamma\colon V \rightharpoonup \mathbb{N}$ used by the type system introduced in B.4.3. From a concrete perspective this means that identifiers of a reified statement should therefore be assigned a higher level by Γ than their subject, object, and predicate. This condition is met if, and only if, the data set is well-stratified. Finally, we remark that finding such levels can be done incrementally while reading the data set or on-the-fly as operations are performed on a well-stratified set; the cost of the former is linear in the size of the base whereas the cost of the second is linear on the number of operations.

The various data levels that can be identified this way can be considered as distinct data "slices" that, though linked, can be considered independent data sets and

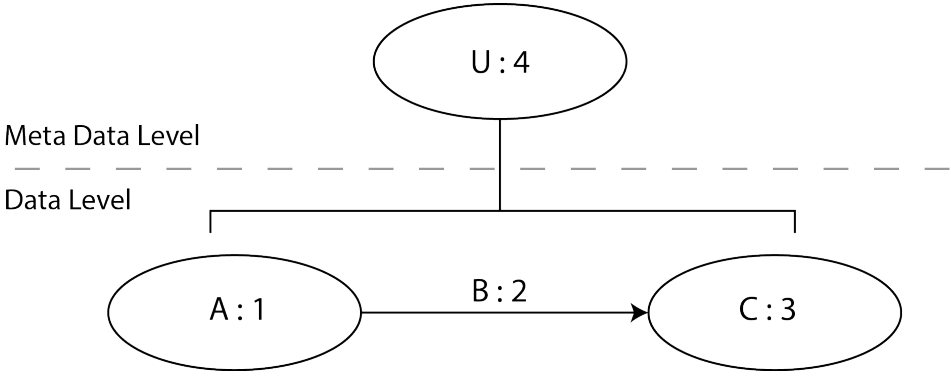

Figure B.2: An example of meta-information levels assignment.

treated accordingly. For instance a large, multi-layered data set could be distributed as a whole, with all layers of meta-information or "reduced" to its sole data level, i.e., without meta-information. Since well-stratification can be checked automatically in a reasonably efficient way and allows the separation of meta-information from ground level information, it addresses the need for such a separation introduced by state of the art data citation methodologies [162]. Such a concept, however, does not exist yet in the Semantic Web stack. The RDF language in fact has several limitations that make well-stratification hard to realise, and here we pinpoint the most evident:

- Checking well-stratification implies, as shown in Section B.4.3, the presence of a type-checking mechanism that does not exist in RDF.
- There is a data-level usage of reification (for instance to express sentences like "Bob says that Alice is kind") that must not be confused with the labelling of triples for meta-information expression purpose we analysed so far, and RDF does not provide a way to discriminate them.
- Assigning an identifier to a triple in RDF is not handy due to the bloated syntax of reification.

To overcome these limitations we strongly advocate the creation of a new language wherein the concept of well-stratification is a first-class citizen. More specifically, such a language should include in its specifications:

- a class for meta-information objects, allowing to explicitly state which triples are to be considered meta-information and which ground level information;
- a *level* property that can be associated to any IRI, serving as explicit annotation of the information level the IRI belongs to;
- a syntax for quad semantics, i.e., switching from a language of triples (like RDF) to a language of quadruples where triples are considered quadruples with a void fourth element;

- a more restrictive semantics for the fourth element of the quadruple, allowing to discriminate between reification for meta-information annotation purpose from actual data level usage;
- a *type system* for data including well stratified data.

Given such a language the actual information would be still expressed in the form of triples, allowing compatibility with the other levels of the Semantic Web, and the meta-information could be handled separately.

B.6 Conclusions

In this paper we briefly outlined a formalisation of data citation over linked data and showed how resolving authorship attribution, subset and version identification are computable in an efficient way as long as the considered data is well-stratified. Because of the relevance of this property, we explored how it can be expressed and verified by means of a type system and ad-hoc algorithms. We showed that checking whether a given NG family (which abstracts over the concrete form of data) is well-stratified requires linear time and proposed a constant time solution for checking if an operation performed by any reasoner preserves or breaks this property.

With respect to the problem of data citation, the expressive power of OWL and RDF is largely overabundant and might be harmful since a misuse of their primitives might break the stratification of information and meta-information, thus making resolving citations an undecidable problem. In our opinion, a more restricted language, designed specifically to grant the stratification of data should be taken into consideration to effectively enable problems such as data citation and provenance assessment to be resolved in practical time, allowing the creation of an effective data trust layer. Attaching meta-information to data published on the Web leveraging such a language might be, in our opinion, Linked Open Data's sixth star, like publishing versioned code is a fundamental quality requirement for Open Source software. The similarity between data meta-information handling and source code versioning is striking since they address similar problems: tracking who and how edited something, identifying subsets of the managed items, and allowing external application or documents to refer to a specific revision. In our opinion this separation is also consistent with the present development of the Semantic Web stack: OWL itself, thought being a logical extension of RDFS, is not built on the top of RDFS but is rather a distinct language sharing concepts and primitives with RDFS. In a similar way a new language for data meta-information management could be built compatibly with RDF and the Linked Data philosophy without being RDF. Finally, this work suggests a deeper connection between formal graph models and knowledge management problems encountered by the Digital Libraries and Semantic Web communities. In our opinion a more formal take on a broad range of non trivial knowledge management tasks and practises might provide relevant insights both on the application side and on the theoretical one as suggested by preliminary works in this directions like e.g. [177, 101, 102].

Summary

Accessing up-to-date and quality scientific literature is a critical preliminary step in any research activity. Identifying relevant scholarly literature for the extents of a given task or application is, however a complex and time consuming activity. Despite the large number of tools developed over the years to support scholars in their literature surveying activity, such as Google Scholar, Microsoft Academic search, and others, the best way to access quality papers remains asking a domain expert who is actively involved in the field and knows research trends and directions. State of the art systems, in fact, either do not allow exploratory search activity, such as identifying the active research directions within a given topic, or do not offer proactive features, such as content recommendation, which are both critical to researchers. To overcome these limitations, we strongly advocate a paradigm shift in the development of scholarly data access tools: moving from traditional information retrieval and filtering tools towards automated agents able to make sense of the textual content of published papers and therefore monitor the state of the art. Building such a system is however a complex task that implies tackling non trivial problems in the fields of Natural Language Processing, Big Data Analysis, User Modelling, and Information Filtering. In this work, we introduce the concept of Automatic Curator System and present its fundamental components.

Bibliography

[1] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131, 1999.

[2] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets with the void vocabulary, 2011. W3C interest group note.

[3] Micah Altman, Christine Borgman, Mercè Crosas, and Maryann Matone. An introduction to the joint principles for data citation. *Bulletin of the American Society for Information Science and Technology*, 41(3):43–45, 2015.

[4] Micah Altman and Mercè Crosas. The evolution of data citation: From principles to implementation. *IAssist Quarterly*, 63, 2013.

[5] Sarawat Anam, Byeong Ho Kang, Yang Sok Kim, and Qing Liu. Linked data provenance: State of the art and challenges. In *Proceedings of the 3rd Australasian Web Conference (AWC 2015)*, volume 27, page 30, 2015.

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

[7] A.L Barabsi, H Jeong, Z Nda, E Ravasz, A Schubert, and T Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(34):590 – 614, 2002.

[8] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence*, pages 40–52. Springer, 2000.

[9] Marco Basaldella, Dario De Nart, and Carlo Tasso. Introducing distiller: a unifying framework for knowledge extraction.

[10] Jeffrey Beall. Criteria for determining predatory open-access publishers. *Scholarly open access. https://scholarlyoa. files. wordpress. com/2015/01/criteria-2015. pdf,(accessed 2015-02-14)*, 2015.

[11] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. Research paper recommender system evaluation: a quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, pages 15–22. ACM, 2013.

[12] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[13] Bo-Christer Björk, Annikki Roos, and Mari Lauri. Global annual volume of peer reviewed scholarly articles and the share available via different open access options. In *ELPUB2008*, 2008.

[14] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[15] Paolo Bouquet, Jérôme Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris. D2. 2.1 specification of a common framework for characterizing alignment. 2004.

[16] A.Z. Broder. On the resemblance and containment of documents. In *Proceedings of Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29. IEEE, June 1997.

[17] Roberto Bruni, Andrea Corradini, Fabio Gadducci, Alberto Lluch-Lafuente, and Ugo Montanari. On gs-monoidal theories for graphs with nesting. In Gregor Engels, Claus Lewerentz, Wilhelm Schäfer, Andy Schürr, and Bernhard Westfechtel, editors, *Graph Transformations and Model-Driven Engineering - Essays Dedicated to Manfred Nagl on the Occasion of his 65th Birthday*, volume 5765 of *Lecture Notes in Computer Science*, pages 59–86. Springer, 2010.

[18] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 29–34, 2001.

[19] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[20] Peter Buneman, Sanjeev Khanna, and Tan Wang-Chiew. Why and where: A characterization of data provenance. In *Database TheoryICDT 2001*, pages 316–330. Springer, 2001.

[21] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16, 2006.

[22] Marine Carpuat, Grace Ngai, Pascale Fung, and Kenneth Church. Creating a bilingual ontology: A corpus-based approach for aligning wordnet and hownet. In *Proceedings of the 1st Global WordNet Conference*, 2002.

[23] Jeremy J Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web*, pages 613–622. ACM, 2005.

[24] Soumen Chakrabarti. *Mining the Web: Discovering knowledge from hypertext data*. Elsevier, 2002.

[25] Kannan Chandrasekaran, Susan Gauch, Praveen Lakkaraju, and Hiep Phuc Luong. Concept-based document recommendations for citeseer authors. In *Proceedings of the 5th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH '08, pages 83–92, Berlin, Heidelberg, 2008. Springer-Verlag.

[26] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[27] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *ACM Sigmod Record*, 35(3):34–41, 2006.

[28] Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):370–383, 2007.

[29] Philipp Cimiano and Johanna Völker. text2onto. In *International Conference on Application of Natural Language to Information Systems*, pages 227–238. Springer, 2005.

[30] Gabriel Ciobanu, Ross Horne, and Vladimiro Sassone. Minimal type inference for linked data consumers. *Journal of Logical and Algebraic Methods in Programming*, 2014.

[31] Tim Clark, Paolo N Ciccarese, and Carole A Goble. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of biomedical semantics*, 5(1):1, 2014.

[32] James Clarke, Vivek Srikumar, Mark Sammons, and Dan Roth. An nlp curator (or: How i learned to stop worrying and love nlp pipelines). In *LREC*, pages 3276–3283, 2012.

[33] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[34] Angus Dalton et al. The boy who never gave up. *Good Reading*, (Jun 2015):18, 2015.

[35] Dario De Nart, Dante DeglInnocenti, Marco Peressotti, and Carlo Tasso. Stratifying semantic data for citation and trust: an introduction to rdfdf. In *12th Italian Research Conference on Digital Libraries, IRCDL 2016, Florence, Italy*, 2016.

[36] Dario De Nart, Dante Degl'Innocenti, and Carlo Tasso. Introducing distiller: a lightweight framework for knowledge extraction and filtering. *UMAP 2015 Extended Proceedings*, 2015.

[37] Dario De Nart and Carlo Tasso. A domain independent double layered approach to keyphrase generation. In *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies*, pages 305–312. SCITEPRESS Science and Technology Publications, 2014.

[38] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer, 2004.

[39] Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. A new multi-lingual knowledge-base approach to keyphrase extraction for the italian language. In *Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval*, pages 78–85. SciTePress, 2014.

[40] Dante DeglInnocenti, Dario De Nart, and Carlo Tasso. The importance of being referenced: Introducing referential semantic spaces. In *Proceedings of the Joint Second Workshop on Language and Ontologies (LangOnto2) and Terminology and Knowledge Structures (TermiKS), Workshop Abstracts, Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portoro, Slovenia, May 23, 2016.*, 2016.

[41] Jörg Diederich and Wolf-Tilo Balke. The semantic growbag algorithm: Automatically deriving categorization systems. In *International Conference on Theory and Practice of Digital Libraries*, pages 1–13. Springer, 2007.

[42] Cory Doctorow. Metacrap: Putting the torch to seven straw-men of the meta-utopia. *Retrieved June*, 10:2003, 2001.

[43] Stephan Doerfel, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Leveraging publication metadata and social data into folkrank for scientific publication recommendation. In *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, pages 9–16, New York, NY, USA, 2012. ACM.

[44] Michael D Ekstrand, Praveen Kannan, James A Stemper, John T Butler, Joseph A Konstan, and John T Riedl. Automatically building research reading lists. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 159–166. ACM, 2010.

[45] Michael D Ekstrand, Michael Ludwig, Joseph A Konstan, and John T Riedl. Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 133–140. ACM, 2011.

[46] Samhaa R El-Beltagy and Ahmed Rafea. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1):132–144, 2009.

[47] Gregor Engels and Andy Schrr. Encapsulated hierarchical graphs, graph types, and meta types. *Electronic Notes in Theoretical Computer Science*, 2:101 – 109, 1995. SEGRAGRA.

[48] Jérôme Euzenat. An api for ontology alignment. In *The Semantic Web–ISWC 2004*, pages 698–712. Springer, 2004.

[49] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM.

[50] Felice Ferrara, Nirmala Pudota, and Carlo Tasso. A keyphrase-based paper recommender system. In *Italian Research Conference on Digital Libraries*, pages 14–25. Springer, 2011.

[51] Felice Ferrara, Nirmala Pudota, and Carlo Tasso. A keyphrase-based paper recommender system. In Maristella Agosti, Floriana Esposito, Carlo Meghini, and Nicola Orio, editors, *Digital Libraries and Archives*, volume 249 of *Communications in Computer and Information Science*, pages 14–25. Springer Berlin Heidelberg, 2011.

[52] Felice Ferrara and Carlo Tasso. Extracting and exploiting topics of interests from social tagging systems. *International Conference on Adaptive and Intelligent Systems*, pages 285–296, 2011.

[53] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and et al. Domain-specific keyphrase extraction. In *PROC. SIXTEENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 668–673. Morgan Kaufmann Publishers, 1999.

[54] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611, 2007.

[55] Eugene Garfield and Alfred Welljams-Dorof. Citation data: their use as quantitative indicators for science and technology evaluation and policy-making. *Science and Public Policy*, 19(5):321–327, 1992.

[56] Bela Gipp, Jöran Beel, and Christian Hentschel. Scienstein: A research paper recommender system. In *Proceedings of the international conference on Emerging trends in computing (ICETiC09)*, pages 309–315, 2009.

[57] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[58] Risto Gligorov, Warner ten Kate, Zharko Aleksovski, and Frank Van Harmelen. Using google distance to weight approximate ontology matches. In *the 16th International Conference on World Wide Web*, pages 767–776. ACM, 2007.

[59] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[60] V. Govindaraju and K. Ramanathan. Similar document search and recommendation. *Journal of Emerging Technologies in Web Intelligence*, 4(1):84–93, 2012.

[61] Somya Gupta, Namita Mittal, and Alok Kumar. Rake-pmi automated keyphrase extraction: An unsupervised approach for automated extraction of keyphrases. In *Proceedings of the International Conference on Informatics and Analytics*, page 102. ACM, 2016.

[62] David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. A closer look at skip-gram modelling. In *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, pages 1–4, 2006.

[63] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

[64] Mounia Haddoud, Aïcha Mokhtari, Thierry Lecroq, and Saïd Abdeddaïm. Accurate keyphrase extraction from scientific papers by mining linguistic information. In *Proc. of the Workshop Mining Scientific Papers: Computational Linguistics and Bibliometrics, 15th International Society of Scientometrics and Informetrics Conference (ISSI), Istanbul, Turkey: http://ceur-ws. org*, 2015.

[65] Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke S Zettlemoyer. Joint coreference resolution and named-entity linking with multi-pass sieves. In *EMNLP*, pages 289–299, 2013.

[66] Sebastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

[67] Patrick Hayes and Brian McBride. Rdf semantics, 2004.

[68] Marti A Hearst. Automated discovery of wordnet relations. *WordNet: an electronic lexical database*, pages 131–153, 1998.

[69] Tom Heath and Christian Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[70] Muhammad Helmy, Dario De Nart, Dante DeglInnocenti, and Carlo Tasso. Leveraging arabic morphology and syntax for achieving better keyphrase extraction. 2016.

[71] Jon Herlocker, Joseph A Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287–310, 2002.

[72] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[73] Ian Horrocks, Peter F Patel-Schneider, and Frank Van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Web semantics: science, services and agents on the World Wide Web*, 1(1):7–26, 2003.

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

[75] Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Data complexity of reasoning in very expressive description logics. In *IJCAI*, volume 5, pages 466–471, 2005.

[76] Tin Huynh, Kiem Hoang, Loc Do, Huong Tran, Hiep Luong, and Susan Gauch. Scientific publication recommendations based on collaborative citation networks. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 316–321. IEEE, 2012.

[77] Paul Jaccard. Lois de distribution florale. *Bulletin de la Socíeté Vaudoise des Sciences Naturelles*, 38:67–130, 1902.

[78] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[79] Yichen Jiang, Aixia Jia, Yansong Feng, and Dongyan Zhao. Recommending academic papers via users' reading purposes. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys '12, pages 241–244, New York, NY, USA, 2012. ACM.

[80] Dhiraj Joshi and Daniel Gatica-Perez. Discovering groups of people in google news. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 55–64. ACM, 2006.

[81] Heung-Nam Kim and Abdulmotaleb El Saddik. Personalized pagerank vectors for tag recommendations: inside folkrank. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 45–52, New York, NY, USA, 2011. ACM.

[82] Su Nam Kim and Min-Yen Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*, pages 9–16. Association for Computational Linguistics, 2009.

[83] Bartek Klin and Vladimiro Sassone. Structural operational semantics for stochastic and weighted transition systems. *Information and Computation*, 2013.

[84] Graham Klyne, Jeremy J Carrol, and Brian Mc Bride. Rdf 1.1 concepts and abstract syntax, 2014.

[85] Dean B Krafft, Nicholas A Cappadona, Brian Caruso, Jon Corson-Rikert, Medha Devare, Brian J Lowe, et al. Vivo: Enabling national networking of scientists. In *Proceedings of the Web Science Conference*, volume 2010, pages 1310–1313, 2010.

[86] M. Krapivin, M. Marchese, A. Yadrantsau, and Yanchun Liang. Unsupervised key-phrases extraction from scientific papers using domain and linguistic knowledge. In *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*, pages 105–112, Nov 2008.

[87] Vijay Krishnan and Christopher D Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1121–1128. Association for Computational Linguistics, 2006.

[88] Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. Unsupervised keyword extraction for japanese legal documents. In *JURIX*, pages 97–106, 2013.

[89] Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. In *Australasian Joint Conference on Artificial Intelligence*, pages 665–671. Springer, 2016.

[90] Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, 7(1):1, 2015.

[91] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL)*, pages 25–32. ACL, 199.

[92] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. Lcc approaches to knowledge base population at tac 2010. In *Proc. TAC 2010 Workshop*, 2010.

[93] J. Leskovec, A. Rajaraman, and J.D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[94] Kenli Li, Wei Ai, Zhuo Tang, Fan Zhang, Lingang Jiang, Keqin Li, and Kai Hwang. Hadoop recognition of biomedical named entity using conditional random fields. *IEEE Transactions on Parallel and Distributed Systems*, 26(11):3040–3051, 2015.

[95] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[96] Patrice Lopez and Laurent Romary. Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 248–251. Association for Computational Linguistics, 2010.

[97] Eddy Maddalena, Marco Basaldella, Dario De Nart, Dante Degl'Innocenti, Stefano Mizzaro, and Gianluca Demartini. Crowdsourcing relevance assessments: The unexpected benefits of limiting the time to judge. In *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[98] Jayant Madhavan, Philip A Bernstein, AnHai Doan, and Alon Halevy. Corpus-based schema matching. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 57–68. IEEE, 2005.

[99] Alexander Maedche and Steffen Staab. Ontology learning. In *Handbook on ontologies*, pages 173–190. Springer, 2004.

[100] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[101] Alessio Mansutti, Marino Miculan, and Marco Peressotti. Distributed execution of bigraphical reactive systems. *ECEASST*, 71, 2014.

[102] Alessio Mansutti, Marino Miculan, and Marco Peressotti. Multi-agent systems design and prototyping with bigraphical reactive systems. In Kostas Magoutis and Peter Pietzuch, editors, *Proc. DAIS*, volume 8460 of *Lecture Notes in Computer Science*, pages 201–208. Springer, 2014.

[103] Luís Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P Neto. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. *arXiv preprint arXiv:1306.4886*, 2013.

[104] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[105] Andrew McCallum, Andres Corrada-Emmanuel, and Xuerui Wang. Topic and role discovery in social networks. *Computer Science Department Faculty Publication Series*, page 3, 2005.

[106] Paul McNamee, Hoa Trang Dang, Heather Simpson, Patrick Schone, and Stephanie Strassel. An evaluation of technologies for knowledge base population. In *LREC*, 2010.

[107] Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 116–125. ACM, 2002.

[108] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[109] Lokman I Meho. The rise and rise of citation analysis. *Physics World*, 20(1):32, 2007.

[110] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[111] Robert K Merton et al. The matthew effect in science. *Science*, 159(3810):56–63, 1968.

[112] Marino Miculan and Marco Peressotti. Bigraphs reloaded: a presheaf presentation. Technical Report UDMI/01/2013, Dept. of Mathematics and Computer Science, Univ. of Udine, 2013.

[113] Marino Miculan and Marco Peressotti. Structural operational semantics for non-deterministic processes with quantitative aspects. *CoRR*, abs/1410.0893, 2014.

[114] Peter Mika. Flink: Semantic web technology for the extraction and analysis of social networks. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):211–223, 2005.

[115] Robin Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.

[116] Andrea Moro, Francesco Cecconi, and Roberto Navigli. Multilingual word sense disambiguation and entity linking for everybody. In *Proceedings of the 2014 International Conference on Posters & Demonstrations Track-Volume 1272*, pages 25–28. CEUR-WS. org, 2014.

[117] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2, 2014.

[118] Adrian Muller, Jochen Dorre, Peter Gerstl, and Roland Seiffert. The taxgen framework: Automating the generation of a taxonomy for a large document collection. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 9–pp. IEEE, 1999.

[119] Cataldo Musto, Pasquale Lops, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. Semantics-aware graph-based recommender systems exploiting linked open data. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 229–237. ACM, 2016.

[120] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[121] Dario De Nart, Dante Degl'Innocenti, Marco Basaldella, Maristella Agosti, and Carlo Tasso. A content-based approach to social network analysis: A case study on research communities. In *Digital Libraries on the Move - 11th Italian Research Conference on Digital Libraries, IRCDL 2015, Bolzano, Italy, January 29-30, 2015, Revised Selected Papers*, pages 142–154, 2015.

[122] Dario De Nart, Dante Degl'Innocenti, Andrea Pavan, Marco Basaldella, and Carlo Tasso. Modelling the user modelling community (and other communities as well). In *User Modeling, Adaptation and Personalization - 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 - July 3, 2015. Proceedings*, pages 357–363, 2015.

[123] Dario De Nart, Dante Degl'Innocenti, and Marco Peressotti. Well-stratified linked data for well-behaved data citation. *Bulletin of IEEE Technical Committee on Digital Libraries*, 12, 2016.

[124] Dario De Nart, Felice Ferrara, and Carlo Tasso. Personalized access to scientific publications: from recommendation to explanation. In *User Modeling, Adaptation, and Personalization - 21th International Conference, UMAP 2013, Rome, Italy, June 10-14, 2013, Proceedings*, pages 296–301, 2013.

[125] Dario De Nart, Felice Ferrara, and Carlo Tasso. RES: A personalized filtering tool for citeseerx queries based on keyphrase extraction. In *User Modeling, Adaptation, and Personalization - 21th International Conference, UMAP 2013, Rome, Italy, June 10-14, 2013, Proceedings*, pages 341–343, 2013.

[126] Dario De Nart and Carlo Tasso. A keyphrase generation technique based upon keyphrase extraction and reasoning on loosely structured ontologies. In *Proceedings of the 7th International Workshop on Information Filtering and Retrieval co-located with the 13th Conference of the Italian Association for Artificial Intelligence (AI*IA 2013), Turin, Italy, December 6, 2013.*, pages 49–60, 2013.

[127] Dario De Nart and Carlo Tasso. A personalized concept-driven recommender system for scientific libraries. In *Pushing the Boundaries of the Digital Libraries Field - 10th Italian Research Conference on Digital Libraries, IRCDL 2014, Padua, Italy, January 30-31, 2014*, pages 84–91, 2014.

[128] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A semantic metadata generator for web pages based on keyphrase extraction. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 201–204, 2014.

[129] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A thin-server approach to ephemeral web personalization exploiting RDF data embedded in web pages. In *Proceedings of the 2nd Workshop on Society, Privacy and the Semantic Web - Policy and Technology (PrivOn 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Trento, Italy, October 20, 2014.*, 2014.

[130] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. Users as crawlers: Exploiting metadata embedded in web pages for user profiling. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization co-located with the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP2014), Aalborg, Denmark, July 7-11, 2014.*, 2014.

[131] Dario De Nart, Carlo Tasso, and Felice Ferrara. Personalized recommendation and explanation by using keyphrases automatically extracted from scientific literature. In *KDIR/KMIS 2013 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing, Vilamoura, Algarve, Portugal, 19 - 22 September, 2013*, pages 96–103, 2013.

[132] M. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Phys. Rev. E*, 64:016131, Jun 2001.

[133] M. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64:016132, Jun 2001.

[134] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[135] Grace Ngai, Marine Carpuat, and Pascale Fung. Identifying concepts across languages: A first step towards a corpus-based approach to automatic ontology alignment. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[136] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-nerd: Joint named entity recognition and disambiguation with rich linguistic features. *Transactions of the Association for Computational Linguistics*, 4:215–229, 2016.

[137] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *International Conference on Asian Digital Libraries*, pages 317–326. Springer, 2007.

[138] Temitope Omitola, Nicholas Gibbins, and Nigel Shadbolt. Provenance in linked data integration. In *Future Internet Assembly*, pages 16–17. Ghent, Belgium, 2010.

[139] Francesco Osborne and Enrico Motta. Exploring research trends with rexplore. *D-Lib Magazine*, 19(9):4, 2013.

[140] Francesco Osborne and Enrico Motta. Inferring semantic relations by user feedback. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 339–355. Springer, 2014.

[141] Francesco Osborne and Enrico Motta. Rexplore: Unveiling the dynamics of scholarly data. In *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, pages 415–416. IEEE, 2014.

[142] Francesco Osborne and Enrico Motta. Klink-2: integrating multiple web sources to generate semantic topic networks. In *International Semantic Web Conference*, pages 408–424. Springer, 2015.

[143] Francesco Osborne, Enrico Motta, and Paul Mulholland. Exploring scholarly data with rexplore. In *International semantic web conference*, pages 460–477. Springer, 2013.

[144] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 85–92. ACM, 2013.

[145] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441–453, 2002.

[146] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. In *International semantic web conference*, pages 30–43. Springer, 2006.

[147] Mohammad Taher Pilehvar and Roberto Navigli. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128, 2015.

[148] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[149] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164. ACM, 2011.

[150] Nirmala Pudota, Antonina Dattolo, Andrea Baruzzo, and Carlo Tasso. A new domain independent keyphrase extraction system. In *Italian Research Conference on Digital Libraries*, pages 67–78. Springer, 2010.

[151] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[152] Giuseppe Rizzo and Raphaël Troncy. Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–76. Association for Computational Linguistics, 2012.

[153] M Andrea Rodríguez and Max J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE transactions on knowledge and data engineering*, 15(2):442–456, 2003.

[154] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

[155] Warren Sack. Conversation map: a content-based usenet newsgroup browser. In *From Usenet to CoWebs*, pages 92–109. Springer, 2003.

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

[157] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213. ACM, 1999.

[158] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[159] Michael Schuhmacher and Christian Meilicke. Popular books and linked data: some results for the eswc14 recsys challenge. In *Semantic Web Evaluation Challenge*, pages 176–181. Springer, 2014.

[160] J.P. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, January 2000.

[161] Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):158–176, 2013.

[162] Gianmaria Silvello. A methodology for citing linked open data subsets. *D-Lib Magazine*, 21(1):6, 2015.

[163] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405, 2005.

[164] Renata Tagliacozzo. Self-citations in scientific literature. *Journal of Documentation*, 33(4):251–265, 1977.

[165] Mike Thelwall and Kayvan Kousha. Researchgate: Disseminating, communicating, and measuring scholarship? *Journal of the Association for Information Science and Technology*, 66(5):876–889, 2015.

[166] Maksim Tkachenko and Andrey Simanovsky. Named entity recognition: Exploring features. In *KONVENS*, pages 118–127, 2012.

[167] Peter Turney. Coherent keyphrase extraction via web mining. 2003.

[168] Peter D Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.

[169] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. Agdistis-graph-based disambiguation of named entities using linked data. In *The Semantic Web–ISWC 2014*, pages 457–471. Springer, 2014.

[170] Anthony FJ Van Raan. Sleeping beauties in science. *Scientometrics*, 59(3):467–472, 2004.

[171] Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharath Reddy, Sudheer Kovelamudi, Srikanth Reddy Vaddepally, Radheshyam Nanduri, N Kiran Kumar, Santhosh Gsk, and Prasad Pingali. Iiit hyderabad in guided summarization and knowledge base population. *International Institute of Information Technology*, 2010.

[172] Paola Velardi, Roberto Navigli, Alessandro Cucchiarelli, and Fulvio D'Antonio. A new content-based model for social network analysis. In *ICSC*, pages 18–25. IEEE Computer Society, 2008.

[173] Suzan Verberne, Maya Sappelli, Djoerd Hiemstra, and Wessel Kraaij. Evaluation and analysis of term scoring methods for term extraction. *Information Retrieval Journal*, 19(5):510–545, 2016.

[174] Nina Wacholder, Yael Ravin, and Misook Choi. Disambiguation of proper names in text. In *Proceedings of the fifth conference on Applied natural language processing*, pages 202–208. Association for Computational Linguistics, 1997.

[175] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[176] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.

[177] Wusheng Wang and Thomas T. Hildebrandt. Dynamic ontologies and semantic web rules as bigraphical reactive systems. In Emilio Tuosto and Chun Ouyang, editors, *Proc. WS-FM*, volume 8379 of *Lecture Notes in Computer Science*, pages 127–146. Springer, 2013.

[178] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[179] D.J. Watts. *Small Worlds: the dynamics of networks between order and randomness*. Princeton Univ Pr, 1999.

[180] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[181] Julie Weeds and David Weir. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475, 2005.

[182] Ian Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[183] Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. ACM, 1999.

[184] Gerhard Wohlgenannt, Albert Weichselbraun, Arno Scharl, and Marta Sabou. Dynamic integration of multiple evidence sources for ontology learning. *Journal of Information and Data Management*, 3(3):243, 2012.

[185] Xiaoyuan Wu and Alvaro Bolivar. Keyword extraction for contextual advertisement. In *Proceedings of the 17th international conference on World Wide Web*, pages 1195–1196. ACM, 2008.

[186] Marcin Wylot, Philippe Cudre-Mauroux, and Paul Groth. Tripleprov: Efficient processing of lineage queries in a native rdf store. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 455–466, New York, NY, USA, 2014. ACM.

[187] Chengzhi Zhang. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.

[188] Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. Keyword extraction using support vector machine. In *International Conference on Web-Age Information Management*, pages 85–96. Springer, 2006.

[189] Antoine Zimmermann, Markus Krötzsch, Jérôme Euzenat, and Pascal Hitzler. Formalizing ontology alignment and its operations with category theory. In *Proc. 4th International conference on Formal ontology in information systems (FOIS)*, pages 277–288. IOS Press, 2006.