POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione

Corso di Laurea Specialistica in Ingegneria Informatica

Automatic meta-research on scientific communities

Relatore: Prof. Carlo GHEZZI

Correlatore: Ing. Giordano TAMBURRELLI

Prof. Maurizio MORISIO

Tesi di Laurea di:

Mario SANGIORGIO

Matricola n. 733085

Anno Accademico 2009–2010

Ringraziamenti

Desidero ringraziare tutti coloro che in questi anni mi sono stati vicini, accompagnandomi nel percorso che mi ha fatto giungere a questo importante traguardo.
Un grazie particolare spetta ovviamente alla mia famiglia, che mi ha sempre aiutato e supportato nelle mie scelte. Gran parte di questo risultato è merito vostro.
Grazie anche ai miei amici di sempre, che mi sono stati vicini anche quando il tempo per vederci non è stato molto.
Un grazie molto sentito anche per i miei compagni di corso, che sono riusciti a rendere questi cinque anni davvero indimenticabili. In particolare vorrei ringraziare i miei coinquilini, la compagnia delle uscite milanesi e chi ha condiviso con me l'esperienza americana.
Infine vorrei ringraziare chi mi ha seguito e chi ha contribuito a questo lavoro, facendo in modo che fosse sempre interessante e stimolante.

Mario

Contents

1 Introduction **1**

1.1 Problem statement and research context 2

1.1.1 Analysis of a scientific community 2

1.1.2 Tools to support research 3

1.2 Motivations for automatic analysis 3

1.3 Outline of this dissertation 5

2 State of the Art **6**

2.1 Related meta-research works 7

2.1.1 Research topics analysis 7

2.1.2 Authors and scientific productivity 10

2.1.3 Social network analysis 12

2.1.4 Geographical analysis 15

2.1.5 Forecasting future trends 16

2.1.6 Digital libraries and public databases 17

2.2 Useful techniques 18

2.2.1 Information Retrieval 18

2.2.2 Text-mining 20

2.2.3 Natural language processing 22

2.2.4 Recommender Systems 23

3 Methodologies **25**

3.1 Motivations 25

3.1.1 Research communities 26

3.1.2 Goal of this work 27

3.2 Identification of the research topics 28

3.2.1 Design choices 29

3.3 Getting the topic of a paper 34

3.3.1 Design choices 35

3.4 Profiling the authors 37

3.4.1 Design choices 38

3.5 Outcomes of this work 41

3.5.1 Research topic evolution 41

3.5.2 Automatic bidding 44

4 Implementation **50**

4.1 Workflow 50

4.2 Data collection 52

4.2.1 Digital libraries crawlers 53

4.2.2 Common data structure 56

4.2.3 DBLP Downloader 58

4.3 Data preprocessing 58

4.4 Research topic identification 62

4.5 Paper classification 67

4.6 Author profiling 71

4.7 Applications 74

4.7.1 Evolution of the research topics 75

4.7.2 Bidding recommender system 77

5 Experimental Results **82**

5.1 Research Field Analysis 83

5.2 Paper classification 91

5.3 Profiling of a researcher 97

5.4 Topic evolution 100

5.4.1 IEEE TSE 101

5.4.2 ACM TOSEM 107

5.4.3 ICSE 113

5.4.4 General conclusions about software engineering . . . 117

5.5 Automatic Bidding 119

5.5.1 ICSM 2010 - Offline Test 120

5.5.2 ICSE 2011 - Live Test 122

6 Conclusion and Future Work **128**

6.1 Conclusions 128

6.2 Future Work 129

List of Figures

3.1 The subdivision of research in subjects, research areas and topics 29
3.2 The data-driven topic identification process 30
3.3 The iterative process and the role of the expert supervisor . . 33
3.4 Classification of the topic of an unseen paper 34
3.5 Training of the classifier from the sets of papers that represent each topic 37
3.6 The creation of researcher's profile 38
3.7 The process of research topic evolution analysis 42
3.8 The bidding process and the assignment of the papers to review 45

4.1 The tools developed in this work 51
4.2 The tools used for the research trends analysis 52
4.3 The tools used in the bidding recommender system 52
4.4 The modular architecture of the crawler 54
4.5 The interaction between the crawler and the web server of a digital library 56
4.6 The entity-relationship diagram of the article database . . . 57
4.7 The preprocessing steps performed to make a document ready for the analysis 61

4.8 Two iterations of the Hierarchical Expectation Maximization (HEM) algorithm with the branching factor set to two 65
4.9 The iterative clustering process 67
4.10 The classification process 70
4.11 The profile creation process 72
4.12 Different kind of author profiles 74
4.13 The analysis process with an example of the graphical outcome 77
4.14 The process to automatically obtain a suggested bid 80

5.1 Precision and recall for the document topic classifier 95
5.2 Results of the survey to check the correctness of the profiles 99
5.3 The distribution of the errors on the profiles after the fix to the model 100
5.4 The distribution of the wrong topics before and after the addition of the two new topics 101
5.5 Distribution of the topics for TSE from 1976 to 1987 104
5.6 Distribution of the topics for TSE from 1988 to 1999 105
5.7 Distribution of the topics for TSE from 2000 to 2010 106
5.8 The clearest trends in TSE 108
5.9 Distribution of the topics for TOSEM from 1992 to 2000 . . . 109
5.10 Distribution of the topics for TOSEM from 2001 to 2010 . . . 110
5.11 Trends for the mainstream topics in TOSEM 112
5.12 Other interesting trends in TOSEM 112
5.13 Distribution of the topics for ICSE from 1976 to 1987 114
5.14 Distribution of the topics for ICSE from 1988 to 1999 115
5.15 Distribution of the topics for ICSE from 2000 to 2010 116
5.16 Trends for the mainstream topics in ICSE 118

5.17 Other interesting trends in ICSE 118
5.18 Answers to the first question for the ICSE 2011 experiment . 125
5.19 Answers to the second question for the ICSE 2011 experiment 126
5.20 Answers to the third question for the ICSE 2011 experiment 126

List of Tables

5.1 The conferences from which the training papers have been taken 85
5.2 The journals from which the training papers have been taken 86
5.3 The topics found for the software engineering research field and the most frequent terms for each of them 88
5.4 List of the terms that could be ignored in software engineering papers 89
5.5 Topics suggested by the domain expert 91
5.6 Topics obtained by the clustering algorithm. The new topics are highlighted with a star. 92
5.7 Some examples of the topic guessed by the classifier 96
5.8 Number of papers retrieved from the TSE archive 102
5.9 Number of papers retrieved from the TOSEM archive 108
5.10 Number of papers retrieved from the ICSE archive 113

List of Abbreviations

ACM Association for Computing Machinery

API Application Programming Interface

ASE International Conference on Automated Software Engineering

ASPEC Asia Pacific Software Engineering Conference

COMPSAC Computer Software and Applications Conference

DBLP Digital Bibliography and Library Project

DBMS Database Management System

ESEC European Software Engineering Conference

ESEM Empirical Software Engineering and Measurement

FASE Fundamental Approaches to Software Engineering

FM International Symposium of Formal Methods

FSE Symposium on the Foundations of Software Engineering

ICSE International Conference on Software Engineering

ICPC International Conference on Program Comphrension

ICSM International Conference on Software Maintenance

ICST International Conference on Software Testing

ISSTA International Symposium on Software Testing and Analysis

IEEE Institute of Electrical and Electronics Engineers

HEM Hierarchical Expectation Maximization

HTTP Hyper Text Transfer Protocol

HTML Hyper Text Markup Language

MALLET Machine Learning for Language Toolkit

MoDELS Model Driven Engineering Languages And Systems

OOPSLA Object-Oriented Programming, Systems, Languages and Applications

PDF Portable Document Format

POPL Symposium on Principles of Programming Languages

RE Requirements Engineering Conference

SoftVis Software Visualization

SWEBOK Software Engineering Body of Knowledge

TOSEM Transactions On Software Engineering and Methodology

TSE Transactions on Software Engineering

UML Unified Modeling Language

WCRE Working Conference on Reverse Engineering

WOSP Workshop on Software and Performance

Summary

The evolution and the growth of scientific communities introduced the need of tools to help the researchers knowing better they research field. This work studied and developed a set of tools to perform meta-research in an automated or semi-automated fashion. In particular, the methodologies developed within this thesis work include: a tool to identify the topic of a research field, an instrument able to tell what a paper is talking about and a technique to determine the topics a researcher is interested in. These tools rely on text mining techniques that made it possible to extract the information from the literature in a data driven process. A huge set of articles from the literature has been used as the data source for the topic identification process and to train the tool that guesses the topic of a paper. The profile of the researchers has been built considering the articles he or she previously wrote.

The analytical tools developed have then been used in the realization of two different kind of application. The first is devoted to perform studies about the evolution of the research topics by first identifying them and later observing how they distributed over the years in some relevant conferences of the research field. The second application developed is a recommender system that wants to help the members of the organizing committee of a conference in the selection of the papers to review by suggest-

ing them the articles the best match with their profile.

The methodology has then been validated by applying it to the software engineering research field. After having generated a model of the research interest, their evolution has been studied in two journals (IEEE TSE and ACM TOSEM) and in a conference (ICSE), highlighting the common trends as well as the peculiarities of each venue. Finally the paper recommender system has been tested with the organizing committee of two conferences (ICSM 2010 and ICSE 2011).

Estratto in Italiano

L'evoluzione e la crescita delle comunità scientifiche hanno reso necessaria la creazione di alcuni di strumenti che aiutino i ricercatori ad avere una conoscenza migliore e più completa del loro campo di ricerca. In questo lavoro di tesi sono stati studiati e sviluppati una serie di strumenti che permettano di effettuare meta-ricerca in modo automatico o semi-automatico, proponendo delle metodologie e implementando una serie di strumenti in grado di portare a termine questo tipo di operazioni.
In particolare, sono stati sviluppati: uno strumento per identificare i temi di ricerca trattati da una comunità scientifica, uno in grado di determinare quale sia l'argomento di un articolo di ricerca e infine una tecnica per conoscere quali sono gli interessi di un ricercatore. Questi risultati sono stati ottenuti sfruttando le tecniche di analisi automatica dei testi che rendono possibile l'estrazione di informazioni a partire dalla letteratura scientifica in un processo basato sui dati. L'identificazione dei temi di ricerca e dell'argomento di un articolo si basano sui dati contenuti nella letteratura scientifica, mentre i profili dei ricercatori sono costruiti considerando gli articoli che hanno scritto in precedenza.
L'identificazione dei temi di ricerca è compiuta mediante l'uso di tecniche di *clustering* che permettono di raggruppare i documenti che presentano delle caratteristiche simili. Il processo sviluppato prevede l'esecuzione in

diverse iterazioni, tramite le quali un esperto del dominio applicativo è in grado di verificare la bontà dei risultati ottenuti. Nel caso questi non siano soddisfacenti, l'esperto può migliorarli effettuando una nuova iterazione con diversi parametri e con un insieme di dati in ingresso più esteso. Quando l'esperto ritiene che i risultati siano corretti, può procedere all'assegnazione di un nome significativo a ciascun gruppo di articoli, ottenendo così un modello per l'ambito di ricerca. Questa operazione è effettuata analizzando le parole più frequenti in ciascun insieme individuato dall'algoritmo.
Gli articoli contenuti nei gruppi così ottenuti e rappresentanti i diversi temi di ricerca individuati, sono utilizzati nella fase di apprendimento di un *classificatore* con il quale è possibile determinare l'argomento di un nuovo articolo. La decisione di utilizzare il modello dei temi di ricerca come sorgente di dati per il classificatore permette di velocizzare enormemente il suo sviluppo, evitando di dover selezionare e classificare appositamente un altro vasto insieme articoli.
La generazione del profilo di un ricercatore è basata sull'assunzione che sia possibile conoscere i suoi interessi sulla base di un'analisi degli articoli scritti nel passato. L'algoritmo sviluppato in questo lavoro cerca di individuare i temi di ricerca più rilevanti per ciascun ricercatore, utilizzando delle tecniche per evitare che siano erroneamente riportati degli argomenti che non fanno più parte degli interessi del ricercatore. Il risultato di questa operazione è, per ciascun ricercatore, un insieme di argomenti di ricerca ordinati in base alla loro rilevanza.

Gli strumenti di analisi implementati sono poi stati utilizzati nella realizzazione di due diverse applicazioni. La prima è dedicata allo studio dell'evoluzione dei temi di ricerca, identificandoli e poi osservando come si

sono distribuiti nel corso degli anni nelle più importanti riviste e conferenze dell'ambito di ricerca. La seconda applicazione è un sistema in grado di aiutare i membri del comitato organizzativo di una conferenza nella selezione degli articoli da revisionare, suggerendo quelli che meglio si adattano al profilo di ciascun ricercatore.
L'analisi dell'evoluzione dei temi di ricerca permette di trarre delle considerazioni specifiche sulla pubblicazione in esame e, unendo i risultati ottenuti dall'analisi di diverse riviste e conferenze, generali sull'ambito di ricerca. Questo strumento può fornire delle preziose indicazioni su quali sono gli argomenti più rilevanti, mostrando anche quando hanno iniziato a manifestarsi. Informazioni di questo tipo permettono di mettere in relazione dal punto di vista temporale le variazioni negli interessi di ricerca con eventi tecnologici o con la diffusione di nuove idee.
Lo strumento per classificare gli articoli sottomessi ad una conferenza, invece, può permettere un notevole risparmio di tempo ai membri del comitato organizzativo che sono messi nella condizione di concentrarsi maggiormente sugli articoli che potenzialmente potrebbero trovare interessanti. Inoltre, le metodologie sviluppate in questa tesi possono essere molto utili anche nella fase di assegnamento vera e propria degli articoli da revisionare: la suddivisione delle sottomissioni in base al tema e una indicazione degli interessi dei diversi membri del comitato organizzativo possono essere delle informazioni molto preziose. Con questi dati l'assegnamento può essere effettuato concentrandosi su un tema alla volta, rendendo tutto il processo più veloce e meno dispersivo.

Le metodologie e i risultati ottenuti dalle loro implementazioni in degli strumenti di carattere generale, sono state verificate applicandole al campo dell'ingegneria del software in modo da poter sfruttare la conoscenza

dell'ambito per un'analisi critica dei risultati ottenuti. Dopo aver generato il modello dei temi di ricerca, la loro evoluzione è stata studiata in delle pubblicazioni di carattere generale, ovvero nelle due riviste maggiore del settore (IEEE TSE e ACM TOSEM) e nella principale conferenza internazionale (ICSE). Durante questo tipo di analisi sono state evidenziante sia le caratteristiche comuni a tutto il campo di ricerca sia quelle specifiche di ciascuna pubblicazione che, come era lecito aspettarsi, hanno mostrato le loro peculiarità. I risultati ottenuti hanno mostrato una buona aderenza delle tendenze riscontrate con quelle già note nel campo dell'ingegneria del software, confermando la validità dell'approccio seguito.
Infine, lo strumento per il suggerimento degli articoli che potrebbero interessare ad un ricercatore è stato utilizzato con i comitati organizzativi di due conferenze, ICSM 2010 e ICSE 2011. L'esperimento sulla prima conferenza ha permesso di mettere a punto l'algoritmo confrontando gli interessi dichiarati dai membri del comitato organizzativo con quanto proposto dallo strumento. La seconda verifica, invece, ha coinvolto direttamente i membri del comitato organizzativo chiedendo loro una valutazione sulla qualità dei suggerimenti proposti dallo strumento. Gli articoli suggeriti si sono rivelati per buona parte dei casi pertinenti con gli interessi dei ricercatori che, in generale, hanno ritenuto che uno strumento simile possa essere utile all'interno di una comunità scientifica.

Chapter 1

Introduction

Research activity is not a static process but it is rather based on the interaction among different people and it is in constant evolution to get a deeper knowledge of the research subject and to build new theories to better explain the studied phenomena.
Considering the number of researchers involved and the quantity of possible research topics, it is clear that a research community is very complex. That complexity originates a lot of different dynamics that lead the researchers to focus on a particular research topic rather than on another, depending on what they have already researched on in the past and on the interactions among the different research groups.
It is also clear that the researchers should know what is happening within their community in order to be able to direct their efforts in the most effective way. For this reason there is the need to perform meta-research analysis to point out the most relevant aspects that can help the researchers to better understand their community and to improve the quality of their work.

A bibliometric research has to discuss the following matters: *what* the

subject of research it and *how* it is evolving in the years, *who* the people involved are and *where* they are dislocated.

1.1 Problem statement and research context

The analysis of a research community has to take into account several different aspects to provide a complete overview of the community that is performing research on a particular subject.

1.1.1 Analysis of a scientific community

Meta-research works have to analyze what the topics of the studied research field are, how they are evolving and which are the interactions among the different research groups.
A first interesting information about a research community is the snapshot of how the researchers' efforts are subdivided among the different topics at a certain year. This information becomes even more useful when it is used to track the evolution of the research subjects by comparing the data taken for different periods of time. With this data it is possible to find out the trending topics and how the interest on some topics changed over the time.
The research activity is deeply related to the interaction among different research groups, so it is useful to analyze the interaction and the collaboration among different researchers. For example it can be interesting to find out which are the topics that have been studied by each research group, what kind of research is developed by universities and what by industrial research and development departments. To get this sort of conclusions there is the need to perform some analysis on the social network of

the co-authors.
The union of the two approaches presented, enables even more sophisticated analysis. For instance it allows to track the spread of the new ideas, showing how influential papers affects the work of the whole community.

This work is focused on the creation of tools to perform the first kind of analysis. For an introduction to the social network analysis techniques that can be useful in the analysis of scientific communities refer to [1] that studied the social network of the software engineering community.

1.1.2 Tools to support research

Meta-research can also provide something more than a valuable analytical outcome: all the tools used in the study of the community can be used to produce application specifically designed support researchers, providing helpful information in the organization of their work.
This kind of tools may range from a recommender system that suggests to the researcher to read a just published paper on a relevant topic to something more sophisticated that helps the organizing committee in the management of a conference.
With this work an automatic bidding tool has been developed to help the members of the organizing committee in the selection of the papers to review that best match with their expertise.

1.2 Motivations for automatic analysis

Scientific communities produce a lot of material in the form of papers published on journals or presented at conferences. This means that there is plenty of data available to base the meta-research on but also that it is

not feasible for a single researcher or a small research group to go through all the available material in order to get some conclusions about what is happening within the community.
Moreover even assuming to have the work-force to deal with all the available material, this process can be quite boring and it would result in the waste of the energy of highly qualified researchers that would have to sacrifice their own research interests to meta-research.

The goal of this work is to provide a general framework that helps researcher understand better their community with an easy to use, fast and scalable tool that exploits text-mining and information retrieval techniques. The creation of an automated general framework also shifts the focus from the topic-related aspects to a more methodological approach to meta research in order to build a tool that can fit the needs of every scientific community and not only the one of the meta-researcher.
The automation of these tasks makes it possible to limit just to the training phase the requirement of the supervision of an expert of the field. All the analysis is performed automatically without any human intervention. This makes also possible to scale the number of publication considered: if a human could study just few of them a machine is able to analyze and compare all the journals and the conferences of the field.

The idea of the development of tools to support this kind of analysis is derived from the Ghezzi's keynote speech at International Conference on Software Engineering (ICSE) 2009 [2] that gave an overview of the evolution of the software engineering community from the beginning up to now. During the collection of the data for that talk it became clear the need of a set of automated tools to overcome the limitation of manual analysis.

1.3 Outline of this dissertation

Chapter 2 describes the state of the art about meta-research analysis and about the topics that has been used to automate it. Chapter 3 explains in the details the methodologies proposed with this work. Chapter 4 describes of how those methodologies have been implemented. Chapter 5 analyzes and evaluates the experimental results. Finally chapter 6 presents the conclusions of this work and explains further possible evolutions.

Chapter 2

State of the Art

In the past a lot of work has been done in the analysis of scientific communities. This chapter contains a review of the most significant meta-research studies, highlighting the different aspects taken into account and the different methodologies adopted. It is possible to identify two different kind of approaches in the reviewed works. On the one hand, there are works that just talk about the results of their analysis. On the other hand, the outcome is a set of methodologies and tools that could be really useful in systematic literature review. The great part of the approaches described in this chapter bases their conclusions on the analysis of the available literature, using data from bibliometric analysis as a guide for their work.
In addition to the overview of the trends in bibliometric studies and meta-research, this chapter also contains an introduction to the well-known techniques that have been used as the basis on which to build other useful tools.

2.1 Related meta-research works

A research community can be analyzed from a lot of different perspectives, each one taking into account a particular aspect of the problem. There are a number of different approaches, each one focusing on a specific feature of the research community.
This section covers the most significant meta-research work and reviews the results obtained with the different approaches. The different perspective of the research on a community includes: the analysis of the research topics and how they evolve, the study of how researchers work, how they interact and how new ideas spread within the community. Moreover there are studies that just aims to take a snapshot of a research community at the analysis is performed, while other focuses on the development of a framework that could be used also to forecast future trends or to extract more general considerations.

2.1.1 Research topics analysis

One of the first aspects that have to be pointed out in the study of a research community is a clear definition of which are the topics the researchers are working on. This is a very fundamental step and a more detailed knowledge of the research subject is the basis for other more sophisticated analyses. For instance it allows to track the interests of the research community, showing the evolution of the topics investigated by the researchers.

The works in this area aims to create a taxonomy of the research area, and a lot of very different techniques to face this problem have been developed. The methodologies proposed range from an analysis performed by

some expert supervisor with a deep knowledge of the field to data-driven approaches that tries to extract the knowledge from the published works. An example of the first approach is the Computing Classification System developed by the Association for Computing Machinery (ACM) [3]. This kind of approaches on the one hand offers the reliability that comes from the knowledge of experts that are into the research field. On the other hand it is quite limited due to the fact that the source of knowledge is in the experts. That makes this kind of methodologies not so flexible, not replicable in other research fields and it is likely that after a while the details of the taxonomy do no longer reflects the actual research topics in a complete manner. Data-driven methodologies are less prone to this kind of issues since it is definitely easier and rather inexpensive to perform again the identification of the research topics for the research area. For this reason a lot of techniques based on the data contained in the literature have been developed. In [4] the taxonomy is generated by analyzing the correlation among the occurrences of keywords in the academic literature. That work is based on the assumption that tightly coupled research areas shares a lot of common and highly coupled keywords while areas with a little in common also uses different words and therefore there is a low correlation. Applying these criterions it is possible to find out which are the research topics that are gathering the interest of the researchers and it becomes clear which are the mainstream matters and the niche subjects studied only by small groups.

The information contained in the scientific literature can be exploited not only by looking at the content of the single papers. In [5] the research area is studied by looking at the references among the papers and analyzing the citation. The idea behind this work is that it is possible to identify

group of papers that share the same topic by looking at the citation graph. The papers with a similar topic will be more connected than papers about different topics. The same principles are also exploited in [6] that also introduces the idea to look for research topics that somehow shares some interest by analyzing the citation among paper of different topics. This way it is possible to find which topics are strictly related and which other only share a little research interest. In [7] the performances of the citation graph based topic identification are compared with other content based techniques. That work found out that the results obtained are equivalent: papers that are declared to be similar by one technique are also considered similar by the other. Citation analysis has also been used in [8] that integrated the clusters representing the different research topics in a retrieval system for scientific publications.
In [9] researchers also tried to exploit natural language processing techniques in order to obtain a generative model able to describe the features that are contained in a paper of a given topic. The information contained in the model can be used to determine the theme of a paper by looking at the features it presents and by finding out the topic that is more likely to have generated them.

Another approach tries to identify the research topics from the information about the collaboration patters and looking at the authors of the papers and their affiliations. This kind of research, as described in [10], is more suitable to perform analysis on the scientific community rather than for the identification of the research topics. Moreover this approach is good for a research field level analysis while it have some issues when it has to deal with the common case of the authors that research on more than a single topic. A similar approach has been adopted also in [11],

where the analysis of the collaboration patterns has been combined with other techniques in order to perform an analysis of the evolution of the research topics and to forecast the deriving technological changes.

Once it is clear which the topics for the research area are, it is possible to go deeper in the studies trying to find the most relevant ones. It is also useful to know how the research topics evolved in the years. This could make clear which topics are gaining interest, which are stable and finally which are mature and no more so appealing to the researchers.
The research described in [12] specifically targets the software engineering field. It relies on the ACM taxonomy and the papers from several venues are classified manually by looking at what their authors declared. From this data the researchers analyzed the statistical distribution of research topics showing which are the leading topics for each of the venues analyzed. The data has then been aggregated in order to extract information that is venue-independent but related to the research area in general.
A similar work has been performed in [13], where there is an attempt to identify the most relevant topics by looking at the number of researcher involved in that subject. This metric is quite different from the previous but it could provide some information about the effort spent by the community on each topic.

2.1.2 Authors and scientific productivity

Bibliometric studies can be performed from different perspective and at several levels. While the methodologies exposed in the previous section are concerned with a general knowledge of the research field, this part describes the techniques developed to have a better knowledge of the researchers and their productivity. There is a number of works that try to

identify the laws and the models that regulates scientific productivity.

One of the first studies on this topic [14] dealt with the analysis of the distribution of the published papers among the authors. It turned out that papers are not evenly distributed but there are a lot of authors with a really limited number of publications and very few researchers that published a lot of works: the highest the number of the published papers the lowest the number of authors that reached that level. This conclusion is described by the Lotka's law that states that the number of researchers who published exactly n papers is about $1/n^2$ of the authors who managed to publish exactly one paper. From this study it is clear that scientific communities are highly fragmented, but there are a very limited number of members that account for a high fraction of the work.

Productivity has not been analyzed only from the Lotka's perspective, but there are also a number of alternative studies that proposes models to represent the way the researchers get to publish a work. For example [15] proposes a minimal model to describe how creative processes take places. In that work the number of papers published within a year is represented as a dynamic system and the evolution of the creative outcome is studies with respect to different set of parameters that models the author and the environment he lives in.

In order to reflect the emerging trend of the creation of teams rather than the past conception of research as a solo work, in the last year the studies about the productivity concentrated more on the effect that grouping together had on the scientific results. In [16] some evidence that show how research is dominated by team-works is reported, in [17] the collaboration among different departments and different universities is studied pointing out how this affect the impact and the relevance of a research. Finally

in [18] the composition of successful team work is studies in order to find out what makes a good group.

Given these premises it comes out that is important to find out the leading researchers and teams, since from them comes out a relevant amount of the whole research work. In [19] the researchers used the data from the literature and text-mining techniques to discover the most prolific authors and teams for each topical area.

There are also some standard metrics to measure scientific productivity. One commonly used is the *Hirsch index* [20] or h-index. That index takes into account the distribution of the citations received by a given researcher's publications and is computed in the following way: a scholar with an index of h has published h papers each of which has been cited by others at least h times.

2.1.3 Social network analysis

The analysis of the scientific productivity focused on the analysis of the amount of work produced by each author and the effects of the creation of research groups in the quantity and the quality of scientific production. The dynamics underneath the aggregation of authors and the evolution of teams can be studied in detail with an analysis of the social network of the community. These kind of studies are really useful to understand the dynamics of the collaboration among the authors and to find out how team works spread within the community.

The study of social networks has been an active research field in social sciences for a long time. Sociologists used social networks as a tool to represent and study the interaction that happens within the members of a community; however the theoretical work proved to be effective also in

the description of other phenomena that are not strictly related to social activities. The concepts and the related properties [21, 22] apply to a really wide range of different contexts: of course they work for the description of social networks, but their scope is much broader as they apply also to power grids, airline time tables, food chain, World-Wide Web and Erdos number, just to name the most known examples.

The formalization of social networks has been first presented in [23]. From that models emerged the well-known *small world* and *six-degrees of separation* phenomenon. The first means that the network presents the same properties regardless of the scale it is observed: the properties that hold for the whole network are also true for a part of it. In [24] this property has been proved to fit also scientific communities, although they are a particular case of social network. In that study each researcher has been represented by a node of the network and the edges modeled the co-author relationship. The result of this study is really valuable since it allows the researchers to use to analyze a scientific community all the work done in the other contexts that have been modeled with social networks.

The second phenomenon is related to the distribution of the members of the network. It states that even for really huge networks there is a very short path that connects two distinct members, even if they apparently don't have anything in common.

The study of the co-authorship network can highlight several patterns that are present within the community. For instance it is possible to know how the relationships among the researchers evolved through the years, how they are teaming up together and also which are the most relevant groups. In [25] the database research community has been studied starting from the co-authorship information retrieved on the Digital Bibliography and

Library Project (DBLP). While that work result is a snapshot of the current situation of the community it is also possible to extract information about the evolution of the relationships. That has been done in [26] where an analytic approach has been presented. In that work, after the study of the properties of the network, the evolution has been studied through simulation to point out how the network could rearrange itself as the time goes on. Other examples of works that takes into account the collaboration patterns are the already mentioned [10] and [27]. The latter focuses on a single research topic and analyzes how the research on that matter is growing and evolving.

Co-authorship is not the only valuable source of information that could be used in the realization of a social network. Another relationship that has always been there in the scientific literature is the citation of a work by another. In such a network, the nodes still represent authors and the links represent the fact that a paper on one researcher cited the work done by the other. While co-authorship is more useful for the identification of the teams, the citation analysis leads to the discovery of relevant works and of the way the ideas spread.
In [28] some mining techniques that could be used to extract some information from the citation graph are presented. That work also discusses the mathematical properties of the citation graph.

Research impact analysis

Citation analysis it is also useful to determine the impact of a work, assuming that the more a work is relevant the more it is cited by others. Moreover it is common that works that opens a new research field get cited for a long time, making it possible to study also how new ideas spread in

the community. The techniques described in [28] that aims to characterize and mine the citation graph come in hand to perform this task.
Citation analysis has always been considered an important tool to determine the impact of a work or the relevance of a journal [29].
Other techniques have been developed to analyze how the ideas spread within a social network. The theme is quite general and has been studied in a general fashion in [30]. The proposed approach applies also to the spread of innovation [31], and it is a good model to describe how ideas become popular on online social networks [32].

2.1.4 Geographical analysis

In bibliometric studies it is important not only to know who is studying one particular topic but it is also useful to discover where the most relevant research centers are located. For instance in [13] the focus is on the geographical origin of the contribution to research. The authors of the paper analyzed the interests of several research groups and later they aggregated the data in order to find out the correlations among the topic and the geographical location.
While geographical information could be relevant, they are not always easily available. The work described in [17] shows that the collaboration is crossing the geographical borders, making it quite hard to assign a work to one specific location. The same work however showed that usually the collaboration happens within a limited region, so it is still possible to perform some macro-area geographical considerations.
Geographical studies could also be focused on a single nation (As happens in [33]) but prove to be more useful when the performances of the countries of a region are compared [34].

Some other research of this kind also discriminates about the nature of the institution (Academic or industrial) that proposed the scientific contribution. The combination of the two sources of information could be really interesting showing for every place whether research is lead by universities or by industries.

2.1.5 Forecasting future trends

All the data gathered in the analysis of a scientific community could be used for more than a snapshot of the current situation. For instance it could be used to get some more information about what will happen in the future. Common question that the bibliometric studies tries to answer are about the topics that will gain more interests, the papers that will become relevant, the location that will be the most important research centers and so on. Of course the answer to these questions cannot be sure, but it is possible to get some hints that could however be useful. While it is not possible to predict sudden revolutionary events, these techniques could be effective in the forecast of the changes that happen smoothly as an evolution of the current situation.
The methodologies are especially applied in the forecast of the new technologies. In [35] data from bibliometrics and new patents in combination with modes of dynamic systems are used to forecast when new technologies will become available. In [36] text mining techniques were used to forecast from the analysis of the literature the new technologies that could emerge in a very specific field.
Generally speaking, the forecast methodologies are more oriented to the prediction about the availability of new technologies. However, there are also studies [5] that dealt with the forecasting of the future trends of re-

search topics.

2.1.6 Digital libraries and public databases

Bibliometrics relies on the public availability of some information about the papers published in journals or at conferences. In order to be able to create an effective process, it is fundamental to have that information organized and easily available. To meet the need of the researchers to easily access the published papers, the associations that manage the journal and the conferences created the digital libraries: web-sites that make it possible to access to all the information about the publication, to search the archives of a journal and to see the links among the different papers.
Such information is really valuable, but each association has his own digital library and there is not any common interface to access easily the data contained in all of them. For instance in the computer science field, the two main libraries are IEEExplore [37] managed by the Institute of Electrical and Electronics Engineers (IEEE) and the ACM Portal [38]. In both of them it is possible to access a lot of information but the lack of a common interface makes it tricky to integrate the data from the different sources. Since each library contains only the complete information about the journals and the conferences sponsored by the association that owns the library there is the need to access several libraries in order to get a complete dataset. This is a big drawback, especially for who wants to base his analysis on the data that comes from the bibliography: the lack of integration makes it not as easy as it should be the work to collect all the data needed for the studies.

Digital Bibliography and Library Project

To overcome the interoperability problem of the digital libraries and to allow the researchers to access to more information and to query the system in a more flexible way the DBLP [39] project has been started by Micheal Lay of the Universität Trier. Its goal is to collect in a single place all the information about the papers published in computer science, providing a lot of details and the links to the respective digital libraries to ensure the availability of the most complete information possible. Moreover the DBLP provides an approach that is less centered into journal and conferences and provides facilities to query the system for information about the authors, making it possible to perform the majority of the data driven analysis: it allows to gather the papers for a given research area as well as to create social networks using citations and co-authorship to define the links among the authors.

2.2 Useful techniques

While the previous section gave an overview of the different approaches to meta-research, this one describes the well-known techniques that could be used in the realization of tools to automatically analyze a research community.

2.2.1 Information Retrieval

The goal of this work is to propose a tool that performs automatic analysis and extract information from the literature produced by a scientific community. Since this task requires the analysis of a large set of

papers it is important to know the techniques to deal with a huge amount of textual document developed by the researchers in the information retrieval field. In fact, information retrieval [40, 41] is the sub-field of computer science that studies effective ways to get from a set of documents the subset that best fits a query representing the informative needs of the user. Although the techniques developed to perform information retrieval tasks are mostly used in the realization of search engines, there are some data structures and some procedures that are really useful to represent, store and manage a huge number of textual documents. In particular really come in hand the techniques used to reduce the dimensionality of the problems on textual documents. These techniques include: the removal of common words, stemming and the measurement of the similarity between documents.

The removal of the most words that are used in a particular domain is useful to reduce the dimensionality of the problem without affecting the quality of the results. On the opposite it improves the results, making the algorithms concentrating just on the words that are somehow useful to discriminate the meaning of the document. There are set of words that are known to be common in a given language, however to improve the quality of the results it is possible to expand that list with some more domain-specific words [42].

Another technique that can be used to reduce the number of words that the algorithms have to take into account is stemming. The goal of that technique is to concentrate on the meaning of the words rather than on the particular form that it is assuming. This process reduces all the words to their stem, grouping together nouns, adjectives and verbs that come from the same root. While this procedure affects the structure of the sen-

tences, making it impossible to distinguish the parts of speech, from the perspective of analysis that works on term-frequency analysis it is a good thing since it groups in one term all the words that derive from it and thus share the same meaning. There are several stemming algorithms; the one that is recognized to work better for the English language is Porter's [43].
Once the dimensionality of the problem has been reduced with stop-lists and stemming, it is possible to compute the degree of similarity among documents. The basic idea is that documents that share the same words are talking about the same topic. The more they have in common the more the two documents are similar. A data representation that allows a definition of similarity among the documents is the *vector space model* [44]: to each document is associated a vector that counts the frequency of each word. A measure of the distance between a couple of documents could be the cosine of the angle generated by the vector representation of the two texts.

2.2.2 Text-mining

Once there is a representation of the textual data that could be efficiently managed by the algorithms it is time to find methods that could be used to extract some information about what the documents are talking about. To perform this task it is possible to use text-mining algorithms which are specifically designed to extract information from a set of textual documents. Text-mining is the sub-field of data-mining that studies how to apply algorithms to the analysis of textual documents. These techniques are the core of a number of bibliometric studies [45] since they allow to speed up the analysis of the data with automated tools.
In general every machine-learning technique, and thus the text-mining al-

gorithms, presents two different kind of approaches: in the first the patterns are completely extracted from the data, in the second all the learning process is directed by an expert supervisor that transfers his knowledge to the machine. In text-mining an example of the first class of algorithm is the *clustering*, that groups together similar documents by looking only at their features. On the other side there are *classification* algorithms that learn the pattern from a set of labeled documents and then are able to classify unseen instances.

Clustering

Clustering could be considered the most relevant unsupervised learning problem. The algorithms developed take as input a set of textual documents and returns groups, each one containing similar documents. There are several clustering techniques: some algorithms try to partition the data by choosing some elements and then iteratively associating the other instances to the group represented by them and changing the element that represents the cluster. Some other algorithms take a different approach and create a hierarchy of clusters until all the dataset is covered in a satisfying way. Hierarchical clustering is the more suitable for textual data and could be performed both in a top-down and in a bottom-up fashion. In the first case the algorithm starts subdividing the dataset in some big clusters and then goes on dividing the sub-clusters until they are meaningful. On the opposite the bottom-up approach at the beginning considers all the documents in the set as they were different and then starts creating the clusters by aggregating together the more similar.

Classification

Classification is the typical supervised learning problem: the algorithm is first trained with a set of instances already labeled by an expert supervisor and then it has to learn from that and generalize it classifying unseen items. There are a lot of different classification algorithms available. Some of them, like the *Bayesian classifier*, are based on statistical properties. Other, like *support vector machines*, rely on particular representation of the data that leads to a new definition of the problem that have to be solved. In classification problem it is crucial to have a training set as complete as possible in order to avoid errors due to the lack of examples of some features that could lead to the misclassification of an item.

2.2.3 Natural language processing

The text-mining approach is not the only way machines could learn how to correctly guess what a document is talking about. Studies on natural language processing come up with some other techniques that are closer to the way humans interpret the meaning of a sequence of words. Rather than on some statistical property they are based on the idea that it is possible to identify a generative model that represents they way the document is structured and how it has been created. An example of the techniques that follows this approach is the *Latent Dirichlet Allocation* [46] that tries to identify the grammar that could have generated a text of a given topic. When it is time to determine which is the topic of a paper, the algorithm looks for the grammar that most likely would have generated the content of the document.

2.2.4 Recommender Systems

For the purposes of this work it is also useful to review the different kind of recommender systems that could be useful in the development of a set of tools to support researchers. A recommender system [47, 48] is a piece of software that gives advices and recommendations about items that can be worth considering for the user, based on what he previously shown interest in. The development of recommender systems had a huge acceleration with the diffusion of e-commerce and on-line shops that to increase the number of items sold wanted a system able to suggest to the costumers items they may find interesting and so that they can purchase. Of course this is not the only field of application and it is possible to create a recommender system for almost every kind of items, included scientific publications.
The best implementation of a recommender system depends on the data available; however it is common to find two different kind of approaches: one based on the comparison of the content of the items and another based on the comparison of the profile of the costumers.

Content based

A content based recommender system keeps track of the item that each user find interesting and generates suggestions looking for a new product that is somehow similar to the one previously purchased. In order to perform this kind of recommendation there is the need of a suitable distance function to discover which are the best matches. These methods tends to suggest only items somehow similar to what the user has already shown interest in. That could be an issue since it does not take into account the possibility that the user could be interested in different kind of items. As

long as he does not reveal the interest in the new category of objects the system is not going to suggest him items of that kind.
An example of a content based recommender system specifically targeted to scientific publications is reported in [49]. In that work the papers previously published by a researcher are used to determine his own interests. This information together with the citation graph of the papers is used to suggest to a researcher the papers that he could be probably interested in.

Profile based

Sometimes there are systems with a lot of data about user preferences available. In such cases it is possible to base recommendations on this information. When a user asks for a suggestion first the system looks for other users with similar profiles. The items suggested are the one that can be found in the profiles of the users with similar tastes but that are missing in the profile of the user who asked for the recommendation. While in general this approach works well, it has some problems dealing with items just inserted in the database: unless an object enters in the profile of some users it won't be recommended.

Hybrid approach

There are also other approaches that try to combine these two different methodologies relying partly on the content of the items and partly on the profile of the users. These approaches aim to use all the information available to make the best recommendation possible, reducing the drawbacks of the two methods. For instance it could recommend new items as happens in the content based approach, but at the same time it could improve the variety with suggestions in a profile based fashion.

Chapter 3

Methodologies

This chapter contains the description of the motivation that led to realization of this work as well as an introduction to the problems faced and to the solutions adopted to solve them.

3.1 Motivations

Meta-research and bibliometric studies are really useful to help researcher to better understand their research community, how it is evolving and who is interested in each research topic.
This kind of information can be really useful also to discover the topics that are becoming relevant or to make the search for someone sharing the same research interests easier. The huge advantage given by this study is the amount of time that the researcher can save by just looking at the results on their research field. Researchers can also perform themselves the studies with the use of automatic tools that with a really little effort provide them all the result they need.
The time saved by the researcher can lead to better results in their research

since they can focus just on their own topics rather than having to spend a lot of time in the exploration of what is happening in the other parts of their research field.

3.1.1 Research communities

Nowadays research communities are extremely complex, each research area presents a lot of sub-topics and each of them has several specific journals and conferences. This kind of structure leads to a very dynamic environment that is in constant growth and evolution: research topics changes over the years, researchers interests may vary and also the research groups are everything but static.
Given the nature of the research communities, it is clear that some sort of meta-research is needed in order to give to the researcher something that may help them not getting lost in such complexity.
It is particularly useful to point out the topics that characterize a research area and how the interest in each of them is evolving. Since research is made by researchers it is also really useful to discover who is doing what, creating a profile for each member of the community. The latter kind of information is particularly useful for whoever wants to find other authors with similar interests and could lead to collaboration between different groups that are studying the same topics.
The same analysis can also be performed on journals and conferences, helping the researchers to discover the most suitable venues for the submission of their works.

There is a lot of information available regarding research communities. Most of them are publicly available and could be the base for a data-driven analysis. All the data collected in the digital libraries and in database such

as the DBLP is really valuable sources to discover the desired information from the research community.

3.1.2 Goal of this work

This work aims to create a set of tools to automatically perform the analysis of a scientific community. In order to be able to do that there is the need of the development of solutions to:

- discover the topics in a research area;
- get the topic of a published document;
- create the profile of a researcher.

The discovery of the topics in a given research area is a key factor in this work since it creates the framework for the other steps of the analysis. Without a clear understanding on the themes the researchers are working on the analysis of topic evolution and the profiles of the researchers are not so significant.
The techniques to retrieve the topic of a paper are fundamental for the development of automatic tools that are based on the data available in the digital libraries: that is the only way to make it possible the analysis of a relevant number of publications.
Once the solutions to these two problems are well defined, it is possible to go on with the analysis finding out how research topics are evolving and to create the profiles of the researchers.

The outcomes of this work are both a framework that automates bibliometric studies and a tool that helps the organizing committee of a conference in the assignment of the papers. The latter it is an example of how re-

search community can directly take advantage of the techniques described in this work even in everyday tasks.

3.2 Identification of the research topics

The identification of the topics of a particular research area is a key factor in meta-research studies: it is the first step that leads to the creation of a taxonomy of the research area and enables the other, more sophisticated, steps of the analysis.
Nowadays it is common to find a subdivision of research in several subjects, each of them organized in several areas. This is somehow useful, but there is the need of a more detailed subdivision that takes in account the topics that are actually studied by the researchers. For instances the classical taxonomy of research presents a subdivision in general subjects (Science, physics, chemistry and so on) and for each subject there are some research areas (Like software engineering, databases, hardware...). While this subdivision is good for some high level considerations it is definitely not enough to perform bibliometric studies: rather than a researcher generally interested in software engineering it is more likely to find someone interested in testing, web-services, requirements engineering or in another specific topic. For this reason it is more meaningful to have a subdivision that takes into account also specific topics like the one displayed in figure 3.1. Such subdivision better reflects the actual research interests. There are some approaches that tried to create a more fine-grained taxonomy. For instance the ACM requires to specify some keywords for the papers published in its journals and in its conferences. While this is appreciable it is not enough because it is limited to a subset of the literature. Moreover

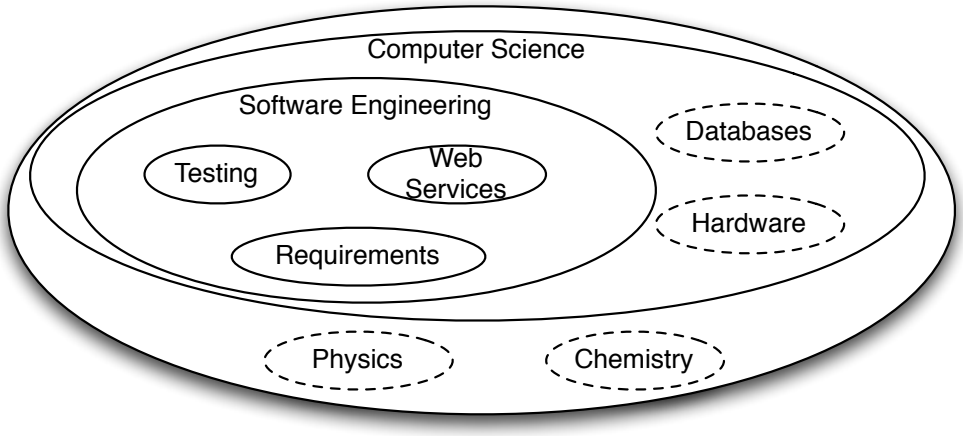

Figure 3.1: The subdivision of research in subjects, research areas and topics

this kind of classification relies on the self-judgment of the authors that sometime may not be too accurate because of the lack of a common set of keywords.

To overcome this issues this work proposes a data driven approach that uses data mining techniques to extract the research topics from the literature.

3.2.1 Design choices

There are several variables that have to be defined in the design of a tool for the identification of the topics of a research area. The first thing that has to be decided is what the analysis will be based on. Since this process requires the interaction with an expert supervisor, it has to be defined which are his duties and how he can influence the results produced by the system. Finally there is the need to properly define which data that will be used to solve the problem.

Data-driven topic identification

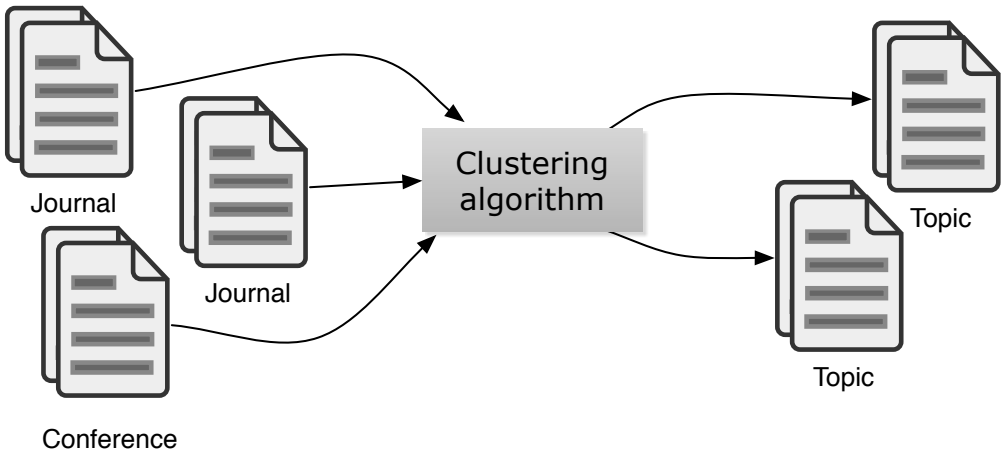

Figure 3.2: The data-driven topic identification process

The fist design choice that has to be taken is about the approach to use for the identification. For instance it would be possible to have the needed information with the analysis of the existing literature or on the opposite by asking it to an expert supervisor that has a deep knowledge of the research field.

Since all the research work are published in a journal or at a conference, the literature is for sure the most complete source of data about a research community one can think of. Although it is a really valuable source of information it does not provide directly the information about the topics of a research area. An expert of the field may be able to provide the needed information but it is really likely that its suggestions are biased by his own interests. Moreover in order to ensure the possibility to extend these methodologies to other research areas it is important to rely as little as possible on the contribution of an expert supervisor.

For this reason the methodology developed in this work proposes the use of some data-mining techniques to unveil the hidden patterns in the literature. The intervention of an expert supervisor is still required, but its role it is limited to the evaluation of the results and to find out the meaning of the subdivision of the papers computed by the algorithm.
The solution adopted in this work uses clustering algorithms (See section 2.2.2) to discover the patterns hidden in a collection of published documents. The algorithm generates a subdivision of the papers in groups representing a research topic. It is based on the assumption that documents that share the same topic also present common distribution of the keywords.
Once the clusters are generated, an expert supervisor has to associate to each group its meaning. This is done by naming the topic that each set of papers represents.

Before thinking at the data-driven approach, the topic subdivision was just asked to an expert supervisor. This turned out in a set of topics that was not really satisfying for the scope of this work since it was not providing a complete coverage of the research area. The reasons of the lack of some topics are due to the fact that a supervisor, even though with a really deep knowledge of a research area, it is somehow biased towards its own interests. It is really likely that the expert would tend to overestimate the relevance of some topic while underestimating some other. This leads to the risk of an incomplete coverage of the research area for the approaches that relies only in the judgment by a single expert. This problem could be solved with the supervision of several experts, but that would definitely decrease the chances of being able to produce a general framework. Finally an expert supervisor would not be able to propose as much examples of

instances for each topic as a data driven approach. He could suggest some specific conferences for each topic, but doing that all the information contained by the more general conference is lost unless the supervisor spends really a lot of time labeling each paper from the non-specialized venues.
These considerations make it clear that the adoption of a solution that avoids these kind of issues is definitely better since it removes or at least tries to limit as much as possible the bias introduced by the personal experiences of the expert supervisors producing at the same time a richer model.
Moreover, the usage of automatic analysis tools shrinks the time required to process the information from a lot of venues. That makes it possible to base the analysis on a huge number of journal and conferences that could only affect positively the outcomes of this process.

Data used in the clustering process

The choice to develop a data-driven solution introduces the need to decide how to create the dataset that is used for the identification of the topics. The definition of the dataset requires to choose the data-source and the nature of the content that is passed to the algorithm.
There are a lot of journals and conferences available in the literature. Some of them publish papers only of a specific topic, other are more general and presents papers about different subjects. The data-source should be based on the less specialized publications to be able to find examples of all the topics. To avoid issues related to the uneven space dedicated to each topic it is important to consider a significant number of publications. Finally, specific journals and conferences may help to provide a good number of instances for the topics that are under-represented.

Talking about the nature of the content, it is possible to use the full-text of a document rather than some other significant parts like the title and the abstract. The full text of the document conveys a lot of more information rather than the title and the abstract that summarize the content in few sentences. For the purposes of this work the effect of the summarization it is really useful because it has the consequence of removing all the implementation details leaving the general meaning of the paper. Moreover the language used in the title and abstract it is more likely to be in common with the other publications of the same research topic. The full-text would introduce a lot of noise that could be harmful for the clustering process: the content of the entire document is filled with a lot of details that are useful for the complete understanding of the ideas that the paper wants to convey but that are not so relevant for an analysis that just wants to discover the general topic of the document.

Role of the expert supervisor

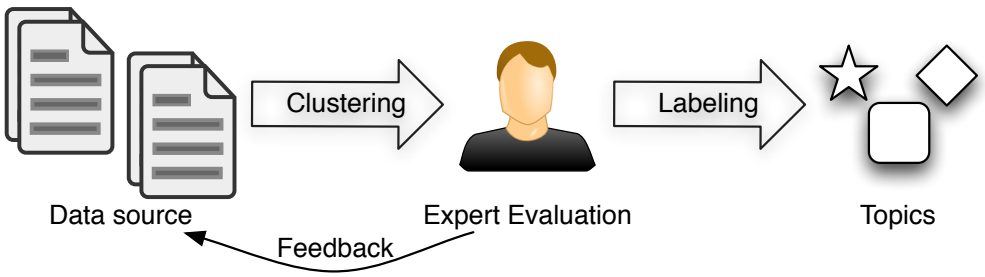

Figure 3.3: The iterative process and the role of the expert supervisor

Section 3.2.1 explains that the role of the expert supervisor it is to evaluate the results of the data-driven analysis and to find out the meaning of the topics identified by the clustering algorithm. This choice has a lot of

advantages, since it allows the development of a general tool that can be used to find out the topic of a given research area.
Although the process does not rely entirely on the expert supervisor, he is still the main actor of the identification process. He does not have to use his experience to name the topics but he still has to direct the algorithm in the proper way and to use his expertise to give a meaning to the clusters. With such approach the expert supervisor becomes a part of the system and has to give a feedback to the iterations of the clustering algorithm (See figure 3.3).

3.3 Getting the topic of a paper

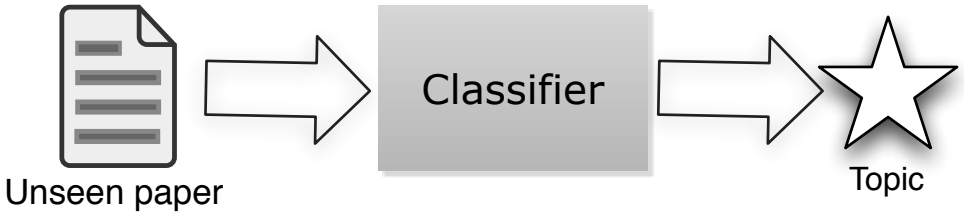

Figure 3.4: Classification of the topic of an unseen paper

Once the topics for the research area have been defined there is the need to produce another automatic analysis tool that has to be able to determine which is the topic a paper is talking about. That kind of information it is fundamental to be able to perform studies that have to deal automatically with the papers published in a huge number of journal and conferences.
This part of the work is based on text-mining techniques that determine the topic of an unseen paper by looking at the keywords that it contains.

It is important to point out that these techniques do not really know anything about the semantics of the content. It is based on the idea that documents with similar terms share the same keyword distribution.

3.3.1 Design choices

The development of a classifier has to take into account two main issues: the kind of the algorithm and the training dataset. The algorithm chosen should be the one that best fits the requirements of the problem. The dataset has to provide a complete set of instances in order to make the classifier effective in the recognition of all the patters.

Kind of classifier

The research on data-mining and text-mining developed a number of different families of classifiers, each one presenting different features that makes it more suitable for an application rather than for another. For this work we are looking for an algorithm that is able to tell which is the class of a paper from the one found in the literature.
This immediately points out a feature that the classifier has to have: it must be able to perform *multi-class* classification. Although it is possible to get it also from binary classifier combining the results of multiple execution of the algorithm it is nicer to have it natively supported by the adopted classifier: this turns out in a significant reduction of the amount of time required by the classification since the results are immediately available.
Another element considered in the choice of the family of classifiers to use is the smoothness of the results produced. There are some classifiers that tend to return a really edgy classification with one single topic prevailing by far on the other. On the other side there are other classifiers that pro-

duce more smooth and fuzzy results pointing out more than a single topic for each document.
Although the smoothness is somehow desirable, for instance in a paper that talk about testing of web-services could be classified both as a *testing* paper and as a *web-services* paper, the improvement in the quality of the results does not justify the increased complexity. Moreover it is likely that a paper that lies across different topics usually analyzes them from a specific perspective. That consideration makes the introduction of the smooth classifier not so useful for the aggregate analysis performed in this work.
The classifier chosen for this work is the *naïve bayesian classifier* that natively supports multi-label classification and that provides a sharp distinction between the selected class and the others. The other approaches considered in the choice of the classifier were not meeting the desired requirements. *Support vector machines* do not directly support multiple label classification and the *vector space model* representation tends to produce smooth results.

Training dataset

The dataset used to train the classifier has to be as complete as possible in order to provide a sufficient number of examples to enable the algorithm to extract the actual patterns present in the data. This means that there is the need to have a dataset that completely covers the research area and that has a significant number of instances for each topic.
The results of the process that lead to the identification of the research topics described in section 3.2 could be a good training set. The clustering process groups the papers into some set, each one representing a topic that could be used as shown in figure 3.5. The possibility to reuse the sets iden-

tified with the clustering is a great advantage since it makes available for free a complete set of instances that presents the information about their topic without requiring an expert to manually label them.

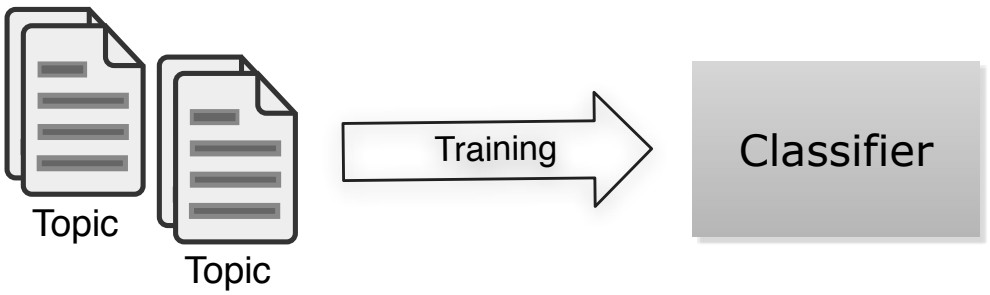

Figure 3.5: Training of the classifier from the sets of papers that represent each topic

Nature of the data

In order to be able to reuse the results of the clustering process, the classification has to rely on the same kind of data: the title and the abstracts of the papers. However this does not make the classification less effective with respect to the analysis of the full-text: all the consideration that led to the choice to consider just the title and the abstract also applies for the classification process.

3.4 Profiling the authors

Getting to know the research topics it is just a part of the work that has to be performed in order to have a better knowledge of what is going on. It is important not to forget that the studies are carried out by the researchers, therefore for a deeper understanding of the state of the research

community it is really useful to find out which are the research interests of each member.
The most effective way to achieve this sort of information is again to extract it from the data available. The DBLP database makes it accessible to everyone a quite complete list of publications for almost every member of the whole computer science community. From this data, using the techniques described in section 3.3, it is possible to get the topic of each publication. The profile is generated from the aggregation of the retrieved information as shown in figure 3.6.

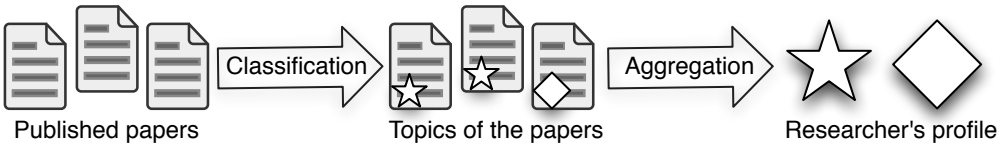

Figure 3.6: The creation of researcher's profile

3.4.1 Design choices

The idea behind the profile generation appears to be straightforward, but there are some issues that have to be faced in order to reach the best accuracy of the results. Of course there is the need to find out an appropriate way to aggregate the results. Moreover research interests are not static but it is really common to find, especially in authors that have been in the community for a long time, a change of the research topics.

Black box versus understandable model

The first thing that has to be considered is the kind of the model the algorithm has to output. The main alternatives are some black box mod-

els that captures the statistical properties of the papers published by each author or a more sophisticated and understandable representation of his interests. The first option would be simpler to implement: it would just require to input all the paper an author wrote to an algorithm and get some statistical property from that. This model could later be used to evaluate the degree of similarity of a new paper with respect to what the researcher previously shown interest in.
While this approach seems straightforward, there is a big drawback on this methodology: it is not possible to control what the algorithm is doing from a high-level perspective. In some applications this is not an issue. For instance it could work to look papers similar to what the researcher published in the past, but to face the problem of the creation of the profile of an author it is better to have information of the highest level possible. That would on the one hand make it easier to check whether the algorithm is performing well with a quick look at some understandable information rather than having to analyze obscure statistical metrics. On the other hand an understandable profile could be useful by itself, telling which are the interests of each researcher. A black box model could hardly be understood and therefore it could probably be useful just to find papers similar to what the researcher previously wrote but could hardly come in hand telling in a clear way which are the topics an author is interested in.

Interests aggregation

The aggregation of the results has to be able to capture the main topics in which the researcher spent his efforts. This means that it is not enough to just check which topics the author wrote about and which he didn't: the aggregation should provide also information about the effort spent in each

topic. This means that the system should be able to recognize the main interests and separate them from other topics that the author worked in just a few times.
In order to obtain such results the aggregation function computes the relative frequency of each topic in the publications of the researcher. That frequency could be used to rank the interests and it is also useful to manage the side topics and the outliers: setting a threshold it is possible to remove from the profile the topics that do not have a significant number of examples and that are likely to be classification errors or the result of collaborations with other researchers.

Interests evolution

In order to design a system that is able to deal with the evolution of the research interest, there is the need to introduce a technique to take into account just the papers that are relevant for the creation of a profile reporting only the current interests.
The naïve approach is to limit the number of papers taken into account to a fixed number of years. Although this may appear the simplest solution at a first glance, it is not the way to go since it is not trivial to determine the number of years to take into account to obtain the best results. That parameter is highly influenced by the number of papers produced by each author and it is highly coupled with the age and the quantity of publications.
A better approach is the weighting of the contribution of the papers. This approach requires to define just once a function that returns a decreasing weight as the date of publication of the paper goes into the past. Papers published in the last year will be considered with their full contribution,

older papers will have a lower impact on the definition of the profile. This approach is able to preserve the completeness of the results since all the data available is taken into account. At the same time it is also able to produce results that reflect the current interests by privileging the most recent publications.

3.5 Outcomes of this work

This section describes how the studied methodologies could be used in the development of tools for the automatic analysis of a scientific community. This work has two different outcomes: a tool to analyze the evolution of the research topics and a system to help the researchers in the assignment of the papers submitted to a conference to the best reviewer according to his interests.

3.5.1 Research topic evolution

The direct outcome of the methodologies described in the previous section is the realization of a tool to automatically analyze the research topic evolution. The application is able to analyze either a single journal or conference and to extract general information about the research field by combining the results obtained for the single venues.
While the analysis of a single venue it is important to discover which are the journal and conferences that publish papers related to a given topic, the combined analysis allows the meta-researcher to get some conclusion about the big picture of the evolution of the research area. The combined analysis is more significant from a meta-research perspective since the conclusions that come from it have a general meaning and could refer to the

entire history of the community. On the other side the analysis of a single venue could make it clear which journal and conference is particularly interested on a research topic.
Like other bibliometric study there is the need of a huge amount of data in order to get statistically significant results. While this is not an issue for the global analysis and for the study of the most relevant venues, it could be a big drawback for journals and conferences that have short history and a limited number of papers published in each edition. For this reason the analysis focuses more on the general aspects rather than on the detailed study of a single venue.

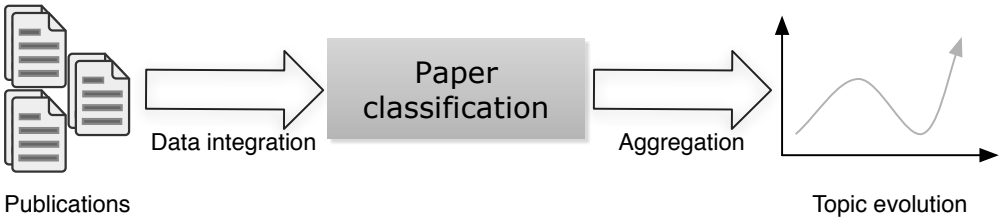

Figure 3.7: The process of research topic evolution analysis

Design choices

The methodology developed to analyze the research topic evolution is deeply related with the solution proposed to automatically find the research subjects and with the techniques to infer what the paper is talking about by looking at its content.
The building blocks for this analysis are the identified research topics and the classifier derived from that subdivision of the research area. Once these elements are defined, there is the need to go on with the development of the analysis framework and to choose the venues that have to be included

in the study. To complete the tool there is the need of a crawler to obtain the data about the publications from the digital libraries. The retrieved information has to be uniformed and stored in database after a data integration process. Finally there is the need of a tool to graphically display the results obtained in order to make the outcome of the process more user friendly and clear at a first glance.
Once all the information are available and integrated together the results should be presented to an expert supervisor that could interpret the data by combining his own knowledge of the research area and the statistical evidences.

The venues considered in the study should meet both qualitative and quantitative requirements. From the qualitative point of view there is the need to choose the venues with a high impact factor, in order to ensure the scientific relevance of the published documents and the freshness of the research themes taken into account. Venues with a high impact factor are more likely to publish works that are really significant and that are exploring the new trends of the research area. This is a really important property when it comes to analyze the evolution of the research topics since it makes it possible to have an idea of which are the hot topics in a given period of time.
On the other side the quantitative requirements have to be respected in order to have a large set of instance that is able to produce statistically meaningful results. The publications should cover the entire time span considered and for each time interval there should be a relevant number of available papers. The requirement of a huge amount of data is one of the strictest requirements of the data driven approach, but luckily in scientific literature there are a number of publications that are able to satisfy them

providing both a good coverage among the years and a number of issues each one with quite a lot of articles every year.

3.5.2 Automatic bidding

The methodologies developed could also be effective in the realization of a series of tools to help researchers in the organization of their community. In this thesis an automatic bidding tool has been developed as a case study to show how the results of this work could have a practical impact in researchers' life.
The goal of the tool is to help the selection of the best reviewer for the papers submitted to a conference.

Bidding

In research communities *peer review* is used to ensure the relevance and the quality of the published works. Every paper has to be positively reviewed by other researchers before it is accepted for the publication in a journal or at a conference. Of course, it is not possible to have all the papers reviewed by all the members of the organizing committee, it would require a lot of time and moreover it is not possible for a reviewer to have a deep knowledge of all the subtopics of the research area. For this reason each paper is assigned to few reviewers rather than having it judged by the entire committee.
It is clear that the choice of the reviewers is fundamental to be able to reach the desired results: the reviewer must know deeply the subject of the research, so it is really important to have each paper assigned to the members of the organizing committee that have the best knowledge of the subject of the work.

The assignment of the papers for review is performed in two steps. At first the members of the organizing committee are asked to make their bids, expressing their interest in the review of a subset of the submitted papers.

The bid of a member of the organizing committee is composed by a set of papers that he finds interesting, a set of papers the he may be interested in and finally the list of the papers that he does not want to review because of conflicts or lack of interest in the topic. A conflict happens when a reviewer may be biased in his judgment because of he is somehow involved with the authors of the paper. The process is called bidding since each committee member makes an offer about the papers he would like to review.

Later the program chair actually assigns the papers to the reviewers taking into account their bids and his knowledge of the researchers. When the assignment is done, the organizing committee reviews the papers, expressing its opinion on it. The judgment could be the acceptance or the rejection of the paper.

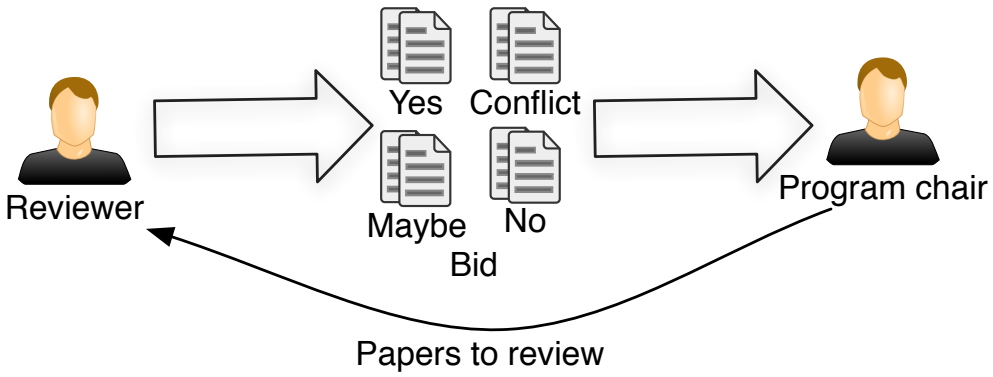

Figure 3.8: The bidding process and the assignment of the papers to review

From the description of the bidding process and the assignment of the submitted papers to each reviewer, it is clear that these operations requires a lot of time to the members of the committee and to the organizers. Each committee member has to read at least the titles and the abstracts of each submitted paper in order to find out whether he could be interested in it. The organizers not only have to read each papers but they have to know the interests of every member of the committee in order to be able to make the best reviewing assignment possible. In international conferences this turns out in a lot of time spent in the bidding process and in the assignment of the papers to the reviewers.

Advantages for the reviewers

Reviewers have to put a lot of effort in the bidding process: they are asked to go through all the submissions and to point out for each of them whether it is interesting or if there is any sort of conflict that makes it better to have the paper review by something else. An automatic bidding tool could save them a lot of time by suggesting the papers that best match with their own interests. The suggestion could dramatically reduce the number of the papers the reviewer has to care about, making it possible a quicker but also more accurate bidding. This is the direct results of the fact that the reviewer could focus on a subset of the submitted papers, concentrating in the one that are about topics he is most probably interested in. Of course the recommendation isn't mandatory and the reviewer would still be able to bid a paper that the tool didn't suggested. However if he first checks the recommended papers and then skims the rest of the submissions, the bidding process could benefit from the suggestion becoming a lot more efficient without any qualitative loss due to the fact that a paper has not

been taken into account.

Advantages for the program chair

Although the automatic bidding tool is mainly for the reviewers, also the program chair could benefit from it when he has to assign each submitted paper to the members of the organizing committee for the review exploiting the information about the profiles of the other researcher and the topic of the papers. The detailed profile the tool is able to extract could be useful to double check the knowledge of the program chair about the interests of each reviewer and to have this information also for the members of the organizing committee that are not well known.
On the other side, the tool to infer the topic of the papers could be useful to have quickly an idea of what the paper is about. This information could be used to partition the submitted papers into some sub-sets accordingly to their topics.
Both these information are particularly useful: they allow the program chair to focus on one topic at time by knowing in advance which are the papers and the reviewers that are researching each subject.

How it works

The bid recommendation is computed in several steps: the system first generates the profile for each reviewer, then it groups the submitted papers accordingly to their topic and finally it matches the interests of the researcher with the content of the papers.
The generation of the profile is performed as described in the section 3.4. With the profiling it is possible to know which are the topics that each reviewer is interested in with a completely data-driven approach that does

not require them to fill in a form to state the subjects they would like to review.
The submitted papers are grouped using a classification system that determines their topic. This step is also performed automatically, without the need of the manual classification by the authors or by the conference organizers.
The recommendation is obtained by looking at the topics in the reviewer profiles and suggesting the submitted papers of the subjects the scholar is more interested in. In order to avoid the suggestion of too many papers there is a filtering stage that tries to recommend only the papers that talk about issues similar to what the reviewer studied in his previous works.

With these steps the system is able to recommend for the bid the subset of the submitted papers that is more likely to match the interest the reviewer. Moreover it ensures that the judgment comes from a scholar that deeply knows the subject of the paper, making the peer review mechanism really effective.

Possible drawbacks

The bidding process is an activity that does not always take into account just the research history of the reviewer, and this is a direct consequence of the choice to use a data-driven approach. Sometimes it happens that a researcher is asked to review a paper for other reason that the fact that he wrote something on the same topic. The retrieval of that kind of information that usually exists only in the mind of the members of the organizing committee it is indeed impossible; therefore the only objective data available to determine who may be interested in the review of a paper is the collection of the articles published by each author. Moreover this

is not a really big issue since the tool presented has not been thought as a replacement for the human judgment. It is rather a tool that could provide helpful information and therefore speed up the process that leads to the assignment of the submitted papers to the most suitable reviewers.

The evaluation of the performance of the recommender system is another major issue, since it is not easy to define a function that tells how good the suggested bid is. The comparison with an actual bid could be not so effective since in a real bid there is no uniformity and reviewers may adopt their own policy to choose the papers. On the other side, surveys are a better way to assess the perception of the reviewer. This could be a better measure of the opinion of the reviewers, but there is the chance that they get biased by what the tool suggests them.

Chapter 4

Implementation

This chapter describes the details of the implementation of the methodologies presented in this work to automate bibliometric analysis. After a brief overview of the whole workflow and a presentation of the tools required at each phase of the process, there is an explanation of how each module has been realized.

4.1 Workflow

The methodologies described could be implemented using a set of tools and connecting them in order to create a complete workflow that is able to automatically produce the results of the meta-research analysis. The system needs to have tools to collect the data from the resources available on-line, then it has to consume that documents and create a model of the scientific community. Finally it has to use the model created to train the tool that determines the topic of a paper.
Once the classifier is set up, there is again the need to collect the data that will be used in the analysis and to feed with the retrieved documents the

tool in charge of the computation of the results. Depending on the results that the system has to obtain there is the need to plug the model in the appropriate tool. Currently there are available a tool that studies the evolution of a research field and a recommender system to help the organizing committee in the choice of the best reviewer for each submitted paper. However the system is designed in a modular way and the single elements developed could be used as building blocks for other analysis tools or could even be integrated in other systems.

Moreover each single component has been designed to be extensible and

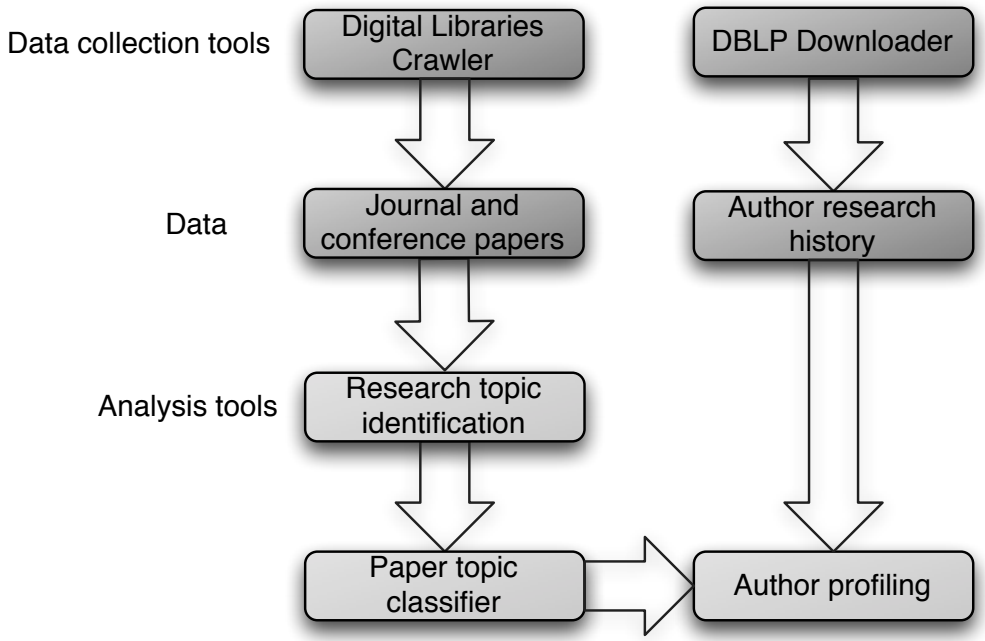

Figure 4.1: The tools developed in this work

to easily allow the addition of new algorithms and data sources. Figure 4.1 shows the tools that have been developed, highlighting the modules related to the data collection and storage as well as the ones devoted to

the analysis. Those tools have been used in the development of two higher level applications. Figure 4.2 shows which are the components used in the development of a tool to analyze the trends in a research area. Figure 4.3 shows the modules that takes place in the realization of a bidding recommender system.

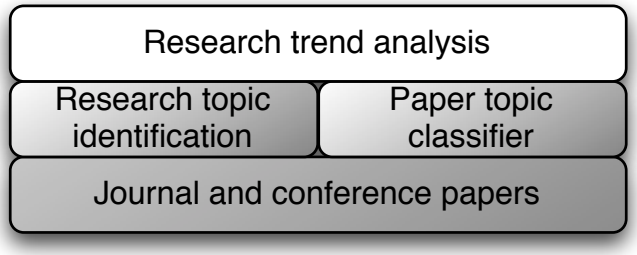

Figure 4.2: The tools used for the research trends analysis

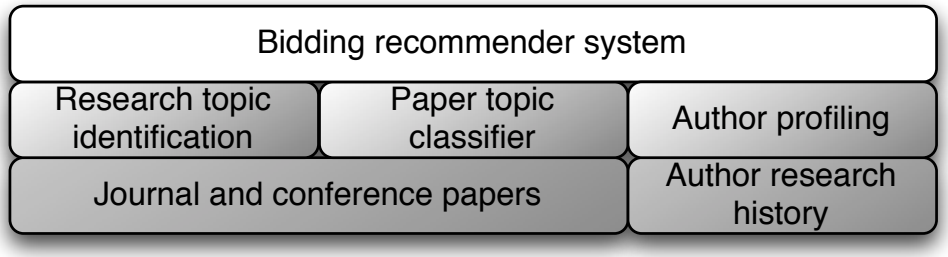

Figure 4.3: The tools used in the bidding recommender system

4.2 Data collection

Being this work data-driven, there is the need to collect the data from the literature and to store it in a common way that is understandable by

the other modules. To accomplish this task there is the need to design a common data structure and to develop a crawler that is able to retrieve from the digital libraries the information about the paper published in a journal or at a conference. In addition to this, the need to have the information about all the authors leads to the development of another crawler that queries DBLP to obtain the desired information.

4.2.1 Digital libraries crawlers

The first part of the analysis required to collect the papers published in several journal and conferences in order to build the dataset used to identify the research topics. This crawler is also useful to retrieve the information about the papers needed to find out the trends in a research field.

The need to get the data from several digital libraries lead to the design of a common structure that has the duty to perform the operations related to the networking and data transfer on the top of which it is possible to plug the parsers that actually get the information. Since the digital libraries make it possible to access to a web interface but does not provide any other Application Programming Interface (API), the crawler has to be able to interact directly with the web pages and to parse their content to retrieve the information. Luckily all the digital libraries share a common structure: there is a page with the list of all the issues of a publication, then there is an index for each issue and finally it is possible to access to the details of a single article. The common structure made it possible to design a general system that deals with the connection to the web server and inserts the data in the database while the modules that target a specific digital library just have to provide the facilities to parse the different web

pages as shown in figure 4.4.

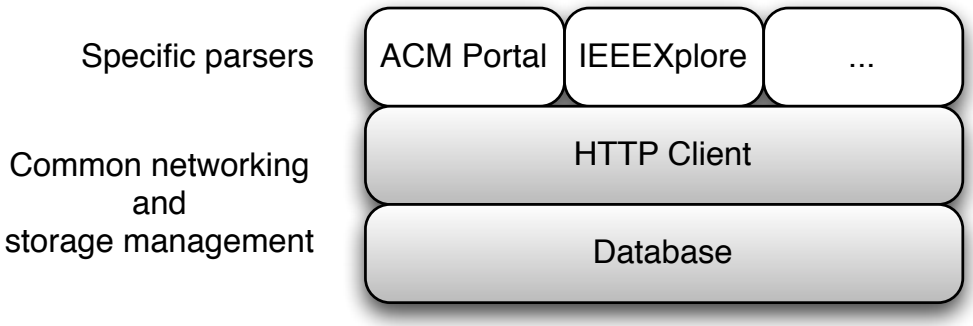

Figure 4.4: The modular architecture of the crawler

The common layer has to take care of the Hyper Text Transfer Protocol (HTTP) connection to the web server, it has to log in in order to have access to the full information, to manage the session and the interaction with the parsers in charge of the data collection for that website. Finally when the data has been collected by a library-specific implementation it has to be persisted into a database.

The network management module relies on the *Apache HttpClient* [50], a library that easily enables the management of a connection and that is able to keep track of HTTP sessions. The functionalities provided by the library have been wrapped into an abstract class that provided the stub of the implementation of a library-specific crawler. The generic class provides the facilities to interact with a generic digital library as shown in figure 4.5. It first obtains the list of the issues of a publication, by accessing to the index page and parsing its content. At this point the crawler knows the addresses of the pages containing the details of each issue. It is now able to query the web server to obtain the list of the list of the papers published

in an issue. Parsing that list the crawler finally gets the links to the pages containing the details of each article. Now the crawler can access to each article page, download the content, interpret it and finally store it into the database.

Library-specific implementation

The description of the workflow that leads to the retrieval of the information of a paper makes it clear that there is the need of a specific implementation to deal with each digital library. In particular there is the need to understand how it is possible to access the index page of each publication, parse the pages containing the list of the issues and the list of the articles in order to find the links to the page reporting the details and finally to get the information from the page of an article.
The first step requires the collaboration of the user that has to manually search in the library for the code of the publication and then pass it as a parameter to the crawler. The other parts are completely automated with the development of a parser for each different kind of page. Luckily the structure of the pages is quite simple and presents Hyper Text Markup Language (HTML) tags, so it is possible to find the content with several regular expressions. The regular expressions that identify the location of the pages containing the detailed information basically have to look for the links that presents a given pattern and find out the identifier of the new items. The page that contains the content is trickier to parse since the content is less structured. However it is possible to discover where the desired information are by looking for some keyword or for the HTML tags and to their style attributes.

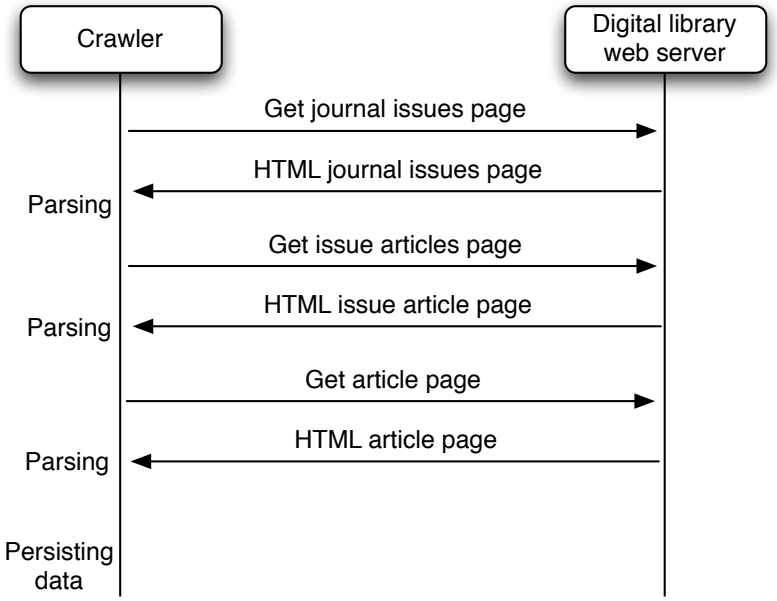

Figure 4.5: The interaction between the crawler and the web server of a digital library

4.2.2 Common data structure

In order to have a clean way to retrieve the data once the crawler downloaded it, all the information is stored in a database. Of course there are some little differences in the information stored in each digital library, so it is not possible to take it as it is but there is the need to perform an integration step that makes the content uniform. The integration is performed directly by the crawler that looks just for the common information and then adds them to the database once all the relevant data has been collected. This required the design of the database that would consider all the possible relevant data. After an analysis of the domain of the problem it has been decided to keep track for each article of its title, abstract, full-text (When available), list of authors, venue that published it and year of publication. That information is stored in a relational database follow-

ing the schema represented in figure 4.6. In order to make it easier the management of the data at the applicative level, all the database related activities are managed by *Hibernate* [51]. With that technology it is possible to have an object-oriented database, making it smoother the transition from the relational world of the Database Management System (DBMS) to the object-oriented application logic. The object/relational mapping is performed automatically by the framework that takes care of creating a relational representation in the database of the classes used by the application. Note that the mapping preserves the meaning of the relationships and could change a bit the data structure. For instance in this case the relationship between the articles and the authors that were modeled with a list of authors within the article class is mapped with a bridge table able to represent its many-to-many cardinality.

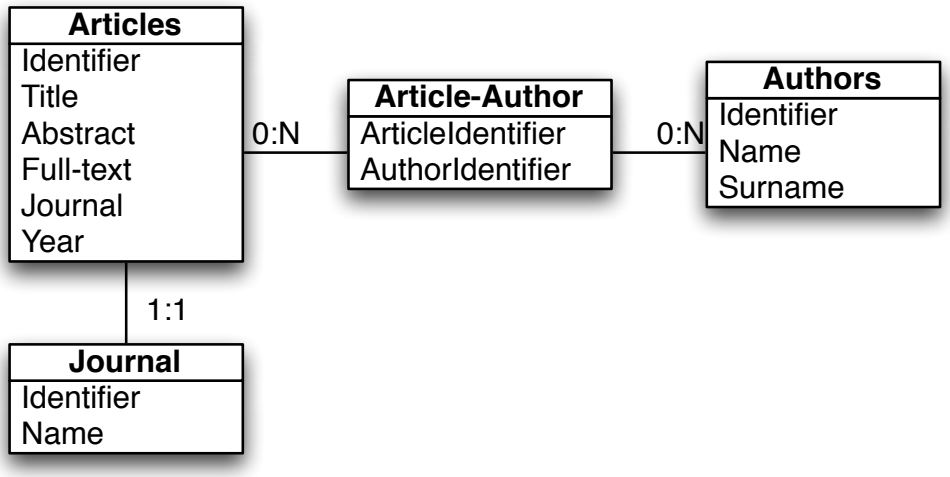

Figure 4.6: The entity-relationship diagram of the article database

4.2.3 DBLP Downloader

Although in the database there are stored the information about the authors of each document, there is the need to develop another tool that guarantees to find all the papers published by each researcher. The digital library crawlers are targeted to the download of all the papers published in a venues, therefore unless all the venues are stored in the database it is not possible to ensure the completeness of the results of a query about the publications of each researcher. For this reason, in addition to the crawlers for the digital libraries it has been developed a tool that retrieves all the papers published by a given author based on the data available on DBLP. Also in this case the tools have a modular structure: there is a main component that reads the list of the papers published by an author from the DBLP web pages. On top of it there are other modules, each one targeted to a specific digital library. DBLP Downloader and the digital library crawlers are designed for different purposes and the difference is reflected also in the output they produce. While the crawler stores the data in a database, the DBLP Downloader keeps just the content of papers and places it in a file named with the title of the paper. It does not keep information about the venue a paper has been published or about the co-authors since that information is not used in the profiling methodology presented in this work and therefore that extra information would be dropped later.

4.3 Data preprocessing

All the data used by the analytic modules is obtained by invoking the crawler and having it to download all the information about the papers from the selected venues or written by a specific author. Once all the data

is collected and stored in the database, there is still need to transform it in order to be able to apply all the analysis algorithm. At this phase take place two major kinds on transformation: one devoted to reduce the dimensionality of the text-mining problem and another that translates the documents in a format that the algorithms can manage.

The dimensionality reduction is required by the nature of textual documents: usually in a paper there are thousands of different words, but only a fraction of them is relevant to determine the meaning. It is clear that there is the need to drop as much as possible the irrelevant words keeping just the ones that actually bring their contribution to the identification of the meaning of the document. That operation is performed in two operations: the removal of the too common words and the unification of the words that come from the same stem.

The first operation aims to remove all the words such as articles, prepositions, pronouns that, although are useful to understand the meaning of the document when a human being reads it, does not have any statistically relevant property that could be exploited by automatic analysis tools based on the study of the term distribution. In addition to the common list of words that can be dropped in English language, it is possible to consider some more domain specific terms that, while in a general context would carry some significant information, won't be too useful. For instance, the common list does not include the *software* word, but narrowing the scope of the analysis to documents about computer science that term becomes useless since almost every piece of writing would probably contain it. The use of the enriched list could practically improve the performance of the analysis tool both from the time and space requirements and from the quality of the results. Time and space consumption is obvi-

ously reduced since there is the need to process a much limited amount of data. On the other side, also the quality of the results improves because the dropped terms carry only irrelevant information and noise rather than a significant contribution that could make the operation more precise.
The other operation that could be relevant in the dimensionality reduction is stemming: that operation makes it possible to recognize the terms that comes from a common lexical stem and therefore have a similar meaning. Take for example the words *document*, *documents*, *documentation* and *documenting*. That simple list of words makes it clear that, while each one is declined in a different form accordingly with their function within the sentence, it would be nonsense to consider each word by itself without taking into account that all of them are actually strictly correlated. To discover that kind of correlation there is the need of procedures that reduces each words to its stem. That operation relies on the well-known Porter's stemming algorithm that is considered to be the most reliable algorithm for the English language. With such an algorithm it is possible to dramatically reduce the dimensionality of the problem by compressing all the terms that comes from the same stem into a single entry. Moreover it improves also the quality of the outcome, making clear to the algorithms the fact that the meaning of these words is strictly correlated.
Once these steps have been performed it is possible to transform the document into a format that is understandable by machine-learning algorithms: the approaches used in this work relies on the *vector space model*, where for each document it is built a vector that keeps track of the number of time each term appears in it. This representation is then used by the algorithms to learn the statistical property of the term distribution that characterize each kind of document. An example of the preprocessing is

reported in figure 4.7.

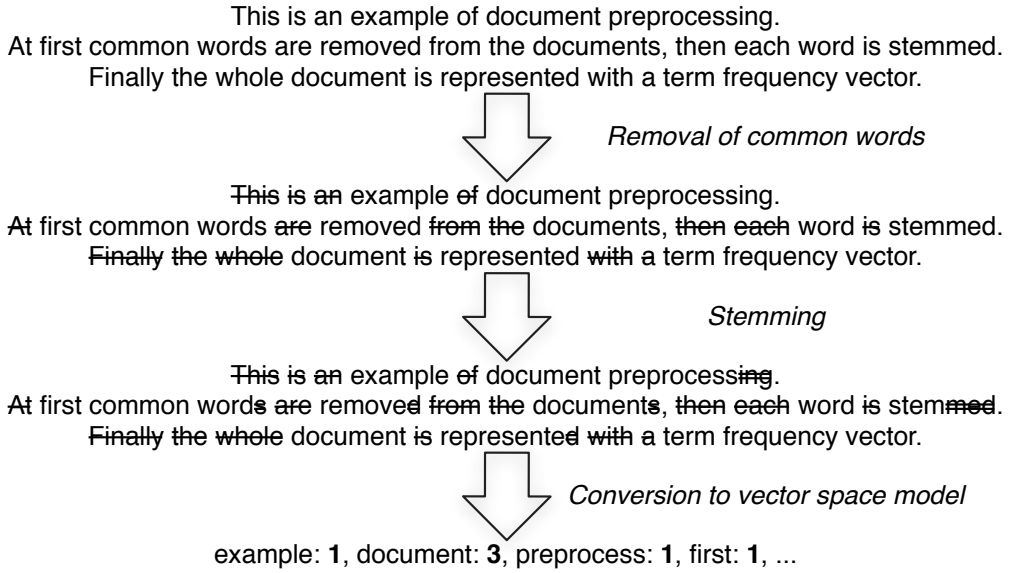

Figure 4.7: The preprocessing steps performed to make a document ready for the analysis

Sometimes another pre-processing step is required depending on the information available on the digital libraries: when the information available does not contain explicitly the abstract there is the need to get it from the full-text. That means that the Portable Document Format (PDF) document have to be converted and, using some heuristics, the abstract has to be extracted from it. Luckily the majority of the digital libraries makes it publicly available the abstracts of their documents, and when that is not true it is possible to look for the common structure shared by almost every scientific paper. The conversion from a PDF document to string representing its content was made possible by *Apache PDFBox* [52], a library that brings to the developers all the facilities needed to deal with PDF files.

4.4 Research topic identification

The first analysis module that needed to be developed is the tool to obtain the topics that are studied within a given research field from the documents published in the scientific literature. Such a module takes as input the papers from several journals and conferences and groups them accordingly to their topic. The methodology presented in this work requires an iterative approach that takes advantage of the feedback of an expert supervisor in order to ensure meaningful results. The outcome of this process is a model of the research area that shows which are its topics and for each of them which are the most relevant keywords. Automatic procedures can produce a subdivision of the papers received as an input but cannot name the topic of each sub-set produced. That taks has to be performed by an expert supervisor who has to associate the correct label to each set. To accomplish this task and assign to each set the name of the topic it is representing, the domain expert could take advantage of the list of the most common keywords contained in each set.

The research identification process takes as input the data from several publications. The data is downloaded by the crawler, but before it is fed to the text-mining tool each document has to be placed into a file containing its title and abstract. In this way it is possible to integrate the database created by the crawler with the data structure required by the third-party text-mining tool.

The data-driven methodology requires an accurate choice of the documents that are going to be part of the set of papers used for the analysis: there is the need to ensure that all the research topics have some represents on the dataset, making the algorithms able to find out all the relevant themes. For this reason the data source used are papers published

both in general and in specific journals and conferences. This allows the algorithm to take in account the differences in style that comes from different venues and also limits the chances to have some minor topic not considered. The choice to include also some general conference makes it possible to find out also some evidence of the presence of minor topics that otherwise would not appear, unless the person who built the dataset already considered it by adding a specific publication on that particular subject into the list of the venues used as data source.

The text-mining technique used in this work relies on the Hierarchical Expectation Maximization (HEM) algorithm [53], often used to perform model-based clustering operations. The algorithm is able to generate a hierarchy organizing the elements in the dataset in a tree, that in the domain of this work could be assimilated to the taxonomy of the research field. An example of the results obtainable from the algorithm is shown in figure 4.8 where there is an initial dataset split in clusters with two iterations of the expectation maximization algorithm. The first iteration operates on the whole dataset; the second is applied to the results of the previous execution of the algorithm. This technique allows to refine the clusters as more iterations take place.

The HEM algorithm is a statistical methodology to find the parameters that presents maximized the likelihood with respect to the input dataset. In the hierarchical variant there is the application of several execution of an expectation maximization algorithm that could be seen as a generalization of k-means to have it work in a probabilistic setting. Such an algorithm consists in the combination of two different phases: the *expectation step* and the *maximization step* that together give the name to the algorithm. The goal of the algorithm is to find the parameters that best describe the

model given a set of observed values and the presence of missing or unobserved values. At first the initial values of the model parameters are guessed in some way or they are chosen randomly. Then the algorithm starts iterating the two steps until it converges to a stable state.
In the within the expectation step computes which are the values that the unobserved items have to assume in order to best fit the model given the current values of the parameters. It is called expectation step because it returns which are the expected values for the unseen items.
Once that has been computed, the maximization step takes place, and the input elements as well as the just computed values for the unseen items are used to obtain a better estimate for the parameters of the model. This phase is called maximization since the value of the parameters is obtained by maximizing the likelihood of the data with respect to the model.

The implementation of the HEM algorithm used comes from *crossbow*, a front-end for the well-known data-mining tool *bow* [54]. Crossbow is specifically designed to use the capabilities of bow in order to perform text-mining tasks. By itself the HEM algorithm takes as input parameter just the desired branching factor for the tree it has to create. The crossbow implementation allows the specification of more parameters, such as the maximum depth that the algorithm has to reach in the clustering process. The result of the clustering operation is a list of the clusters found for each level of depth in the tree. In addition to that, the crossbow implementation also returns other useful data like the ten most relevant words in each cluster.

The output of the HEM algorithm still needs to be checked and interpreted by a domain expert that has to try to name the topic of each cluster. He has to do that by having a look at the most relevant words in each

cluster and at the documents that are part of each set. It is important to notice that the expert supervisor may merge together some of the leaves that talks about a similar topic. The nature of the algorithm, which goes at the same depth in all the branches, makes it really likely that while some topics emerge earlier, some other takes longer to be assigned to their own cluster. Therefore the domain expert has to take a look at the whole taxonomy rather than just at the leaves of the tree, aggregating the nodes that got split but that actually refers to the same topic and should be grouped together. Once a suitable labeling is found, the domain expert has to judge it. If it is satisfying the process ends, otherwise there is the need to tune the parameters and the dataset and go on with another iteration.

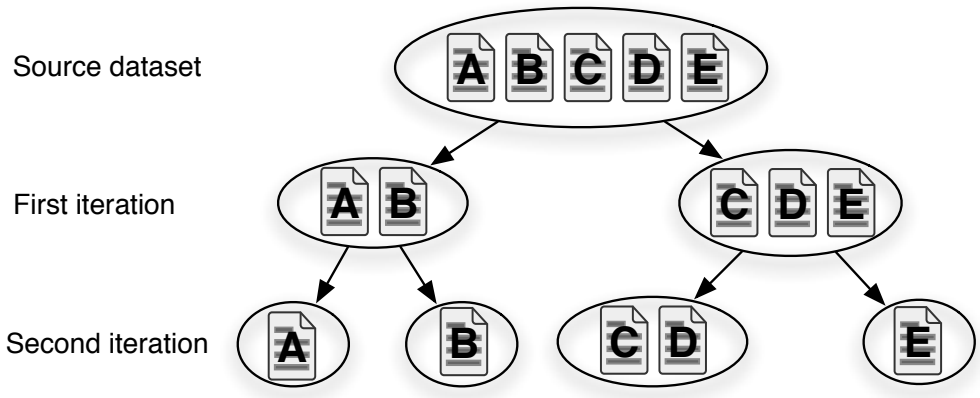

Figure 4.8: Two iterations of the HEM algorithm with the branching factor set to two

In order to obtain satisfying results there is the need to fine tune the algorithm parameters and to choose the best venues that will compose the dataset on which the clustering algorithm will run. The right setting of these aspects is a key factor for the performance of the algorithm. The branching factor and the maximum depth that has to be reached and it

affects how the tree is built and thus the composition of the clusters. The branching factor determines how quickly the number of clusters increases. On the other side, the depth the algorithm has to reach affects the number of time each cluster has to be partitioned in smaller groups.
The need to find the best value for these parameters and the optimal input dataset lead to the design of an iterative process that puts the domain expert in a feedback loop with the algorithm as shown in figure 4.9. After each iteration the domain expert has to try to give a meaning to the taxonomy. Unless a meaningful model is found the operations have to be reiterated with some modification to the inputs of the algorithm. A change in the parameters could lead to the increase or to the decrease of the number of cluster generated. It also affects the structure of the tree and therefore the way the clusters are generated. Since the branching factor is applied at each level of the tree, it is important not to use a too high value that would turn into a number of nodes that grows exponentially with the depth. For this reason it is better to keep a low branching factor and then allow the algorithm to run through more levels. That would guarantee the creation of a number of clusters that fits the purposes of the work without having to deal with an exponential number of potentially uninteresting groups.
The change in the dataset could be effective in driving the algorithm toward the desired direction by providing more examples of papers from under-represented topics that otherwise would be considered as outliers rather than as a relevant cluster. For instance, if it is clear that a topic is a missing, it is possible to make it appear by adding to the dataset a specific journal or conference that is known to deal only with that subject.
The presence of a domain expert that supervises and directs the whole process is also valuable when it comes to evaluate how good the results are. In

such a process the validation happens within the execution itself, making it possible to discover and fix the errors in the model directly when it is built. This kind of process leads to the construction of a meaningful and sound model.

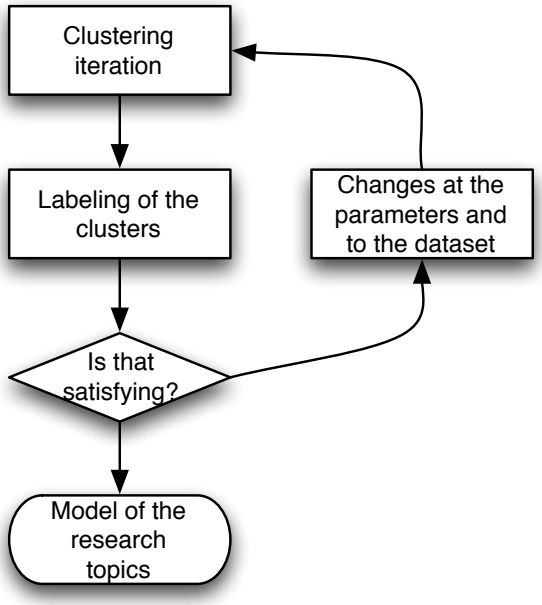

Figure 4.9: The iterative clustering process

4.5 Paper classification

Once a model for the research area has been defined, it could be used as the basis to build upon other tools. For instance, the model could be used to train a classifier that is able to guess which is the topic of an unseen paper. In order to do that this work relies on text-mining techniques. Is uses the instances that are part of each clusters and the names of the topic given by the domain expert as the training instances. This makes it possible to

use all the knowledge extracted by the tool for the identification of the research topics also as training set rather than having to spend a huge effort in the realization of a brand new collection of documents labeled with the name of the topic they are talking about. Using the model generated as described in the previous section, the training set for the classifier comes for free and thus there is the need only to define the details about the algorithm that is going to be used.

The technique used in this work is based on the naïve bayesian classifier. The approach relies on the Bayes theorem (Reported in equation 4.1) that shows the relationship between the a-priori and a-posteriori probability of an event given certain conditions.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{4.1}$$

That theorem still holds when there is the need to consider some complex events that are composed by more than a single feature rather than the simple formulation with just atomic events. Of course a more complex formulation introduces some problems when the variables are correlated each other since that makes it impossible to analyze each feature independently. To overcome this issue making the problem more easily tractable with a more compact set of training instances it has been introduced the naïve assumption that treats each feature as if it were independent from the other. In that way it is possible to obtain a formulation of the problem where each variable could be treated separately as shown in equation 4.2.

$$P(C|x_1, x_2, ...x_n) = \frac{P(x_1, x_2, ...x_n|C) * P(C)}{P(x_1, x_2, ...x_n)} = \frac{\prod_i P(x_i|C) * P(C)}{\prod_i P(x_i)} \tag{4.2}$$

That formula could be interpreted in a text-mining context by thinking at C as the topic of the document and at the variables x_i as the terms considered

in the analysis. It is then possible to find out which is the class that best fits with the words contained in the document by finding out the one that maximizes the likelihood function as shown in equation 4.3.

$$\text{Topic} = \underset{C}{\text{argmax}}(\prod_{i} P(x_i|C) * P(C)) \tag{4.3}$$

Note that it is required just to know which are the probability to find a feature for each topic and the probability to find each topic. All these values can be computed from a training set of instances and then are used to guess the topic of unseen papers.

The nature of the classifiers raises some sort of problems related to the probability computation involved. One is the *zero-frequency problem* that arises when there are some features that are not present in any instance of the training set for each class. That absence turns out into some probability that are equal to zero and therefore the contribution of the other features is not considered at all. To reduce the impact of this problem, it has been used a *Laplace estimator* which adds up a small fixed value to all the features in order to make them greater than zero. Another property of this kind of classifier is that the results tend to be very clearly separated, with a single class with high probability values and the other with very low values. On the one hand it is good since it leads to a clear guess, on the other hand it does not deal well with documents that spans across more than a single topic. However this drawback does not affect too much the outcome of this work since there are considered documents that tend to have a clear and rather specific topic.

Despite its simplicity that algorithm is quite good and works very well in text-mining tasks even though it makes such a strong assumption on the correlation of the features: that assumption does not affects in a negative way the results of the algorithm that could hence used to get meaningful

results.

From a higher level perspective the algorithm reads all the terms in each document and uses them to compute the probability to find each word in a text that is talking about a specific topic. Once the probabilities have been computed, the algorithm takes the terms that appears in a new document and computes the probability of the document to talk about each topic and then it returns the theme that is most likely according to the words found. An example of the classification process is reported in figure 4.10.

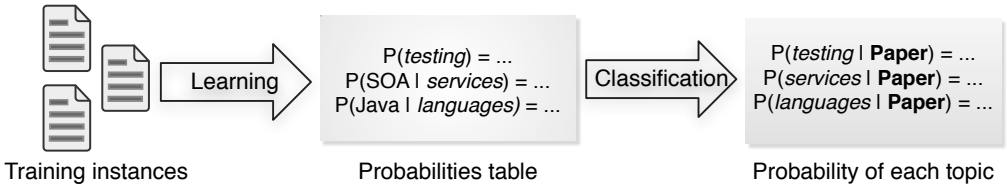

Figure 4.10: The classification process

However, the implementation is not bound to the usage of a particular classifier, and it is quite easy to plug another classifier into the system. It is required to extend an abstract class, overriding the methods to train the new classifier and to obtain the classification for a new instance as well as the facility to store and load it from the disk.

In this work it has been used the naïve bayesian classifier implemented in the Machine Learning for Language Toolkit (MALLET) [55] library that provides a lot of tools to perform analysis and operations on textual document. MALLET provides both the implementation of the analysis methodologies and the tools to preprocess the textual document transforming it in a format that could be interpreted and processed in an efficient way by

the algorithms. In MALLET, the preprocessing is based on a pipeline that makes it possible to combine all the steps required to transform the format of the textual document. Another point that leads to the choice of that library is the fact that it makes it easy to customize all the settings of the algorithm, making it possible to control and customize as desired every aspect.

In order to ensure the quality of the results, the training phase has been followed by a validation step that aims to double check the results by comparing what the algorithm guessed with the actual topic of the papers. The data used in this step has been taken apart from the original dataset: the bigger part of it has been used to actually train the classifier, a small part has been used to check the performance of the classifier. In that way it has been possible to have an immediate validation of the result based on real world data that are very similar to the documents that will be passed as input to the classifier. This step is even more significant considering that the classifier does not provide an easily understandable model: it just computes all the probabilities that do not carry a clear high level meaning. For this reason it is important to check early and in an automated fashion the correctness of the results.

4.6 Author profiling

The technique developed to obtain the topic of a paper is not just useful by itself but could be used also as the basis for other analysis tools. An example is the creation of the profile for each researcher involved in a scientific community. In order to do that there is the need to combine the classification technologies with the data gathered for each single re-

searcher. That data is then processed by the classifier that determines the topic of each paper. Once all this information are available it is required to gather them in some way that makes it possible to summarize the interests of the researcher. While the data gathering process have already been discussed, this section is going to describe the details of the summarization process.

The interest summarization algorithm has to find out which are the most relevant topics for each author by looking at the content of the papers he published. The goal of the algorithm is to extract only the topics that match with the actual research interest, dealing with possible outliers and taking into account the fact that the topics one researcher is interested in changes as the time goes on. The outliers could come from the publication of a work on an unusual topic written after some sort of collaboration with another research group that have slightly different research interests. Another requirement about the profile is that it should be meaningful not only in some obscure statistical term, but it rather has to be easily understandable at a first glance. For this reasons the selected output is a list of topics ranked according with their relevance with respect to what the author published.

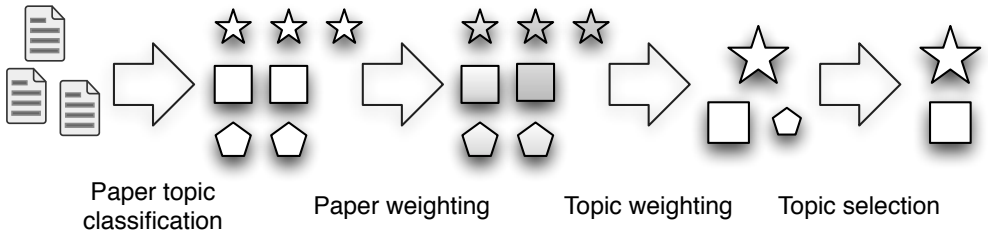

Figure 4.11: The profile creation process

The implementation of the interest summarization algorithm has to

deal with all the details that would make it possible to meet the requirements mentioned above. At first there is the need to guess the topic of each paper the author published. That is possible by using the classifier that has been specifically designed for a task like that. Once all the papers have been classified, there is the need to aggregate the results in order to discover which are the most relevant topics. That is done by associating to each topic a value that reflects the number of the papers published by the author that talk about it. Since there is the need to take into account also when a paper was published, that index cannot be just the number of papers classified for each topic, but there is the need to introduce some weighting to make the older papers have a lower impact than the newer. The final weight of each topic is therefore computed with the formula 4.4 that is based on the weight function defined in 4.5. Note that the definition of the weight of a paper depends only on the topic guessed by the classifier and on its year of publication. The parameter α has to be between 0 and 1 in order to make the weight decrease as the year of publication of the paper goes deeper in the past.

$$\text{Weight}_{\text{Topic,Year}} = \sum_{\text{Paper} \in \text{Papers}} \text{weight}(\text{Paper}, \text{Topic}, \text{Year}) \tag{4.4}$$

$$\text{weight}(\text{Paper}, \text{Topic}, \text{Year}) = \begin{cases} 0 & \text{if topic(Paper)} \neq \text{Topic} \\ \alpha^{\text{Year}-\text{year(Paper)}} & \text{if topic(Paper) = Topic} \end{cases} \tag{4.5}$$

The weights computed for each topic are a quite objective measurement of the relevance of each topic in the interest of the author. Moreover it is possible to rank the interests by sorting with a descending ordering the weights of the topics. While that would able to provide an effective indicator of the relevance of each topic, it is not going to filter out the out-

liers due to some misclassification or to some uncommon collaboration with other researchers. In order to avoid this problem, there is the need to keep just the most significant topics. Since it is not possible to determine a-priori which is the number of topics to keep, the filtering algorithm works adaptively: it continues to pick from the list the most relevant topic until the weight of the picked topics goes over a threshold. Being the threshold referred to the weight it makes possible to keep a significant number of topics for each author without having to care if he focused on few topics or whether its interests were quite broad and spanning over a lot of themes. Figure 4.12 shows how the different composition of the interests could lead to different topic selection.

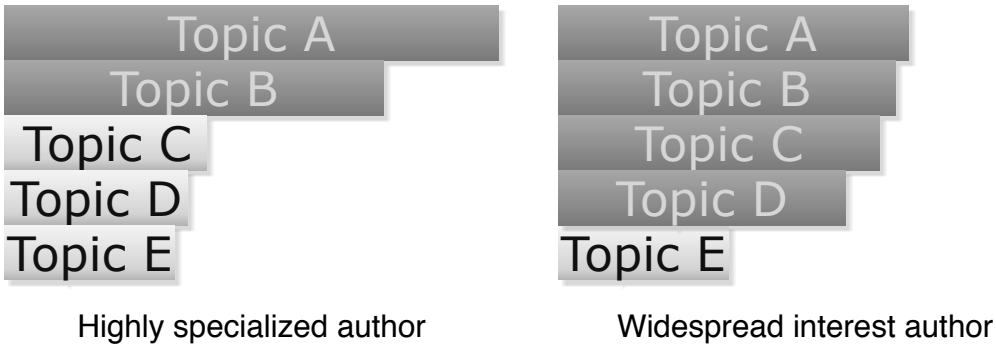

Figure 4.12: Different kind of author profiles

4.7 Applications

The analysis tools described so far can be used and integrated in order to produce some applications that are able to provide useful information to the researchers. In this thesis work the analysis techniques turned into two different tools that have also been used as the case studies to test in

a real life environment the effectiveness of the methodologies developed. The first case study is more meta-research oriented and it aims to make the trends in the evolution of the research topics emerge. The second is somehow more practical and it has been developed to help the researchers in the choice of the papers they could be interested in for a review in a conference.

4.7.1 Evolution of the research topics

To track the evolution of the research topic there is the need to define at first which is the subject of the analysis. That is particularly important since each venue is biased towards some topics rather than others. While this is not an issue when the analysis focuses on a single venue and therefore the goal is to find out these biases, when it comes to analyze the whole community it has to be taken into account to obtain meaningful results. Once the journals and the conferences that have to be considered are chosen, the first step is to run the crawler to get all the data that have to be fed to the analysis tools.

The analysis phase consists of two steps: at first all the downloaded papers are classified and later the data is summarized and reported into some user friendly charts. The classification step relies on the model and on the tool to guess the topics of the papers described before. On the other side the part that is in charge of displaying the topic evolution has to gather all the information about the topics produced by the classifier and output some charts to make the trends evident. The implementation of this step takes as input the classified papers and builds the charts accordingly to the settings specified by the user. Those settings are basically related to the granularity of the data aggregation that has to be performed. It is possible to specify

the initial and final years that have to be taken into account as well as how many years has to be grouped together. Grouping is fundamental to obtain statistically relevant results, especially when the amount of data for each year is not so high: in such cases it is better to sacrifice the details in order to have significant trends. Moreover the trends we are looking for covers several decades, to it is not actually an issue to group the data in set of a two or three years rather than considering each year alone.
In order to have some significant data, both the absolute and the relative number of papers have been considered. The absolute data are useful to have a general perception about the interest on each topic, but does not help too much in knowing how it is relate with respect to the interest in other topics. The introduction of the second measurement makes it possible to deal and make comparable the data that comes from different sources and from different historical periods. For instance it would make a little sense to compare the absolute numbers of papers of a journal that publishes very few articles in each issue with other venues that publish hundreds of documents each year. Another result from this tool are chart representing the distribution of the research topics at a given time that could help in getting at a first glance an idea about what a venue is mainly focused in. The whole process is reported in figure 4.13 that shows the phases and the outcome of the analysis of the literature.

The outcome of the process could be quite useful in the analysis of a research area: with that data it is possible to know which were the hot topics in the past and how the situation is now. Moreover the analysis could be performed both for a single journal or conference and for the whole research field, combining the data from several venues. That kind of information could be effectively used to discover which publication follows

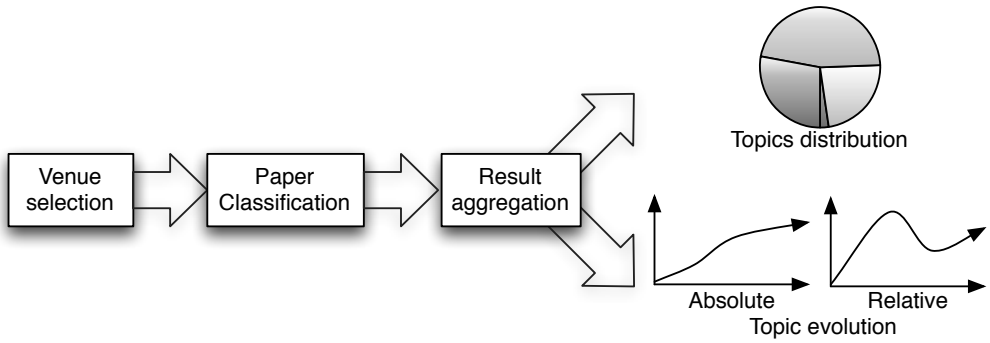

Figure 4.13: The analysis process with an example of the graphical outcome

the mainstream topics and which other are focused more into a particular niche.

4.7.2 Bidding recommender system

The second case study involves all the analytical tools developed in this work and aims to help during the assignment of the papers submitted to a conference to the member of the organizing committee that would know better the issues the paper is talking about. To make an effective assignment, there is the need to both suggest to the committee members which are the papers he could be interested in, and to provide the program chair with information about the topic each author is interested into and the topic of each submission. This information could be quite useful and could be effective in speeding up the assignment of the papers to the reviewers. The two tasks have been faced separately with the development of a tool that is able to present the relevant information both to the reviewers and to the program chair.

Automatic bidding

The implementation of the tool that automatically proposes a list of papers that could be interesting for each reviewer relies on the methodologies that generate the profile of an author as well as to the technology that guesses the topic of a paper. Moreover there is the need to deal with the vector space model representation of the documents in order to make it possible to compute the degree of similarity of a couple of documents.
The process as it is reported in figure 4.14, relies on the profile of the author, where there is a ranked list of the interests of the reviewer and aims to suggest to him a set of papers that reflects his own interests.
To succeed in this task, at first there is the need to compute the profile of the reviewer. The next step is the classification of all the papers by their topics that leads to the creation of several groups of papers that shares the same theme. Finally there is the need to match the interest found in the profile of the reviewer with the submissions. This is performed by associating to each author the set of papers belonging to the topics of his interest. The matching makes a new problem emerge: usually the number of paper that a reviewer wants to put into his own bid is limited from ten to twenty papers. It is clear that keeping all the papers in each group would imply the presence of a higher number of papers in the bids of each committee member, making the automatic bidding process not too much useful since it would not be as selective as it should be. For this reason there is the need to introduce a filtering phase that selects the most relevant papers from the groups of articles about the relevant topics. The filtering passage should also preserve the fraction of papers for each topic in order to better reflect the interest of the reviewer.
The only thing left to define is how to determine which are the papers that

best match with the profile of the author. That problem has been solved by extending the definition of similarity in the vector space model to deal with the need to compare a single submitted paper with the set of the papers the author wrote about a similar topic. The solution consisted in computing and summing up, as reported in equation 4.6, the similarity between the submitted paper and every single paper written of the same topic of the submission by the author.

$$\text{similarity}_{\text{Reviewer,Paper}} = \sum_{P \in W} \text{similarity}(P, \text{Paper}) \tag{4.6}$$

In the equation 4.6, W is the set of the papers written by the reviewer about the same topic of the submitted paper. It is formally defined in equation 4.7.

$$P \in W \iff \text{isWrittenBy}(P, \text{Reviewer}) \wedge \text{topic}(P) = \text{topic}(\text{Paper}) \tag{4.7}$$

This measurement of the similarity was not the only one considered, however it was the more robust and the one able to guarantee the better performances. Other measures considered included the use of the minimum distance rather than the sum of all the distance, but that one did not fit well with the scope of this work since it looks for an example of a very similar paper rather than for a paper that fits in general with the profile of the reviewer.

With such a definition of the similarity between a submitted paper and the profile of the author it is possible to rank the submissions from the most relevant to the ones that do not have too much in common with the themes studies by the committee member. Having the ranked list of the papers, there is left to pick the most relevant for each topic taking care of keeping the desired number of papers and maintaining the same proportions of themes that is present in the reviewer profile. It is important to

keep the proportions that are present in the reviewer profile since it reflects the actual interest of the researcher.

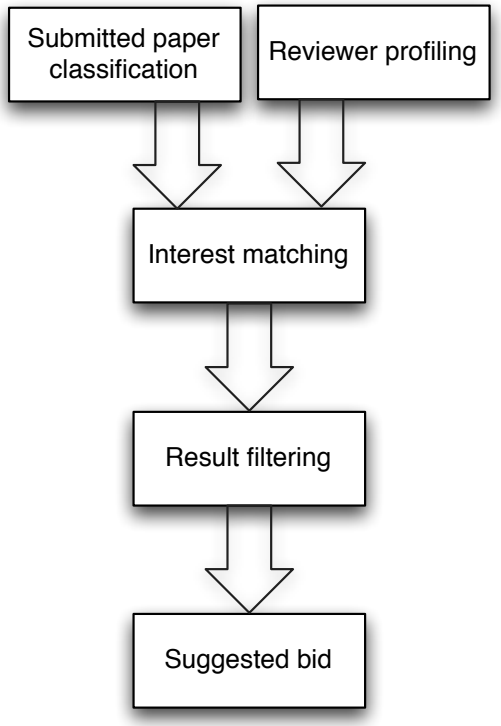

Figure 4.14: The process to automatically obtain a suggested bid

Tools for the program chair

The tools developed could be really useful also for the program chair when he has to assign the papers to the members of the program committee that best know the topic of each article. In that phase it is quite important to have an idea of the content of each paper without having to read carefully a lot of papers. Considering that in the most important conferences there could be hundreds of submitted papers it could save a

lot of time. Moreover it is possible to have an idea of what the committee members are interested in from their profiles. That is quite useful, especially for the researchers that are not personally known by the program chair. All this information in addition with the actual bids could turn into a more efficient assignment of the papers to review. For instance it would be possible to focus at a topic at time, avoiding to waste time that comes from picking a random paper and looking for a researcher that is going to review it.

From the implementation perspective, that does not add much to what has already been discussed regarding the automatic bidding tool. It just presents the same results from a different point of view, so that they could be used effectively by the program chair.

Chapter 5

Experimental Results

This chapter reports the experimental results obtained both during the set up and the tuning of the algorithms and from the case studies applications. At first there is a description of the model of the topics obtained for the *Software Engineering* research community. Then the performances of the classifier derived from the model are described focusing on the tests carried on in order to find out its reliability when it comes to guess the topic of an unseen paper. Finally the test of the analytical tools is concluded with the validation of results obtained from the author profiling algorithm.

The validation of the applications developed upon the analytic tools has been performed on some real life scenarios. The tool that automatically track the evolution of the research interests of the community has been tested analyzing the history of two journals and a conference, and then looking whether the results obtained made sense. The venues selected for this work are the IEEE Transactions on Software Engineering and the ACM Transactions On Software Engineering and Methodology journals, and the International Conference on Software Engineering.

The work related to the suggestion of the bids to the committee member has been performed in two different manners: offline with the actual bids from the International Conference on Software Maintenance (ICSM) 2010 and with the use of surveys in collaboration with the program chair of ICSE 2011.

5.1 Research Field Analysis

The clustering process aims to learn from the data contained in the papers published in different journals and conferences which are the different topics studied by the Software Engineering research community. Being this a data driven process, the model has to be extracted from a huge amount of papers. The dataset used in this part of the work consists in the twenty conferences and the two journals reported in table 5.1 and 5.2. The venues selected for the dataset aims to provide the most complete coverage possible of the research field presenting both generic and specific publications. In addition to this kind of variety, the selected venues also tries to cover different geographic areas taking into account both international and local publications. This set of journal and conferences is the result of several iterations of the clustering algorithm. The original set was composed only by venues that dealt with software engineering in general, but later the more specific publication were added to reflect the feedback of the expert supervisor that were reporting missing topics.
For the selected venues, the data collection phase gathered from the digital libraries 19,509 papers. Unfortunately the abstract was not available for all of them, but the crawler has been able to retrieve the complete data for only 11,868 papers. This problem was due to missing information on the

digital libraries that, especially for the oldest papers, reported incomplete data or in some cases just the index of the issue without presenting more details. In order to deal with the missing information some experiments were performed to determine whether it is better to drop the papers with some missing information or whether, on the other side, the title alone carries still some useful information. The tests showed that it is better to have a huge dataset, even though it could be incomplete rather than to have a limited but complete dataset. For this reason the papers with a missing abstract have been kept in the dataset for the clustering operations.

Other experiments have been performed to determine the best parameters of the HEM algorithm in order to have it work at its best producing the most meaningful results possible. After some iterations it turned out that, for this instance of the problem, the best value for the branching factor is 3 while there is the need to reach the third level of the tree. With these settings the algorithm produced 27 different clusters that then had to be merged and labeled by the expert supervisor. The final labeling resulted in the eighteen different topics reported in table 5.3. While that model has already been validated by the expert supervisor who took part to the clustering process and gave his feedbacks, it is possible to double check the results by looking at the frequent terms. In this way it is possible to agree on the correctness of the labels by looking at the most common words in the set of documents. Unfortunately there isn't any other validation technique that could come in handy when it comes to determine if the selected topics are a good coverage of the research field, or if there is still some research theme which is not taken into account by the model. That kind of validation could only come from the experience of a supervisor who notices that a topic is missing.

Conferences	
ASE	International Conference on Automated Software Engineering
ASPEC	Asia Pacific Software Engineering Conference
COMPSAC	Computer Software and Applications Conference
ESEC	European Software Engineering Conference
ESEM	Empirical Software Engineering and Measurement
FSE	Symposium on the Foundations of Software Engineering
ICPC	International Conference on Program Comphrension
ICSE	International Conference on Software Engineering
ICSM	International Conference on Software Maintenance
ICST	International Conference on Software Testing
ISSTA	International Symposium on Software Testing and Analysis
FASE	Fundamental Approaches to Software Engineering
FM	International Symposium of Formal Methods
MoDELS	Model Driven Engineering Languages And Systems
POPL	Symposium on Principles of Programming Languages
RE	Requirements Engineering Conference
SoftVis	Software Visualization
OOPSLA	Object-Oriented Programming, Systems, Languages and Applications
WOSP	Workshop on Software and Performance
WCRE	Working Conference on Reverse Engineering

Table 5.1: The conferences from which the training papers have been taken

Journals	
TOSEM	Transactions On Software Engineering and Methodology
TSE	Transactions on Software Engineering

Table 5.2: The journals from which the training papers have been taken

Topic	Frequent terms
Aspect Oriented	Aspect, Class, Design, Object, Oriented, Programming
Education	Computer, Design, Education, Language, Object, Oriented, Programming, Student, Teaching
Empirical Studies	Accuracy, Bug, Cost, Data, Development, distributed, Effort, Empirical, Estimation, Model, Prediction, Project, Quality, Requirements, Study
Formal Methods	Analysis, Checking, Concurrent, Formal, Logic, Methods, Model, Properties, Program, Proof, Protocol, Semantics, Specification, System, State, Time, Verification
Middleware for Distributed Systems	Agent, Aware, Computing, Context, Devices, Information, Mobile, User, Pervasive, Web
Mining	Clone, Clustering, Detection, Information, Program, Refactoring, Source, Tool

Models	Architecture, Design, Development, Diagrams, Domain, Language, Model, Object, Performance, Semantics, Time, Transformation, UML
Programming Languages	Analysis, Api, Class, Collection, Compiler, Data, Dynamic, Flow, Garbage, Heap, Information, Java, Language, Machine, Memory, Object, Performance, Program, Programming, Rules, Slicing, Static, Time, Type, Virtual
Program Comprehension and Visualization	Classes, Comprehension, Coupling, Developers, Program, Source, Visualization
Requirements	Based, Behavior, Formal, Requirements, Scenario, Security, Specification, System
Reverse Engineering and Maintenance	Change, Cohesion, Evolution, Information, Maintenance, Program, Source
Security	Analysis, Critical, Privacy, Requirements, Risk, Safety, Secure, Security, System, Vulnerability
Software Components	Access, Application, Architecture, Based, Component, Control, Distributed, Object, Policies
Software Metrics	Complexity, Coupling, Fault, Measures, Metrics, Model, Object, Oriented, Quality, Reliability
Software Processes Agile	Agile, Development, Improvement, Management, Practices, Process, Project, Requirements
Software Product Lines	Architecture, Development, Feature, Line, Model, Product, Reuse, Variability

Testing	Analysis, Based, Bug, Coverage, Criteria, Data, Debugging, Execution, Failure, Fault, Generation, GUI, Input, Localization, Path, Program, Regression, Search, Suite, Symbolic, Techniques, Testing
Web Services	Data, Distributed, Fault, Performance, Service, SOA, System, Time, Web, WS

Table 5.3: The topics found for the software engineering research field and the most frequent terms for each of them

During the iterations that brought the algorithm to converge to a meaningful list of topics, the words that are too common in the documents about software engineering were also tracked. That list is reported in table 5.4. These terms were not considered during the analysis phase and were not fed to the clustering algorithm. The fact that some of these words appears into the list of the frequent terms could be a little misleading, but it is related to the fact that the tool used for the analysis produced that results in two steps. In the first phase the algorithm takes care of the computation of the clusters; in the second step, the tool extracted some information from the content of the papers of each group in order to give to the supervisor some more information to base his feedback on. While in the first step the list of the words to ignore has been taken into account, the second step computes the statistic considering the original documents. This choice does make sense since, while these common words does not contribute too much from the machine learning point of view, they could be useful for the supervisor that otherwise would have to guess which is the context the other uncommon keywords are placed into. The removal of the common words from the list of terms associated with each topic would

make it quite hard to give to each cluster an appropriate label.

software, program, code, approach, system, paper, research, test, case, service, application, engineering, object, type, language, tool, fault, failure, agent, pattern, model, aspect, goal, workshop, research, researcher, result

Table 5.4: List of the terms that could be ignored in software engineering papers

A comparison of these results with the list of topics that were defined by the domain expert without any interaction with the system makes it clear that the introduction of the clustering process is really useful for at least two reasons. On the one hand, the use of the tool makes it possible to discover quickly a higher number of topics: in the case studied the clustering process doubled the research themes that were identified by the domain expert as reported in table 5.5 and 5.6 which respectively shows the topics suggested by the domain expert alone and what it has been identified after the clustering process. On the other hand it provides a more complete source of instances for each topic. In fact it makes a broader dataset available since it took into account and then labeled both the papers from specific venues and also the papers from the journals and conferences about software engineering in general. A comparison of the topics reported in table 5.5 and 5.6 highlights that the clustering algorithm helped the definition of the research themes in two different manners: it made it possible to give a more meaningful name to some interests and showed the presence of a higher number of interests. As an example to the first kind of improvement take the *Design and Software Architectures* topic turned out to be more focused on *Middleware and Distributed Systems*, while the studies about *Performances and Reliability* could be included into

the more general field of interest of *Software Metrics*.

The list of topic retrieved is by far more useful than the one that have been presented in literature. For instance, in the Software Engineering Body of Knowledge (SWEBOK) there is a taxonomy which tends to identify only a subset of the relevant areas and then each interest is studies in the details reporting the different methodologies. The list of topics considered in SWEBOK is the following: *Requirements*, *Design*, *Software Construction*, *Testing*, *Maintenance*, *Software Configuration Management*, *Software Engineering Management*, *Software Engineering Process*, *Software Engineering Tools and Methods*, and *Software Quality*. Just reading the list, it is clear that it is too focused on the methodological aspect that leads to the actual production of software rather than to the topics taken into account by the research community. However the fact that the list retrieved by the tool includes the majority of the topics reported in SWEBOK shows that the application is performing well. Looking at the missing topics, it is possible to observe that they are only the most technical, such as *Software Configuration Management*, which are the less appealing for the research community. Similar results could be obtained from a comparison with the software engineering section of the ACM Computing Classification Systems. That list reports the following topics: *Requirements and Specifications*, *Design Tools and Techniques*, *Coding Tools and Techniques*, *Software and Program Verification*, *Testing and Debugging*, *Programming Environments*, *Distribution, Maintenance, and Enhancement*, *Metrics*, *Management*, *Design*, *Software Architectures*, *Interoperability*, and *Reusable Software*. The issue with this list is the space left to the more technical aspects and specifically to the tools that may help the programmers in the realization of software systems. However the list substantially agrees with the results suggested by the tool.

Manual approach	
1	Empirical Software Engineering
2	Formal Methods
3	Design and Software Architectures
4	Models
5	Languages
6	Requirements Engineering
7	Security and Privacy
8	Performance and Reliability
9	Testing and Analysis
10	Services

Table 5.5: Topics suggested by the domain expert

Such a similarity may not be too evident because of the different labels chosen to describe the topics but it should be clear that, just to make a couple of examples, *Formal Methods* and *Software and Program Verification* or *Models* and *Design Tools and Techniques* actually refer to the same research interest.

5.2 Paper classification

The data used for the creation of the model of the research field has then been used to train the classifier in order to make it able to guess the topics of an unseen software engineering paper. The labels that the supervisor gave to each cluster have been used as the name of the classes that the classifier is going to deal with. The dataset has been split into two different

Automated approach	
1	Empirical Studies
2	Formal Methods
3	Middleware Distributed Systems
4	Models
5	Programming Languages
6	Requirements
7	Security
8	Software Metrics
9	Testing
10	Web Services
11	Aspect Oriented*
12	Education*
13	Program Comprehension and Visualization*
14	Reverse Engineering and Maintenance*
15	Software Components*
16	Software Mining*
17	Software Processes and Agile*
18	Software Product Lines*

Table 5.6: Topics obtained by the clustering algorithm. The new topics are highlighted with a star.

parts in order to make it possible to use a fraction of the papers to actually train the classifier and another to verify the results it produced. Since it is better to have the higher number of instance possible for the training phase, the two sets have been defined by splitting the original dataset in a bigger training set containing the 85% of the papers and in a smaller one with the remaining 15% to be used for the validation. The composition of the two sets was chosen randomly and, in order to ensure the statistical relevance of the classification performance measurement, the experiment has been repeated several times in order to ensure a fair distribution of the papers.

The precision and the recall of a classifier are the measurements that are usually used to determine how good it is. Precision represents how good is the algorithm in the assignment of a label only to the documents that should actually have it. On the other side, recall measures how good is the algorithm in labeling correctly all the documents that belong to a class. A more formal definition of precision and recall is given in equations 5.1 and 5.2 where a *true positive* is a correctly classified document, a *false positive* is a document classified as belonging to a class while it does not, and finally a *false negative* is an element that is not classified as a member of the class it actually belongs to.

$$\text{precision} = \frac{\text{truePositives}}{\text{truePositives} + \text{falsePositives}} \tag{5.1}$$

$$\text{recall} = \frac{\text{truePositives}}{\text{truePositives} + \text{falseNegatives}} \tag{5.2}$$

Figure 5.1 shows the average precision and the recall for all the topics identified in the software engineering research area. The precision is about 70% in the worst cases and a bit more than 80% in the average case. The recall performances are lower ranging roughly from the 50% to the 80%. At a first

glance the results could seem fairly good but not as good as one should expect to use the tool in an effective and reliable manner. In practice the tool works well and there are some explanations and remarks that should pointed out to better understand what could affect the performance of the classifier.

A first remark is that in some cases the papers lies between two topics and it is possible that while the clustering algorithm privileged one of them, the classification algorithms tends to be more biased towards the other. This of course does not affect too much the results from a high level perspective but it has a relevant impact on precision and recall since the metrics are not aware of the possibility that the paper could be possibly classified in a different way than the one reported on its label. Moreover there are some topics that present a lot of similarities with other, so there is the chance of the presence of some overlapping that the algorithm has somehow to solve picking a label rather than another.

Finally there is the need to point out that, while the usage of the data obtained from the clustering process is helpful since it provides a lot of labeled data almost for free, it introduces also some noise because not all the instances were double checked by an expert supervisor. That is not going to produce relevant biases in a real application since the few wrongly labeled instances that originates the noise gets covered by the thousands of correctly labeled documents, however it is possible that some noisy document falls into the validation set affecting the performance measurements. However, the repetition of the validation step for several times with different validation sets should have limited this effect.

A second validation step has been performed by double checking the results on a number of unseen instances, and even in that case the perfor-

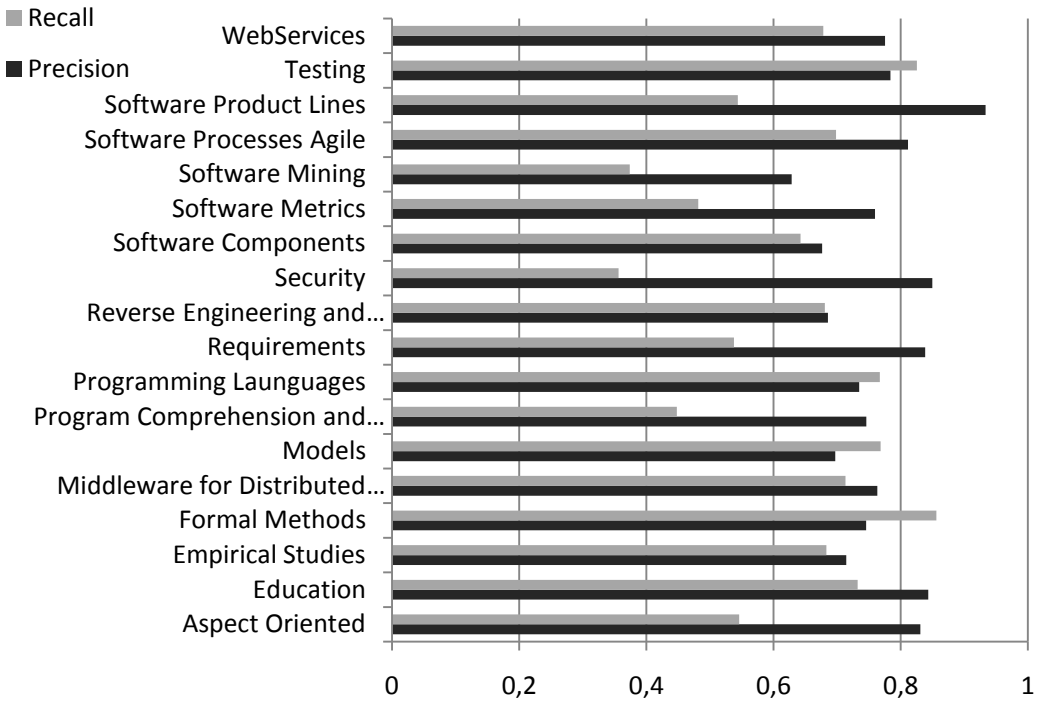

Figure 5.1: Precision and recall for the document topic classifier

Title of the paper	Topic
Exception Handling Patterns for Process Modeling	Models
A Quantitative Investigation of the Acceptable Risk Levels of Object-Oriented Metrics in Open-Source Systems	Software Metrics
A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search	Testing
Directed Explicit State-Space Search in the Generation of Counterexamples for Stochastic Model Checking	Formal Methods
Determining inspection cost-effectiveness by combining project data and expert opinion	Empirical Studies

Table 5.7: Some examples of the topic guessed by the classifier

mance of the classifier were quite good: the domain expert agreed with the topic selected by the automatic classification tool for the majority of the documents. An example of the topic classification for several papers is reported in table 5.7 that shows both the title of the paper and the topic guessed by the classifier.

The sampling of some classified papers is really important to double check the results obtained from the algorithms. That is the only way to spot the possible issues with the model that could be wrong or incomplete. Since an error in the model is going to heavily affect all the subsequent phases, there is the need to look at the results produced by the tools in a

critic way trying to find the possible misbehaviors of the algorithms and associating the problems with the issue in the model that introduced it. Once the issue is fixed there is the need to step back, fixing the model and running again all the tool-chain in order to have meaningful results.

5.3 Profiling of a researcher

The validation of the tool to generate the profiles representing the interest of a researcher is a little bit tricky since there isn't available anywhere an updated description of what each members of the community is researching on. For this reason the validation step has been performed by asking to several researchers whether the profile generated reflected his research interests. With the survey it is possible to know directly from the researchers whether the tool performed well. The survey requires to clarify the setting of the problem, avoiding that the researchers misunderstand the question and therefore looking for a too detailed topic classification. For this reason there is the need to propose both the profile obtained by the tool and the options from which the topics have been selected. The researchers that contributed to this work with their answers on the profiles are the committee members of ICSE 2011 as a part of the experiment on the bids recommender system.

The survey presented interesting results that on the one hand confirmed that the tool is able to produce meaningful results but on the other hand remarked how the model of the research area is fundamental in order to obtain accurate results. In fact during that phase emerged that there were missing topics that turned out into incomplete profiles or into profiles with wrong topics due to the misclassification of some papers. How-

ever a fix to the model with the addition of some examples for the missing classes allowed the training of a *correct* classifier that made the missing topics appear in the profiles.

Looking at the numbers, from the 39 replies to the survey 18 researchers agreed with the profile generated by the tool. This could seem a very little fraction, but that number represents the profiles reported as completely correct. In the remaining 21 profiles some were claimed to be wrong because of the absence of a topic, other for the presence of a topic that used to be in the research interests of the researcher but that are no more studied, and finally some profiles contained topics that have never been studied by the author. Looking with more attention at the data it turn out that 8 were correct but for few missing topics, 5 presented some wrong topics but at the same time the other themes presented covered correctly the interests of the researchers and finally 8 presented both missing topics and wrong topics. These data are reported in figure 5.2.

The errors in the profiles present two different natures. Missing topics come from the fact that there isn't enough information in the literature for topics that the researcher just started working at. This kind of issue could also be due to the fact that there is a lower number of publications on that theme with respect to the other relevant topics reported in the profile. While it is hard to solve the first aspect of the problem, it is possible to avoid the missing topics by improving the heuristic that decides how many topics are relevant for the profile.

The presence of wrong topics comes from errors in the underlying model that leads to the misclassification of the papers. This issue could be solved by improving the model in order to reflect better the research community.

To better validate the methodology, the model has been fixed to take

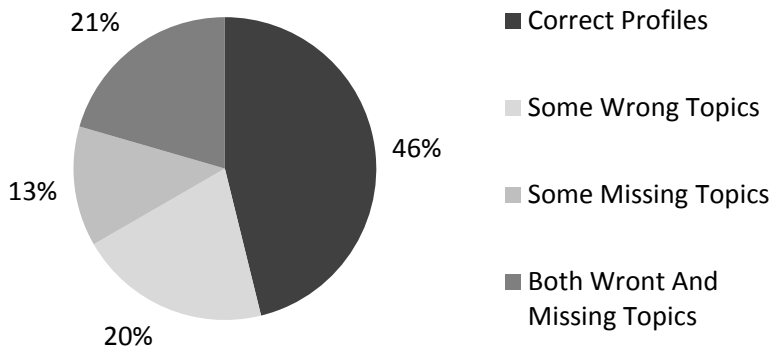

Figure 5.2: Results of the survey to check the correctness of the profiles

into account the topics that were ignored. The introduction of two new topics improved the quality of the results that, at that point, reflected in a better way the feedbacks given by the researchers: a lot of the profiles with both missing and wrong topics turned into correct profiles. The right classification on the one hand made the wrong topic disappear and on the other hand replaced it with a topic that previously was reported as missing. As the data reported on figure 5.3 shows, there is still need to fix something in the model but at least the work is on the right direction as a little improvement to the model lead to a significant reduction of the wrong topics.

The fix to the model performed at this stage took into account the addition of the two classes that were presenting the highest rate of misclassification: *Reverse Engineering and Maintenance* and *Program Comprehension and Visualization*. After the insertion of these two topics the distribution of the errors in the profiles changes as reported in the chart of figure 5.4. That result of the addition of the new classes is a significant improvement: the topics that were wrongly classified as belonging to *Models*, *Programming*

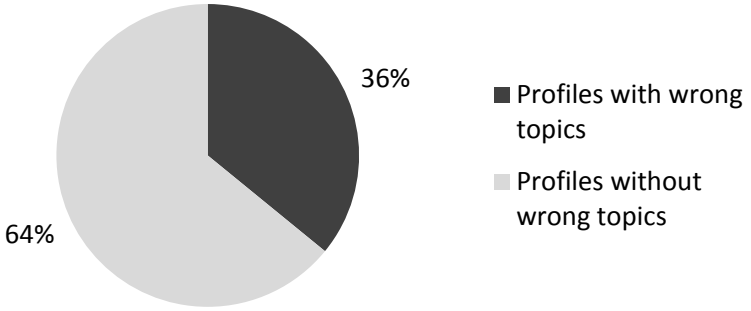

Figure 5.3: The distribution of the errors on the profiles after the fix to the model

Languages, *Aspect Oriented*, *Software Processes and Agile*, and *Software Components* completely disappeared. The errors due to the misclassification of papers belonging to *Software Mining* and *Testing* presented a reduction. Finally some other topics (*Education*, *Empirical Studies* and *Middleware and Distributed Systems*) were not affected by the introduction of the new topics. That means there is still the need to improve the model inserting some other significant topics that so far have not been taken into account. These results justify the confidence towards the approach since they shows that, given a model that actually reflects the research community interests, it is possible to generate an accurate profile for the researchers.

5.4 Topic evolution

The first example of the application of the tools that have been shown to work in the previous sections is the analysis of the research interests in some of the most relevant venues of the software engineering community. The analysis span through two journals, Transactions on Soft-

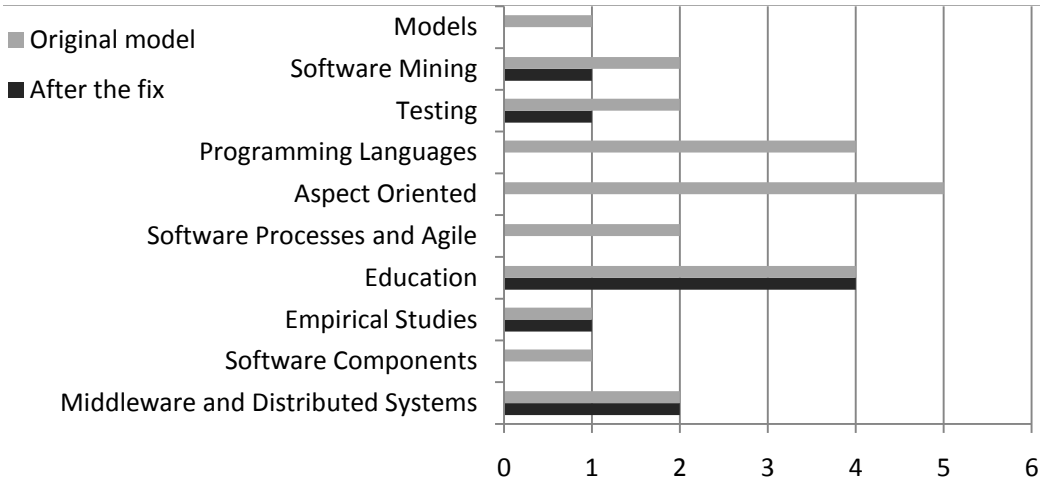

Figure 5.4: The distribution of the wrong topics before and after the addition of the two new topics

ware Engineering (TSE) and Transactions On Software Engineering and Methodology (TOSEM), and a conference, ICSE. The choice of these three venues made it possible to have a significant insight on the interest of the software engineering community in general. The significance of these three venues comes from their importance as well as to the fact that by looking at the paper published here it is possible to get an overview of the whole history of the research field from the beginning. Moreover the presence of two journals and a conference makes it possible to see whether there is any difference in the topics that are published in the different kind of publication. This section at first reviews singularly each venue and later there are some conclusions about the software engineering field in general.

5.4.1 IEEE TSE

The IEEE Transactions on Software Engineering is one of the oldest journal dealing with the topics researched by the software engineering

Year	Papers	Year	Papers	Year	Papers	Year	Papers
1976	35	1977	47	1978	58	1979	65
1980	66	1981	60	1982	59	1983	80
1984	87	1985	149	1987	130	1988	148
1989	133	1990	111	1991	94	1992	83
1993	85	1994	75	1995	77	1996	65
1997	56	1998	73	1999	53	2000	68
2001	63	2002	76	2003	86	2004	70
2005	67	2006	61	2007	55	2008	52
2009	48	2010	24				

Table 5.8: Number of papers retrieved from the TSE archive

community. Its first issue was published back in 1976 and until today several issues have been published every year. The collection of the articles published in the journal contributes to the creation of a relevant dataset for bibliometric studies. Table 5.8 reports the number of the TSE articles that have been retrieved for each year. These numbers make clear that there is a lot of information available that could be used to base the data driven analysis on. While there are some fluctuations in the number of the papers published, a significant amount of documents is available for every year. This also enables the possibility to know better the current situation and to compare it with the research interests of the past. To improve the effectiveness of the analysis the data set has been analyzed in two steps: a first phase taking into account the trends on the macroscopic scale followed by a more detailed and fine grained study of the most interesting trends.
For a first rough analysis aimed to identify the macro-trends of the topic

evolution the dataset has been split into three temporal subsets which respectively covered the years from 1976 to 1987, from 1988 to 1999 and finally from 2000 to 2010. While it is clear that this subdivision is quite rough since it spans over very long temporal intervals, it is still useful to identify which are the trends on the macro-scale and to start getting an insight of which are the interests of the community. This preliminary step is required since the model consists in a very high number of topics but only a part of them show interesting and clear trends.

In the early ages of software engineering, represented by the data of the 1976-1987 papers of the TSE reported in figure 5.5, there is a dominance of the studies that makes is possible to create solid basis for the discipline. For that reasons it should not be surprising the fact that there are a lot of theoretical studies about the definition of the formal properties of software as well as the definition of the tools to write programs. The typical research papers were following a more mathematical oriented approach on software engineering that directed research towards the definition of the formal and theoretical properties of software systems. The more application oriented research had to work on the definition of tools and techniques to effectively write programs: in this context, the studies focused on the one hand on programming languages and paradigms, on the other hand on methodologies, such as testing, able to lead to the development of programs that actually works. Finally, as the first software systems went into production, it immediately came out the need to take care of all the lifecycle of an application keeping it up to date with respect to changes in the specification and providing effective way to get rid of the errors. The lack or the minor relevance of some topics should not be a surprise. For instance, in the early years, there were not available networking technolo-

gies that could have lead to studies on distributed systems and service oriented applications. A similar problem happened with empirical studies that were not diffused since there a very limited availability of data from existing software systems.

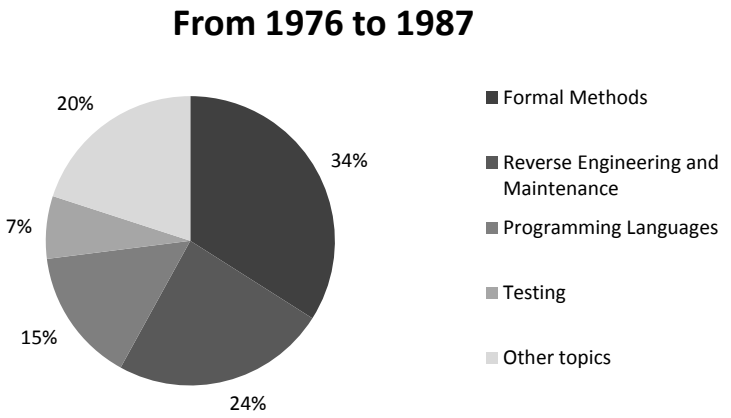

Figure 5.5: Distribution of the topics for TSE from 1976 to 1987

The next period, which covers the years from 1988 to 1999 and that is reported in figure 5.6, still shows the dominance of the themes that have been discussed in the first years but is also the setting for the rise of other topics. For instance the widespread availability of networking techniques shifted the interest on distributed systems and more recently on web services. Moreover the need to develop software systems of increasing complexity made the researchers to come up with sophisticated modeling techniques able to represent in a clear way both the requirements and the structure of the applications. Finally in this period the diffusion of open source and free software made it possible to start investigating how real life systems are developed. Although the chart does not show any topic related to that, summing up the shares of articles gained by *Empirical Studies*, *Software Visualization* and *Software Metrics* it is possible to find an

initial interest to this kind of approach even in the articles published on the TSE.

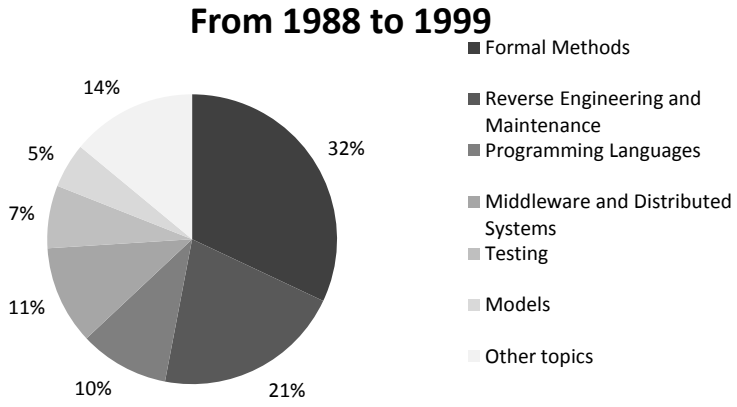

Figure 5.6: Distribution of the topics for TSE from 1988 to 1999

Finally, the trends observed in the previous period went on also for the years from 2000 to 2010, as reported in the chart in figure 5.7. It is possible to observe a decline of the historically relevant topics that are leaving room for a higher variety of topics and research interests. This is due to the fact that the more theoretical knowledge reached some confirmed results and on the other side the object oriented paradigm became the dominant programming abstraction. The data shows that the community confirmed its interest in the topics that emerged in the previous period, while the availability of software repositories both private and public determined the explosion of studies based on the analysis of existing applications and their evolutions.

A final remark has to be spent on the topics such as *Security* or *Education* that apparently always had a little attention. This is mainly due to the fact that TSE is a venue more devoted to the core aspects of software engineering rather than to these research interests which are mainly published

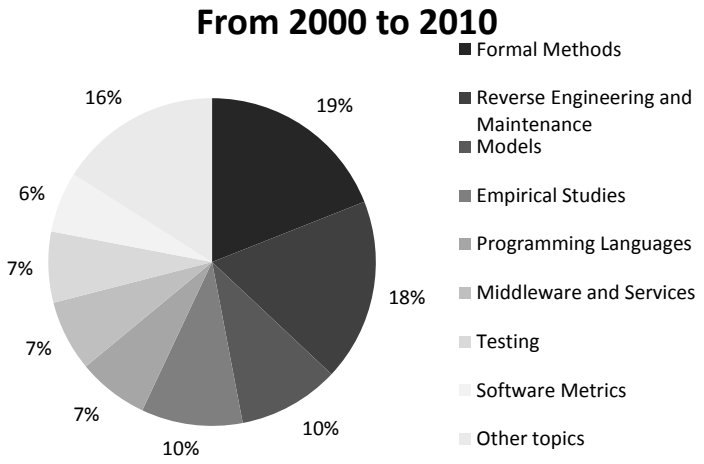

Figure 5.7: Distribution of the topics for TSE from 2000 to 2010

on more specific publications.

Another interesting general trend that emerged is that the number of topics covered grew in each period. This is a signal that the focus of the researchers is shifting from the topics that have become mature to new research themes.

The general discussion showed that it is the case to investigate deeper the evolution of *Formal Methods*, *Programming Languages*, *Models* and *Empirical Studies* which are the topics more likely to present some clear trends. Figure 5.8 shows the evolution of these research interests with a finer three-years granularity.

The interest on programming languages had a peak in the late seventies when the object oriented paradigm started to spread and later presented a constant decreasing share of articles published.

As already mentioned, the interest on empirical studies showed up in the late nineties as repositories like Sourceforge made a lot of data from various open source programs available. Since then that topic constantly

gained interests in the software engineering community.
The formal methods approach always covered an important share of the TSE papers but, as the approach shifted from a more mathematical oriented to a different perspective, it is now gathering less research interest.
Even though with some small variations, the theme of software maintenance is keeping a relevant share of the TSE papers. This is happening because the maintenance has always been an important part in the software engineering process. In the first year it was mainly focused on the techniques to fix bugs in the programs and to adapt it to the changing requirements, while not it is more focused on the retrieval of some information from the code that could be useful to find possible problems.
Finally the evolution of *Models* shows how that topic is consistently gaining interest. This is mainly due to the fact that to manage the complexity of current software systems there is the need to be able to represent them with models that makes it easier to understand how they work and how it would be possible to combine together elements from different applications. It is possible to find out a first peak after the introduction of the Unified Modeling Language (UML) which now represents the standard way to describe software architectures. After that tool became available the research interest on the topic presented a constant though slow growth.

5.4.2 ACM TOSEM

TOSEM is another very relevant journal in software engineering, and shares some of the features of TSE. Unfortunately the data available is much lower both in terms of temporal extension and for the number of articles published each year. In fact, as reported in table 5.9, the journal has been published only since 1992 and for years there is about a dozen

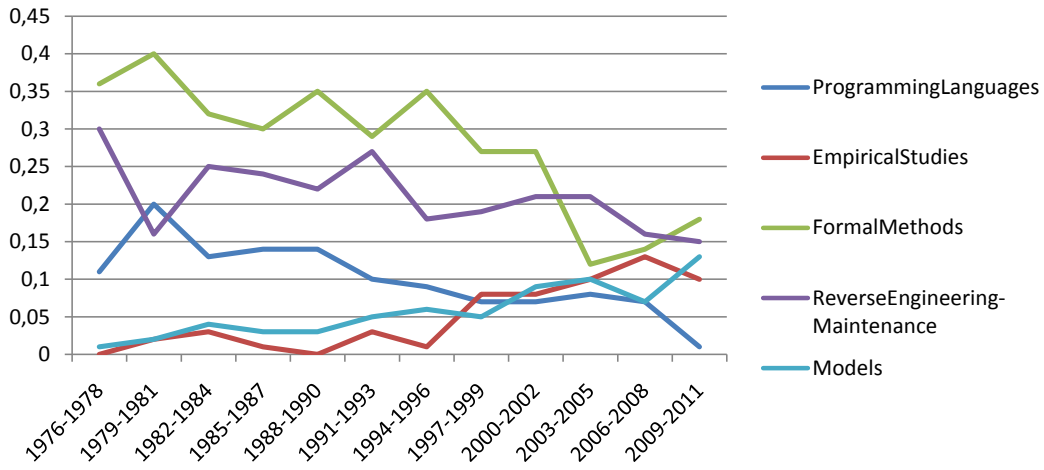

Figure 5.8: The clearest trends in TSE

Year	Papers	Year	Papers	Year	Papers	Year	Papers
1992	15	1993	13	1994	9	1995	10
1996	10	1997	13	1998	14	1999	13
2000	14	2001	10	2002	13	2003	12
2004	10	2005	12	2006	12	2007	15
2008	20	2009	13	2010	6		

Table 5.9: Number of papers retrieved from the TOSEM archive

of papers. While the amount of data available is limited, the analysis of TOSEM is a good test bench to determine how good the performances of the algorithm are when it has to deal with a small dataset.

The analysis methodology followed is the same that has been already described for TSE: at first the general trends were identified from a coarse grained subdivision of the dataset in two parts, one from 1992 to 2000 and another from 2001 to 2010. The trending topics were then analyzed using a finer three year granularity.

In general, the topic distribution is quite similar to the one observed in TSE but, of course, there are some peculiarities related to the nature of the venue. For instance in the period that ranges from 1992 to 2000, which is reported in figure 5.9, TOSEM also shows a high number of significant topics. Moreover there are also strong similarities in the topics which cover the most relevant fraction of article published. However this does not mean that the two journals have exactly the same distribution of the topics. For instance while the emerging topics in TSE are more focused on distributed systems and service oriented architectures, TOSEM shows more interest in the study of software components. Differences like that one were expected since it is clear that while the mainstream topics presents more or less the same distribution in all the publications, the other research interests tends to cluster to a particular venue. This turns out into different distribution for the niche topics which could have a relevant presence in a publication rather than in another.

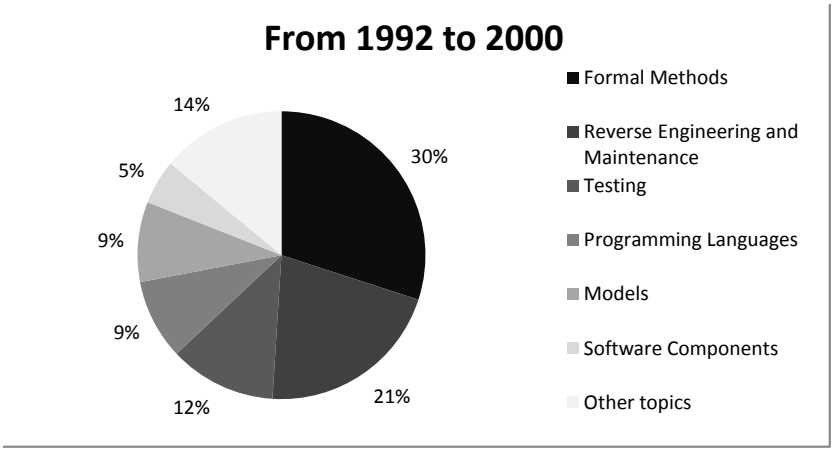

Figure 5.9: Distribution of the topics for TOSEM from 1992 to 2000

In the next term considered, the years from 2001 up to now, TOSEM shows two different trends: on the one hand the mainstream topics are

still present, on the other there is a very high fragmentation as reported in figure 5.10. A possible explanation to these phenomena is the low number of papers published in the journal that could be the cause of the explosion of the minor research interests: with few articles each one has a greater weight, and since TOSEM tries to cover a lot of different interest each of them gets a little but still relevant part of the overall share.
Looking with more attentions at the topics that come out it is possible to find similarities with what has been experienced in TSE. For instance the appearance of topics like *Empirical Studies*, *Software Metrics* and *Middleware and Services* has already been observed and explained. TOSEM also introduces some novel topics: *Aspect Oriented Programming* and *Requirements Engineering* that actually started to develop in the first years of the new millennium as it is possible to observe from the data gathered.

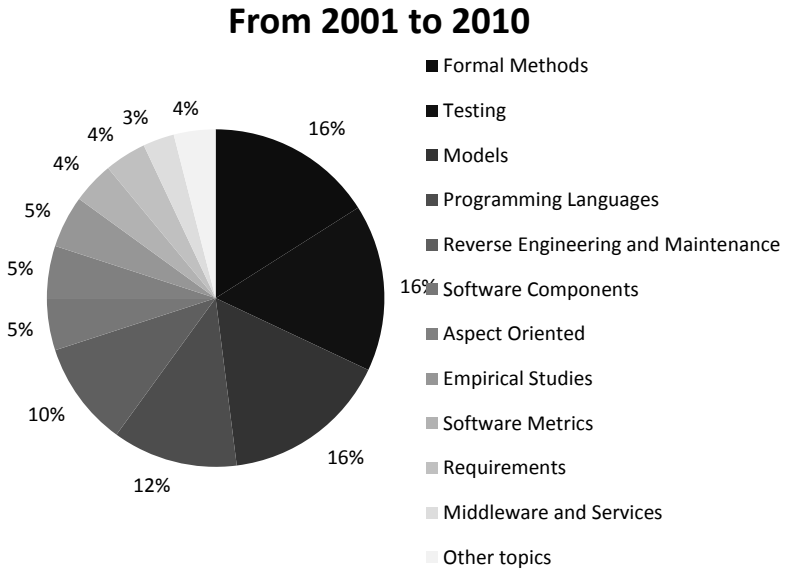

Figure 5.10: Distribution of the topics for TOSEM from 2001 to 2010

While the low number of papers available only affects in a negligible

manner the more coarse grained analysis, it has a more significant effect on the fine grained analysis. If the TSE graphs were somehow smooth, the results for TOSEM show a really different behavior. In particular there are a higher number of topics and that makes each of them to have a very little fraction of the published papers. Moreover the limited number of articles available turns into distributions characterized by bursts and peeks since the presence or the absence of a few papers is able to change a lot the fraction of the papers of a topic. Finally this effect is amplified by the fact that TOSEM is a mainstream journal and has to publish papers about a lot of different topics. Given the very limited amount of space available, it is possible to notice the presence of a topic just on a few years.

Figure 5.11 shows that the mainstream trends identified in TSE have a major role also in TOSEM. While two of them, *Formal Methods* and *Reverse Engineering and Maintenance*, are presenting a clearly decreasing trends the others have a much more constant behavior that could be related to their lower overall share.

Looking at the other interesting topics, which are reported in figure 5.12, there are some confirmations and some other interesting trends that were not clear in TSE. While it were already known that *Models* gained a lot of interest from 2001 to 2006 and that the *Empirical Studies* are emerging, the analysis of TOSEM highlighted the interest on *Aspect Oriented Programming* that has been gathering some interest since the idea first come out, and on *Requirements Engineering* which is slowly but constantly gaining interest.

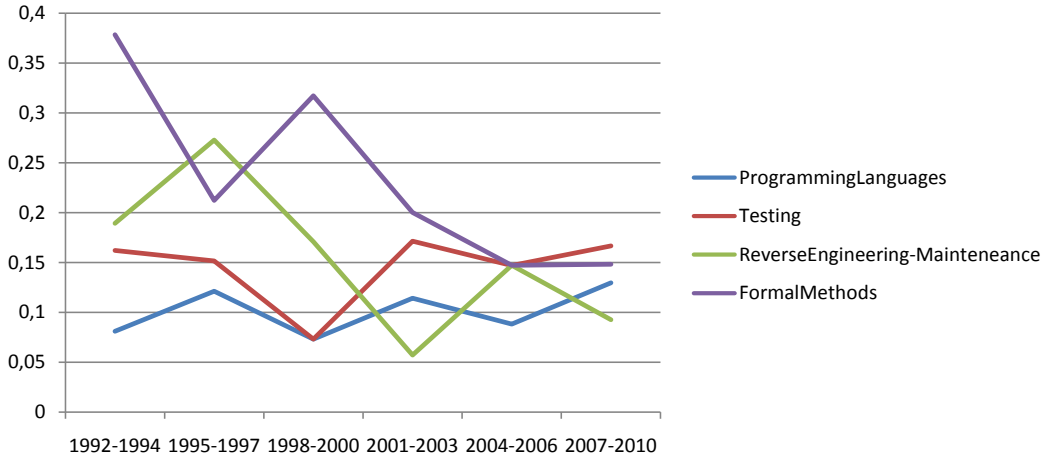

Figure 5.11: Trends for the mainstream topics in TOSEM

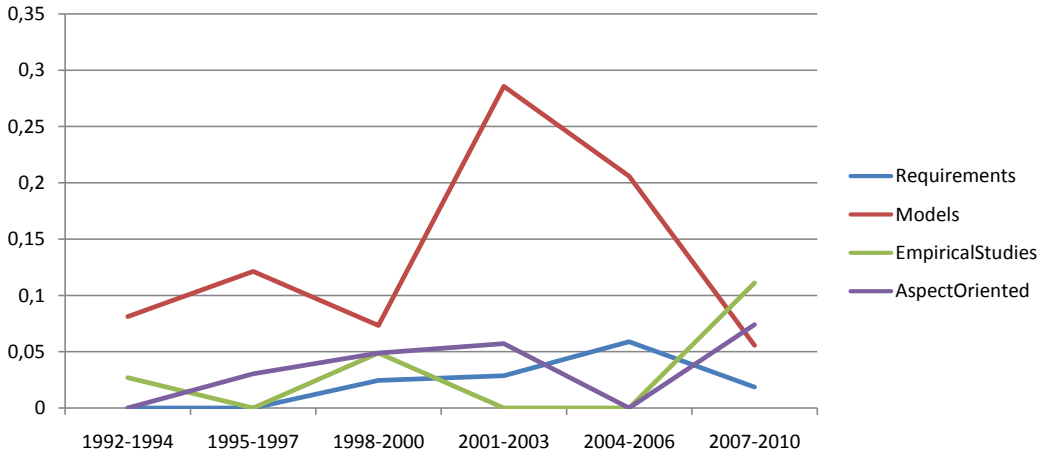

Figure 5.12: Other interesting trends in TOSEM

Year	Papers	Year	Papers	Year	Papers	Year	Papers
1976	103	1978	45	1979	56	1981	50
1982	47	1984	51	1985	54	1987	37
1988	39	1989	34	1990	48	1991	39
1992	25	1993	49	1994	42	1995	32
1996	69	1997	97	1998	65	1999	82
2000	126	2001	120	2002	87	2003	108
2004	105	2005	152	2006	176	2007	89
2008	102	2009	45				

Table 5.10: Number of papers retrieved from the ICSE archive

5.4.3 ICSE

After having discussed the performance of the analysis tools on the journals, it is time to see whether the presented methodology works also for conferences and to try to find how many similarities there are between the two different kind of venues. For the time interval covered and the quantity of papers available, the data of ICSE is more similar to the information gathered from the TSE archive. Table 5.10 reports the number of papers download from the archive of the main track of ICSE. Notice that while the first editions did not happen regularly but sometimes every year and sometimes every two years, from 1988 the conference has been organized yearly.

The analysis has been performed on the same time intervals used for TSE. This has been made possible by the fact that the two venues started the same year, and it also allows the direct comparison of the topics of the papers published in the different publications.

From a first glance at the data gathered for ICSE from 1976 to 1987, which is reported in figure 5.13 it comes out that the in general the topics of the papers of the conference reflects what has been observed in both TSE and TOSEM. Even though the most common topics are the same, there is a substantial difference in how they are distributed: while the journals seems to have a more formal approach to software engineering, at ICSE there is more focus on the aspects concerning with testing and maintenance. Another interesting difference is the relevance of aspects such as *Education* which usually did not have much relevance in both TSE and TOSEM.

These consideration could be a signal that, while both journals and conferences reflects the interests of the community, there are some differences in the specific interests and approaches that guides the selection of the papers that appears in the two different kind of venues. However the differences are not directly about the general interest but rather to the space left to each topic.

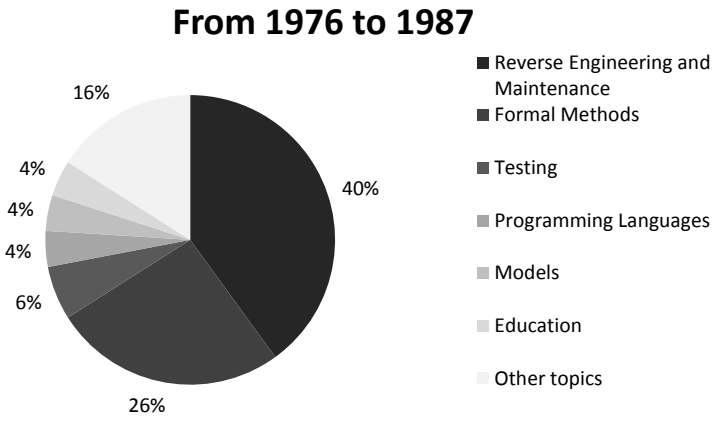

Figure 5.13: Distribution of the topics for ICSE from 1976 to 1987

The differences highlighted for the early years of ICSE are still present

in the following period which is represented by the chart in figure 5.14. In the years which spans from 1988 to 1999 there is still the predominance to maintenance related topics over the more formal aspects of software engineering. Along with this trend it is possible to observe the rise of the study of the processes underlying software production. That interest could be explained by the fact that ICSE leaves more space to the methodological aspects of software engineering. It is interesting to observe how the other trends still appears a bit early. For instance *Empirical Studies* gained interest earlier and also *Models* had a more relevant fraction of papers. This trend could be related to the fact that ICSE tends to leave some space for research in their early stages while journals requires more solid and confirmed results.

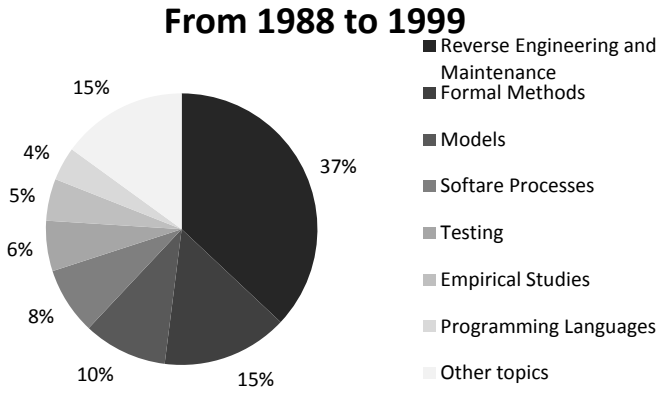

Figure 5.14: Distribution of the topics for ICSE from 1988 to 1999

In the last period taken into account, which spans from 2000 to 2010 and that is reported in figure 5.15, there is the confirmation of both the trends derived from the analysis of the journal and the consideration obtained from the comparison between ICSE and TSE for the previous periods.

A look at the chart shows how the ICSE community still has a huge interest in the methodology and the processes behind software development. That interest increased a lot in this period after the introduction of *Agile methodologies* which started spreading at the beginning of the new millennium. The analysis also reported the presence of the other research themes which started spreading after the year 2000 and that have been showed also in the studies performed on the journals. As an example of this trend, it is possible to observe the presence of *Aspect Oriented Programming* and *Middleware and Services* which, as expected, started to show up only in the most recent period.

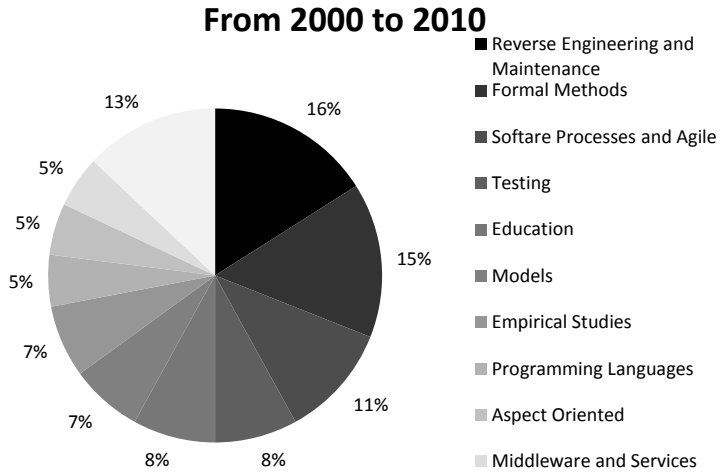

Figure 5.15: Distribution of the topics for ICSE from 2000 to 2010

A more fine grained look at the evolution of the research interest shows similar results to what has been observed in both TSE and TOSEM. As it has already been done with TOSEM, the analysis split the trends in two groups: one containing the mainstream topics and the other which shows how the new trends emerged and gained interest. Regarding the topics which have always had interest in the software engineering community,

ICSE presented the lead of *Reverse Engineering and Maintenance* and *Formal Methods* but in the most recent years there is the tendency to see more or less the same space allocated to each topic. On the other side *Testing* and *Programming Languages* have held the same share but small variations through all the history of ICSE. The data of the mainstream software engineering topics is reported in figure 5.16.
The decline in the share held by the mainstream topics reflects in the rise of a set of new research themes which are getting more interest as the time goes on. These trends are reported in figure 5.17 which shows how the different topics emerged. For instance, the interest about *Software Processes* grew steadily and had a jump in with the introduction of agile methodologies in the last decade. *Models* had a peak in the end of the eighties and then returned at the end of the nineties with the introduction of UML, but after that the topic is presenting a decline. As happened in the journals, it is possible to observe the diffusion middleware, service oriented architecture and to analysis of software repositories such as *Empirical Studies* and *Software Metrics*.

5.4.4 General conclusions about software engineering

The results obtained from the data of TSE, TOSEM and ICSE are coherent and therefore could be used as the basis for some more general conclusions about the field of software engineering regardless of the specific venue. A first result is the identification of the topics which contributed, in the early ages, to the definition of the setting for the software engineering field. These topics include on the more theoretical side the studies about *Formal Methods* and *Programming Languages*, and from a more practical perspective the development of *Maintenance* and *Testing* techniques. The evo-

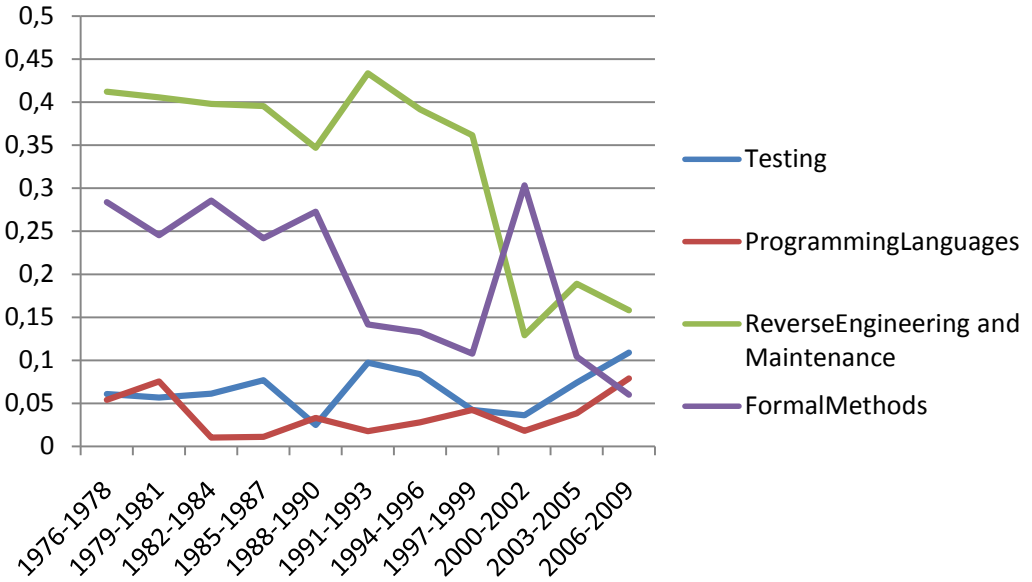

Figure 5.16: Trends for the mainstream topics in ICSE

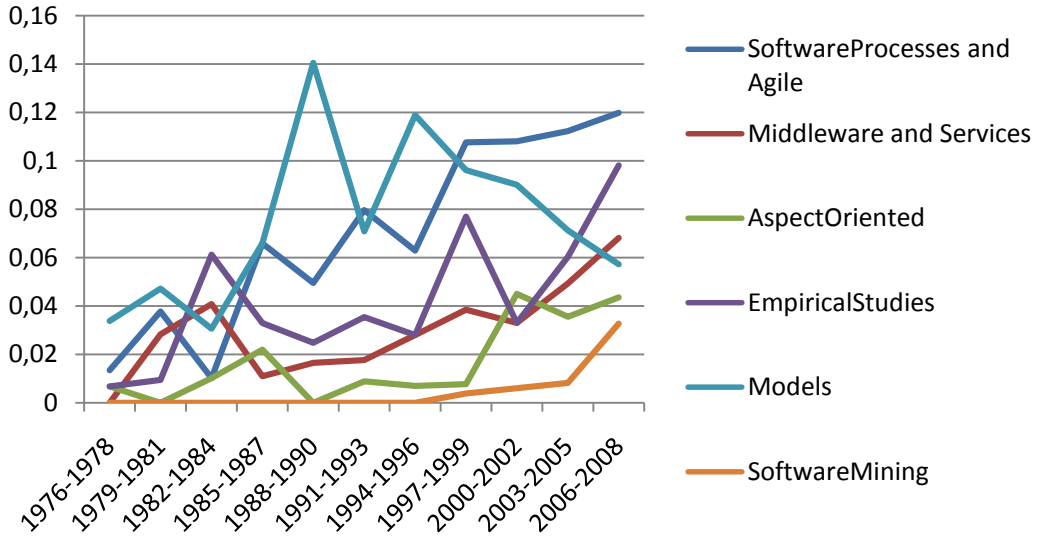

Figure 5.17: Other interesting trends in ICSE

lution of these topics shows that, while they are keeping having a significant role within the community, the interest of the researchers is moving to a much more various set of themes. The introduction of new themes were made possible by the introduction of some new technology, as happened with *Middleware and Services*, or by the spread of some tools and methodology, such as UML for *Models* and Agile for *Software Processes*, or finally by the availability of a wide set of data, like the open source repositories used in *Empirical Studies*. It is really interesting that the charts show quite clearly the beginning of trends in the years corresponding to the introduction of these innovations.
Finally it is important to remark how each venue has its own peculiarities. The analysis performed showed that generally the journals deal more with the theoretical approach while in conferences there is more space left for problems related to the development and the maintenance of software. Even the different journals present some differences in the space left to each topic, but this is mainly related to the minor topics rather than to the more general issues which have a comparable amount of papers published.

The most important result is that the trends obtained with the automatic analysis tools are coherent with what is was expected from the study of the software engineering community, so this test confirms that the application is working properly and gives meaningful results.

5.5 Automatic Bidding

The second application of the tools developed in this work is the recommender system that should help the members of the organizing com-

mittee of a conference to select for the review the papers that best matches with their knowledge. The validation of the tool consisted in two different analysis: one based on the actual bids from ICSM 2010 and the other by asking an opinion to the committee members of ICSE 2011.
Both the test proved to be really useful. In particular the first helped to tune the parameters of the algorithm while the second provided the actual feedback on the quality of the suggestions.

5.5.1 ICSM 2010 - Offline Test

The test performed with the data about the actual bids of ICSM 2011 has been really useful to tune the parameters algorithm but was not able to provide any significant measurement of the quality of the suggestion. This happens mainly because there is no standard way to make a bid but each committee member is free to pick the number of papers he want and does not have to follow any fixed strategy. This freedom left to the committee members turns out in very poorly structured data that are not so good to make a comparison with the results of the algorithm. In particular, the difficulties come from the different number of papers picked and the distribution of the submitted articles between *High* interest, *Low* interest and *Conflict* which does not always follows a clear strategy. Moreover this kind of validation does not take into account the fact that the tools could be able to discover some interesting papers that were not inserted in their bids by the committee members.

This kind of analysis consisted in the computation of the *precision* and *recall* measures. In order to simulate the different number of paper chosen by each committee member, the tool was suggest to get the same number of papers that the user inserted in the actual bid. This attenuates the issue

related to the lack of a well defined structured in the bid. To overcome issues related to the distinction of the reported papers in *High* interest, *Low* interest and *Conflict* it has been decided to aggregate the three values in a single measure. In fact for our purposes what really matter is whether the tool is able to find out interesting papers. Conflicting papers are considered similarly to the interesting papers because usually are written by people in the same research group and therefore if the tool spots one of those papers it is a good sign.

The experiments showed that, as usual, there is a strong conflict between the two measures of precision and recall. Moreover for the not so structured nature of the bid it is quite hard to obtain a bid that exactly resembles the one proposed by the authors. Finally the metrics proposed does not take into account one of the possible advantage that could come from the use of the tool, the discovery of papers that the reviewer missed to place in the bid.

For these reasons the data from ICSM 2010 were used to set up the parameters of the algorithm in order to have a good recall and a fair precision. The focus has been pointed on the recall since that would make it possible to leave more space for the identification of the missed papers. On the other side, a high value recall should ensure the presence of all the papers that were spotted by the reviewer. This is perfectly acceptable remembering that the final purpose of the tool is not to substitute the reviewer in the bidding process, but rather to give them a set of probably interesting papers. For this reason it is far more important to ensure that the highest number possible of interesting articles are retrieved even though that could turn in into some false positives.

5.5.2 ICSE 2011 - Live Test

To overcome the limitations of the test on the data of an actual bid that were highlighted in the ICSM 2010 example, it has been decided to perform the test by asking an opinion about the goodness of the suggestions directly to the committee members. This makes it possible to have an idea of how well the tool performes and also about how useful it could be.
Since the bidding process is really important for the outcome of the conference it has been decided to avoid the risk to bias the committee members with the suggestions produced by an untested tool. For this reason the evaluation of the results of the tool took place after the regular bidding process, in order to not interfere with it. This solution makes it possible to have a meaningful feedback without affecting the regular review process for the conference.

The data gathering method consisted in a survey in which the members of the organizing committee were asked to answer the following questions:

1. Did those recommendations reflect the research profile that we previously provided?

2. Did we recommended papers in your area of expertise or that you would be interested in reviewing?

3. Did those recommendations contain interesting papers that you missed while placing your bids?

4. Would you be satisfied if a conference management system such as Cyber Chair provided you with those recommendations?

5. Would a recommender system like this be beneficial?

Question 1 and 2 are generally focused on the pertinence of the suggestions with respect to the reviewer profile, without referring specifically to the ICSE 2011 bid performed by the researcher. In particular, the first question wants to ensure that the algorithm picks the papers from the right topics accordingly to the author profile, while the second is aimed to discover how good it is the filtering algorithm that has to find out which are the papers that fits the best the expertise of the reviewer. These questions are particularly aimed to avoid that the committee members just compares their bids with the suggestions provided by the tool. This aspect is very important and it is the only viable way to obtain results which are qualitatively different from what it has been observed in the ICSM 2010 test.
The third question aims to discover whether the usage of the tool could turn into better bids since it suggests papers which otherwise would be missed by the committee member.
Questions 4 and 5 want to ask for a direct feedback about the perceived utility of the tool in terms of the benefit it could bring into the bidding process in general and after the integration of one of the software used to manage conferences.

The results for this part of the work are tightly coupled with what it has been described in section 5.3: this application is the natural follow up of the researcher profiling techniques. Unfortunately that coupling means that the problems reported in the previous experiment are going to affect also the result of this experiment. For this reason it is important to look at the reported numbers keeping that in mind for a more critical analysis of the data.
Unfortunately this experiment gathered a lower number of responses than the previous survey. This is probably due to the fact that while the answer

to the survey about the profile just took few minutes, going through all the suggestions and determine whether they were interesting or not is much more time consuming. However the responses were still good and there were collected responses from 23 members of the organizing committee.

The answers to the first question reported that about a half of the researchers found that the suggestions reflected the profile we provided. The other half of the researcher responded that the suggestions matched only partially the proposed profile. The survey did not provide any information about the degree of correctness for the suggestion that partially matches the profiles, so there is no way to know whether that partially should be interpreted as a bad or a good signal. An encouraging result is that no one of the 23 committee members reported that the suggestions did not match the profile generated. The last consideration at least proves that the methodology is aiming in the right direction. Moreover the absence of completely wrong suggestions makes us confident that the partial correctness reported in some responses is the indicator for some minor flaws in the suggestions. The data related to the first question in reported in figure 5.18.

The responses to the second question, which is reported in figure 5.19, is useful to determine the effectiveness of the filtering algorithm in the selection of the papers which best fits the committee member expertise, and the results are encouraging, especially taking into account the answers to the previous question. The data shows that the committee members who reported suggestions pertinent with the profile also observed a set of suggestions that matched well with their expertise. On the other side, the people who reported a partial match with their profile, as it was possible to expect, also answered that the suggestions was composed only partially

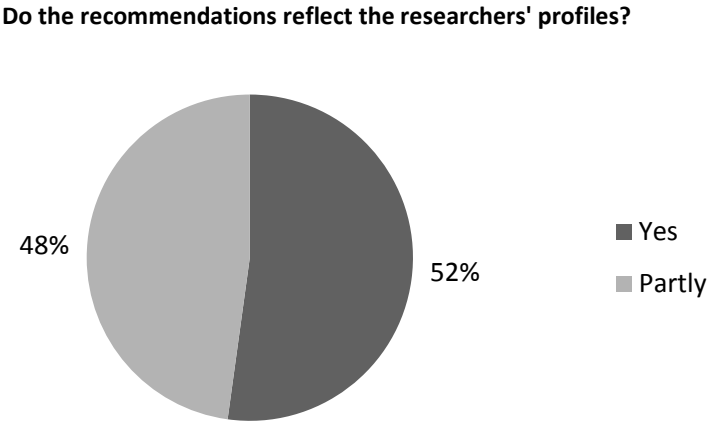

Figure 5.18: Answers to the first question for the ICSE 2011 experiment

by relevant papers and in some cases it did not contained any interesting paper.

These answers show how the presence of interesting papers is tightly coupled with the quality of the profile: if the profile is good then the suggestions are also good. This fact is a good validation for this phase of the automatic bidding process that is able to provide useful suggestions when it is fed with meaningful profiles of the researchers.

The last question that directly targeted the content of the proposed bid is the third, which wants to discover whether the suggestion contained any papers that the reviewer missed to place in his bid. The 13% of the reviewers reported that the tool proposed at least one interesting papers that they missed when they placed their bid. This number is only apparently low but, after considering that the committee members already went through all the submission looking for something interesting, it is clear that, after the reviewers have already spotted the papers they found relevant, there were only a little number of possibly missed papers.

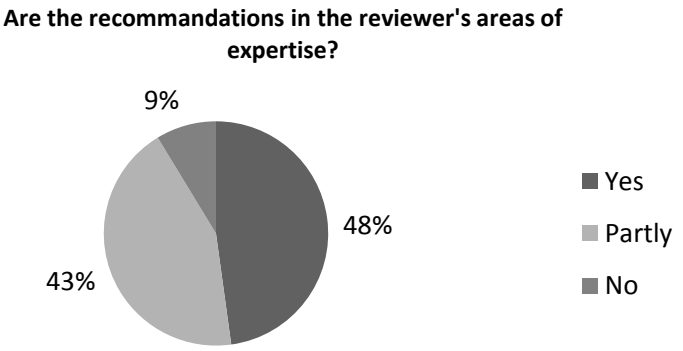

Figure 5.19: Answers to the second question for the ICSE 2011 experiment

The combination of this results with the one obtained from the previous question is a clear signal that, after the required fix in the model that led to the generation of partially correct profiles, the tool could really come in hand in the research of the papers that best match with the expertise of the reviewer. Figure 5.20 reports the answers to the third question.

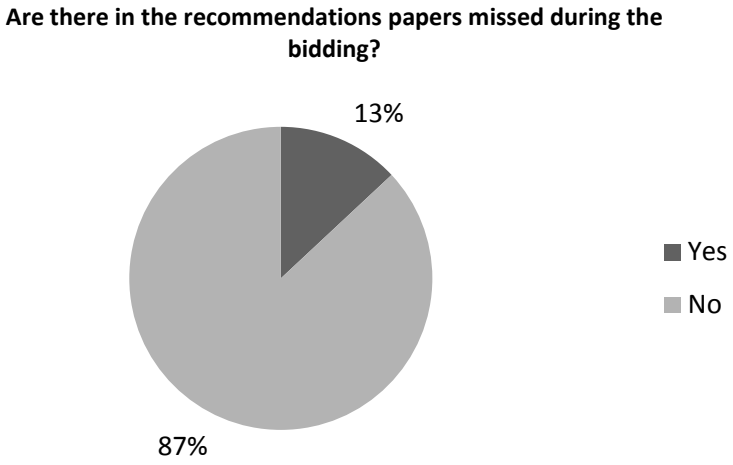

Figure 5.20: Answers to the third question for the ICSE 2011 experiment

The fourth and fifth questions wanted to perceive the general feeling

of the committee members with respect to the tools and the vast majority of them found that it could be beneficial and should be integrated, after the correction of the problems highlighted, in the tools commonly used to manage a conference. The 65% of the committee members who took part in the survey think that an automatic bidding tool could be useful, while the 61% would like to have it integrated with a conference management tool. These data is really encouraging since they show that not only the committee members who had completely satisfying results believe that the tool could lead to beneficial results.

Chapter 6

Conclusion and Future Work

This chapter reviews the results obtained in this thesis work and suggests the directions for further developments and applications that could benefit from the usage of the described tools.

6.1 Conclusions

This thesis work demonstrated the effectiveness of automatic tools in meta-research and bibliometric analysis. In particular it showed that it is possible to develop a data driven methodology able to identify the research topics, to determine what an article is talking about, and finally to generate profiles reporting the interest of researchers. The analytic tools have been validated studying their ability to get meaningful results applying the developed techniques on the software engineering research field. At first a model of the research topics studied by the researchers has been created and after that the paper classifier has been trained, preparing the tools for further and more detailed analysis.

The developed techniques have then been employed in the realization

of two different applications. A first application of the analysis tools is a program to track the evolution of the topics in the software engineering community by looking at the papers published in the literature. In this work were considered two journals and a conference which represent the most important venues of the research field. The results obtained were coherent with what the knowledge of the research field suggested, and the methodology developed proved to be an efficient way to perform the meta-research analysis.
The other case study developed showed that the analysis tools can be effectively adopted in the construction of a recommender system to suggest to the committee members of a conference which are the submitted papers that best matches his profile. This case study showed the feasibility of this approach and demonstrated that it is possible to get meaningful suggestions from the application. The experiment also highlighted the need to have a model as complete as possible in order to be able to produce the most accurate profiles and then to generate suggestions that are actually relevant for the researcher.

6.2 Future Work

A first step that has to be performed to improve the outcome of this work is a refinement of the model, ensuring that it includes a greater number of topics. That could be very beneficial: it is going to positively affect the results of all the applications and it would turn into a significant improvement in the performance of recommender system. That improvement could be very beneficial also for the topics evolution analysis allowing the observation of even more accurate and specific trends.

Another possible evolution is the integration of the analysis tools with other meta-research techniques. For instance, it would be nice to integrate the data about the profiles of the researchers with a co-authorship social network. That could lead to more interesting consideration about how the authors groups together by taking into account they research interests.

It would also be interesting to complete the automatic bidding work by providing an application specifically targeted to the program chair of a conference. He could save a lot of time with an application helping him in the assignment of the papers to the committee members. Such a tool would have to take into account the papers suggested to each researcher and then to find the best assignment possible, associating each paper to the most suitable committee member.

Bibliography

[1] Gianluca Staffiero. Applicazione di metodi per l'analisi di reti sociali alla comunità scientifica di ingegneria del software. Bachelor's thesis, Politecnico di Milano, 2009.

[2] Carlo Ghezzi. Reflections on 40+ years of software engineering research and beyond. an insider's view. ICSE '09 Keynote Speech.

[3] Neal Coulter. Acm's computing classification system reflects changing times. *Commun. ACM*, 40(12):111–112, 1997.

[4] Wei Woon and Stuart Madnick. Asymmetric information distances for automated taxonomy construction. *Knowledge and Information Systems*, 21(1):91–111, October 2009.

[5] Henry Small. Tracking and predicting growth areas in science. *Scientometrics*, 68:595–610, 2006.

[6] A. Saka and M. Igami. Mapping modern science using co-citation analysis. *Information Visualization, 2007. IV '07. 11th International Conference*, pages 453 –458, jul. 2007.

[7] Wangzhong Lu, J. Janssen, E. Milios, N. Japkowicz, and Yongzheng Zhang. Node similarity in the citation graph. *Knowl. Inf. Syst.*, 11(1):105–129, 2006.

[8] Yulan He, Siu Cheung Hui, and Alvis Cheuk M. Fong. Mining a web citation database for document clustering. *Applied Artificial Intelligence*, 16(4):283–302, 2002.

[9] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[10] K. Anuradha and Shalini Urs. Bibliometric indicators of indian research collaboration patterns: A correspondence analysis. *Scientometrics*, 71:179–189, 2007.

[11] Donghua Zhu and Alan L. Porter. Automated extraction and visualization of information for technological intelligence and forecasting. *Technological Forecasting and Social Change*, 69(5):495 – 506, 2002.

[12] Kai-Yuan Cai and David Card. An analysis of research topics in software engineering - 2006. *Journal of Systems and Software*, 81(6):1051 – 1058, 2008. Agile Product Line Engineering.

[13] Guilherme Vale Menezes, Nivio Ziviani, Alberto H.F. Laender, and Virgílio Almeida. A geographical analysis of knowledge production in computer science. pages 1041–1050, 2009.

[14] A. J. Lotka. The frequency distribution of scientific production. *J. Walsh. Acad. Sci.*, 1926.

[15] Sergio Rinaldi, Roberto Cordone, and Renato Casagrandi. Instabilities in creative professions: a minimal model. *Nonlinear dynamics, Psychology and Life Sciences*, 2000.

[16] Stefan Wuchty, Benjamin F. Jones, and Brian Uzzi. The Increasing Dominance of Teams in Production of Knowledge. *Science*, 316(5827):1036–1039, 2007.

[17] Benjamin F. Jones, Stefan Wuchty, and Brian Uzzi. Multi-University Research Teams: Shifting Impact, Geography, and Stratification in Science. *Science*, 322(5905):1259–1262, 2008.

[18] John Whitfield. Collaboration: Group theory. *Nature*, 455(7214):720–723, October 2008.

[19] R. N. Kostoff, D. R. Toothman, H. J. Eberhart, and J. A. Humenik. Text mining using database tomography and bibliometrics: A review. *Technol. Forecast. Soc. Chang.*, 68(3):223–253+, 2001.

[20] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, November 2005.

[21] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152, October 2000.

[22] Duncan J. Watts. Networks, dynamics, and the small-world phenomenon. *American Journal of Sociology*, 105:493–527, 1999.

[23] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[24] Xiaoming Liu, Johan Bollen, Michael L. Nelson, and Herbert Van de Sompel. Co-authorship networks in the digital library research community. *Inf. Process. Manage.*, 41(6):1462–1480, 2005.

[25] Ergin Elmacioglu and Dongwon Lee. On six degrees of separation in dblp-db and more. *SIGMOD Rec.*, 34(2):33–40, 2005.

[26] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3-4):590 – 614, 2002.

[27] T Braun. Growth and trends of fullerene research as reflected in its journal literature. *Chem Rev*, 100(1):23–38, 2000.

[28] Yuan An, Jeannette Janssen, and Evangelos E. Milios. Characterizing and mining the citation graph of the computer science literature. *Knowl. Inf. Syst.*, 6(6):664–678, 2004.

[29] Eugene Garfield. Citation analysis as a tool in journal evaluation. *SCIENCE*, 178(4060):471–479, 1972.

[30] Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[31] H. Peyton Young. *The Diffusion of Innovations in Social Networks*, volume III. Oxford University Press, 2003.

[32] Muhammad A. Ahmad and Ankur Teredesai. Modeling spread of ideas in online social networks. In *AusDM '06: Proceedings of the fifth Australasian conference on Data mining and analystics*, pages 185–190, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[33] Marcio de Miranda Santo, Gilda Massari Coelho, Dalci Maria dos Santos, and Lelio Fellows Filho. Text mining as a valuable tool in fore-

sight exercises: A study on nanotechnology. *Technological Forecasting and Social Change*, 73(8):1013 – 1027, 2006.

[34] Mee-Jean Kim. A bibliometric analysis of the effectiveness of korea's biotechnology stimulation plans, with a comparison with four other asian nations. *Scientometrics*, 72:371–388, 2007.

[35] Tugrul U. Daim, Guillermo Rueda, Hilary Martin, and Pisek Gerdsri. Forecasting emerging technologies: Use of bibliometrics and patent analysis. *Technological Forecasting and Social Change*, 73(8):981 – 1012, 2006.

[36] N. R. Smalheiser. Predicting emerging technologies with the aid of text-based data mining: the micro approach. *Technovation*, 21(10):689 – 693, 2001.

[37] IEEExplore. http://ieeexplore.ieee.org/.

[38] ACM Portal. http://portal.acm.org.

[39] DBLP, Computer Science Bibliography. http://www.informatik.uni-trier.de/ ley/.

[40] Amit Singhal. Modern information retrieval: a brief overview. *BULLETIN OF THE IEEE COMPUTER SOCIETY TECHNICAL COMMITTEE ON DATA ENGINEERING*, 24:2001, 2001.

[41] William B. Frakes and Ricardo Baeza-Yates, editors. *Information retrieval: data structures and algorithms*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[42] Masoud Makrehchi and Mohamed Kamel. Automatic extraction of domain-specific stopwords from labeled documents. In Craig

Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 222–233. Springer Berlin / Heidelberg, 2008.

[43] C.J. van Rijsbergen, S.E. Robertson, and M.F. Porter. *New models in probabilistic information retrieval*. Computer Laboratory, University of Cambridge, 1980.

[44] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[45] Paul Losiewicz, Douglas Oard, and Ronald Kostoff. Textual data mining to support science and technology management. *Journal of Intelligent Information Systems*, 15:99–119, 2000.

[46] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[47] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17(6):734–749, 2005.

[48] Miquel Montaner, Beatriz Lopez, and Josep Lluis de la Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19:285–330, 2003.

[49] Kazunari Sugiyama and Min-Yen Kan. Scholarly paper recommendation via user's recent research interests. In *Proceedings of the 10th annual joint conference on Digital libraries*, JCDL '10, pages 29–38, New York, NY, USA, 2010. ACM.

[50] Apache HttpClient. http://hc.apache.org/.

[51] Hibernate. http://www.hibernate.org/.

[52] Apache PDFBox. http://pdfbox.apache.org/.

[53] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. Chapter 9: Mixture Models and EM.

[54] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[55] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.