UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Expert Finding for Requirements Engineering

Matthieu Vergne

Advisor

Dott. Angelo Susi

Università degli Studi di Trento

April 2016

Abbreviations

We centralize here the principal abbreviations used in this thesis report for an easy reference. This list does not include the abbreviations used in formulae and other highly specific cases, which are only presented at their first use.

EF	Expert Finding
GA	Genetic Algorithm
ICT	Information and Communication Technologies
IR	Information Retrieval
KM	Knowledge Management
MN	Markov Network
OSS	Open Source Software
RE	Requirements Engineering
RS	Recommender System
SE	Software Engineering

Abstract

Requirements Engineering (RE) revolves around requirements, from their discovery to their satisfaction, passing through their formalisation, modification, and traceability with other project artefacts, like preliminary interviews or resulting source codes. Although it is clear for many that involving knowledgeable people is an important aspect of many RE tasks, no proper focus has been given to Expert Finding (EF) systems, leading to have only few related works in the field. Our work attempts to fill this gap by investigating several dimensions of EF: conceptual by analysing the literature about expertise and its evaluation, formal by revising the usual representation of expert rankings, and practical by designing an EF system. As a result, we provide (i) a meta-model grounded in literature from Psychology to identify requirements for EF systems, (ii) a novel formalisation of experts rankings which solves limitations observed in usual EF measures, (iii) two variants of an EF system which builds on usual RE indicators (accessible knowledge and social recognition), and (iv) an enriched evaluation process which investigates deeper the consistency and correctness of an EF system.

Keywords
Expert Finding, Requirements Engineering, Expertise Meta-model, Ranking Formalisation

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Context

In Information and Communication Technologies (ICT), information is a central matter, and managing it correctly is a major concern, whether we speak about obtaining the right information, processing it the right way, or channelling it towards the right destination. Although ICT is often highly technical, it takes its value through the ability people have to improve their own capacity in retrieving and exploiting information through the technologies developed in this field and its sub-fields. One of the most relevant pieces of information is the knowledge people have gathered through their life and experience, an information of particular importance for example when we need to know what should be done to obtain specific outcomes or what should be considered to take specific decisions. This is in the light of this information that we design and implement ICT systems intended to solve problems that people face everyday.

Requirements Engineering (RE) is one of those sub-fields of ICT which focuses on ensuring that a system satisfies the requirements of its *stakeholders*, which include the final users of the system, the people operating it, the people who implement it, and so on. In RE, it is of first importance to identify what are the right requirements to satisfy, and ensure that they lead to proper implementations, which implies to have a deep knowledge about the system, the environment in which it operates or will operate, and who will use it or maintain it. Therefore, many domains are involved depending on the situation and we should ensure that we obtain the right information for each of them, so it should be based on broad knowledge in these domains and free of unmanageable conflicts. One way to obtain such information is to rely on domain experts, which means people having a good experience in at least one of the relevant domains of the system.

Expert Finding (EF) focuses on this task: it looks for people who show the highest levels of expertise in order to recommend them to other people who need them. EF is not always about expertise only, because when you look for experts this is to obtain something from them, so you also want them to be able to provide what you are looking for, not just to *be* an expert. In particular, people may want to obtain information from the expert, so the expert should accept to share her knowledge and be able to convey it in a way people can understand. Experts are also busy, and recommending someone less expert might be acceptable depending on the situation. In this thesis, we focus on the *expertise* aspect common to many situations, while we consider other aspects like availability and will to share as additional dimensions to involve depending on the specific use of the EF system. In particular, RE involves a broad set of tasks with each its own specificities, but one of the common aspects involved in these tasks is that they would find great support by obtaining knowledge from domain experts.

To illustrate how an EF system might help in RE, we can first think about a freshly hired requirements engineer in a company managing an enterprise resource planning (ERP), a kind of software which involves various domains like manufacturing, sales, shipping, salaries, etc. This engineer may be an expert on requirements but lack the relevant expertise in the different domains the software is about, and as such he needs to find out who are the knowledgeable people to involve to understand better the requirements of the ERP. Nowadays, such an engineer would need to go around and ask to people of the company who to involve, which might work well in small companies but becomes harder when we need to decide from hundreds of people, even more if they are geographically distributed. In such a context, the requirements engineer would have to trust on his colleagues, who might prefer to recommend a good friend or simply someone they know better rather than thinking about the most knowledgeable people in the company. Such limitations are worsen when extending the boundaries out of the company: in open source communities, people join and leave, often anonymously, which makes it infeasible to know enough about everyone to recommend the most knowledgeable people. Some people might provide breakthrough ideas and highly relevant comments and only need the opportunity to do so, which is what the requirements engineer can exploit if he uses some RE tools integrating expert recommendation features. Not only he could complement the recommendations of his colleagues, but he could obtain on-demand, customised support to identify the right people to involve, while reducing the time spent in searching for them.

1.2 Problem

The problem we identified through our analysis of the state of the art, which we detail in Chapter 3, is that the EF task, although it has been recognised as a relevant and important one, has been poorly addressed in RE. Indeed, we could relate only two main works (with their various publications) to a goal similar to identifying experts for helping in RE tasks, and actually with a focus on the requirements elicitation only. More explicitly, [Castro-Herrera and Cleland-Huang, 2009] focus on the knowledge and personal interests of the stakeholders, while [Lim et al., 2010] target powerful or influencing people. If expertise does have an effect in each case, other aspects are also clearly involved: one can be interested in domains which are far from his main expertise, and be powerful or have influence because of having money or a strategical position. In this thesis, we focus on the expertise aspect only, which consequently covers a broader set of RE tasks, but is also more restricted regarding its direct application, because each task need to consider other aspects too.

1.3 Contributions

Because we intend to solve the problem of the poverty of support for finding experts in RE, we need to enrich it with new approaches. Our research approach first focused on inferring informational capability of experts, before to try to understand better what identifies those experts, which lead us to investigate deeper what is expertise and how to evaluate it. Consequently, our work does not target *only* the EF process, but also contributions to design and validate such processes based on what expertise is and how it can be evaluated, thus covering both the fundamental and empirical aspects of the EF task. Consequently, and with some humour, we may say that another relevant title for this thesis could have been the reversed *Requirements Engineering for Expert Finding*, but it would neglect the fact that we designed and refined our contributions mainly within a RE context. In the following, we give a summary of each main contribution of this thesis.

The contribution which focuses the most on both EF and RE aspects is described in Chapter 6, where we inspire from those two existing RE works to design a novel approach combining them. This approach considers at the same time indicators about the knowledge of the stakeholders, in particular the one provided in their messages like [Castro-Herrera and Cleland-Huang, 2009], as well as their recognised roles, like [Lim et al., 2010]. This combination of indicators, rather than the combination of the complete approaches, allows us to be more comprehensive by considering also the relations between the two aspects. We design one version based on Markov Networks (MNs), for computing the probability for each stakeholder to be an expert, as well as a version based on a Genetic Algorithm (GA), which exploits the advantages of optimization techniques to fix some issues observed with the MN version. The MN approach has lead to several publications [Vergne et al., 2013, Morales-Ramirez et al., 2014, Vergne and Susi, 2014] while the GA approach gave us the opportunity to participate in the improvement of an existing, state of the art Java library [Nebro et al., 2015].

In order to design our approach, we need first to find out what are the important elements to consider in each RE work, which implies to have a broad understanding of the concept of expert and what it relates to. If we dedicate two sections of the state of the art (2.1 and 2.2) to experts and how to find them, we also contribute to this state of the art by providing a meta-model relating all these concepts together in Chapter 4. This meta-model relates not only practical concepts like the outcomes produced by a performer and used by an evaluator to infer the performer's expertise, but also more abstract aspects like the knowledge and skills of the expertise or the absolute/relative aspect of the evaluation. This model allows us to highlight which elements of the two existing works should be exploited in our own approach, but also to analyse existing EF solutions to identify what they cover and what could be investigated for improvements. An excerpt of this meta-model and some coverage analyses have been published in [Vergne and Susi, 2015].

If having a sounding approach due to the consideration of a broad literature is nice, having an approach which provides sounding results is even better, which means that a particular care should be provided to the evaluation of our approach, which is why we dedicate it the whole Part III.

Usual validation methods for EF approaches focus on comparing the rankings produced by their approach to some gold standards built using other methods, but often by taking the correctness of these “gold standard” rankings as granted. Not only we can argue the validity of these gold standards depending on the situation, but there is also other issues that we faced but for which we found no concrete help. Consequently, we provide in Chapter 7 a detailed and systematic evaluation process, which extends the single gold standard validation with three other assumptions to satisfy, covering correctness as well as consistency, and which adds a preliminary phase of stability evaluation to ensure that the generated results can be safely exploited. This evaluation process is used on three different contexts to evaluate our EF approaches and described in Chapter 8. One of these contexts exploits the broad dataset of an existing project, which required to design a procedure for building its gold standard, described in [Vergne, 2016a].

This revision and extension of the evaluation procedure has been possible because of a last contribution of this thesis, which is our revision of the formalism of a ranking of experts in Chapter 5. Experts rankings are usually compared to the documents rankings of the Information Retrieval (IR) field, leading to use usual measures designed for the latter. However, we saw that some aspects are not thoroughly covered for documents rankings, like having rankings which are partially ordered or incomplete, leading to design dedicated measures to mitigate these issues. Additionally, usual ranking conventions happen to be problematic for rankings of experts, like the fact to consider two people at the same rank as equal: if it can be true for races, or even for documents because the total access to their content allows to confirm proper equality, the expertise of two persons is far to be that explicit, leading to prefer an assumption of lack of information, meaning of inability to order them. This shift of interpretation has an impact on the comparison of rankings, and our contribution revises this formalism and reuses IR basic measures (precision and recall) in a novel way, showing that we can fully consider these properties rather than making mitigation procedures for when we face them. A preliminary analysis of existing IR metrics has been done in [Vergne, 2016b], which provides examples of mitigation procedures to adapt these metrics to our interpretation.

1.4 Structure of the Thesis

In order to simplify the reading, this thesis has been organised into three main parts.

Part I focuses on the current state of the art and is separated into two chapters. Chapter 2 describes the literature of interest for our work, which involves fundamental notions related to experts and expertise, as well as a description of the EF task and the RE field in which we work. Then, Chapter 3 focuses on the gap we found in this RE field in regard to EF, and what are the research questions which motivated us for working on it.

Part II is broader and presents three of our main contributions. Our meta-model of expertise, which centralises what we learned from the literature about expertise, is presented in Chapter 4, and we use it in different manners to show how it can support designers of EF systems. With a more formal perspective, Chapter 5 introduces our novel formalisation of experts ranking, providing revised definitions as well as measures to compare rankings, including for compliance purpose. The part closes on our EF approaches, detailed in Chapter 6, which consist in extracting relevant data to build a weighted graph, and exploit this graph to infer who are the most experts based on the user's query, with two different inference techniques (MN and GA).

Part III focuses on the evaluation of our EF system. As such, it provides another main contribution, which is a systematic evaluation process using the measures designed through our formalisation of experts rankings. This evaluation process, presented in Chapter 7, adds to the usual query-specific gold standard three other criteria to satisfy, which are able to stress the consistency and correctness of the EF system without depending on costly and hard to validate gold standards. The following Chapter 8 is dedicated to the application of this process in three different contexts: one focuses on the formal aspect by evaluating our approach with synthetic data, the next one introduces some noise by using real data generated in a controlled environment, and the last one is based on a completely open situation by using the archives of an Open Source Software (OSS) project.

Finally, we close this thesis on Part IV, which gives a summary of our contributions and the answers to our research questions, and lists some future works that we think to be among the most relevant.

Part I

Literature

Chapter 2

State of the Art

Through this chapter, we present the different works we considered to identify and understand the problem we want to tackle and the main concepts and techniques we could build on. This state of the art has been composed in a rather repetitive manner through successive backward snowballing processes, similarly to [Wohlin, 2014]. Shortly, once some literature has been read about RE to identify an interesting problem to tackle, additional references have often been identified by (i) facing a specific issue, (ii) searching for works dealing with this specific issue, and (iii) looking iteratively at relevant cited works until the issue appears as being solved or solvable. Consequently, we have first been interested in a better involvement of domain experts in RE, before to search for more insights on the EF task in particular, which lead us to consider even deeper the core notion of expertise and how it is evaluated. It is particularly worth noting that, when the issue was to define or understand a particular concept, such a process lead us to give preference to books and other works providing a dedicated chapter to introduce core concepts, giving us a proper grounding to rely on.

Consequently, we can now present to the reader the basics about expertise, how is it built, and how is it evaluated in Section 2.1. We then build on it to present the techniques used for finding experts in Section 2.2, which involves several fields having each its own perspective. Finally, we describe the main field of our work in Section 2.3, thus RE, and show why the EF task is of high relevance. We hope that, by ordering this chapter with our main field last, we facilitate the understanding of the problem we tackle by providing to the reader, in a progressive manner, all the relevant knowledge that we also needed.

2.1 Experts and Expertise

In this section, we focus on the basis of expertise: what is it, how is it built, and how is it evaluated. As such, we first look at existing definitions in Section 2.1.1 to clarify what we are speaking about. Then, we rely on references in the literature about expertise in Section 2.1.2 to understand how people improve their performances, from the mere amateur to the top expert. Rich of this knowledge, we finally investigate how the achieved level of expertise can be assessed in Section 2.1.3, highlighting also in which aspects experts do not perform well and which properties should be investigated to properly assess their expertise.

2.1.1 Definitions

In this section, we look at the definitions of the two main concepts which are *expert* and *expertise*, in order to have a robust basis for our work by avoiding misleading personal interpretations. In particular, we estimate that two complementary perspectives need to be taken to ensure that we cover these terms in the most robust way: popular as well as scientific. The *popular definitions* allow to rely on a broad understanding of the terms, which means that we maximize the understandability and generalizability of our work by building on a common ground. The *scientific definitions*, on the other hand, are usually the most refined definitions providing the key properties to consider for having a reliable work. Consequently we retrieve, in the following, definitions from both popular dictionaries and reference literature in expertise and expert performance in order to establish the basis on which we build our work.

Expert

By looking at different dictionary definitions of *expert*, we can identify a broad agreement on the concept. The Collins[1] and Oxford Dictionaries[2] speak similarly about "a person who has extensive skill or knowledge in a particular field". The Cambridge Dictionaries[3] are a bit more precise with "a person with a high level of knowledge or skill relating to a particular subject or activity". The Merriam-Webster Dictionary[4] is even more precise with "having or showing special skill or knowledge because of what you have been taught or what you have experienced". This last one is of particular interest because it shows better two perspectives: when one *has* special skills or knowledge, whether people assess them or not, and when one *shows* special skills or knowledge, whether he actually has them or not. Because we are focusing on EF, we focus mainly on what is *shown* by these people, which is the basis for us to evaluate their expertise.

For a scientific perspective, through the Cambridge Handbook of Expertise and Expert Performance, [Ericsson, 2006a] investigates more deeply the expert identification by pointing three criteria (p. 14): a "lengthy, domain-related experience", supporting the presence of *extended* knowledge and skills, like how long someone have worked in a given field ; a "reproducibly superior performance", supporting the presence of *advanced and consolidated* knowledge and skills, like a manager able to react particularly quickly and efficiently

[1] Collins dictionary: http://www.collinsdictionary.com/

[2] Oxford Dictionaries: http://www.oxforddictionaries.com/

[3] Cambridge Dictionaries: http://dictionary.cambridge.org/

[4] Merriam-Webster Dictionary: http://www.merriam-webster.com/

to problematic situations ; and a "social criteria", based on evidences arising from a community of people, like an obtained degree or the endorsement functionality we can find on online social networks, like LinkedIn[5]. While the popular definitions identify the need to consider both knowledge and skill, the scientific literature gives us these three assessment notions: quantitative, qualitative, and socially recognized.

Expertise

Similarly, we can inspect the definitions of *expertise*. While the Collins Dictionaries speak about a "special skill, knowledge, or judgment", the Cambridge Dictionaries focus on the "high level of knowledge or skill", adding the *level* dimension. In other words, we can speak about expertise as the specific knowledge and skills which compose it, or as some level of performance achieved based on them. The Oxford Dictionaries rely on the *expert* definition by describing an "expert skill or knowledge in a particular field", as well as the Merriam-Webster Dictionary with "the skill or knowledge an expert has". With a thorough thinking, we can see different interpretations: the expertise required to achieve an expert level (domain-specific), or the actual skills and knowledge of someone independently of their level (performer-specific). Because in our work we intend to evaluate the expertise of people, we favour the individual performance rather than the domain level, which allows us to work with a relative perspective, as described later.

On the scientific side, [Sonnentag et al., 2006] differentiate two perspectives on the notion of *expertise* (p. 375): "years of experience" and "high performance", which can be easily related to, respectively, the lengthy, domain-related experience and the reproducibly superior performance of [Ericsson, 2006a]. They also highlight that even people having less practice can have more expertise due to other factors: [Ericsson et al., 1993] show for example

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

how the motivation, leading to deliberate practice, is one of these factors. Going even deeper into the mind of experts, [Bédard and Chi, 1992] mention that they better structure their knowledge, allowing them to identify better solutions faster. However, for our scope, we focus on tasks where we can evaluate the expertise of potentially huge groups of people, which means that we probably cannot afford something as detailed as measuring how the knowledge of each of them is organised. However, already at this stage, we can mention that there is matter to go deeper in the analysis to strengthen the design of a fully featured EF system.

2.1.2 Expertise Building

Independently of the definitions, we are also interested in processes, in particular how people build their own expertise, in order to find what are the relevant indicators to consider. [Ericsson, 2006b] summarizes a broad literature on this purpose. In particular, an “acceptable level of proficiency” requires some months of experience during which the performer will focus on the actions to perform while avoiding gross mistakes, like in school or any other training course. A “stable, average level of performance” is then necessary to perform in an autonomous way, and requires often several years, what we called a *lengthy, domain-related experience*, to become fluent in the domain-relevant activities. However, [Ericsson et al., 1993] highlight that what differentiates the average professional, who maintains his level by executing routine work, from the domain expert (or master) is the continuation of “deliberate practice” to fix weaknesses.

If an average performer simply continues to perform without aiming for improvements, behaviours became automatic, leading to loose the awareness of how to perform in favour of reflexes. This loss of awareness could explain the observation of [Chi, 2006] (p. 24) that experts often cannot articulate their knowledge because it is tacit. If the performer seeks for improvements,

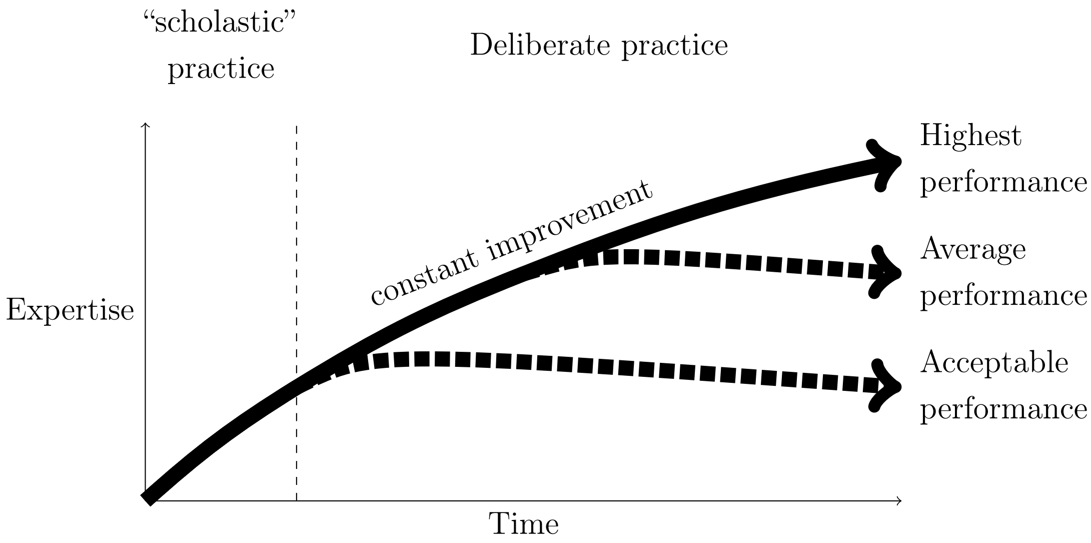


Figure 2.1: How the expertise is built over time, with the improvement phase in solid line and the automation phase in dotted line. Each arrow corresponds to a different time from which the performer stops seeking for improvements through deliberate practice.

deliberate activities take place by concentrating on the task, having a personal will to improve, performing on non-mastered cases, comparing to references, and searching for explanation to refine mental representations. At some points, the performer could give up in this constant effort for improvement, leading to automation of routine work and stagnation. This process is illustrated in Figure 2.1, adapted from [Ericsson, 2006b] (Figure 38.1, p. 685). The interested readers can also look in [Ericsson, 2006a] for dedicated chapters about deliberate practice in different domains, like in chapters 14, 39, 40 and 42.

2.1.3 Expertise Evaluation

Once we have understood how expertise can be built, it is even more relevant for our work, which aims at recommending expert people, to know how to evaluate this expertise. For this purpose, we can consider the review of [Chi, 2006] which presents the two main approaches used to study expertise: absolute and relative. The *absolute approach*, on one hand, studies exceptional people to understand how they perform, in order to identify the properties which pertain to the top experts. The advantage of the absolute approach is to identify key properties to strengthen in order to reach the top, but there could also have innate capacities and nothing says that the methods applied by known top experts are the only methods able to improve expertise. The *relative approach*, on the other hand, focuses on distinguishing people within a common, domain-related group, in order to identify what can be provided to the less experts to reach the level of the more experts. It complements the absolute approach by identifying iteratively the lacks to fix to obtain a higher level of expertise, although it does not claim to support further increase than the most expert of the group.

[Chi, 2006] also summarizes the *properties* which seem to characterize experts, who excel for example by generating better solutions faster, perceiving deep features, identifying lacks and errors, and managing better their resources (e.g. skill, knowledge, sources of information). Excepted these properties, which are usually expected from experts, experts also spend significant time in qualitative analysis of the problem to represent it with domain-specific and domain-independent constraints. They also have more facility to apply forward analysis (find the rules applicable to the current data, independently of the final goal) while non experts rely exclusively on backward analysis (check and refine hypothesis based on the final goal). However, [Chi, 2006] also highlights that the excellence of experts decreases with less context information, when it is not aggravated by overlooked details. Experts also tend to fail in having similar excellence in different domains, especially due to the bias inculcated by their actual domain of expertise and a potential over-confidence in their own abilities. Finally, experts also tend to fail in judging non-expert abilities, thinking that some tasks requiring a reasonable amount of expertise can be achieved by pure novices.

While many indicators have been considered in the literature about expertise, [Ericsson, 2006b] notices that people evaluating the expertise of a performer often rely on simple experience-based indicators, which do not help in finding the highest experts. In these *good but not best* indicators, we can find the length of experience in the domain, the accumulated accessible knowledge, the completed education and the social reputation. In order to identify the highest experts, one need to look at *reproducibly superior performance* on representative, authentic tasks which require domain-specific experience, like a chess master should find the best move on a chess board already set up [Sonnentag et al., 2006, Ericsson, 2006b].

2.2 The Expert Finding Task

Once we know how expertise can be built and evaluated, it is worth looking at the techniques used for finding experts in different situations. However, EF did not come as a properly identified field, but rather like a specific task going across several existing fields. In the following, we show how the difficulties of managing “experts’ knowledge” in Knowledge Management (KM) lead to focus on retrieving experts themselves in Section 2.2.1. Then, how the need to automatise the profiling of these experts has lead to consider IR techniques in Section 2.2.2. Additionally, the need to recommend experts came also with the need to consider the particular context of the information seeker, which lead to inspire from Recommender Systems (RSs) as shown in Section 2.2.3. Finally, we give some concrete examples of EF systems in Section 2.2.4, showing how the different expertise criteria described in the previous section (skill, knowledge, and social) have been exploited.

2.2.1 Knowledge Management & Expertise Location

Following [Groff and Jones, 2012], KM can be defined as the set of tools, techniques, and strategies to retain, analyse, organise, improve, and share business expertise. From [Marwick, 2001], we get a broader definition, as the set of systematic and disciplined actions that an organization can take to obtain the greatest value from the knowledge available to it. In other words, we may summarise the aim of KM as the maximisation of profit –financial or other– made on the knowledge available within a group of people. The notion of "knowledge" can be defined strictly based on mind artefacts, so the experience and understanding of people [Groff and Jones, 2012], or rather loosely by adding also external artefacts, such as documents and reports [Marwick, 2001], although it is more common to consider them as information or data. KM has been established and developed in the 1990s, with early approaches mainly focusing on how to unify disparate databases of the organization into a data warehouse that can be easily mined, like yellow pages, expert locator systems, or expertise management systems [Balog, 2012].

A particular focus was given to what is called *tacit knowledge*, as opposed to the explicit knowledge obtained from documents, which lies in the beliefs, perspectives, and values of people, and tend to be hard to put in words [Groff and Jones, 2012]. As we saw in Section 2.1.2, this is a primary component of expertise once habits have been developed to make a task become a routine work, and one of the goals of KM is to make this tacit knowledge explicit [Groff and Jones, 2012]. A first way was by storing the relevant knowledge (or information) into knowledge bases that people in the organisation can use to retrieve what they are interested in. However, not only it might be hard to retrieve exactly what is needed, but these knowledge bases require to be kept up to date, which imposes a significant maintenance effort hard to guarantee [Marwick, 2001]. Consequently, the focus transferred from the knowledge itself to its container, trying to retrieve people who know rather than directly what they know, called *expertise location*.

Consequently, expertise location aims at identifying who are the people having some relevant expertise, based on the needs of the expertise seeker, whether it is for working directly on some tasks or providing topic-specific information. [Balog, 2012] mentions that early systems often relied on employees to manually judge their skills against a predefined set of keywords, which was, again, a laborious and time-consuming process, with stored data being soon obsolete. These drawbacks have motivated the need for automation, involving the field of IR to find automatically experts based on text corpora.

2.2.2 Information Retrieval & Expertise Retrieval

[Manning et al., 2008] define IR as finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). As such, at the opposite of KM which focuses on the knowledge of *people*, IR focuses on the information stored in *documents*, which lies on fairly different constraints. [Manning et al., 2008] mention, in particular, how the clear preference to rely on people to obtain information in the past, the core of KM, has been reversed to rely today on computers to find the relevant information, the core of IR, due to the significant increase in computation power. IR covers different kinds of information-related tasks, like finding a relevant document or piece of information, filtering the whole set to keep only the most relevant, or clustering in order to arrange them by similarity. All these tasks can be considered from the individual scale, like spam management and e-mail retrieval, to the web scale, which implies to retrieve few items from billions of them efficiently. Like KM, IR also deals with the organization scale, in order to retrieve documents or pieces of information from the material of the organization.

Like KM aims at retrieving available knowledge, IR aims at retrieving information, but they differ in the "container" from which they need to retrieve it [Balog, 2012]. Because IR is dealing with documents, nothing is "tacit": if a relevant information is present, then it lies in the sentences of a report, in the relations of a database, or in the structure of a program available in the store. As such, IR focuses on exploiting at best the available material to obtain the information, which implies also to retrieve information about people, like who has written what and how people are described within available documents. By retrieving such information, it is then possible to link people to information about them, in particular information about their expertise, what we call *expertise retrieval*.

[Balog, 2012] defines *expertise retrieval* as linking humans to expertise areas, which involves two directions. *Expert finding*, on one hand, aims at retrieving people for queried topics of expertise, thus answering questions like "Who is an expert in X?". *Expert profiling*, on the other hand, aims at retrieving topics of expertise for queried people, thus answering questions like "What are the areas of expertise of the person X?". Due to the document-centred approach of IR, expertise retrieval has first focused on content-based approaches, which means approaches focusing on the content of the documents to infer its relevance, and thus the relevance of its authors. Progressively, the human-centred perspective of KM has also been considered, giving room to factors like physical proximity, work overload, or reliability of the expert [Hofmann et al., 2010]. After this additional focus on the personal properties of the expert, logically follows the consideration of the personal properties of the expert seeker, a usual perspective in RSs.

2.2.3 Recommender Systems & Expert Recommendations

[Ricci et al., 2011] define RSs as software tools and techniques providing suggestions for items to be of use to a user. [Felfernig and Burke, 2008] give even more focus on the *personalized* aspect of these recommendations, which also have to be selected from a *large space* of possible options. Today, RSs are widely applied in daily tasks and we retrieve them in common places, like Amazon recommends products and Yahoo Answers recommends questions which might interest the user. RSs have been also considered in more technical contexts, like in Software Engineering where we can use them to obtain pieces of code to reuse, or to write effective bug reports [Robillard et al., 2010]. As developers, we are ourselves used to the auto-completion features of our Integrated Development Environment (IDE) and to the automatic writing of patterns like `for` and `while` loops, all being based on RSs.

[Adomavicius and Tuzhilin, 2005] and [Felfernig and Burke, 2008] provide usual classifications of RSs, including the types of recommendation techniques that we summarize below. With a *content-based* RS, the user is recommended items similar to the ones the user usually prefers, in other words we compare directly the properties of the items to compute their relevance. With a *collaborative* RS, the user is recommended items that people with similar preferences like, so we compare the users to infer indirect evidences of relevance for each item. In the case where we use different sources of information than the ones above, like user requirements (used as complex queries) and knowledge about the domain, we speak about *knowledge-based* RSs. To the best of our knowledge, these categories are the most commonly referred in the literature, although we might find additional ones like demographic (deal with niches) and community-based (focus on the preferences of friends), as mentioned by [Ricci et al., 2011]. Finally, any combination of the techniques previously described fall in the category of *hybrid* RSs.

In the context of EF, it is important to remember that items are also people, so we need to differentiate the *user*, who is using the RS, and the *experts*, who can be recommended to the user by the RS. A content-based technique, in such a context, would focus on various expertise evidences found in the outcomes of the experts and in documents describing them in order to compute their relevance, like [McDonald and Ackerman, 2000] identify who modified a specific piece of code or who has solved a similar problem to know who has more expertise on it. A collaborative technique, on the other hand, would focus on how other users perceive or interact with the experts, thus giving a more social aspect to the RS, like [Spaeth and Desmarais, 2013] relies on who has read an expert profile, who has sent messages to her, and who has met her. Although we did not find any example to illustrate it for experts, a knowledge-based technique might exploit specific properties that the user is looking for or properties which seem particularly relevant for the specific domain targeted, like focusing on source code for developers.

2.2.4 Existing Expert Finding Systems

As we saw through the previous sections, rather than being a field itself, EF is a task transversal to several fields, each providing its own perspective on it. With KM, we take an expert-centred perspective, looking at what people provide and how they perform in the organisation to evaluate their expertise. With IR techniques, we are more document-centred, giving more importance to the information stated in documents and other materials. With RSs, we finally take the user-centred perspective, with a better consideration of user-specific properties to better design the final recommendation. Consequently, various approaches have been designed and, to not favour any specific perspective, we give examples of works classified by type of expertise evidences they rely on, as described in Section 2.1. However, as we show later in Section 4.5.2, it is not trivial to identify whether or not an approach identifies *lengthy domain-related performance* or *reproducibly superior performance.* So we classify these techniques based on *skills*, *knowledge*, and *social recognition.*

Skill-based An approach based on the skills of the performer should rely on domain-specific activities, like programming activities in Software Engineering. This is what [Mockus and Herbsleb, 2002] analyse by looking at the amount of code written in a piece of a software to identify knowledgeable programmers. They rank programmers relatively to the number of changes they made on the source code, possibly restricting the counting to a given period of time. The advantage of such an approach is to build an evaluation based on reliable evidences of performance, but it requires to be highly domain-specific to identify them.

Knowledge-based In order to be more generic, one can try to see how the apparent knowledge of a person corresponds to the knowledge expected from an expert in the domain. For instance, [Serdyukov and Hiemstra, 2008] analyse the content of many documents to identify the contributions of their different authors, which helps in identifying their potential knowledge. They compute the probability that a given document or a given term relates to a given author and, when looking for experts related to a specific term, sum up the corresponding probabilities to rank the authors correspondingly. There is many approaches using this kind of representation, whether they build a single textual representation for each expert (candidate models) or find the relevant documents before to look for the experts related to them (document models). Rich reviews on these particular techniques are provided by [Balog, 2008, Balog, 2012] and appear to be particularly efficient in heavily textual environments, like the Web.

Recognition-based For situations where we lack in documents to retrieve the information from, we can rely on other techniques based on recognition, especially through the exploitation of social interactions. [Zhang et al., 2007], for instance, look at question-answer forums in an online community to identify people seeking and providing knowledge. At the opposite of the previous works, they do not look at *what* is said, but *who replies to who*, leading to recognise people having more knowledge than others. In their work, they compare several algorithms to rank people, like using the number of answers as evidences of expertise, and the number of questions as counter-evidences. They also use a PageRank-like algorithm, which propagates these values over the community so that people answering questions from experts are themselves considered as more experts, which reinforces the social aspect of this approach.

Hybrid While each of these works illustrates well its own kind, other EF systems have been designed to combine several of these aspects. For instance, [Karimzadehgan et al., 2009] exploit the content of e-mails of employees, to retrieve their potential knowledge, as well as hierarchical relations, so a social aspect, to smooth their values. This “knowledge-social” combination seems rather popular, leading to further works like exploiting “following” relations found in public social networks [Bozzon et al., 2013], using various relations including questions-answers [Liu et al., 2013], or decreasing redundancy by identifying synonyms [Omidvar et al., 2014]. Although they appear to be rare, we can also find “skill-knowledge” works, like [Vivacqua, 1999] which retrieves the particular classes and methods used by a programmer by looking at the source codes, thus showing that the programmer *knows* about them, but also that he *is able to use* them through frequency and quality metrics. Many other EF systems exist in various domains, and the reader can refer to more comprehensive works about EF to have a broader list [Maybury, 2006, Balog, 2012].

2.3 Expert Finding in Requirements Engineering

Once the notions of expertise and EF have been investigated, we have to consider the field in which we plan to use them. We target here the field of Requirements Engineering (RE), where requirements engineers, the users we target, aim at discovering and managing the requirements or specifications of a project. We give in Section 2.3.1 a thorough description of the field of RE and how domain experts could be helpful. Then, in Section 2.3.2, we investigate the existing approaches which can be related to the EF task for RE purpose. This is from these works that we plan to build on to improve the support of RE processes.

2.3.1 Requirements Engineering

The [IEEE Standards Board, 1990] has standardised the definition of *requirement* as:

1. A condition or capacity needed by a user to solve a problem or achieve an objective.

2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

3. A documented representation of a condition or capability as in (1) or (2).

As highlighted by [Loucopoulos and Karakostas, 1995], although it was written with a perspective of *software* requirement (the definition of *requirements phase* is explicitly linked to a software product), the definitions above can be applied to non-software systems as well. RE, by extension, is the set of activities revolving around these requirements, which has been stated in a detailed way by [Zave, 1997] as *the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.* Once again, we might argue the reduction to Software Engineering (SE) only, although it can be motivated by the aim of dealing with requirements in a scalable way, leading to use ICT tools, which often involve software components. People interested in a more generic definition might consider the one from [Pohl, 1994] who define RE as *the systematic process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained.*

It is important to us to mention both these definitions because they have another fundamental difference, similarly to [IEEE Standards Board, 1990] which defines a requirement with a user-centred perspective (1) and a system-centred one (2). [Zave, 1997] takes the system-centred perspective by providing a definition of RE which focuses on the software system, its behaviours, and its specification, while the people involved are reduced to providing *real-world goals*. [Pohl, 1994], on the other hand, take the user-centred perspective by stating that it is a *co-operative* process, which requires to analyse and document *observations* and check the *understanding* gained. This difference is important to us, because [Zave, 1997] seems to be more cited (551 citations identified by Google Scholar[6]) than the original work of [Pohl, 1994] (318 citations, 187 more with its re-publication in 2013[7]). On a personal basis, we

[6] [Zave, 1997] on Google Scholar: https://scholar.google.com/scholar?q=Classification+of+research+efforts+in+requirements+engineering

[7] [Pohl, 1994] on Google Scholar: https://scholar.google.com/scholar?q=The+three+dimension

also heard and read repeatedly about RE colleagues based on Zave's works, or on [Nuseibeh and Easterbrook, 2000] who reused his definition. We think that –and this is supported by the citations of the re-publication– it is of particular importance to give back proper focus to the people involved in the RE process, which is an obvious claim if we plan to deal with people-centred tasks like EF.

More than a personal opinion, [Dutoit and Paech, 2003] highlight that a lot of knowledge need to be retrieved to properly achieve RE tasks. One kind is the "application domain knowledge not accessible to developers", which allows to understand why specific requirements are considered but not others, and also why some are more important than others. A second kind of important knowledge is the "solution domain knowledge not accessible to the client", which is about understanding the trade-off enforced by external constraints, like cost limitations. The third one is the knowledge about "relationships between the requirements and the design of existing systems", in particular to understand current systems well enough to minimize the impact of changes due to the evolution of requirements. They also highlight that obtaining this knowledge is costly and difficult, and that we should focus on the relevant parts only, which is not trivial either.

These issues lead [Cheng and Atlee, 2009] to state that RE involves extensive human interaction, while [Maalej and Thurimella, 2009] went as far as calling for the need to "recommend experts" to help in RE tasks. We cannot agree more with them because, as we saw in Section 2.1.3, experts spend significant time in qualitative analysis of the problem to represent it with domain-specific and domain-independent constraints. They also excel in perceiving deep features, identifying lacks and errors, and managing better their resources [Chi, 2006]. If a requirements engineer want to obtain the knowledge described by [Dutoit and Paech, 2003] easily and with minimal

s+of+requirements+engineering:+A+framework+and+its+applications cost, by focusing on the relevant parts only, experts appear to be among the most suited sources to consider.

2.3.2 Stakeholders Recommendations

Once we know what is RE and why recommending experts can be relevant, it is interesting to know what has been done so far to support this aspect. [Mohebzada et al., 2012] provide us a good reference through their systematic literature review of recommender systems applied in RE tasks. They have identified 23 works among which 5 were dealing with stakeholder recommendations, which includes –but is not limited to– EF. Unfortunately, these 5 works concern only 2 series: one is on recommending topics in a forum to stakeholders who could be interested, and the other is about identifying core stakeholders to involve for establishing the list of requirements. We describe in more details these two works to understand better their contribution to RE and how they can be related to EF.

The first approach comes from [Castro-Herrera and Cleland-Huang, 2009, Castro-Herrera and Cleland-Huang, 2010], where the participation of stakeholders in a forum is exploited to evaluate their knowledge on relevant domains. Since several threads can be related to the same domain or one thread can mix several of them, they cluster the messages by domain, or *topic*, depending on their common *terms*, which results in generating abstract topics represented as vectors of terms. Consequently, the stakeholders are related to some topics depending on the content of their messages in all the threads of the forum. The result is for some stakeholders to be recommended to participate in a new thread by identifying a high topic similarity with previous threads. From the EF point of view, we can see this approach as a way to exploit the *knowledge* provided by the stakeholders through their contributions to identify their topics of expertise.

In the second approach, StakeNet [Lim et al., 2010] aims at prioritising the requirements to implement depending on how the stakeholders rate them. For this aim, starting from a reduced set of well-identified stakeholders, each of them suggests people that he or she assumes to have some influence on the project. A *role*, like student, security guard or director, and a level of *salience*, a value on a scale between 1 and 5, are provided to describe how and to which extent the suggested stakeholders influence the project. Based on these suggestions, a social network is built and usual measures are applied to evaluate the global influence of each stakeholder. From the EF point of view, we can see this approach as a way to evaluate the expertise of a stakeholder by aggregating the suggestions of other stakeholders, thus providing *social recognition* evidences.

Chapter 3

The Ugly Duckling of Requirements Engineering

3.1 The Poor Support of Expert Finding

From some of our first publications, we saw that interviews was among the most important sources in some projects [Morales-Ramirez et al., 2012a, Morales-Ramirez et al., 2012b], providing a hint on the need to involve domain experts. From the state of the art, especially Section 2.3, we can see that this observation is actually a common one, and yet we assess a huge gap within the field of RE: although the need to exploit domain experts is clear, poor support has been given to the EF task. There is two main works we can link to this aspect: [Castro-Herrera and Cleland-Huang, 2009] who identify knowledgeable or interested stakeholders based on their posts in a forum, and [Lim et al., 2010] who identify stakeholders having the most influence in the project based on recommendations. If the first work is actually more a matter of *expert profiling* than *expert finding*, as described in Section 2.2.2, the second is clearly not intended to deal with expertise itself. Yet, we can see that they both use typical expertise indicators, as identified in Section 2.1.3: the first work relies on *accumulated accessible knowledge* retrieved from the posts of the stakeholders, while the second relies on *social reputation*.

Consequently, the main problem we want to tackle here is *the poverty of support for dealing with the EF task in the field of RE*. From that point, we have to make something clear to the reader: a recurrent feedback we received was that EF for RE in general is too broad, often adding that we should focus on the elicitation of new requirements, a particular RE task. What we have to make clear is that requirements elicitation is the most obvious, but not the *only* RE task which deserves to be supported with EF. But in order to properly understand this point, we need to describe further what are these RE tasks.

Different people have identified different tasks, which tend to evolve with the field: [Pohl, 1994] identifies the three dimensions of specification, representation, and agreement ; [Nuseibeh and Easterbrook, 2000] speaks about requirements elicitation, modelling, analysis, communication, agreement, and evolution ; [Cheng and Atlee, 2009] focuses on elicitation, modelling, analysis, validation, verification, and management. We can see that there is a general consistency: the first is completed by the second and third, who focus each on different levels of details (communication and evolution vs. management, agreement vs. validation and verification). We analysed the tasks described by [Nuseibeh and Easterbrook, 2000] and finally came to the conclusion that EF can be applied in most of them, but here we will focus on [Cheng and Atlee, 2009], which is more recent and provides us statements to quote:

- Requirements elicitation is about *the understanding of the goals, objectives, and motives for building a proposed software system*, which requires to build *deep* and *precise* requirements, which *fit* the environment, with the help of *positive and negative feedback*. Thus, it is inherently relying on broad and reliable information, which is what domain experts can provide.

- Requirements modelling is about having *precise models* to *evoke details* *that were missed in the initial elicitation* and to *communicate the requirements to downstream developers.* The fact that more details may be needed and that adapting the models for developers should not loose nor alter these details is where domain experts can help.

- Requirements analysis is about *evaluating the quality of recorded requirements*, including *trade-off decisions* and *understanding*, and can involve *negotiation* and *inspection.* As such, domain experts are among the most suited to tell why a given trade-off is worth, and to feed or lead negotiations.
- Requirements validation is about *subjective evaluation of the specification with respect to informally described or undocumented requirements*, which *usually requires stakeholders to be directly involved.* In other words, this is where the tacit knowledge of experts and there ability to identify lacks and errors can be of first importance.
- Requirements verification is about using a *formal description* to *prove that the software specification meets these requirements.* Consequently, it requires less domain experts than the previous tasks, although they can be considered if the description is not completely formal.
- Requirements management is about *ease, and partially automate, the task of identifying and documenting traceability links* or analyse the *maturity and stability of elicited requirements.* Like verification, it requires less involvement of experts, but still has to integrate well with the workflow of the stakeholders, which is where feedback from experts can be helpful.

Long things short, obtaining broad, precise, relevant, domain-specific information is often among the top needs in many RE tasks, making EF a generic need. Yet, we may argue that it does not always lead to *problems*, in the sense that we can face situations where experts are easy to retrieve, for example in companies we can rely on well-identified people having well-identified responsibilities. Once again, if we agree that it is a relevant situation to consider, it is not the *only* one: [Cheng and Atlee, 2009] for instance mention two important trends, which are globalization and scalability. Globalization concerns geographically distributed organizations, thus involving people having different time-zones but also different cultures and languages, which makes the access to and understanding of people more difficult, in which case it might be better sometime to rely on some "local experts" which still have to be identified. The scalability issue seems to us even more important, because it also includes the ability to consider a growing amount of stakeholders to satisfy and domains to involve, in which case it might be better to involve few experts for each domain to keep it manageable. Open Source communities are a good example: contributors to a same project can come from different places on the globe, and they evolve constantly with (often anonymous) people coming and leaving, thus involving both globalization and scalability properties. In such a situation, the experts we brought for early elicitation can have leaved the community, leading to rely on new experts who are not yet well-identified, or they may simply have been considered because they were the most expert at that time, which can be now false because more expert people have joined the community.

Finally, we might still argue that the problem we try to deal with is too broad *because* it intends to deal with all the RE tasks, which are still subject to different constraints. The point is that we do *not* aim at dealing with all the specificities of each RE task, but with *a single common part* of most of them, which is the need to find expert people. For instance, [Cleland-Huang and Laurent, 2014] highlight that requirements gathering (i.e. elicitation) requires CRACK stakeholders (coined by [Boehm and Turner, 2008]): people who are *Committed* to the project, *Representative* of a group of stakeholders, *Authorized* to make decisions, *Collaborative* team members, and *Knowledgeable* of the domain. If we consider requirements validation for example, we might not require stakeholders to be committed to the project nor authorized to make decisions, because we would like external feedback to stress the specifications, yet we want to minimize the feedback to the most relevant one by asking only potential experts to participate. Clearly, the fact that a stakeholder is an expert is only one piece of the requirements elicitation puzzle, and if we also need stakeholders to fulfil several properties for requirements verification, expertise is again only a single piece of it. In our work, we are not concerned with the commitment of a stakeholder, her representativeness, or collaborative capabilities, but we want to know whether or not this person is an expert, thus knowledgeable. And if it is important to focus on a single aspect, it is because at the end each property must be properly assessed: a stakeholder cannot properly be identified as a CRACK stakeholder unless, among other things, he or she *is* knowledgeable.

3.2 Research Questions Driving this Thesis

Given that the problem we want to tackle is the poverty of support for dealing with the EF task in the field of RE, our research objective is to improve this support. The main path we investigated was by building on RE approaches in order to improve our ability to measure expertise. The fact that these approaches build on complementary aspects (knowledge for one and reputation for the other) allows to design a more comprehensive approach by inspiring from them. Thus, we were first interested in the following question:

RQ 1. *Can we design an EF process able to consider the core artefacts (topics, terms, and roles) of the two RE approaches?*

This question is not about a mere combination of [Castro-Herrera and Cleland-Huang, 2009] and [Lim et al., 2010], for instance by building a ranking of experts from both systems and place them behind some merging or voting strategies. Our aim is to design a comprehensive approach able to consider topics, terms, and roles *together* (e.g. by using also relations between roles and topics or terms), in order to build a more complete evaluation of the expertise of the stakeholder. To answer this question, we design such an approach in Chapter 6 and we evaluate it in Chapter 8.

This evaluation is of particular importance to ensure that the designed process is a proper EF system, so provides a correct ranking of experts. From the state of the art (Section 2.2), we know that EF systems are based on techniques from KM, IR, and RSs. A deeper investigation shows that usual measures used for comparing rankings of experts and evaluate EF systems are the very same measures used for rankings of documents in IR [Balog, 2012]. Through our work, we saw that these measures are based on assumptions which poorly reflect the reality of expert rankings, which can be not only partially ordered but also incomplete [Vergne, 2016a, Vergne, 2016b]. Moreover, having two documents at the same rank is interpreted as providing *equal satisfaction* to the user, which is not compatible with the idea of ranking people by level of expertise. Indeed, not only other factors not related to the domain of expertise are involved in order to satisfy the user, like explication skills or the ability to understand the user, but having two experts at the same rank seems to us more a matter of inability to differentiate them rather than a strict equality. Consequently, we were interested in a better way to compare rankings of experts, leading us to the following question:

RQ 2. *How can we compare incomplete and partially ordered rankings of experts?*

This question implies not only to inspire from existing IR measures, to preserve there advantages, but also to think about how to *represent* a ranking of experts in the right way. We do this by revising the usual formalisation of rankings in Chapter 5, which provides definitions and measures that we think to fit better the evaluation of expertise, and we use this framework in the remaining of the thesis, especially in Part III to evaluate our approach.

Finally, if a given EF system cannot properly rank people by levels of expertise, it is important to know why, which implies to analyse the system not only from a technical point of view, but also from a conceptual one. Indeed, it is important to assess that one has used right indicators in a right way to properly evaluate the expertise of a stakeholder. Thus, we are also interested in answering this question:

RQ 3. *How can we support the correction of an existing EF system?*

This question aims at bringing the insights obtained from dedicated literature on expertise, presented in Section 2.1, to analyse an EF system in a convenient manner. Indeed, usual EF systems do not appear as building on literature about expertise itself, focusing more on perspectives brought from IR and similar fields ranking documents. In order to answer this question, we build a meta-model based on this literature in Chapter 4, in which we also use the meta-model to evaluate the coverage of existing EF systems.

Part II

Framework

Chapter 4

Meta-model of Expertise

In this chapter, which has been partially published in [Vergne and Susi, 2015], we intend to give a more structured representation of the knowledge acquired in our state of the art about expertise (Section 2.1). We first start by modelling in Section 4.1 the overall context: the *domain* which relates both the performer (the one to be evaluated) and the expertise evaluator. Then, we go in a finer granularity by considering the perspective of the *performer* in Section 4.2, which deals with obtaining skills and knowledge to build ones' own expertise. Similarly, Section 4.3 models the perspective of the *evaluator*, who has to rely on sources of information to perceive this expertise in the best manner. Because we focus on EF systems, we go even further in the granularity of the evaluator by modelling its *evaluation* in Section 4.4. Finally, we illustrate different uses of our meta-model in Section 4.5 to show the different kinds of support it can provide to EF designers.

Each subsection presenting a part of the meta-model follows a systematic plan which (i) describes the model by referring to the corresponding literature, (ii) provides the graphical representation of the model, and (iii) illustrates it with a concrete example of a recruiter who wants to hire a Database (DB) programmer. In order to differentiate common terms from the concepts introduced by the meta-model, we use THIS FONT for the concepts of the meta-model.

4.1 The Domain

When saying that someone is an expert, we should precise *in what*, so we first need to speak about the *domain of expertise* and how it relates to this expert or other people we would like to evaluate. Figure 4.1 shows how we model this domain of expertise, starting from the root concept of Domain, which relates to a set of people that we call the Domain Community. The Domain Community is broad, no specific filter is considered out of the ability to relate a person to the corresponding Domain. For example, someone currently working in this Domain or who has worked in it in the past, or someone who knows about it, or simply who is involved in such a way that he could, at some point, be influenced by or influence what happens in this Domain. So rather than thinking about this community as the set of *important* people to consider for this Domain, which involves some pre-filtering, it seems to us better to consider a community representing the set of people we are able to evaluate, which gives it a more practical interest for EF. For instance in a company we may consider a specific Domain relating to a specific department, but depending on the coverage we target the Domain Community can be all the people of this department as well as the whole set of employees of the company. Indeed, it is not guaranteed that people out of this department lack the expertise we are searching for.

Within this Domain Community, one relevant kind of person to consider is the Performer, who is the one who produces something related to the Domain, what we call an Outcome. The concept of Outcome, once again, should be considered in an open manner: the Performer can make specific products, like a book about the Domain or a piece of software usable in this Domain, but also a service, like teaching about the Domain, or even ideas, like how to improve existing processes used in the Domain. In order to perform well, the Performer may try to obtain Relevant Domain Knowledge, which can be achieved by looking at what has been already done in the past, so Outcomes produced by another (or the same) Performer. Particularly relevant Outcomes to inspire from are what we call Domain Prior Achievements, meaning Outcomes which have been considered as providing a significant added value to the Domain. A particular interest the Performer can have by learning about these Domain Prior Achievements is the higher chances of producing a successful and creative Outcome [Ericsson, 1999]. For example, the Performer can inspire from Recognized Masterpieces, so Outcomes which have advanced the state of the art by solving new problems, providing original solutions, or improving existing ones. People who have already identified some Relevant Domain Knowledge may have compiled them into Domain Teachings, like courses, and may have shared them with the Domain Community, so a Performer can improve her expertise more efficiently [Ericsson, 1999].

Another relevant kind of agent to consider in the Domain Community, and actually one of the most central concepts for EF, is the Evaluator, meaning an agent which tries to evaluate the expertise of a Performer. In order to evaluate this expertise, the Evaluator should look at the Outcomes of the Performer, which are the most direct evidences of what the Performer is able to do. This evaluation can then be communicated to the rest of the Domain Community, thus building the Social Recognition of the Performer, which allows other people, in particular other Evaluators, to refine or complete their judgements. This Social Recognition is the way by which a specific Outcome can be recognized as a Domain Prior Achievement, which helps the Performer to inspire from the most relevant Outcomes. Often, an Evaluator is or has been a Performer, because as we will see later the Evaluator needs some expertise to be able to judge the Outcomes properly, added to the fact that a Performer can also act as her own Evaluator. Yet,

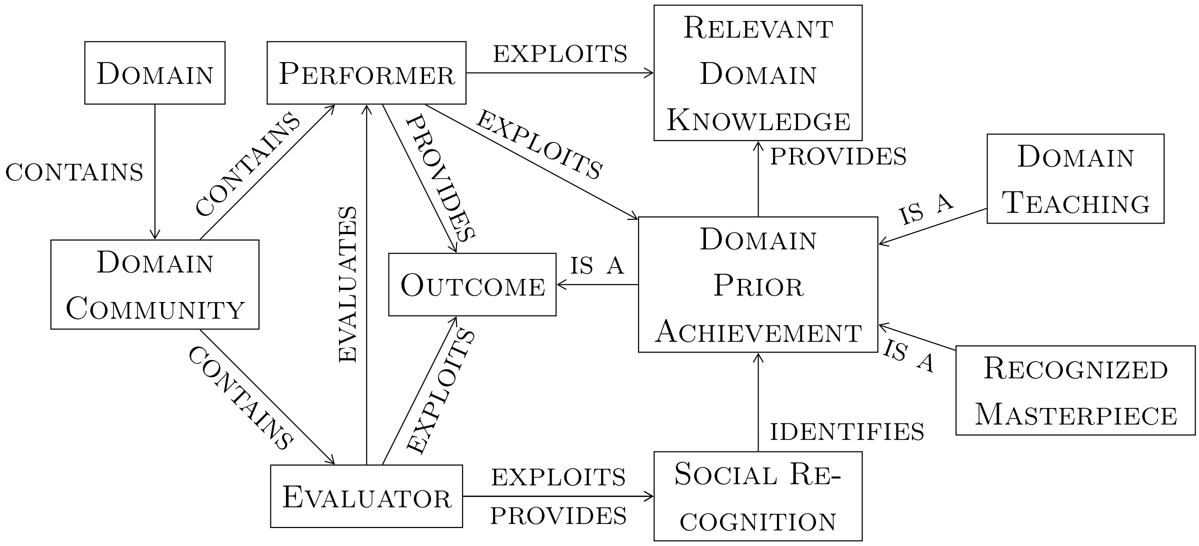


Figure 4.1: Conceptual model of the Domain.

it is not a requirement: if an Evaluator lacks in expertise to evaluate the Outcomes of a Performer, she can entirely base her evaluation on Social Recognition. For example, a recruiter might have heard that one of her problems requires having expertise in a particular Domain that she never heard before in order to solve it, leading her to look for Performers that *other people* have relied on.

As an illustration, we can take the example of the database (DB) Domain in a context of hiring for a company, so the Domain Community can be composed of the employees of the company and the job candidates, but also of other people well known in the field of DB. In particular, we can consider DB programmers and researchers, who produce and use techniques and tools (Performers producing Outcomes), but also academic teachers, who are relevant sources of Relevant Domain Knowledge (through Domain Teaching) as opposed to practitioners who can have produced Recognized Masterpiece but may have difficulties to verbalize their own knowledge ([Chi, 2006]). The recruiter (Evaluator) should at least know what are DBs and which aspects he wants to evaluate, like mastering the SQL language or abilities in building schemas, but can also rely on Social Recognition to enrich her evaluation.

4.2 The Performer

In order to know how to evaluate the expertise of a Performer, so what to infer from her Outcomes, it is important to know what composes her expertise, which is well described in the literature exposed in Section 2.1. By exploiting this literature, we go deeper in the modelling of this Performer in Figure 4.2, where we relate the Performer to her Owned Expertise. As mentioned in Section 2.1.1, expertise builds on skills and knowledge, which is what we represent through the Owned Domain Skill and Owned Domain Knowledge of our model. The Owned Domain Skill corresponds to what the Performer is able to do, like running fast or programming a software, while the Owned Domain Knowledge corresponds to what is known by the Performer, like how long is the race or what should be implemented. In order to improve her expertise, the Performer should improve her Owned Domain Skill or Owned Domain Knowledge, which can be achieved in two ways, as mentioned in the literature: experience and teaching.

Experience is the expertise which has been built by the Performer herself, what we model with her Practice. This Practice can be of several kind, but we saw in Section 2.1.2 that an important property is whether the deliberateness of the Performer. Although any practice allows to build basic expertise, the highest levels cannot be reached without Deliberate Practice, meaning a Practice where the Performer identifies her own limitation and try to overcome them. When the Performer starts to be used in doing a given task and no external stimulus stresses the remaining weaknesses, unless the Performer deliberately searches for further improvement, it becomes Routine Work which leads to stagnation.

The other way to improve one's expertise is through teaching, meaning by

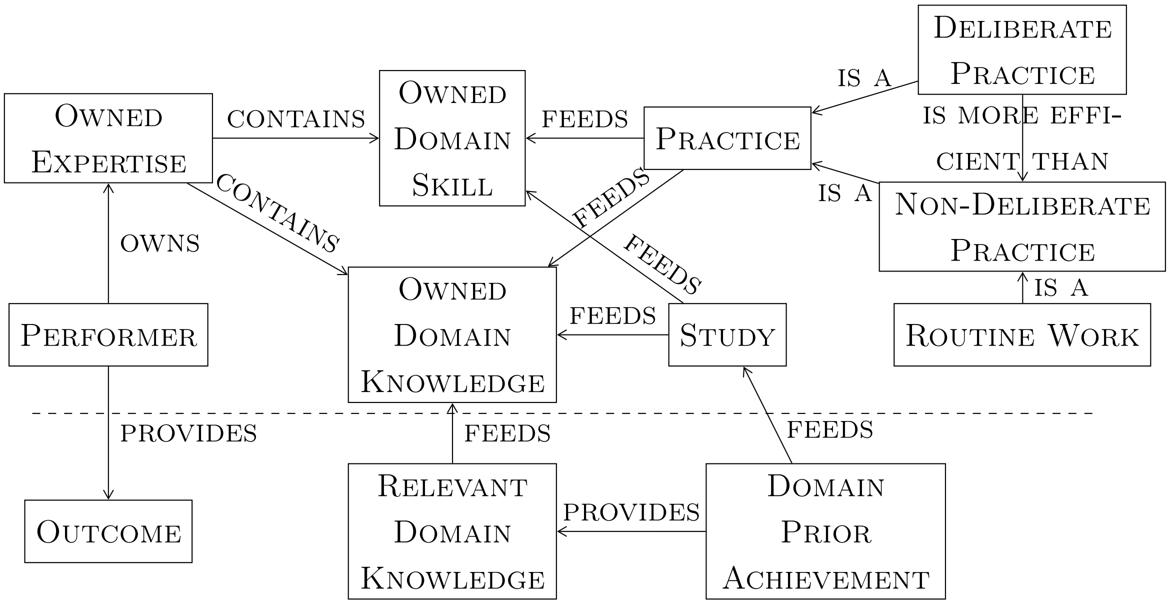


Figure 4.2: Conceptual model of the PERFORMER (above the dashed line) and her interactions with the DOMAIN (below the dashed line).

acquiring knowledge and skills that other people accept to share. We model this through the STUDY that the PERFORMER may do, which is fed by elements which are external to the PERFORMER. These elements are the DOMAIN PRIOR ACHIEVEMENTS described in the previous section, in which we find the DOMAIN TEACHINGS allowing to obtain RELEVANT DOMAIN KNOWLEDGE. One might prefer to model the STUDY in a decomposed manner: the *communicative part*, where the other person provides the teaching, and the *practical part*, where the PERFORMER practices on her new knowledge or her new skill. With such a model, we may argue that it is always a matter of personal PRACTICE, and the STUDY is just a specific way to obtain it. Although this approach is interesting, this is not the one used for this model, in which we prefer to highlight the fact that the PERFORMER could improve her OWNED EXPERTISE from two kind of sources: by herself or with the help of other people. This highlight is interesting because an EVALUATOR (including an EF system) cannot directly access to the first source, while she could have access to the second.

Going further in our illustration with the company hiring, we could have a DB programmer as candidate (Performer) who produces websites and applications with DBs (Outcomes). The expertise of this DB programmer builds on his Owned Domain Knowledge, such has knowing about the SQL language and other programming languages, and Owned Domain Skills, like building DB schemas and website interfaces. This DB programmer has built websites and applications for different clients during several years, allowing him to master the most common techniques (which are now a Routine Work), and developed his own applications to try and improve customized techniques (Deliberate Practice). In parallel, he has learned the basis of SQL from some courses in school, and searched for original ideas to represent data in a DB by looking at Open Source codes available on the Web (Study).

4.3 The Evaluator

Once the expertise of the Performer is considered, what we are interested in is how to evaluate it, which is the role of the Evaluator, modelled in Figure 4.3. This is the most important concept for us because it is the one representing the EF system we intend to build. While the Performer can improve her level of expertise, the aim of the Evaluator is to perceive this level, which leads us to mirror what we have modelled in the Performer into the Evaluator. Consequently, the Evaluator builds a Perceived Expertise, which is composed of both the Perceived Domain Skill and Perceived Domain Knowledge. However, in order to perceive them, the Evaluator should be correctly prepared: as someone who is unfamiliar with tribal cultures would not get the particular interest of a tribal dance before someone explains her, the Evaluator cannot properly identify a relevant skill or knowledge if she does not know about it. This is why the Perceived Domain Skill and Perceived Domain Knowledge need to be supported by the

Owned Domain Knowledge of the Evaluator. If the Evaluator is prepared enough to perceive them, then she can properly analyse the Outcomes of the Performer.

However, identifying what the Performer is able to do and knows about is not enough: not only they should be owned, but they should also be used efficiently. The most direct way to evaluate how good the Performer uses them is to look how she performs in domain-relevant tasks, but this is not always possible, especially for a requirements engineer, who is supposed to work on specifications rather than on the field. Thus, indirect evidences need to be used to infer the level of expertise of the Performer. In the literature reviewed in Section 2.1.3, the most common way to do so is to look at evidences about Lengthy Domain-Related Experience, such as the time spent working in the Domain or the number of Outcomes produced. Nevertheless, this kind of evidence only supports that the Performer is able to do her job, so the presence of a reasonable or average expertise. In order to be sure that the highest levels of expertise have been reached, the Evaluator should look at evidences of Reproducibly Superior Performance, like being always the first to finish a race or achieving the best performances with several pieces of software she has developed.

Going even further, while the Lengthy Domain-Related Experience is often traced in some way for common purposes, like in a CV, it might provide only superficial information. Moreover, evidences of Reproducibly Superior Performance tend to be rare, with the focus being often simply on whether or not the task has been achieved in a reasonable time. In the case where the Evaluator is a requirements engineer, she has other priorities than spending time studying the field to find out the missing information, so she needs to rely on other sources to refine her evaluation. This is where the Social Recognition comes in play: other Evaluators may have already evaluated the Performer, and if they accept to share their evaluations in some way (e.g.

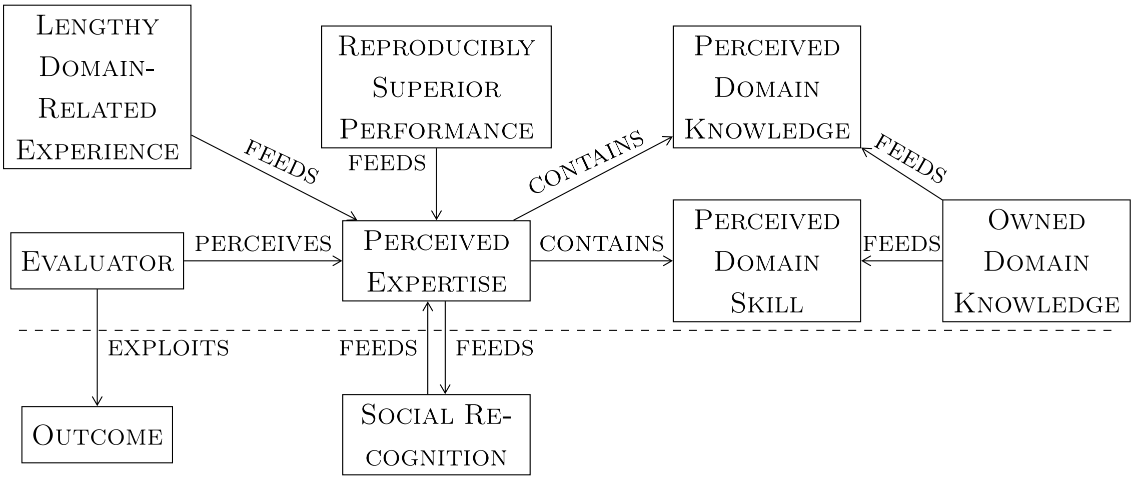


Figure 4.3: Conceptual model of the Evaluator (above the dashed line) and his/her interactions with the Domain (below the dashed line).

feedback, official endorsement) the current Evaluator can exploit them to refine her judgement. Nevertheless, if the Social Recognition allows to assess average performance, we can see from the literature described in Section 2.1.3 that evidences of Reproducibly Superior Performance remain required to ensure that the Performer has obtained the highest levels of expertise.

Extending our illustration with the DB example, we take the perspective of the recruiter (Evaluator) who wonders whether he should hire our DB programmer or find someone else. For that, the recruiter looks at the CV of the candidate (Outcome) and, if the time permits and the interests justify it, looks at specific websites and applications the DB programmer has worked on (other Outcomes). By checking the CV, the recruiter notices the 12 years experience of the candidate in building websites (Lengthy Domain-Related Experience) and identifies his participation in current projects involving advanced technologies (potential Reproducibly Superior Performance). Additionally, he relies on social networks and other public recognitions (e.g. specialized press articles, comments, awards) to check the opinion of other people regarding this DB programmer and his works (Social Recognition).

Interested by the profile, the recruiter plans an interview with the DB programmer during which they enter in more details to know on which parts of the projects the candidate was involved, what was his responsibilities, and what he was able to achieve (Perceived Domain Knowledge and Perceived Domain Skills). Obviously, to properly evaluate the CV and perform the interview, the recruiter should have enough expertise in the domain to identify the right characteristics to look at and the right questions to ask (Owned Domain Knowledge), or rely on someone else to make the evaluation (e.g. expert already employed, external specialists).

4.4 The Evaluator's Evaluation

In the absolute, once the Evaluator has built her Perceived Expertise of the Performer, she is able for instance to decide who to recruit or who to trust more, so we could consider the job to be done. But we can also think about the Social Recognition which requires some Evaluators to share their Perceived Expertise, leading to transform this perception into an exploitable, concrete expertise evaluation. This is particularly true for an EF system, which is not aimed to decide who to choose, but to recommend people to decision makers who will make the final selection. Consequently, we also provide a model of the Performance Evaluation made by the Evaluator in Figure 4.4, which should help the decision makers to infer how much expertise the Performer has in the Domain they are interested in. And like in the literature presented in Section 2.1.3 about expertise evaluation, this can be done in two ways: absolutely or relatively to other Performers.

When looking at the literature, it appears to us that the most common is to try computing an Absolute Performance Value, which means to describe the performance of the Performer independently of other Performers. This is usually done by assigning to the Performer a Performance Level, typically by selecting a scale and telling where this Performer is located on this scale based on her Perceived Expertise. Usual scales look like the Table 2.1 of [Chi, 2006], which assigns to the Performer a level from Novice to Master, although variants exist (e.g. from *Newbie* to *Top Java expert* for [Zhang et al., 2007]). But we might also go for more abstract levels, like the Average Performance Level and Highest Performance Level that we already mentioned, which are levels that we can directly link to other elements of our model. In particular, evidences of Lengthy Domain-Related Experience can confirm an Average Performance Level, while additional evidences of Reproducibly Superior Performance are required to confirm the achievement of the Highest Performance Level. We are convinced that other types of Absolute Performance Value –not using scales– can be included, like *labelling* (e.g. which jobs of the Domain the Performer seems to be able to handle), although we did not include it in the current model due to lack of literature support.

In some cases, it might be that the Evaluator is unable to state where the Performer stands on a given scale, by lack of information or because she does not have a relevant scale to exploit. Nevertheless, the Perceived Expertise she has acquired from several Performers gives her the possibility to describe them with Relative Performance Values. In such a situation, we do not know where a specific Performer stands between Novice and Master for example, but we know whether or not she is more expert than other Performers, so the Evaluator can provide a Performers Ordering. It is worth noting that if a Performance Level can be given to each Performer, then we can build the corresponding Performers Ordering (Performers at a master level are more expert than Performers at an expert level, and so on), so Relative Performance Values can be considered as generalizations of Performance Levels (but not of *labelling*, so it is not a generalization of Absolute Performance Values as a whole). This is why in our work we

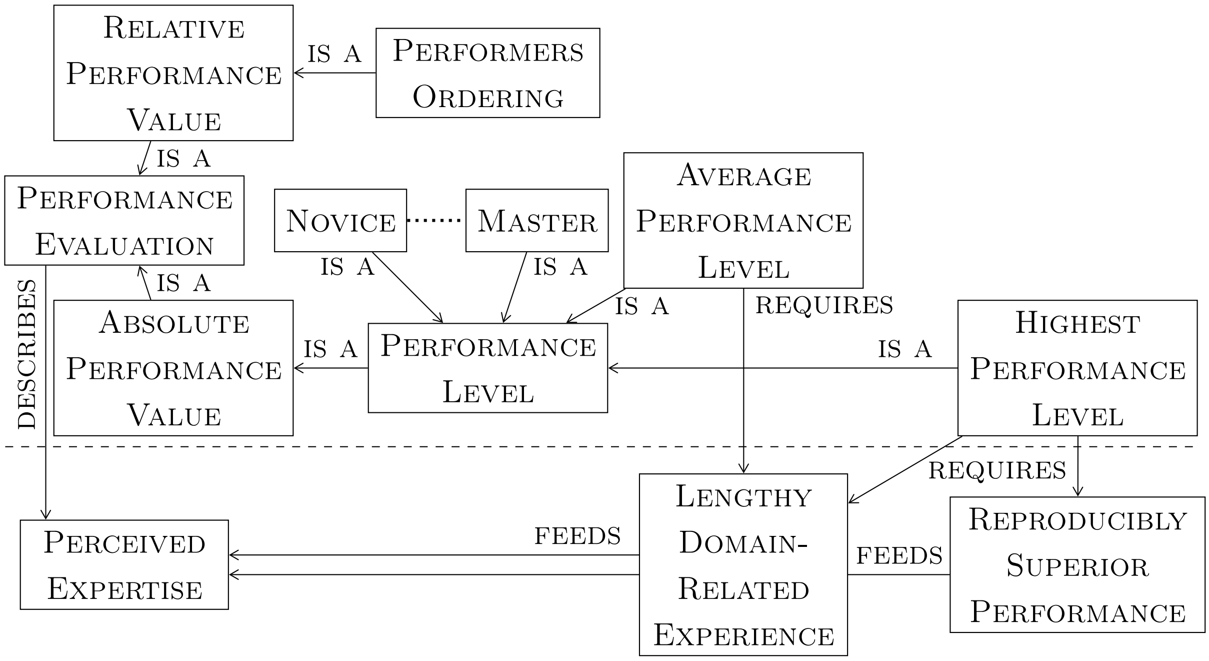

Relative Performance Value
is a
Performers Ordering
is a
Performance Evaluation
Novice
Master
Average Performance Level
is a
is a
is a
is a
requires
Absolute Performance Value
is a
Performance Level
is a
Highest Performance Level
describes
requires
Lengthy Domain-Related Experience
Reproducibly Superior Performance
Perceived Expertise
feeds
feeds

Figure 4.4: Conceptual model of the Performance Evaluation (above the dashed line) and its relations with the Evaluator's Perceived Expertise (below the dashed line).

favour this relative perspective, and we describe in Chapter 5 all what we need in order to deal with Performers Orderings for evaluating similarity and compliance to gold standards, rather than using more usual measures designed for Absolute Performance Values.

With this last model, we can close our DB illustration by looking in the details of how the recruiter evaluates the DB programmer. We can imagine that he evaluates him by giving him some marks, which represent his perceived level for some key characteristics of the job (Performance Level), or by ranking him in regard to other candidates (Performers Ordering). By using marks, the recruiter could consider simple numbers on a given scale (e.g. from 1 to 10) or more explicit levels (e.g. Novice–Master scale) or even dedicated labels which correspond to available jobs (e.g. DB administrator, schema designer). While a recruiter who is already knowledgeable on DBs could do it, a recruiter lacking the relevant knowledge –thus giving a good reason to hire– could be unable to properly identify levels of expertise. In such a case, the recruiter can compare the candidate to other people he may know, like the current employees, and build a comparison based on what the candidate was able to show in his CV and interview. For instance, the candidate could seem to know better some interesting SQL features that a team member had difficulties to use, leading to rank the candidate higher than this employee, thus considering the candidate as a potential opportunity.

4.5 Meta-model Application

By building this model, our first goal is to centralize the knowledge retrieved from the literature about expertise, which makes it a relevant support for understanding what is expertise and how to evaluate it. Particular uses that we consider here are the support for *creating* new EF systems, as shown in Section 4.5.1, but also *analysing* existing ones to identify improvements, as shown in Section 4.5.2. A third application, done in Section 4.5.3, is to map concepts from this model to usual indicators used in RE works. This is particularly interesting for us because it helps us to identify what should be used in an EF system to be applicable to RE contexts.

4.5.1 Design of an Expert Finding System

In order to create a new EF system, the designer should first think about the categories of Domains to cover, which can go from the most specific ones to the most generic. For instance, [Mockus and Herbsleb, 2002] targets the development of a specific software, while [Zhang et al., 2007] relies on question-answer patterns which can be applied anywhere. In parallel to this, it is important to identify the classes of Performers to consider, especially the Outcomes they provide and which can be exploited by the Evaluator (the new EF system) to evaluate their expertise. Additionally to the Outcomes, which provide direct information about the Owned Domain Knowledge and Owned Domain Skill of the Performer, the designer might also consider to exploit sources providing information about Social Recognition. Outcomes and Social Recognition are complementary, but because Outcomes can be highly domain-specific (e.g. scientific publications, pieces of software, architectural projects) they are more suited for EF systems dealing with restricted Domains, while the most generic ones would find more interest in exploiting Social Recognition evidences, although it is recognised as a poor indicator [Ericsson, 2006b].

While the Outcomes of a Performer can be exploited to build the Perceived Expertise of the EF system, this perception should be based on some Owned Domain Knowledge, as shown in Figure 4.3. Although we did not model how to feed this knowledge from the Evaluator perspective, this property is shared with the Performer, represented in Figure 4.2, which builds on personal Practice and Relevant Domain Knowledge obtained through some Study. For an automated Evaluator, common designs focus on the Study part: pre-defined data (already identified Relevant Domain Knowledge), or pre-defined processes to retrieve it (from Domain Prior Achievements, like sets of publications for EF systems based on language-models), are often developed to design the system. It might be interesting to investigate how an automated system could improve through “personal Practice”, for instance by learning through neural networks or by using other Artificial Intelligence (AI) techniques to revise the criteria to measure expertise. In particular, mirroring the Deliberate Practice of the Performer, the automated system could seek for personal improvements by revising data leading more often to poor or contradictory results.

Finally, the designer of the EF system should think about the kind of Performance Evaluation to build in order to describe the Performers. In particular, if it is hard to identify evidences of Performance Levels for a Performer, the designer might prefer to use Relative Performance Values, which still allow to say who appears to be among the most expert. It could also be interesting to investigate hybrid solutions, for instance by computing Relative Performance Values and enriching them with Performance Levels when it is possible. We may rely for instance on well-identified Performance Levels for some people, while other people would have a level which depends on how they relatively rank compared to these reference people. We also think that, if Performance Levels can be used, they should be designed based on the *users* of the EF system. Different users might have themselves different levels of expertise in the Domain, giving them more or less ability in understanding the various Performance Levels. Simple and broad levels might be used for ignorants, while detailed levels might be used for more advanced users, or we could also think about using different categories of Performance Levels to cover various uses of the EF system.

4.5.2 Analysis of Existing Expert Finding Systems

Another way to exploit our meta-model is by mapping its concepts to the elements of an existing EF system, which allows to identify potential improvements. A mapping with the elements of the Domain (Figure 4.1) might help to identify lacks in the external resources exploited, for instance by using exclusively Outcomes or exclusively Social Recognition. The elements of the Evaluator (Figure 4.3) can help highlighting an unbalance between the Perceived Domain Knowledge and the Perceived Domain Skills. Or it might show that the exploited evidences of performance do not properly support Reproducibly Superior Performance, thus limiting the ability of the system to identify experts having the Highest Performance Level. As we mentioned before, the elements of the Performer (Figure 4.2) provide further insights on how the Owned Domain Knowledge of the Evaluator can be built, although it remains rather superficial so we ignore it here. Finally, the elements of the Performance Evaluation (Figure 4.4) relate to the final value provided to the user of the EF system. In the following, we analyse several existing works to illustrate these different kinds of support that our meta-model can provide. Some of these analyses are represented also in a graphical way to have an efficient summary of what is covered by the EF system analysed.

By analysing the work of [Serdyukov and Hiemstra, 2008] (Figure 4.5), an approach evaluating the contributions of people based on the documents they write, we can see that they focus mainly on Perceived Domain Knowledge items by identifying the terms used. In particular, by evaluating how much a person contributes compared to all the others (via normalization), this approach infers Absolute Performance Values (i.e. probabilities) and recommends the people having the highest ones. While we could imagine that the values computed could help to infer Performance Levels, this approach would need to be completed with correlations between their values and proper levels. Moreover, while such an approach is probably efficient to build the Perceived Domain Knowledge, it lacks the Perceived Domain Skill dimension. We can argue about redaction skills, which are obviously necessary to redact documents, but they are not always the most relevant skills for the targeted Domain, like research papers in regard to the research topic they are about. Going further, these approaches probably identify evidences of domain-related experience but not necessarily of Lengthy Domain-Related Experience, making it difficult to assess even an average level, unless the assumption of a lower bound expertise can be supported by the specific type of documents considered (e.g. peer-reviewed papers accepted for publication). Such assumptions, however, would probably not help in discriminating good from exceptional Performers, meaning finding evidences for Reproducibly Superior Performance. Additionally, no use of Social Recognition is made, although it could help refining the values.

The work of [Zhang et al., 2007] build social networks by linking questioners to answerers in a Java forum, each network being based on a given topic. They measure the number of answers (higher expertise) against the number of questions (lower expertise) to infer directly the PERCEIVED EXPERTISE (i.e. no content analysis to infer any PERCEIVED DOMAIN KNOWLEDGE or PERCEIVED DOMAIN SKILL). They also exploit PageRank-like algorithms, which propagate these values over the network, to weight each individual based on the overall network configuration, leading to consider SOCIAL RECOGNITION indicators. Regarding the representation used, each social network relates questioners to answerers, thus allowing to build PERFORMERS ORDERINGS where a questioner has less expertise than an answerer. However, for their evaluation, they asked to identified experts (although they do not mention how they assessed their expertise) to relate participants of the forum to five PERFORMANCE LEVELS, from *Newbie* to *Top Java expert*. Once again, this approach lacks the identification of PERCEIVED DOMAIN SKILLS, but could exploit the PERFORMANCE LEVELS used in their evaluations to correlate to the results of their PERFORMERS ORDERINGS. They also suffer the same difficulties than [Serdyukov and Hiemstra, 2008] to identify clear evidences for LENGTHY DOMAIN-RELATED EXPERIENCE as well as REPRODUCIBLY SUPERIOR PERFORMANCE.

We retrieve these difficulties in approaches combining documents and social analysis, like [Karimzadehgan et al., 2009]. Although they combine SOCIAL RECOGNITION (hierarchy of employees) with PERCEIVED DOMAIN KNOWLEDGE (terms and topics), they ignore the PERCEIVED DOMAIN SKILLS.

In the literature presented so far, only [Mockus and Herbsleb, 2002] (Figure 4.6) provide a rather complete approach by considering the commits (changes on a software) made by programmers. Commits are at the same time good indicators of PERCEIVED DOMAIN SKILLS (coding skills are major skills in software) as well as PERCEIVED DOMAIN KNOWLEDGE (module modi-

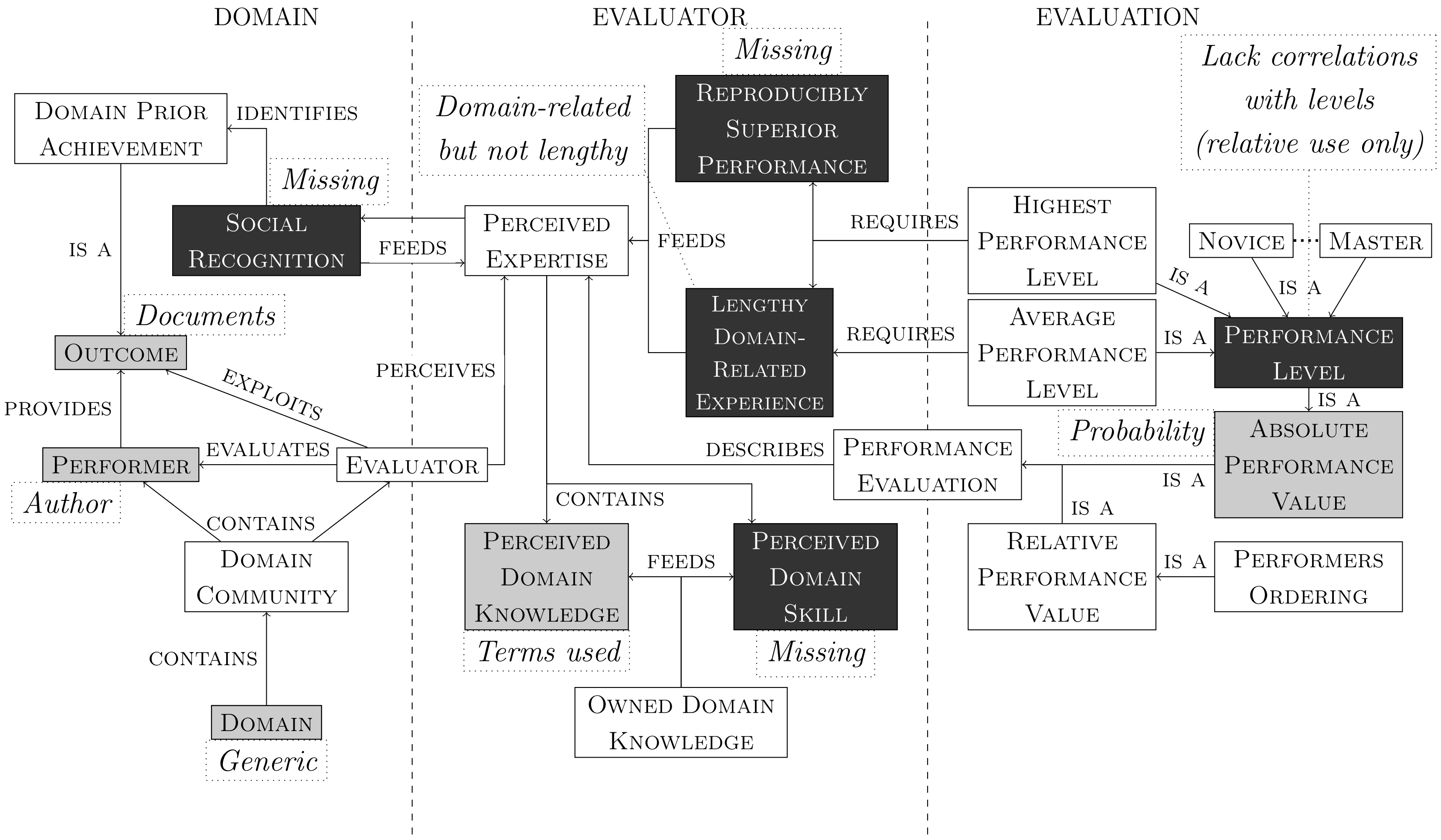


Figure 4.5: Graphical analysis of the EF system designed by [Serdyukov and Hiemstra, 2008] with a reduced aggregation of the complete meta-model of expertise. The light-grey elements are managed, with annotations to tell how. The dark-grey elements are absent or poorly managed, with annotations to tell why.

fied, names of the variables added/removed/changed, etc.). The number of commits made over time can also show a Lengthy Domain-Related Experience, while frequencies of commits per month could show reproducible performances, although it does not necessarily support the high quality required by Reproducibly Superior Performances. Thus, while they already provide strong supports and results, our models allow us to understand quickly why they are able to do so and also to identify the potential improvements to perform (i.e. identifying the highest levels of expertise). Though, these good results should be contrasted to the fact that this approach targets a specific Domain (software programming) while the other approaches try to be more generic, making the task more difficult.

4.5.3 Mapping to Requirements Engineering

The two previous applications of our meta-model show how it can help in building new EF systems as well as analyse existing ones to identify improvements. But a derived way to use it is to map, in our case, usual concepts of RE works to this expertise meta-model. Indeed, in our work we design an EF system to use in RE tasks, so experts can be recommended to requirements engineers to help them elicit and refine requirements. In order to do so, not only we need our meta-model to know what is required at the abstract level, but we also need to know what are the concrete indicators that are used in usual RE works and that we can build on. Consequently, we identify usual RE concepts from the works described in Section 2.3.2 and map them to our meta-model of expertise. Although they are not EF systems themselves, they are the closest works we could find in RE to inspire from: they share the goal of recommending relevant people based on their outcomes or social relations. All the concepts and relations established in this section are summarized in Figure 4.7.

Because we aim at recommending experts for RE tasks, we have to iden-

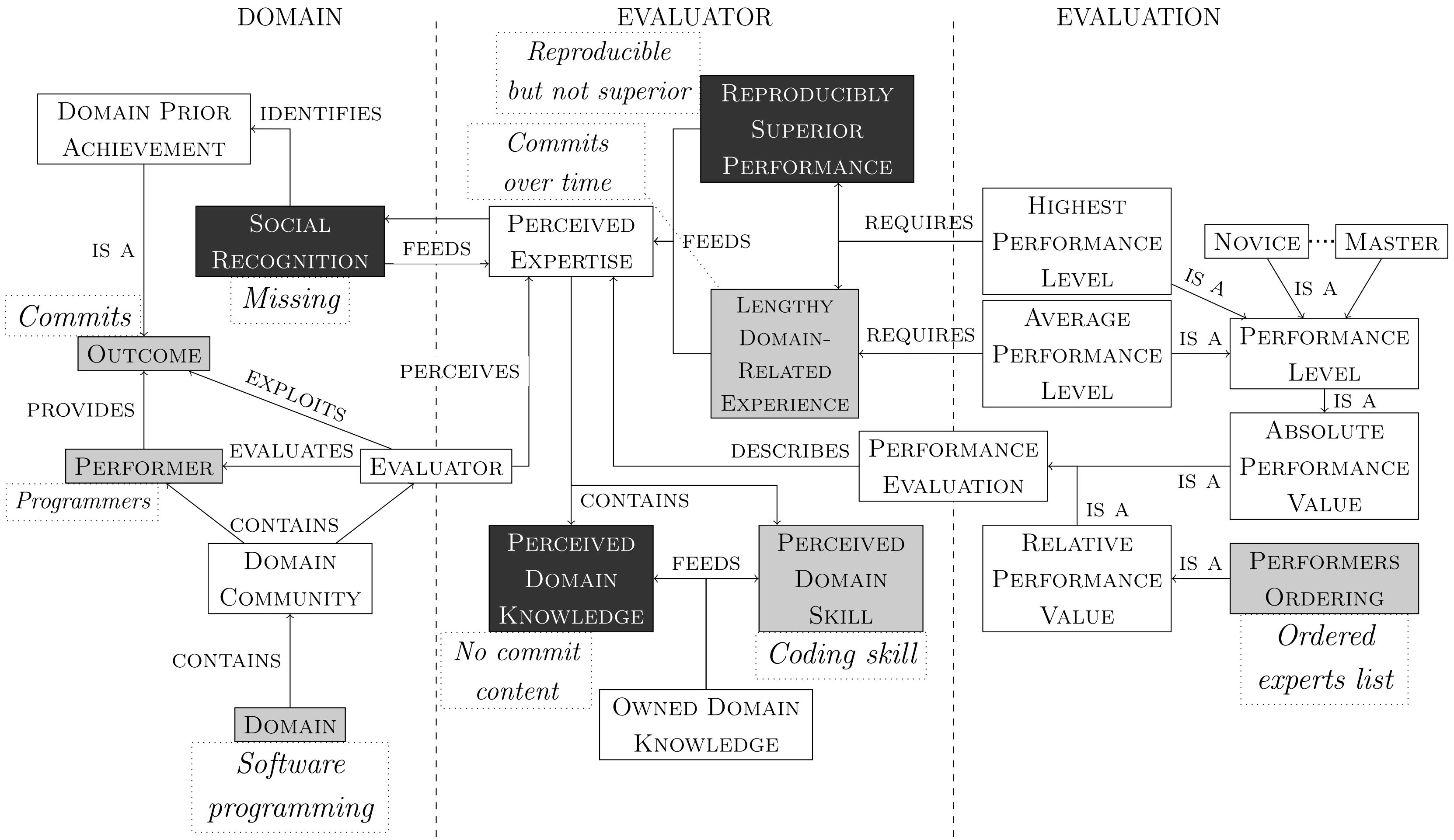


Figure 4.6: Graphical analysis of the EF system designed by [Mockus and Herbsleb, 2002] with a reduced aggregation of the complete meta-model of expertise. The light-grey elements are managed, with annotations to tell how. The dark-grey elements are absent or poorly managed, with annotations to tell why.

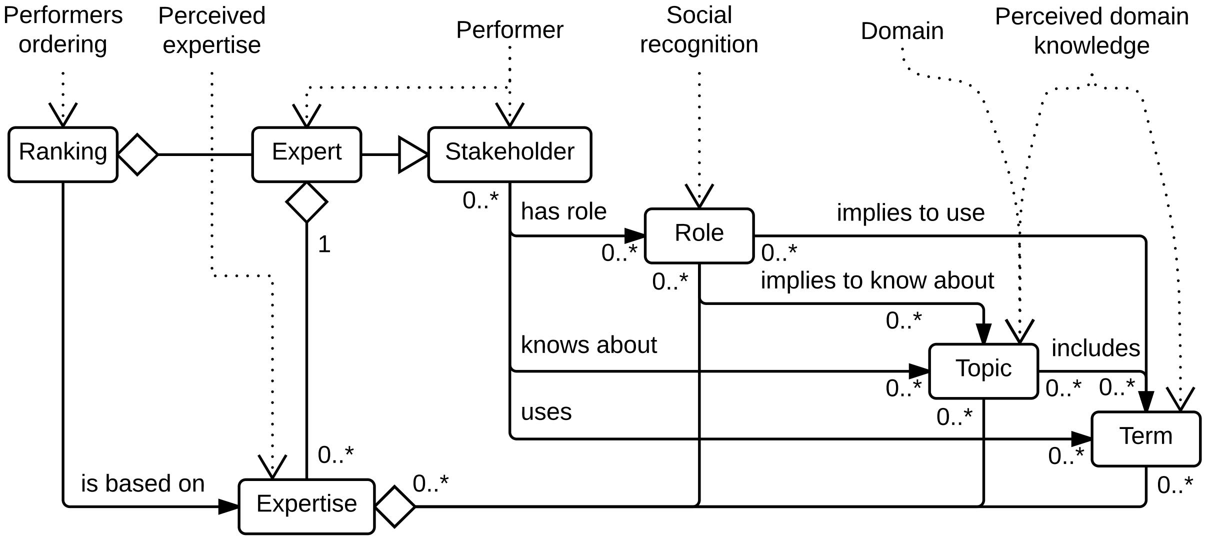


Figure 4.7: UML model of the concepts and relations of our approach, mapped to concepts of our meta-model at the top. The relations between stakeholder, role, topic and term are directed only to simplify the reading: we exploit them as correlations.

tify the Performers who will be recommended. In RE, the people involved in a project are usually called *stakeholders* and, because we consider that any person involved in a project is a potential expert to work with, we will use the same term in our approach. Each stakeholder can have one or several *roles*, such as being a developer or a manager in a company, but also being a contributor in the forum of an OSS, which are evidences of Social Recognition. Each stakeholder can also know about some *topics*, such as security, community management, interface or more specific products of the company or even specific features of these products. Going further, we can see that each stakeholder uses *terms*, whether it is in his contributions in some forums or in official documents he redacts. Both correspond to indicators of Perceived Domain Knowledge, with different levels of granularity.

All these concepts can be retrieved from existing recommender systems in RE, like [Lim et al., 2010] who builds on social networks by identifying stakeholders through their *roles*, while [Castro-Herrera and Cleland-Huang, 2009] work on forums by exploiting *topics* and *terms*. Going further, we can see that these concepts more broadly relate to our meta-model of expertise, starting from the Domain of interest which can be considered as an equivalent to a *topic* (EF systems are usually queried on topics of interest). Then, by taking a stakeholder as a Performer, the messages written in a forum, including their *terms*, correspond to the Outcomes of this stakeholder. By taking other stakeholders as Evaluators, they can evaluate these Outcomes to build their Social Recognition, and by extension to allow the Performer to endorse specific *roles*, like to be a moderator of the forum or a contributor in an Open Source project. There is many more types of Outcomes that one could exploit, like the code produced by the stakeholder, but more technical Outcomes tend to be more specific of the Domain considered, while we want here to focus on the generic part of RE tasks. Consequently, we focus here on the types of sources exploited in the previous RE works, while additional types of sources can be considered for future works.

At this point, we have stakeholders who are related to roles, topics and terms. In our approach, we go further by exploiting the fact that knowing about a topic, like *interface*, implies generally to know some terms related to this topic, like *interface* (the name of the topic itself), *button*, *screen* and so on. In the same way, having a specific role, like *developer*, implies generally to know about some specific topics, like *interface* and *programming*, and to use specific terms, usually some topic-specific jargon. We exploit all these relations in our model to describe and evaluate the expertise of each stakeholder.

These relations can also be mapped to our meta-model of expertise, starting from the stakeholder who, as a Performer who produces Outcomes, do it within the scope of one or several Domains, so topics. Restarting from a topic as a specific Domain, the Relevant Domain Knowledge would correspond to the topic-specific terms, thus supporting the explicit relations between topics and terms. Similarly, a role being mapped to some Social Recognition, it should support the identification of Domain Prior Achievements, and by extension the identification of Relevant Domain Knowledge, thus relating roles and terms. Once again, starting from the stakeholder as a Performer, he should exploit this Relevant Domain Knowledge to produce relevant Outcomes, which gives another explicit relation between stakeholders and terms. We can also highlight that, like a Social Recognition (role) occurs depending on Domain-specific Evaluators, thus within the scope of a specific Domain (topic), roles can be topic-specific. We could also mention, although it is more implicit, that a Domain can have sub-Domains, so a topic can have sub-topics, which naturally relates the relevant roles to these sub-topics, thus justifying that a role-topic relation is not just a one way relation.

Additionally to all the concepts and relations presented so far, we have to consider the final recommendation we build from them. First of all, we define an *expert* using a relative point of view: being more expert than another person means having more expertise compared to this person. This definition takes the side of *relative* experts rather than of *absolute* ones, as described by [Chi, 2006], while the latter point of view considers people *above a threshold* to be experts even if nobody reaches this threshold in the considered community. This relative position is not often taken in practice (at least not explicitly) because people usually refers to some kind of scoring function assigning a specific value to a single person [Balog, 2012], but we think that the relative perspective provides a more robust way to deal with expertise, especially when we assume incomplete knowledge (individual scores may change with additional knowledge even when relative orders are preserved). To describe the *expertise* of a stakeholder, we exploit the topics she knows and the terms she uses, which are evidences of knowledge, but also the roles she has, which supports her social recognition.

Coming back to our meta-model of expertise, we focus now on the Performance Evaluation which is based on the Perceived Expertise of our Evaluator (the EF system). Actually, because we consider how the stakeholders relate to roles, topics and terms, rather than to other stakeholders, our approach produces absolute values, so one could think that we should map our rankings to the Absolute Performance Value. However, our meta-model is about expertise, and its concepts have to be interpreted in this scope: although we compute absolute values, we do not have a proper way to map them to objective Performance Levels, which is a limitation of our approach. Because of this limitation, we anyway build our rankings based on absolute values, but this is the comparison between these absolute values which allows us to compare the stakeholders, in other words to build Performers Orderings. This is why we take a relative point of view, and why our approach should be considered to produce Relative Performance Values.

4.6 Discussion

Although we were able to analyse existing approaches with our meta-model, there is some pieces of these works that we cannot link to it, showing that it can be completed. For instance, we were not able to relate the different formulae used in [Serdyukov and Hiemstra, 2008] and [Zhang et al., 2007] to the meta-model, highlighting the lack of concepts about *processes* (e.g. how to link the Perceived Domain Knowledge to evidences of Lengthy Domain-Related Experience or to the Performance Evaluation). We also miss the notion of time, which is required to properly assess a Lengthy Domain-Related Experience or Reproducibly Superior Performance, while [Mockus and Herbsleb, 2002] already consider it. Furthermore, they do not retrieve extensive Perceived Domain Knowledge and Perceived Domain Skill items, but still identify Lengthy Domain-Related Experience and reproducible performances. This lack of dependency highlight the need to study deeper their relations and, in particular, how the former items could help to find the latter ones, which is a main intuition for the other approaches, in particular the intuition that extended knowledge should support extended experience.

Even if we do not focus on the works analysed, we can notice some unbalance in the development of our concepts: the Non-Deliberate Practice specializes only into Routine Work while we could consider also exceptional job duties (which are clearly out of routine, yet non-deliberate); Deliberate Practice can specialize into hobbies and probably others ; we could also add different types of Absolute Performance Values, like the labelling we mentioned, and of Relative Performance Values, like classes of equivalence (highlight similarity instead of greater expertise). We could also argue whether or not Domain Prior Achievement can provide "relevant domain skills" (like Relevant Domain Knowledge) to allow the Performer to feed his Owned Domain Skill, or if it comes only as Relevant Domain Knowledge that the Performer should put in Practice.

Going back to the literature already cited, we did not consider the expert properties provided by [Chi, 2006] (i.e. generate better solutions faster, fail in judging non-expert abilities, etc.), while it could provide relevant indicators to exploit. We might also add that evidences of Deliberate Practice from the Performer may support an expert-like behaviour, complementing the simple experience-based indicators criticized by [Ericsson, 2006b]. We also rely exclusively on literature in Psychology to identify the main concepts (top-down), while it could be complemented with systematic literature reviews of existing EF techniques to identify relevant lower level concepts (bottom-up, like [Yimam-Seid and Kobsa, 2003]). Other perspectives could also be considered, like creativity [Ericsson, 1999] (i.e. producing something new and useful), which seems to be a way to identify some of the highest experts.

We can also argue the interpretation of the concepts and relations chosen to build our meta-model. The concept of expertise introduced (Perceived Expertise) represents the knowledge and skills of a given person, while the literature also considers expertise as the knowledge and skills required to reach an expert level, which is Domain-specific and not related to a single Performer. We could also discuss the need to have the relation "Evaluator is a Performer" because, like the Performer, the Evaluator needs to build his Owned Domain Knowledge. We might consider a more structured specialization, for example a Performer should have some Practice *or* Study, while an Evaluator (e.g. employer) should at least have some Study and a "practitioner" (e.g. DB programmer) should at least have some Practice. We can also discuss how the Lengthy Domain-Related Experience and Reproducibly Superior Performance can be obtained directly through the Domain Community, for instance when someone explicitly describe his perception about the work of a Performer.

Although the limitations are numerous, motivating the need for future works, we have shown all along this chapter that our meta-model of expertise, even in its current state, provides an interesting support, especially to identify the relevant concepts to re-use in our own EF approach and their concrete indicators. As we mentioned, when we want to validate an EF system, we enter a recursive loop: in order to show that the recommended people are experts, we need to get the confirmation from experts, thus we need to use an EF system, which itself need to be validated, and so on. By having such a generic meta-model of expertise, which builds on literature and applies in any domain, we break the need to rely on domain-specific experts by providing another way to evaluate the system. This advantage appears to us as a significant one to justify spending efforts on such a meta-model, and the fact that it can already highlight improvements in existing works supports its current usability.

As a concluding word for this chapter, we can provide a preliminary answer to RQ 3: *How can we support the correction of an existing EF system?* Indeed, we saw that our meta-model can support us in identifying what is cover and what is not covered by existing approaches, thus giving hints on potential improvements. Although we miss the empirical confirmation that these hints are proper requirements for an EF system, making us unable to offer a definitive answer to RQ 3, this offers an additional support as a check-list for designing new techniques. Moreover, it also starts to answer RQ 1: *Can we design an EF process able to consider the core artefacts (topics, terms, and roles) of the two RE approaches?* While it does not say anything regarding the concrete design of such an approach, it appears that topics, terms, and roles can indeed act as concrete indicators of knowledge and social recognition, which are relevant expertise indicators to be considered in an EF system.

Chapter 5

Formalisation of Experts Ranking

A limitation we noticed is that IR metrics, used for evaluating EF systems, consider rankings of experts as exhaustively informed: all expert candidates should be considered (if we consider only top experts, we know they are the top from the whole set of candidates) and people at the same rank have strictly equal expertise. We consider such an assumption to be arguable because the knowledge of people, one of the main components of expertise, is hard to evaluate exhaustively, which is one of the causes making RE tasks hard. In particular, such an assumption generates inappropriate constraints: if humans are unable, because of a lack of information, to decide who is more expert between two people, the ranking produced has them at the same rank, and using it as a gold standard forbids a more automated approach to give one as more expert than the other, although it is able to consider more information than humans. This exhaustive interpretation enforces the production of complete and totally ordered rankings, which (i) give an incentive to humans for selecting arbitrary orders, and (ii) can lead to reject automated rankings because of artificial disagreements with equal ranks, without differentiating them from actual reversed orders. We give a deeper analysis to this problem in [Vergne, 2016b], where we provide mitigation procedures to deal with incompleteness and partial orders with usual IR metrics.

In this chapter, we design metrics naturally adapted to incomplete and partially ordered rankings of experts to avoid additional mitigation procedures. We first introduce the core concepts and procedures in Section 5.1 to deal with rankings, a more generic structure that we call ordering, and the ordered pairs of stakeholders they are composed of. Then, Section 5.2 introduces the notion of "agreement" that we use to compare orderings together, and measures building on it for symmetric as well as asymmetric comparison. Finally, we discuss some limitations of our definitions and measures in Section 5.3.

However, we saw that some aspects are not thoroughly covered for documents rankings, like having rankings which are partially ordered or incomplete, leading to design dedicated measures to mitigate these issues. Additionally, usual ranking conventions happen to be problematic for rankings of experts, like the fact to consider two people at the same rank as equal: if it can be true for races, or even for documents because the total access to their content allows to confirm proper equality, the expertise of two persons is far to be that explicit, leading to prefer an assumption of lack of information, meaning of inability to order them. This shift of interpretation has an impact on the comparison of rankings, and our contribution revises this formalism and reuses IR basic measures (precision and recall) in a novel way, showing that we can fully consider these properties rather than making mitigation procedures for when we face them.

5.1 Definitions and Procedures

In this section, we provide the definitions we use to represent stakeholder rankings, which is the core concept to build our recommendations. We introduce several notions which, while they build on common vocabulary, differ significantly from usual definitions as provided in common order and statistics theories. We highlight these differences and justify them, based on the practical situations that we face in our context of ranking stakeholders having incomplete information, in order to clarify the way we use them and why they deserve to have these differences. We start by defining, in Section 5.1.1, the order atom which orders two stakeholders, the ordering over a set of stakeholders, our main structure, and the more restrictive ranking of stakeholders, which is required to build proper recommendations. Then, we describe in Section 5.1.2 how we build the centroid of a group of orderings, which is the main way for us to obtain a single ordering to represent a group (e.g. represent the average result over several runs). Finally, we show in Section 5.1.3 how we build a ranking from an ordering, which implies to take additional assumptions if the data is incomplete, leading to use it only when we require to use a ranking.

For the redaction of the following sub-sections, we take a generic stance and do not define our concepts on the sole purpose of ordering stakeholders, although we precise what we mean for the specific application to stakeholders. This generic vocabulary allows us to easily link to common order and statistics theories, in which we use the same terms, in order to highlight the similarities and differences. This way, we try to facilitate as much as possible the reading by exploiting the common knowledge provided by these theories, while highlighting how and why we adapt them to our specific purpose.

5.1.1 Order Atom, Ordering, and Ranking

We start by defining an **order atom** as how two elements a and b compare to each other, such as:

- a is *Superior* to b, also written $a{>}b$
- a is *Inferior* to b, also written $a{<}b$
- a and b are *Unordered*, also written $a?b$

We use an order atom to tell which stakeholder between two appears to be more expert than the other, so $a{>}b$ if a appears as more expert than b, $a{<}b$ for the opposite, and $a?b$ if we cannot tell which one is more expert than the other. We do not use the equality because we consider as more natural to think, at least in most of the cases, that being completely informed would allow us to tell which one is more expert than the other, even if it comes from a tiny difference. This thinking is also supported by an observation we made in [Vergne, 2016a], where we gave the possibility to our subjects to build expert rankings which are partially ordered and where all the rankings produced (20) were actually partially ordered. We could observe for instance that the 5 rankings produced for one of our tasks ranked 10 to 13 people into 4 to 6 ranks only, what we consider to be more reasonable to explain through a lack of information than through an equal level of expertise. If we limit ourselves to this thinking, the equal case would make sense mostly to compare a stakeholder to himself ($a = a$), which is not really useful, so we prefer to simplify our definitions by ignoring the equality case and using $a?a$, so we focus on the comparison between different stakeholders. Still, we think that the equality could be useful in some cases, so we discuss it further in Section 5.3, but from a general perspective we consider that the *Unordered* case should be present.

Then, we define an **ordering** o over a set of elements $E = \{e_1, ..., e_n\}$ as the function $o : E \times E \rightarrow \{>, <, ?\}$. Consequently, some pairs of elements can be *Unordered*, which means that an ordering can be *partially ordered.* Additionally, nothing forbid elements to be *Unordered* with *all* the others, so we can remove them without losing information, which means that an ordering can be *incomplete.* Their can also have loops ($a{>}b$, $b{>}c$, and $c{>}a$), so an ordering is *not necessarily transitive.* However, a single ordering should provide a consistent order atom for each pair of elements, so if an ordering returns a given order atom for a couple (a, b), it should return always the same order atom for the same couple, and return the reversed one for the reversed couple (b, a):

$$o(a,b) = \textit{Superior} \Leftrightarrow o(b,a) = \textit{Inferior}$$
$$o(a,b) = \textit{Unordered} \Leftrightarrow o(b,a) = \textit{Unordered}$$

We can write an ordering o over the set $E = \{a, b, c\}$ by enumerating each pair of elements, like $o = (a{>}b, a{<}c, b?c)$, or in a more simple way by ignoring the *Unordered* orders, like $o = (a{>}b, a{<}c)$, optionally choosing a specific direction to highlight, like $o = (a{>}b, c{>}a)$.

Our definitions of order atom and ordering diverges significantly from the usual notion of order as used in order theories, where we usually speak about *partially ordered sets* (among other kinds of ordered sets) building on an order relation $\leq$ ([Simovici and Djeraba, 2008] p. 127). Such sets should be reflexive ($a \leq a$), anti-symmetric ($a \leq b \wedge b \leq a \Rightarrow a = b$), and transitive ($a \leq b \wedge b \leq c \Rightarrow a \leq c$). In our case, we do not consider a single relation ($\leq$) but two:

- "<" which is neither reflexive, anti-symmetric, nor transitive (usually it is transitive),
- "?" which could be considered as reflexive ($a?a$) because we do not use the equality case, although such a design would be arguable, and which is symmetric ($a?b \Rightarrow b?a$), but not transitive.

Notice that $a{>}b \Leftrightarrow b{<}a$, so we can reduce to two relations instead of three, and we reuse a natural interpretation to say that a and b are ordered if $o(a, b) \neq \textit{Unordered}$.

The fact that we use strict orders ($<$) is because we want to exploit evidences that one element *is* greater than another (e.g. a stakeholder a is more expert than another b based on some evidences). Weak orders ($a \leq b$) allow difference as well as equality and need additional information to properly differentiate them (e.g. $b \leq a$ for equality), so it is not adapted to our needs. Additionally, the *Unordered* order (?) allows to explicit the *absence of evidence*, meaning that we miss the information to give any order at all, while usual representations would ignore such lacks or take arbitrary assumptions to complete the data. One could notice that we do not use the *Equal* (=) relation for the complement evidences, saying that no one is greater than another, and could wonder why not using *Equal* instead of *Unordered*. The reason of this choice is because $a = b$ explicitly means that a is, in some way, equivalent to b, for instance a gold standard telling that we should produce an ordering $\hat{o} = (a{>}b, b = c, c{>}d)$ means that we *must* have b and c at the same level. In our case, we want to explicit the fact that we are *not able* to tell which one is greater than the other, meaning that additional information could allow us to tell $b{>}c$ or $c{>}b$. In other words, a gold standard $\hat{o} = (a{>}b, b?c, c{>}d)$ has no conflict with an ordering $o_1 = (a{>}c, b{<}c, c{>}d)$ nor another ordering $o_2 = (a{>}c, b{>}c, c{>}d)$. Finally, discarding the transitivity property is interesting because (i) we exploit some procedures which do not need nor preserve transitivity, like the centroids described in Section 5.1.2, so we can describe these procedures by using our ordering concept, and (ii) we exploit a more restrictive concept of ranking, described below, when we need a transitive property.

We finally define a **ranking** v (we already use r for roles) as an ordering in which the elements can be organized by ranks, with a definition which once again differs from more usual ranking definitions (e.g. $v : E \rightarrow \mathbb{N}^+$, so assigning a rank or ordinal value to each element of the set[1]). Indeed, each element can be assigned an ordinal value (e.g. $1, 2, ...$) telling which rank it belongs to, or no rank at all ($\varnothing$). Whether a rank is assigned to an element a depends on how a relates to the other elements: if it is not ordered at all,

[1] Mapping an element to an ordinal value is a usual definition of ranking in statistics: https://epilab.ich.ucl.ac.uk/coursematerial/statistics/non_parametric/ranking_data.html

then no rank can be assigned to a, while being ordered with at least one other element b allows to infer the rank of a relatively to the rank of b. More formally, we have $v : E \to \mathbb{N}^+ \cup \{\varnothing\}$, and the comparison between a ranking v and its equivalent ordering o can be described as:

$$\begin{aligned} v(a) < v(b) &\Leftrightarrow a{>}b \\ v(a) > v(b) &\Leftrightarrow a{<}b \\ v(a) = v(b) &\Rightarrow a?b \\ v(a) = \varnothing &\Leftrightarrow \forall e \in E, a?e \end{aligned}$$

The one-way implication of the third line is the reason why a ranking is a specific kind of ordering: if any ranking can be represented through an ordering, all orderings *cannot* be represented as rankings. Indeed, with an ordering $o = \{a{>}b, c{>}d\}$ we could assign a rank to a, say 1, which implies to give a higher rank to b, say 2. If we give a different rank than 1 to c it means that we should have $a{>}c$ of $a{<}c$, which is not the case, so we need to give it the rank 1 too. The conflict arises because d should have then a higher rank than c, and so a higher rank than a, which implies to have $a{>}d$ or $a{<}d$, which is not the case either.

For the notation, we write a ranking v over the set $E = \{a, b, c, d, e, f\}$ by ordering each of the elements having a rank, like $v = a{>}b?c{>}d$, which means that $v(a) = 1$, $v(b) = v(c) = 2$, $v(d) = 3$, and $v(e) = v(f) = \varnothing$ because they are not listed. In particular, a ranking which gives no order at all ($\forall e \in E, v(e) = \varnothing$) can be written $v = \emptyset$, which is preferred to $v = a?b?c?d?e?f$ for its clarity (no element is ordered). Applied to stakeholders, we take the usual interpretation regarding differing ranks: being at a *lower rank* (closer to 1) implies to be *more expert* and vice-versa, so the ranking orders the stakeholders by *decreasing expertise*. However, at the opposite of usual representations, being at the *same rank* means that we are *not able* to tell which one is more expert, and not being able to assign any rank means that we have no information at all about the stakeholder.

In short, we adapt the usual ranking definition to consider lacks of information instead of concrete equality, which leads not only to consider that two elements having the same rank are *Unordered* (instead of *Equal*), but also adding the case of not being ordered at all, which leads to have no rank for this specific element (while being equal with all would lead to everyone having the same rank). The first difference (no equality) is important for us because, as we described before, we assume that a complete information would allow us to assign different ranks to each stakeholder in most cases. The second difference (have no rank) is also an advantage, because it provides a natural way to compare rankings which do not rank the exactly same sets of stakeholders (incomplete rankings). For instance we can have some human-made rankings which gives the top 10 stakeholders, with some stakeholders being absent or different between each ranking [Vergne, 2016a], or we can compare human rankings to rankings produced by an automated approach, for which it is a lot easier to rank all the known stakeholders. By considering *Unordered* cases, no mitigation action needs to be taken to align the rankings (e.g. reduce all of them to the intersection, like the homogenisation presented in [Vergne, 2016b]) because we can directly exploit the explicit lacks of information given by the *Unordered* orders.

Despite these differences, we can see that $>$ (resp. $<$) is, as one would expect, transitive for a ranking because $<$ (resp. $>$) is transitive for $\mathbb{N}^+$:

$$\begin{aligned} a{>}b \wedge b{>}c &\Rightarrow v(a){<}v(b) \wedge v(b){<}v(c) \\ &\Rightarrow v(a){<}v(c) \\ &\Rightarrow a{>}c \end{aligned}$$

In brief, we consider here a set of definitions which inspire from usual concepts of order theories, but adapted to a more flexible context where we explicitly consider lacks of information. This flexibility allows us to compute measures which do not require to have total nor complete rankings, neither to make arbitrary assumptions on the missing information.

5.1.2 Centroid of Orderings

In this section, we aim at building a single ordering $\hat{o}$ to represent a group of orderings O. It is built by considering, for each couple of elements (a, b), the most probable order atom depending on the different orderings in O. In such a way, we build a *centroid* for O, meaning an ordering which is "in the center" of O. To compute the order atom of a given couple (a, b), a 2D vector representation is used with Euclidian coordinates (x, y), such that $x, y \in [0; 1]$. In particular, as illustrated in Figure 5.1, we associate a specific vector to each type of order atom:

- *Unordered* $= (0, 0)$ or the null vector
- *Superior* $= (1, 0)$
- *Inferior* $= (0, 1)$

To identify the centroid order atom of a couple (a, b), we compute a weighted average of these vectors, with the weights corresponding to the distribution of these vectors among the orderings of O. More formally, for a set of n orderings, n_s orderings return a *Superior* order for the couple (a, b), n_i return *Inferior*, and n_u return *Unordered*, with $n_s + n_i + n_u = n$. We compute the average vector (x, y) such that $x = \frac{n_s}{n}$ and $y = \frac{n_i}{n}$, which makes it fall between the three cases, and we consider the closest vector to obtain the centroid order atom. In the case of conflicts ($x = 0.5$ or $y = 0.5$), we use *Unordered* as a default, leading to three dedicated areas as shown in Figure 5.1.

Some advantages of this model are the following:

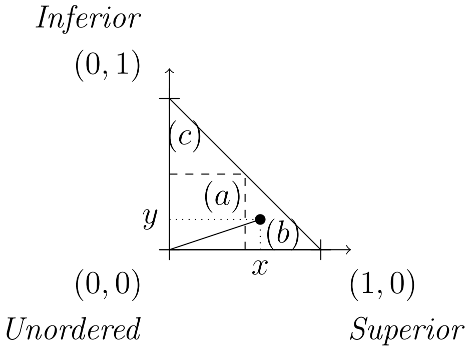


Figure 5.1: Distribution of the three order atoms in a 2D space with their respective areas (a), (b), and (c). An example of vector (x, y) falls in the area (b), thus being interpreted as a *Superior*.

- If a majority of orderings agree on having a given order atom, this order atom will be the one used for the centroid.

- If some order atoms are not represented within the orderings, they are prone to not be used for the centroid too: having only *Inferior* and *Superior* evidences lead to remain on the diagonal between the two, which does not mix with *Unordered* (excepted on the extreme case of the center, which follows the conflict resolution described above).

- This model can be easily extended with an *Equal* order: by assigning it the vector $(1, 1)$ and by computing $x = \frac{n_s+n_e}{n}$ and $y = \frac{n_i+n_e}{n}$, with n_e the number of orderings providing an *Equal* order, we can apply the same procedure while maintaining the previous properties.

It is worth noting that, if O contains transitive orderings (e.g. rankings), the transitivity property is not necessarily preserved. Indeed, by having the rankings $r_1 = a{>}b{>}c$, $r_2 = c{>}a{>}b$, and $r_3 = b{>}c{>}a$, we obtain the centroid $c(\{r_1, r_2, r_3\}) = (a{>}b, b{>}c, c{>}a)$ which is not transitive.

5.1.3 From an Ordering to a Ranking

In our approach, we aim at recommending stakeholders, so at some point we have to decide how to rank them in order to consider the top stakeholders, leading us to build a *ranking* of stakeholders. As it is preferable to apply rounding policies at the end of the computation of numbers, to reduce the rounding errors, it can be preferable to deal with orderings during the computation process, and "round" to a proper ranking only at the end, but in any case we have to obtain a ranking. In order to do so, we use a procedure which is described in Algorithm 1 and contains four steps:

(1-9) retrieve all the order atoms of the ordering (*Superior* orders here),

(11-15) infer the transitive orders ($a{>}b \wedge b{>}c \Rightarrow a{>}c$),

(17-21) reduce the loops to a single rank ($a{>}b \wedge b{>}a \Rightarrow a?b$),

(23-32) build a ranking by looking iteratively for dominant stakeholders.

The first phase retrieves the explicit data, the second phase infers the implicit one, the third phase resolves the over-constrained pairs, and the last phase resolves the under-constrained ones (add arbitrary order atoms to produce a proper ranking). In particular, during this last phase, if the information inferred so far shows that $a{>}b{>}c$ and $d{>}e{>}f{>}g$, without having any information between the elements of the two subsets, then the final ranking arbitrarily merges them into $v = a?d{>}b?e{>}c?f{>}g$. Even if some relations occur, like for $a{>}x{>}b{>}c$ and $d{>}e{>}x{>}f{>}g$, the final ranking arbitrarily merges them into $v = a?d{>}e{>}x{>}b?f{>}c?g$, while it could have been $v = d{>}a?e{>}x{>}b{>}f{>}c?g$ as well as many others. These arbitrary choices having an effect on how the stakeholders are ranked (so how we recommend them), it is important to obtain sufficient information to be able to build a proper ranking (at least for the top stakeholders). This procedure should be

Algorithm 1 Procedure used to build a ranking from an ordering.

```
**Input** $o$: ordering to exploit
**Output** $r$: ranking built
$SUP \leftarrow \emptyset$
$E \leftarrow elementsOf(o)$
for each $(a, b) \in E \times E$ do
    if $o(a, b) = \mathit{Superior}$ then
        $SUP \leftarrow SUP \cup \{(a, b)\}$
    else if $o(a, b) = \mathit{Inferior}$ then
        $SUP \leftarrow SUP \cup \{(b, a)\}$
    end if
end for

for each $(a, b, c) \in E \times E \times E$ do
    if $\{(a, b), (b, c)\} \subset SUP$ then
        $SUP \leftarrow SUP \cup \{(a, c)\}$
    end if
end for

for each $(a, b) \in E \times E$ do
    if $\{(a, b), (b, a)\} \subset SUP$ then
        $SUP \leftarrow SUP \backslash \{(a, b), (b, a)\}$
    end if
end for

$v \leftarrow \emptyset$
$rank \leftarrow 0$
while $|SUP| > 0$ do
    $\mathit{top} \leftarrow \{e \in E | \exists x \in E, (e, x) \in SUP \wedge \nexists y \in E, (y, e) \in SUP\}$
    for each $e \in \mathit{top}$ do
        $v(e) \leftarrow rank$
    end for
    $SUP \leftarrow SUP \backslash \{(e, x) \in SUP | e \in \mathit{top}\}$
    $rank \leftarrow rank + 1$
end while
```

used at the very end, when all the information has been gathered, such that the last phase minimizes the arbitrary choices.

5.2 Measures on Orderings of Stakeholders

[Balog, 2012] presents usual metrics, established by the TREC community, for evaluating EF methods, which are evaluated in exactly the same way as document retrieval systems. From his point of view, this is a reasonable choice, since "*the quality of rankings can be estimated independently of what we rank if quality measures for individual items are alike*". We confirmed from one of the authors that this sentence essentially means it doesn't matter whether we rank documents or we rank experts (or other objects), we can use the same measures. Although we might agree on the feasibility of applying the same measures, we don't see clear evidences that the measures cited are such well-fitted measures. Indeed, we investigated usual IR measures in [Vergne, 2016b], where we highlight their limitations in dealing with incomplete and partially ordered rankings. With incomplete rankings, which means rankings having different sets of stakeholders, measures like precision and recall can be used but they only measure the intersection, so they miss the order, while other measures rely on the rank of the stakeholder, which is unknown for a missing stakeholder. With partially ordered rankings, stakeholders at the same rank lead to conflicts if they are compared to a ranking in which they are ordered, which is justified only if the stakeholders *must* be at the same rank, which is not the case if it comes from a lack of information.

In this report, we proposed different mitigation procedures to fix these issues, but they remain limited and imply to add several levels of computation (one per mitigation), while we consider to be more interesting to use adapted measures which naturally fit to our requirements. In this section, we go a step ahead by designing novel measures able to do so, based on the lessons learned from this previous work. We provide *agreement measures* between orderings in Section 5.2.1, giving us the possibility to compare incomplete and partially ordered rankings (which is just a specific type of ordering). Then, we exploit them to design *symmetric measures* in Section 5.2.2, for comparing two rankings without considering any as a reference. Finally, Section 5.2.3 describes *reference-based measures*, so we can properly assess how much a ranking complies to some gold standards.

5.2.1 Agreement Between Orderings

By comparing two orderings o_1 and o_2, we can look at how they order their different elements, and see how much they agree or disagree. For instance, if both o_1 and o_2 provide *Superior* for the couple (a, b), then they have an *Agreement*, while if one of them provide *Superior* and the other *Inferior*, then they have a *Disagreement*. Given that both orderings order a set of stakeholders S, we can count how many *Agreement*s and *Disagreement*s occur for computing comparison statistics. However, because we can have *Unordered* orders (at the opposite of a proper equality), we also have to consider unknown agreements when it occurs. These unknowns, moreover, can be considered in several ways: we can simply assess the unknown agreement (*Indifference*), or use an optimistic (resp. pessimistic) perspective to use *Agreement* (resp. *Disagreement*) as a default value. The complete logics is represented in Table 5.1 and more concisely illustrated in Figure 5.2.

One should notice that it is possible to extend our definitions with proper equality cases by extending correspondingly the table, without changing anything regarding the agreement values we consider: *Agreement*, *Disagreement*, or *Indifference*. It would be then the responsibility of the designer to assign *Agreement* or *Disagreement* to the additional combination, or *Indifference* when the choice is not straightforward. The optimistic (resp. pessimistic) perspective can be easily extended for that purpose, the idea being of using

$o_1(a,b)$	$o_2(a,b)$	Comparison	Optimistic	Pessimistic
Unordered	*Unordered*	?	$\surd$	$\times$
Unordered	*Superior*	?	$\surd$	$\times$
Unordered	*Inferior*	?	$\surd$	$\times$
Superior	*Unordered*	?	$\surd$	$\times$
Superior	*Superior*	$\surd$	$\surd$	$\surd$
Superior	*Inferior*	$\times$	$\times$	$\times$
Inferior	*Unordered*	?	$\surd$	$\times$
Inferior	*Superior*	$\times$	$\times$	$\times$
Inferior	*Inferior*	$\surd$	$\surd$	$\surd$

Table 5.1: Comparison between two orderings o_1 and o_2: looking at how they order a given couple (a,b) leads to *Indifference* (?), *Agreement* ($\surd$), or *Disagreement* ($\times$) for this couple. An optimistic measure assumes that *Indifference* is by default an *Agreement*, while a pessimistic measure assumes that *Indifference* is by default a *Disagreement*.

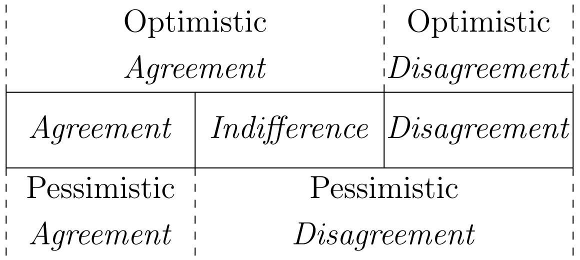


Figure 5.2: The three types of order comparison (*Agreement* for same order atom, *Disagreement* for reversed order atom, and *Indifference* if at least one *Unordered*) and how they are interpreted with an optimistic or pessimistic perspective.

Agreement (resp. *Disagreement*) instead of *Indifference*.

5.2.2 Measures for Symmetric Comparisons

From the different measures provided in Section 5.2.1, we obtain absolute values identifying the number of *Agreement*s, *Disagreement*s, and *Indifference*s between two orderings. By trying to show how far two orderings are, we can compute different distances based on their *Disagreement*s. In order to make such a distance meaningful, it is important to normalize it, so that we can identify when two rankings are close (distance close to zero) or far (distance close to one) without needing additional knowledge about them. Thus, we can design distances based on $A(o_1, o_2)$, $D(o_1, o_2)$ and $I(o_1, o_2)$ (we remove the arguments in the following to simplify the reading) which are respectively the numbers of *Agreement*s, *Disagreement*s and *Indifference*s between two orderings o_1 and o_2:

$$DD = \frac{D}{A + D} \tag{5.1}$$

$$ODD = \frac{D}{A + I + D} \tag{5.2}$$

$$PDD = \frac{I + D}{A + I + D} \tag{5.3}$$

DD (Disagreement Distance) translates the basic comparison, where we consider only the explicit *Agreement*s and *Disagreement*s, ignoring the *Indifference*s. ODD (Optimistic DD) translates the optimistic perspective, where we assume that *Indifference* is like *Agreement*, thus adding it to the denominator but not to the numerator (which only cares about *Disagreement*s). PDD (Pessimistic DD) translates the pessimistic perspective, where we assume that *Indifference* is like *Disagreement*, thus adding it to both the denominator and the numerator.

We can already identify some limitations for each distance. DD does not count *Indifference*, so it is hard to see when a ranking is close to (resp. far from) another because it has a lot of *Agreement*s (resp. *Disagreement*s) or just because there is a lot of unknowns. ODD counts all the available data but, as its name implies, it does not count *Indifference* as a *Disagreement*, thus we can be artificially close to any other ordering by giving one which is completely uninformative ($\forall e_1, e_2 \in E, o(e_1, e_2) = Unordered$), leading to have no explicit *Disagreement*. Although PDD counts *Indifference* as a *Disagreement*, it does not make the difference between an ordering which is simply less informative or actually disagreeing. In other words, both the limitations of ODD and PDD correspond to a different specialization of the limitation of DD: we could rewrite $ODD = \frac{D}{A'+D}$ and $PDD = \frac{D'}{A+D'}$ with $A' = A + I$ and $D' = D + I$, thus one corresponds to DD with a given assumption and the other to DD with the opposite assumption.

However, while we see that DD can be advantageously replaced by ODD or PDD to consider more information, it appears that, because of their complementarity, exploiting both ODD and PDD is even more interesting to properly mitigate their limitations. In particular, ODD gives a lower bound of DD, PDD gives it an upper bound, and the difference between them provides a normalised indicator of the amount of *Indifference*. We provide more details in appendices, please refer to Section A.1 for proofs of these three claims.

Thus, by providing both ODD and PDD, we can have a meaningful evaluation of the distance between two orderings. For instance, assuming that we have an automated technique which creates rankings by using some randomization, we can expect it to produce completely different rankings at the start, leading to have high ODD and PDD between them. Then, after some rounds of improvements, we might start to see some convergence, leading to the disappearance of obvious *Disagreement*s which decreases ODD but not PDD. With some additional rounds, they might converge further to proper *Agreement*s, forcing PDD to decrease as well, until we reach a final state where all the rankings produced by the approach are the same. By plotting such an evolution, as illustrated in Figure 5.3, it is easy to evaluate the amount of *Agreement*, *Disagreement*, and *Indifference*, so 3 parameters, just by plotting the 2 curves of ODD and PDD.

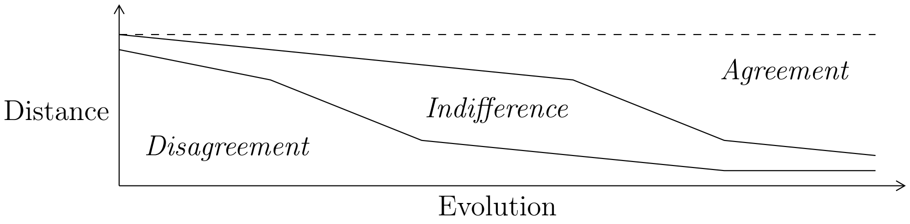


Figure 5.3: Example of graph showing the evolution of ODD (bottom curve) and PDD (top curve) when the agreement increases. We expect this kind of curve when the rankings produced by an automated technique converge to a stable, unique ranking.

Thinking back about the more usual case where no *Unordered* order is considered, leading to no *Indifference* agreement (i.e. $I = 0$), we have $DD = ODD = PDD$. Thus, by considering the specific case where no *Unordered* order is used, the two boundary measures ODD and PDD can be reduced to the single measure DD, which shows that we deal with a generalization of this specific case. For further investigation on these measures, the interested reader can refer to Appendix A, where we show equivalences with measures based on *Agreement* (Section A.2) and a comparison to usual IR measures (Section A.3). We also show a comparison to Kendall's τ coefficient (Section A.4) which is used in the tau test, a statistical test which does not rely on any assumption regarding the distribution of the data (non-parametric statistics).

5.2.3 Measures for Compliance to a Reference

In complement to distances, which assume symmetry (or commutativity) such that $d(o_1, o_2) = d(o_2, o_1)$, we also need to consider the cases where we want an ordering to comply to some references, so making an argument "more important" than the other. For example, if a reference $\hat{o}$ provides an *Unordered* order atom, then an ordering o may provide any order atom, which is an optimistic perspective as shown in Table 5.1, while if $\hat{o}$ orders two elements, then o must comply, which is a pessimistic perspective. So we cannot directly use measures like ODD or PDD, which take only one perspective at a time, neither we can directly use DD, which just ignores *Unordered* order atoms. Consequently, in this section we define functions to measure the quality of an ordering o based on a reference ordering $\hat{o}$.

We start by defining two basic functions:

$$Orders(o, \eta) = |\{(a, b)|o(a, b) = \eta\}| \tag{5.4}$$

$$Shares(o_1, o_2, \eta) = |\{(a, b)|o_1(a, b) = o_2(a, b) = \eta\}| \tag{5.5}$$

Equation 5.4 simply counts the number of times a given ordering o provides order atoms of a given type $\eta \in \{\mathit{Unordered}, \mathit{Superior}, \mathit{Inferior}\}$, while Equation 5.5 counts the number of times two given orderings share the same order atoms of a given type η. In Section 5.1.1, we already mentioned that $o(a, b) = \mathit{Superior} \Leftrightarrow o(b, a) = \mathit{Inferior}$, thus it is worth noting that $Orders(o, \mathit{Superior}) = Orders(o, \mathit{Inferior})$:

Proof.

$$\begin{aligned} Orders(o, \mathit{Superior}) &= |\{(a, b)|o(a, b) = \mathit{Superior}\}| \\ &= |\{(a, b)|o(b, a) = \mathit{Inferior}\}| \\ &= |\{(b, a)|o(b, a) = \mathit{Inferior}\}| \\ &= |\{(a', b')|o(a', b') = \mathit{Inferior}\}| \\ &= Orders(o, \mathit{Inferior}) \end{aligned} \qquad \square$$

Similarly, we have $Shares(o_1, o_2, \mathit{Superior}) = Shares(o_1, o_2, \mathit{Inferior})$, which can be proved with the same strategy. This is important to know because, in order to design our quality measures, we do not want to count several times the same pairs, so we can arbitrarily choose *Superior* or *Inferior* to consider all the ordered pairs instead of considering both. Additionally, it is also worth noting that $Shares(o_1, o_2, \eta)$ is commutative on the orderings, so $Shares(o_1, o_2, \eta) = Shares(o_2, o_1, \eta)$, because equality is also commutative, so $o_1(a, b) = o_2(a, b) \Leftrightarrow o_2(a, b) = o_1(a, b)$. This commutativity implies that $Shares(o_1, o_2, \eta)$ makes no difference between the two orderings, so this is through the use of $Orders(o, \eta)$ that we can differentiate the reference $\hat{o}$ to the other ordering o.

In this work, we compose these functions in several reference-based measures. We replace the order terms by their corresponding symbols for clarity (e.g. *Superior* by $>$):

$$TotalComp(\hat{o}, o) = \frac{Shares(\hat{o}, o, >) + Shares(\hat{o}, o, ?)}{Orders(\hat{o}, >) + Orders(\hat{o}, ?)} \tag{5.6}$$

$$OptimComp(\hat{o}, o) = \frac{Shares(\hat{o}, o, >) + Orders(\hat{o}, ?)}{Orders(\hat{o}, >) + Orders(\hat{o}, ?)} \tag{5.7}$$

$$OrderComp(\hat{o}, o) = \frac{Shares(\hat{o}, o, >)}{Orders(\hat{o}, >)} \tag{5.8}$$

Equation 5.6 aims at ensuring total compliance, so the order atoms provided by the reference $\hat{o}$ must all be provided by the ordering o to reach the maximal value ($TotalComp(\hat{o}, o) = 1$), while providing any different order atom decreases this value until no one is actually shared ($TotalComp(\hat{o}, o) = 0$). Equation 5.7 is more optimistic, in the sense that any ordered pair (*Superior* or *Inferior*) provided by the reference $\hat{o}$ must be found in o, while an *Unordered* order atom in $\hat{o}$ allows any order atom in o. Having only a few ordered pairs in $\hat{o}$ may lead to dramatic values if we *ignore* the *Unordered* order atoms, while it is in fact negligible because the reference is just extremely permissive, which is why we consider them in this equation. At the opposite, Equation 5.8 ignores the *Unordered* order atoms, which may be useful if one wants to focus on the constrained part only, for instance if a gold standard is aimed at providing only few constraints which *must* be satisfied.

Although we provide new measures, like the symmetric measures of Section 5.2.2, we actually build on existing ones: both the equations 5.6 and 5.8 are also *recall* measures, the former on the full set of order atoms and the latter on the ordered ones only. By considering both *Superior* and *Unordered*, we make it possible to have different variants (with a precise semantic) and to add new measures like Equation 5.7. It may also be of interest to see that the notions defined in the previous section, namely the *Agreements* $A(o_1, o_2)$, the *Disagreements* $D(o_1, o_2)$, and the *Indifference* agreements $I(o_1, o_2)$, and by extensions the equations building on them, can be redefined with the same functions:

$$\begin{aligned} A(o_1, o_2) &= Shares(o_1, o_2, \mathit{Superior}) \\ D(o_1, o_2) &= Orders(o_1, \mathit{Superior}) - Shares(o_1, o_2, \mathit{Superior}) \\ I(o_1, o_2) &= Orders(o_1, \mathit{Unordered}) \end{aligned}$$

While we do not exploit these properties in this work, it helps to support the consistency of our formalism by showing that it can be applied more broadly. In particular, we can see that summing the three provides, as expected, the total number of pairs of o_1 ($Orders(o_1, \mathit{Superior}) + Orders(o_1, \mathit{Unordered})$), and because we can also do the same by replacing o_1 by o_2 and o_2 by o_1 in the right-hand side of the equations, it is also the total number of pairs of o_2. One could wonder about the case where o_1 and o_2 provide order atoms on different pairs, but this would lead to an inconsistent comparison which needs additional assumptions to deal with the misalignments. In such a case, both orderings should be extended correspondingly to provide the same pairs, with the additional pairs being *Unordered*, which is the complementary process of removing uninformative elements from an ordering, as described in Section 5.1.1.

5.3 Discussion

A remaining limitation of our measures is on the compliance of an incomplete ordering if compared to an incomplete reference. A scenario similar to one we face our evaluations (Chapter 8.3) is the following: we use as reference a human-made ranking of n stakeholders based on a part of the full dataset, so it does not necessarily include the best stakeholders but provide the right orders for the ones considered, while we generate a computer-made ranking over the full dataset but with a limited size of n for performance purpose. Let assume now that 1 of the stakeholders of the full dataset is more expert than any stakeholder of the reference, and that the computer-based ranking is so good that it provides a fully compliant ranking, but with this additional stakeholder at the top, thus ignoring the last stakeholder of the reference. In such a case, the $n-1$ pairs related to the best stakeholder and the $n-1$ pairs related to the ignored stakeholder cannot be shared with the reference, leading even $OptimComp(\hat{o}, o)$ –which is designed to be optimistic– to significantly decrease its value. In the worst case, if there is n or more better stakeholders than the ones provided by the reference, then the compliance level would reach zero because of the size limitation rather than because of the wrong orders. Consequently, a particular care should be taken in such a situation to clarify what are the best achievable values, otherwise even good rankings might not appear as so.

Another arguable point is the addition of ordered pairs when transforming an ordering to a ranking. Although we agree that it should be avoided, it comes with the requirement of having a single ranking as a result of the EF process. If we allow to have several complementary rankings or a graph, then we can produce a different representation which is fully compatible with the original orderings, but which is more complex to read for the user.

We can also imagine other centroid definitions, for instance we used a variant to build some gold standard rankings for Chapter 8.3 which gives priority to ordered pairs by using *Unordered* only when we have a strict equality between *Superior* and *Inferior*. From our experience and thinking, it seems that different weights should be given to the *Unordered* pairs depending on the situation. For instance, if a human provide an *Unordered* pair because he does not know whether or not the available information allows to order them, then this pair should have less weight than if he is certain that the available information is not sufficient to order them. Indeed, while the first case does not take any position, the second tend to be like an equality case, claiming that other people should not provide *Superior* nor *Inferior*. Probably the use of an equality case should be investigated further with this idea in mind.

Despite these limitations, we shown that our formalism is a good way to answer to our RQ 2: *How can we compare incomplete and partially ordered rankings of experts?* Indeed we shown, especially with the help of Figure 5.3, how the combination of PDD and ODD allows to compare rankings while having also an overview of the amount of information lacking through the amount of *Indifference* ($PDD - ODD$). we also shown that our measures build on broadly used measures, like recall and Kendall's τ coefficient [Kendall, 1938], what we consider to be evidences of usefulness and robustness. However, we should highlight the lack of measures giving priority to top stakeholders, as it is usually done in some common IR measures. Consequently, we consider our formalism to be a good way to deal naturally with incomplete and partially ordered rankings, although more work is needed to provide a set of measures as rich as the current state of the art.

Chapter 6

Expert Finding Approaches

We present in this chapter our main approach, which aims at recommending experts by integrating indicators already used in RE works like [Castro-Herrera and Cleland-Huang, 2009] and [Lim et al., 2010]. We first give an overview of the approach in Section 6.1 to make the reader aware of each component of the approach. Each of them is then detailed, starting from Section 6.2 with the extraction of relevant data from sources to build a weighted graph, before to focus on the query building in Section 6.3. We dig further in the details in Section 6.4, which describes the two kinds of inference that we decided to investigate (MN and GA), highlighting the motivations behind our choices and providing the necessary formalism to understand how they work. Finally, we conclude on the advantages and limitations of our two variants and on more general aspects of our approach in Section 6.5.

Because we intend to design a generic process, trying to abstract from the specificities of the various contexts and RE tasks, we did not make a deep requirements elicitation and analysis for designing our EF approaches. We exploited evidences found in the literature in Psychology and existing RE works to design our system, but a deeper investigation on the precise requirements to implement would be of great interest, in particular to establish the implementation to use for a specific context and a specific RE task.

6.1 Global Picture

In order to recommend stakeholders as experts, we extract information from available *sources of data*, which can be written documents, like forum messages, e-mails or reports, or models, like goal-models or social networks built from social recommendations. We use two kinds of extractors on these sources: a *node extractor*, which retrieves the relevant entities to consider (i.e. stakeholders, roles, topics and terms), and a *relation extractor*, which retrieves the amount of co-occurrences of these artefacts. We split the extraction process because we do not make any assumption on which source will provide the relevant nodes and relations and in which order they will be parsed: by extracting the nodes first, we ensure that the following relation extraction step will consider all the relevant nodes. Once the information is extracted from each source, we aggregate the nodes and relations into a common weighted graph that we can then process to obtain quantitative results on the stakeholders.

Once the weighted graph is extracted, we build the query based on the properties searched for the experts to recommend (having some roles, knowing about some topics, or using some terms). For instance, if the network contains a topic *security*, it is possible to query for an expert in this topic, possibly combining it with other topics, but also with roles and terms. If the artefact is not present, it cannot be queried and an equivalent need to be found, e.g. *cryptography*, which is in the network. In our approach, when we look for an expert in a topic which is not in the network, we replace this topic by the corresponding term if it exists, otherwise we ignore it. Notice that querying for an expert with a given role does not necessarily mean that only people having this role will be considered (it is not a filtering function), but that people being more related to this role (directly or indirectly, as described in the model) will be considered as more experts. The interpretation

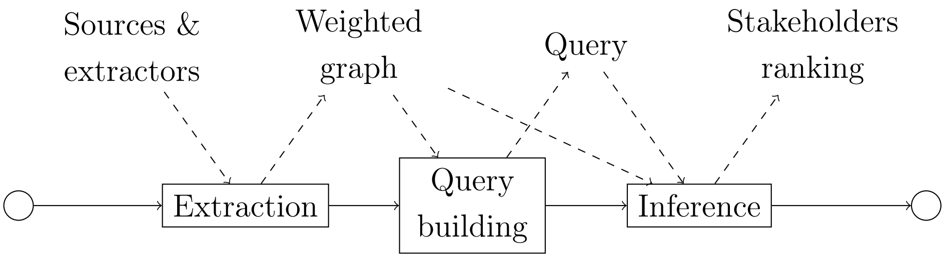


Figure 6.1: Recommendation process, from left to right, with the artefacts on the top and the tasks on the bottom. The directed arrows shows the inputs and outputs of each task.

here is that the person has more expertise in the role queried, whether or not she is officially assigned to this role: maybe she were assigned to it in the past, or maybe she were particularly involved with people having this role, or other reasons justifying that she could have some relevant expertise.

Once the weighted graph is built and the query known, we search for the relevant stakeholders: we investigate here the use of a MN in Section 6.4.1 and a GA in Section 6.4.2. The MN approach has been the first studied and lead to several publications [Vergne et al., 2013, Morales-Ramirez et al., 2014, Vergne and Susi, 2014], while the one based on GA came later to investigate other aspects not considered with the first approach, like performance. In both cases, the process can be described as (i) starting from the nodes corresponding to the artefacts of the query, (ii) exploiting the weighted relations to retrieve the most relevant nodes, (iii) rank the stakeholders by decreasing relevance. The way in which the relevance is computed for each case is detailed in the corresponding section, and the complete process is illustrated in Figure 6.1.

6.2 Extraction Process

6.2.1 Sources & Extractors

Sources can be structured, like goal models and other diagrams, or not, like e-mails or forum messages, but what is important for us is the kind of information they provide. In particular, because we want to infer a ranking of stakeholders, we should have at least one source describing the people for which we can evaluate the expertise. Then, because we want to infer such ranking from a set of roles, topics, and terms, we should have some sources from which we can extract them, at least one in which we have the topics/terms we want to query. Finally, because the relations between these elements will be used to establish the ranking, we need to have a graph with reliable data. This reliability can be achieved by exploiting highly trustful sources (e.g. official documents, established experts), or by relying on numbers to make the irrelevant information negligible by using a lot of reasonably trustful data (e.g. involvement in several projects, rich social network). In this work, we do not investigate the notion of trust (e.g. document obsolescence, expert biases) nor how the data should be scaled (we use weighted relations), but we investigate inference techniques providing some freedom in choosing such scales. This means that the sources used should be preliminary assessed as trustful by some authority, and that this trust should be represented by the weights used in the extractors described below.

Our approach is designed for dealing with available resources, rather than exploiting specific ones, what we do by using the notion of *extractor*: this is the combination of sources and adapted extractors which allows to extract the relevant information. Extractors are of two kinds: a *node extractor* intends to extract the relevant stakeholders, roles, topics and terms, while a *relation extractor* intends to count the amount of co-occurrence of two artefacts (e.g. how much a given stakeholder is prone to appear with a given term). For instance, we have implemented an extractor for e-mails, which exploit the authors and terms used to identify stakeholders, topics, and terms for the graph, and an extractor for goal models which exploit the architecture of the model to identify roles, topics, and terms. We present the first one in our evaluation in Part III (algorithms 6 and 7), but other extractors can be used depending on the context and we can imagine to have a library of generic or specific extractors to reuse in practice. These extractors take some decisions on which content to exploit from the sources, how to exploit it, and in particular how to scale the weights extracted. As a summary, given the available sources and the extractors designed to exploit them based on the trust we have, the objective is then to extract the relevant artefacts and weighted relations in order to build our weighted graph.

Algorithm 2 shows how we first extract the nodes before to extract their relations. Notice that one extractor can be used on several sources, for instance using a noun extractor on any textual document to retrieve its terms. Similarly, several extractors can be used on one source, in particular to combine extractors dedicated to specific types of nodes/relations. For the extraction of the relations, the evidences (relation weights) are summed over the whole set of sources, so the final weight of a relation correspond to the sum to the weights of all the similar relations extracted. Thus, a particular attention should be given to redundant data in case one does not want to exploit such a redundancy.

6.2.2 The SRTC Graph

In order to model the experts and their expertises, we use a *weighted graph* representing the instances of the concepts and relations previously defined. We have a set of stakeholders S which are the Performers we want to evaluate (if we refer to our meta-model of expertise in Chapter 4), a set of roles R supporting the Social Recognition, a set of topics T and a set of terms C

Algorithm 2 Sources extraction process.

```
Input $IN$: Sources of data
Input $E_n$: Node extractors
Input $E_r$: Relation extractors
Input $M_n : IN \to 2^{E_n}$: Function providing the node extractors applicable to each source
Input $M_r : IN \to 2^{E_r}$: Equivalent function for relation extractors
Output $S, R, T, C$: Extracted Stakeholders, roles, topics and terms
Output $L$: Extracted weighted relations
// Extract nodes
$S, R, T, C \leftarrow \emptyset$
for each $in \in IN$ do
    for each $ex \in M_n(in)$ do
        $\{S_{in,ex}, R_{in,ex}, T_{in,ex}, C_{in,ex}\} \leftarrow ex(in)$
        $S \leftarrow S \cup S_{in,ex}$
        $R \leftarrow R \cup R_{in,ex}$
        $T \leftarrow T \cup T_{in,ex}$
        $C \leftarrow C \cup C_{in,ex}$
    end for
end for

// Extract weighted relations
for each $in \in IN$ do
    for each $ex \in M_r(in)$ do
        $L_{in,ex} \leftarrow ex(in, S, R, T, C)$
        $L \leftarrow L \uplus L_{in,ex}$
        // The symbol $\uplus$ (multiset sum) acts as a union
        // symbol which sums the weights of similar relations,
        // so $\{\langle a,b,2\rangle, \langle a,c,1\rangle\} \uplus \{\langle a,b,3\rangle\} = \{\langle a,b,5\rangle, \langle a,c,1\rangle\}$.
    end for
end for
```

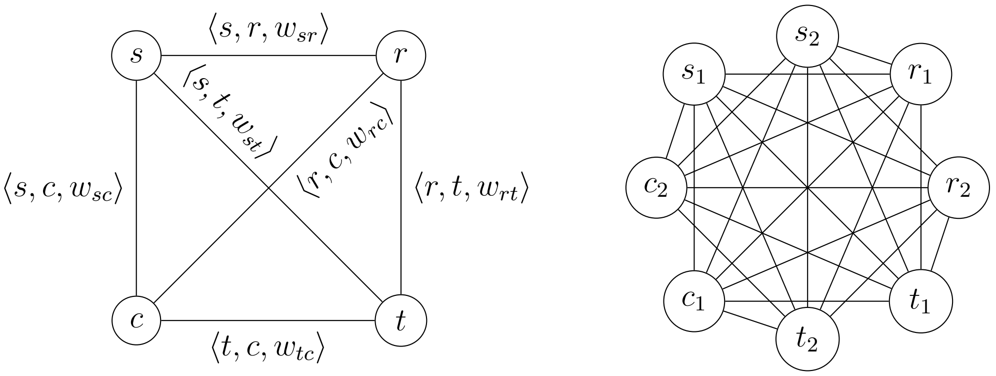

$\langle s, r, w_{sr} \rangle$
s
r
$\langle s, t, w_{st} \rangle$
$\langle r, c, w_{rc} \rangle$
$\langle s, c, w_{sc} \rangle$
$\langle r, t, w_{rt} \rangle$
c
t
$\langle t, c, w_{tc} \rangle$
s_2
s_1
r_1
c_2
r_2
c_1
t_1
t_2

Figure 6.2: Examples of graphs with 1 node (left) or 2 nodes (right) for each S, R, T and C, showing the different relations and the 4-partite structure (only nodes of the same type are not related).

which can be used to build the Perceived Domain Knowledge of the Evaluator. These sets provide the nodes of a graph, and we use weighted edges between these nodes to represent the extent to which they are correlated. Basically, each stakeholder in S is related to all elements in R, T and C, each role in R is related to all elements in S, T and C and equivalently for each topic in T and each term in C, forming a complete 4-partite graph, as shown in Figure 6.2. The weight of an edge represents the amount of evidences supporting the co-occurrence of the two nodes it relates. For instance, if we have no evidence that a stakeholder $s \in S$ knows about a topic $t \in T$, these nodes are related by an edge with a zero weight written as a tuple $\langle s, t, 0 \rangle$. Having the tuples $\langle s_1, t, 5 \rangle$ and $\langle s_2, t, 10 \rangle$ describes two relations showing that we have twice the amount of evidence that s_2 knows about the topic t compared to s_1.

The actual value of the weight depends on the interpretation of *evidence*. [Lim et al., 2010], in their social network, use the *salience* (i.e. power or influence) elicited from the stakeholders to weight their edges, while [Castro-Herrera and Cleland-Huang, 2010] exploit the *frequencies* of appearance of terms and normalise them in vectors. Both these approaches as well as others can be exploited, with the main challenge being to have comparable weights for proper inference. For instance, the salience values of [Lim et al., 2010] are in $\{1, ..., 5\}$ and can be used as weights for relating stakeholders to the roles they have. The frequencies of [Castro-Herrera and Cleland-Huang, 2010] can be used as weights for the relations between stakeholders and terms (if built from messages from these stakeholders) or between topics and terms (if built from documents related to identified topics). We do not intend to constrain the scale of the weights more than required: the only constraint which seems justified to us is to have the same scale for all the relations of the same type (i.e. relating the same types of nodes), so we can safely compare information of the same kind. Due to that, we selected the inference techniques to investigate also depending on the freedom they provide on these scales.

6.3 Query Building

Once the stakeholders, roles, topics, and terms are identified and related, our goal is to identify the expertise need of the user, thus which stakeholder to recommend as an expert. By expert, we mean that the user is looking for someone being knowledgeable on something, and he has several ways to look for it: a specific role (having knowledge which is normally delegated to this role, such as responsibilities), knowing about a specific topic (having a broad knowledge on a given subject) or knowing some specific terms (using the corresponding vocabulary). The need can also be composed, for instance looking for people covering some roles and more specifically knowing about several topics. Consequently, the query we consider is the set of roles, topics and terms the user is looking for $Q \in 2^{R \cup T \cup C}$, for instance $Q = \{t_{cryptography}\}$ or $Q = \{r_{developer}, c_{encrypt}, c_{RSA}\}$ with $r_x \in R$, $t_x \in T$, and $c_x \in C$. We do not investigate the possibility to give more influence to some query nodes by weighting them, for example if we are interested about a topic in particular but additional knowledge on some other topics would be appreciated, so we let this aspect for future works and focus on queries as simple sets of nodes.

To obtain this query, the user may explicitly select the artefacts to query, otherwise we need to infer them from another kind of input, like natural language questions. For the implementation of our approach, we use a simple keyword-based system, so the user provides a sequence of terms from which we retrieve the relevant artefacts in our weighted graph. All the topic nodes corresponding to query terms are used, then for the remaining query terms we use the corresponding term nodes, and finally for the last query terms we consider the corresponding role nodes. Any term not found as a topic, term or role in the graph is ignored.

This query parsing process, although simple, appears to us as a natural one, because most of the EF techniques use topic-based queries. A more advanced building could be of interest, like relying on synonyms if the exact term is not part of the graph, or using annotations to explicit the kind of node to use (e.g. for a query term *developing*, should we use the term node $c_{developing}$, the topic $t_{development}$, or the role $r_{developer}$?). We did not investigate such kinds of improvements because we remained in an experimental setting, giving us the opportunity to focus more on the approach itself while keeping the interface with the user minimal. Further investigation on this aspect would be needed to design a usable tool, what falls out of the scope of this thesis and relate to future works.

6.4 Inference Techniques

Once the artefact network has been built and the query is known, we can search through the artefact network the elements which are the most relevant to the query, until we find out the most relevant stakeholders to recommend.

For this process, we investigated two different approaches: the *MN* in Section 6.4.1, which was the technique introducing the least assumptions to process our weighted graph, and another based on a *GA* in Section 6.4.2, which is more customized and more adapted to huge computation.

6.4.1 Markov Network

Initially, we were looking for a structure which is as close as possible to our weighted graph, to minimize the adaptation effort and minimize the additional assumptions, and existing computation methods on this structure. Our research lead us to consider graphical models, like Markov and Bayesian Networks, and the most adapted one appeared to be the MN [Kindermann and Snell, 1980] due to its ability to deal with our undirected graph. They are often used as Markov Logic Network [Richardson and Domingos, 2006], which adds a layer of first order logic to make it more expressive among other advantages. However, we saw that this additional layer implies a specific use of the MN layer (i.e. specific potential functions, as described later) which did not fit our objectives. Consequently, we focused on the MN alone and investigated different ways to use it, which is what we describe in this section.

From the SRTC Graph to the Markov Network

Before to introduce the technique we use to build expert rankings, we introduce some basic notions of the MN, also called Markov random field.

A *random variable* is a variable having several possible states, each with a specific probability. For instance, a random variable x can have a binary state in $V_x = \{\top, \bot\}$ with probabilities $P(x = \top) = 0.8$ and $P(x = \bot) = 0.2$. By linking several variables in a graph, like in Figure 6.3, we can identify groups of nodes completely linked, i.e. all the nodes of the group are linked to all the other nodes of the group. The nodes composing such a complete sub-graph is called a *clique*. In our example, interesting cliques are all the

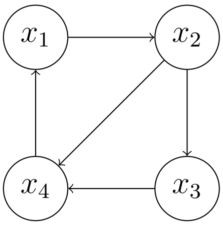


Figure 6.3: Example of simple MN.

pairs of related variables (any link makes a clique of two nodes) but also $\{x_1, x_2, x_4\}$ and $\{x_2, x_3, x_4\}$ (x_1 and x_3 are not linked, so they cannot be in the same clique).

Once the variables are defined and linked in a graph, one can take a clique $g = \{x_1, ..., x_n\}$, where x_i can take any state in V_i, and define a *potential function* $f_g : V_1 \times ... \times V_n \rightarrow \mathbb{R}^+$ which returns a real value based on the states of the variables in the clique. For instance, we may define a potential function on the clique $g = \{x_1, x_2, x_4\}$ such that $f_g(\bot, \bot, \bot) = 0$, $f_g(\bot, \bot, \top) = 5$, ..., $f_g(\top, \top, \top) = 3$. Finally, a MN $N = (X, F)$ is defined via a set of random variables $X = \{x_1, ..., x_n\}$ and a set of potential functions $F = \{f_1, ..., f_m\}$ over cliques in X. While nodes and weights are extracted from the sources of data, we also need to define the potential functions, what we do in details later in this section.

In the specific case where all the potential functions are defined on pairs of nodes, the MN represents a weighted graph, where the weights of the links depend on the states of the nodes. For example, a simple translation of a tuple $\langle n_1, n_2, w \rangle$ (two nodes n_1 and n_2 related with a weight of w) into a potential function f over $\{n_1, n_2\}$ can be to assign w to $f(\top, \top)$ and zero to the other states, or any other transformation of w. Consequently, we can represent our SRTC graph as a MN, where the artefacts are represented by binary random variables and the weighted relations are used to define the potential functions. Regarding the interpretation of the MN, the binary state of each node tells whether the node (stakeholder, role, topic or term) is relevant ($\top$) or not ($\bot$). In the specific case of a stakeholder, a relevant stakeholder is interpreted as an expert, and this state should be computed depending on the query.

Probability Computation

By computing a MN, one infers the probabilities of each state of each random variable based on the potential functions. Considering the nodes $X = \{x_1, ..., x_n\}$, where x_i is assigned a state $v_i \in V_i$, and each clique g_i assigned to a potential function f_i, the probability to be in a specific state $\chi = \{v_1, ..., v_n\}$ is computed as:

$$P(\chi) = \frac{\prod\limits_{i=1}^{m} f_i(g_i)}{Z}$$

With $Z = \sum\limits_{\chi} \prod\limits_{i=1}^{m} f_i(g_i)$ the normalisation factor which allows to build a probability ($\sum\limits_{\chi} P(\chi) = 1$).

An interesting property of a MN is its *global scale independence*: if we apply a scaling factor α on the potential functions $f'_i = \alpha.f_i$ and compute the probability P' based on these functions, we can see that we get the same results:

$$P'(\chi) = \frac{\prod\limits_{i=1}^{m} f'_i(g_i)}{Z'} = \frac{\prod\limits_{i=1}^{m} \alpha.f_i(g_i)}{Z'} = \frac{\alpha^m \prod\limits_{i=1}^{m} f_i(g_i)}{Z'}$$

$$Z' = \sum_{\chi} \prod_{i=1}^{m} f'_i(g_i) = \sum_{\chi} \prod_{i=1}^{m} \alpha.f_i(g_i) = \alpha^m \sum_{\chi} \prod_{i=1}^{m} f_i(g_i) = \alpha^m Z$$

$$P'(\chi) = \frac{\alpha^m \prod\limits_{i=1}^{m} f_i(g_i)}{\alpha^m Z} = \frac{\prod\limits_{i=1}^{m} f_i(g_i)}{Z} = P(\chi)$$

This property is of particular importance because it reduces the initial constraints on the extracted data, so one can choose any arbitrary scale. However, this is a global scale: the scale should be the same for every type of relation, which seems too much constraint to us.

Going further in the computation of a MN, if we are interested in a subset of variables $X' \subset X$, one can compute a *partial probability* $P(\chi' \subset \chi)$ by computing all the cases where χ' holds and summing them. For instance, with $X = \{x_1, x_2\}$, $P(x_1 = \top) = P(x_1 = \top, x_2 = \top) + P(x_1 = \top, x_2 = \bot)$. If some variables have a known state, we can compute a *conditional probability* $P(\chi'_1|\chi'_2)$, with the computation (including the normalization factor Z) being done only on the states where χ'_2 holds. For instance with $X = \{x_1, x_2\}$ and a single potential function $f(x_1, x_2)$:

$$P(x_1 = \top | x_2 = \top) = \frac{f(\top, \top)}{f(\bot, \top) + f(\top, \top)}$$

This conditional computation is of first interest for us, because what we want to know is whether or not a given stakeholder s is an expert given the query $Q = \{q_1, ..., q_n\}$, which can be translated as the probability for s to be relevant given that the queried nodes are relevant, so $P(s = \top | q_1 = \top, ..., q_n = \top)$. The probability resulting from this computation allows us to know the likeliness of this stakeholder s to be an expert, so we can recommend the stakeholders by decreasing probability. For instance, by looking for someone knowing about the topics $t_{security}$ and $t_{cryptography}$, we need to compute for each stakeholder s the probability $P(s = \top | t_{security} = \top, t_{cryptography} = \top)$. It is possible to combine as much topics as wanted for the query, as well as other nodes like roles and terms: the MN works the same independently of the kind of nodes queried.

For the implementation, we have used libDAI [Mooij, 2010], a free and open source C++ library made to compute graphical models. It was chosen because, among all the tools or libraries able to compute graphical models like Bayesian networks and MNs, it was one of the few able to compute MNs in particular and the only one explicitly supporting loops, which is a major constraint considering our complete 4-partite graph. We used the implementation designed for the UAI 2010 Approximate Inference Challenge[1] and which was in the three winners of the challenge.

Potential Functions Used

Potential functions are defined for each weighted relation, so they take two arguments (the binary state $\top/\bot$ of the two related nodes) and return a real value based on the weight of the relation w. Several potential functions are investigated in this work. We first have a reference function with 5 variants, composing with prior and normalization strategies, and a last function using a totally different method. In the following, we first describe the reference function, then we describe the variants applied on it, and we finish with the last function, for a total of $(1 + 5) + 1 = 7$ potential functions, listed exhaustively in Table 6.1.

The reference function, which we call *Id*, is a trivial function assigning the weight w to the co-occurrence cases $(\top, \top)$ and $(\bot, \bot)$, and zero to the others. The interpretation is simple: the weight w aims at representing the amount of evidence that the two nodes are prone to co-occur, so when one is relevant the other also appears as relevant, which is the $(\top, \top)$ case. Similarly, a non-relevant node would lead to make the co-occurring nodes non-relevant as well, thus the $(\bot, \bot)$ case. This assumption makes the function symmetric, and we could argue that only $(\top, \top)$ should be assigned with w, which was actually the first version of our reference function. With early investigations, we obtained better results with the symmetric version, which is why we focused on this one with later investigations. However, although we do not investigate the non-symmetric version (and its corresponding variants), we

[1] UAI 2010 Challenge: http://www.cs.huji.ac.il/project/UAI10/

consider that they might be worth to investigate as well.

A first variant applied on this reference function is the priorised function *Id+5*: we add 5 to each case as a prior value. Prior values are often used to provide a default value to feed where the data lacks, assuming that the actual data will make it negligible where the data is big enough to be reliable. A second variant is the normalized function *Norm*, which divides the values by their sum, or enforce $\frac{1}{4}$ everywhere if this sum is null (because there is 4 cases). In the absolute, this is a scale modification, but this is a local one: each potential function is normalized based on its own values, which are different for each potential function, so it could have an effect on the computed probabilities. A third variant is the semi-normalized function *S-Norm*, which applies a normalization on half the values (i.e. first node in state $\top$) and another normalization on the other half (i.e. first node in state $\bot$). For the symmetric reference function, it is equivalent to a global scale change ($\times 2$), but for the non-symmetric version used initially it implies to give more weight to the half having no data, which is worth investigating to see how the concentration on $(\top, \top)$ could be affected. Because we finally did not investigate the non-symmetric case, this variant only offers to stress the global scale independence property. The two last variants apply the prior value and then normalize the result with one of the two described strategies, resulting in the functions *Norm+5* and *S-Norm+5*.

Finally, the last function investigated is based on the *weight of evidence* of [Good, 1960], initially introduced by [Berkson, 1944] as the logit function, which is a bijective function offering to pass from a probability $p \in [0; 1]$ to a weight $w \in \mathbb{R}$. More formally, given some amount of good outcomes e and bad outcomes $\bar{e}$, so the probability to have a good outcome is $p = \frac{e}{e+\bar{e}}$, then the weight of evidence WoE is computed as follows:

$$WoE = ln\left(\frac{e}{\bar{e}}\right) = ln\left(\frac{p}{1-p}\right)$$

This formula gives us a direct relation between a real value WoE, similar to our relation weight w, and a probability p, which is a normalized value. The main difference with a typical normalization is that this formula "zooms" on the middle of the function, when the probability is close to 0.5, so the evidence is close to 0. This means that with high values, in order to make a difference in the probability, the weight should not have just some more evidence, but have a factor higher. This is interesting to investigate to see how this decreasing of the impact of high values can influence the final results, for instance compared to other types of relations which have a lower scale. Consequently, our last potential function *WoE* exploits this formula to translate the weight w of a relation into a normalized value p by using the inverse function of the weight of evidence, the logistic function:

$$p = \frac{e^w}{1 + e^w}$$

This value p is applied to both $(\top, \top)$ and $(\bot, \bot)$, while the other cases use the complemented value $1 - p$.

6.4.2 Genetic Algorithm

By using a MN, we were looking for a theoretically robust model able to exploit our graph-based data. However, we quickly faced a practical limit: having some dozens of nodes is enough for the computation to take a significant time to compute, and the approximative computation did not lead to convincing results. Consequently, we targeted a more efficient kind of algorithm, which focuses on finding good solutions quickly, leading us to investigate heuristics used in optimisation techniques. Many such heuristics have been designed, from simple ones like Hill Climbing or Tabu Search, to more advanced ones involving complex parameters, like GAs, Simulated Annealing, or Particle Swarm Optimisation [Michalewicz and Fogel, 2004]. In this section, we describe a version based on a GA, which appeared to us as

Function	$f(\top,\top)$	$f(\top,\bot)$	$f(\bot,\top)$	$f(\bot,\bot)$
Id	w	0	0	w
Id+5	$w+5$	5	5	$w+5$
Norm				
$w=0$	0.25	0.25	0.25	0.25
$w\neq 0$	0.5	0	0	0.5
S-Norm				
$w=0$	0.5	0.5	0.5	0.5
$w\neq 0$	1	0	0	1
Norm+5	$\frac{w+5}{2w+20}$	$\frac{5}{2w+20}$	$\frac{5}{2w+20}$	$\frac{w+5}{2w+20}$
S-Norm+5	$\frac{w+5}{w+10}$	$\frac{5}{w+10}$	$\frac{5}{w+10}$	$\frac{w+5}{w+10}$
WoE	$\frac{e^w}{1+e^w}$	$\frac{1}{1+e^w}$	$\frac{1}{1+e^w}$	$\frac{e^w}{1+e^w}$

Table 6.1: List of potential functions investigated, with w the weight of the relation represented by the potential function f.

the most natural to use because of its notion of fitness that we can relate to our notion of Perceived Expertise, and the use of "genes" that we can relate to the artefacts at the source of the Perceived Domain Knowledge and Social Recognition, in other words the nodes of our SRTC graph.

General Description

A GA focuses on finding one or several solutions to a problem in a search-based manner: by starting from trivial solutions (usually random ones) and looking for variations, the solutions are progressively improved until some search budget is consumed, leading to return the best solution(s) found. GAs in particular inspire from natural evolution of species, which evolves by reproduction subject to mutations. More formally, a GA starts with a population of individuals, each individual being described by its "genes", meaning the specific information stored in the individual. Then, it uses a fitness function to select the most interesting individuals and make them reproduce by producing one or more new individuals based on these parents, typically half of the genes from one parent and the other half from the other. Additionally, some random changes can be applied to "mutates" these new individuals, changes which allow to try new solutions not tried before. After the new individuals are produced, the initial population (parents) is replaced by a new one having the most interesting individuals (parents + children) which includes the ones having the best fitness as well as other individuals for future interesting reproductions. Finally, by repeating this reproduction process until some stopping criteria are met, a final population is returned, from which the best individuals found so far can be retrieved.

The genes of the individuals and their evaluation to identify the best ones is the core of the work to use a GA. In our case, because we want a technique able to work on a reduced amount of information to not face the heavy computation faced with MNs, an individual should give the reduced part of the SRTC graph to work with. We call this sub-graph the *constrained query* because it should represent at best the user query, which usually means that the user query is part of it but enriched with other nodes, so that we obtain a more informative "query" to consider. Additionally, we want at the end to recommend stakeholders, so the individual should also provide the list of stakeholders to recommend. This list is called *constrained ranking* because it focuses, like the constrained query, on a subset of stakeholders to evaluate based on the constrained query previously selected.

More formally, given the user query $Q \subset R \cup T \cup C$, each individual solution should provide a *constrained query* $\hat{Q} \subset R \cup T \cup C$ with a limit of N_R roles, N_T topics, and N_C terms, and a *constrained ranking* $\hat{S} \subset S$ of N_S stakeholders. Regarding the selection of the best individuals, we want to maximize the relevance of $\hat{Q}$ ($r(\hat{Q})$), which should be representative of the query Q, and maximize the relevance of $\hat{S}$ ($r(\hat{S})$), which is computed based on $\hat{Q}$. We describe different relevance functions investigated in this work later in this section, which are based on the weights of the relations linking the nodes of Q, $\hat{Q}$, and $\hat{S}$. In short, we want to maximise the relevance of $\hat{Q}$ and $\hat{S}$ given a size of sub-graph to analyse, which corresponds to the following optimisation problem:

$$\begin{aligned} \underset{\hat{Q},\hat{S}}{\text{maximise}} \quad & r(\hat{Q}), r(\hat{S}) \\ \text{subject to} \quad & |\hat{Q} \cap R| = N_R \\ & |\hat{Q} \cap T| = N_T \\ & |\hat{Q} \cap C| = N_C \\ & |\hat{S} \cap S| = N_S \end{aligned}$$

As an example, assuming that the user's query is $Q = \{t_{cryptography}\}$ and that a relevant sub-graph is considered to be composed of 1 role, 2 topics and 2 terms, then we might have tow individuals having different constrained queries:

$$\hat{Q}_1 = \{r_{developer}, t_{security}, t_{cryptography}, c_{encrypt}, c_{RSA}\}$$

$$\hat{Q}_2 = \{r_{manager}, t_{cryptography}, t_{marketing}, c_{build}, c_{employee}\}$$

for which the data extracted should show that $\hat{Q}_1$ is more relevant, because it exploits nodes which are more related to $t_{cryptography}$ than $\hat{Q}_2$. Each individual having also a constrained ranking, we could have for instance two individuals having the same $\hat{Q}$, let say for instance $\hat{Q}_1$, but different sets of stakeholders. The individual having the stakeholders which relate the most to the nodes of $\hat{Q}_1$ would then be the most relevant individual, because recommending more relevant people in regard to the initial query.

From a process perspective, once we get the query, we generate a set of random individuals, each being a couple $(\hat{Q}, \hat{S})$, and we run the GA to maximize two values at the same time: the average relevance of the nodes of $\hat{Q}$ and the average relevance of the nodes of $\hat{S}$. Once the GA has finished running, we obtain the population of individuals and retrieve the single best one by searching first the individuals having the highest average relevance for $\hat{Q}$, then the individual having the highest average relevance for $\hat{S}$ among them. Then, from this single individual, we sort the stakeholders of $\hat{S}$ and recommend them by decreasing relevance. For the implementation, we used NSGA2 for the GA [Deb et al., 2002], which is implemented in the library jMetal [Durillo and Nebro, 2011] and able to deal with both the values we need to maximize (multi-objective GA). We have run it with a population of 100 individuals, a crossover probability of 0.9, a mutation probability of 0.1, and we kept the number of iterations as a configurable parameter for the evaluation. For integrity, we inform the reader that we participated in the design of the new 5.0 architecture of jMetal [Nebro et al., 2015], so we had an additional motivation in using it. Furthermore, we fixed our version to 5.0b34[2], not the final 5.0 version which was released later and removed some experimental features.

Type-specific Relevance Functions (STx)

The relevance of the nodes should be computed for two sets of nodes: $\hat{Q}$, based on Q, and $\hat{S}$, based on $\hat{Q}$. We describe in the following the first case in details, the second case being the same adapted to $\hat{Q}$ instead of Q, adaptation that we describe at the end of this section. Additionally, the computation of the relevance is split into two steps: we first compute the *type-specific relevance*, meaning the relevance of a node for each type of relation separately (e.g. for a topic node, one value for its relations with roles and another for its relations with terms), before to compute the *overall relevance* which merges all of them into a single value. We describe here the different functions used for computing the *type-specific relevance*, and describe the merging functions

[2] jMetal 5.0b34: https://github.com/jMetal/jMetal/releases/tag/jmetal-5.0-Beta-34

later in the section.

Type-specific functions focus on a specific set of nodes $X \in \{R, T, C\}$, and aim at computing the relevance of a node $x \in X$ based on its relations with nodes of other types. Thus, the general strategy is, given x, to look at another set of nodes $Y \in \{R, T, C\} \backslash \{X\}$ to see how it is related to each nodes $y \in Y$. By exploiting the weight of each relation w_{xy} as the relevance of x based on the single node y, we compute the type-specific relevance of x based on all the nodes of Y by summing the weights in a specific manner. Thus, a type-specific function is a function $rel_{XY}(x)$ able to compute the relevance of a node $x \in X$ based on another set of nodes Y.

Our first function, the weighted average *ST1* (Specific Type 1), builds on the weights directly related to the query Q:

$$rel_{XY}(x) = \frac{\sum\limits_{y \in Y} w_{xy}.rel_Q(y)}{\sum\limits_{y \in Y} w_{xy}} \tag{6.1}$$

With $rel_Q(y)$ being the relevance of y based on Q, which means 1 if $y \in Q$ and 0 otherwise, which is equivalent to filter the numerator to consider only the weights for $Y \cap Q$. Consequently, for each y queried in Q, its weight is added to increase the relevance of x if they are related, meaning that if x is related to nodes which are all queried, its relevance is 1. A problem with this function is that, if we also query nodes unrelated to x, the relevance of x does not decrease because the corresponding weight ($w_{xy} = 0$) does not increase the denominator either. To fix this, instead of using the weight of the relation w_{xy}, we can use the maximal weight expected, which can be found by looking at other nodes related to y. This is what is done by the maximised average function *ST2*:

$$rel_{XY}(x) = \frac{\sum\limits_{y \in Y} w_{xy}.rel_Q(y)}{\sum\limits_{y \in Y} max_{x' \in X}(w_{x'y})} \tag{6.2}$$

The problem with this function is that we increase significantly the amount of information to consider: not only we should look at all the nodes in Y, but for each of them also all the nodes in X. As we want to design a function which minimizes the amount of information considered, what we can do is to look only at nodes which are queried. This is what we already do for the numerator (we can ignore the nodes y which are not queried), and because we take a maximized weight for the denominator (which is necessarily superior, so it has a normalization effect), we can also restrict to queried nodes in the denominator, what is done with the filtered-maximised average function *ST3*:

$$rel_{XY}(x) = \frac{\sum\limits_{y \in Y} w_{xy}.rel_Q(y)}{\sum\limits_{y \in Y} max_{x' \in X}(w_{x'y}).rel_Q(y)} \tag{6.3}$$

Like the MNs, all these functions have an interesting property with scaling, but at the opposite of MNs, we have here a *local scale independence*: if we apply a scaling factor α on the weights $w'_{xy} = \alpha.w_{xy}$ and compute the relevance of the node x, in each function the scaling factor α can be factored in both the numerator and the denominator, leading to a simplification giving the original function. This property, like for MN, is important to reduce the initial constraints on the extracted data, so one can choose any arbitrary scale. The additional advantage here is that, because these functions are specific to a given type of relation, a different scaling factor can be chosen for each type of relation, which is more flexible than having a single global factor. Moreover, this scale independence composes with the fact that these functions always return a normalized value in $[0; 1]$, so we can merge them without having one type of relation having more impact than another in an uncontrolled way. This merging is the goal of the functions described in the following.

Overall Relevance Functions (MTx)

Once the relevance of a node $x \in X$ is computed based on its relations with the queried nodes Q, with one value $rel_{XY}(x)$ for each type of relation $Y \in \Psi = \{R, T, C\} \backslash \{X\}$, we need to merge these values into a single, overall relevance value $rel(x)$ which represents the global relevance of the node for the query Q.

Our first overall function, the balanced average *MT1* (Merged Type 1), simply takes the trivial average of the different types of relations:

$$rel(x) = \begin{cases} 1 & x \in Q \\ \frac{\sum\limits_{Y \in \Psi} rel_{XY}(x)}{|\Psi|} & \text{otherwise} \end{cases} \tag{6.4}$$

A particularity is that, if the node x is a queried node ($x \in Q$), we don't compute its relevance: it is relevant by definition, because it has been queried. The relevance is computed only if it is not queried, so it is part of $\hat{Q}$ but not Q. Although it is simple, a problem is that it is not guaranteed that all the categories of nodes are present or numerous enough. For instance, we faced cases where there was only few topics or no roles, while we usually have many terms: in such a case, if the limits of roles N_R, topics N_T, or terms N_C are above the number of available nodes, it seems unfair to give as much weight to each category of node. To improve the balance, the availability-based average *MT2* considers the number of available nodes to weight the type-specific values:

$$rel(x) = \begin{cases} 1 & x \in Q \\ \frac{\sum\limits_{Y \in \Psi} |Y| rel_{XY}(x)}{\sum\limits_{Y \in \Psi} |Y|} & \text{otherwise} \end{cases} \tag{6.5}$$

However, practice shows that terms are usually way more numerous than any other category of nodes, giving them a natural priority. A better weighting can be to consider the limits (N_R, N_T, and N_C) representing the interest we have in each category, what we do with the selection-based average *MT3*:

$$rel(x) = \begin{cases} 1 & x \in Q \\ \frac{\sum\limits_{Y \in \Psi} |Y \cap \hat{Q}| rel_{XY}(x)}{\sum\limits_{Y \in \Psi} |Y \cap \hat{Q}|} & \text{otherwise} \end{cases} \tag{6.6}$$

With this function, the weights are the number of nodes actually used for building $\hat{Q}$, which depends on the available nodes in Y but also on the limits specified above.

Stakeholder Relevance Functions

As mentioned earlier, the type-specific and overall functions described above are designed to compute the relevance of a node of $\hat{Q}$ based on its relations with Q. But once we know which part $\hat{Q}$ of the whole graph to focus on, we are also interested in computing the relevance of a stakeholder of $\hat{S}$ based on that. This can be achieved easily by using the very same functions where we replace the initial $rel_Q(y)$ by a $rel_{\hat{Q}}(y)$. So instead of returning 1 if $y \in Q$ and 0 otherwise, it should return $rel(y)$ (the overall relevance computed based on Q) if $y \in \hat{Q}$ and 0 otherwise. The similar naming has been chosen specifically to show this chaining:

1. we start from a trivial relevance ($rel_Q(y)$) which simply returns a binary value (0/1) depending on which node is queried,
2. we extend the query ($Q \rightarrow \hat{Q}$) and compute a refined relevance ($rel(x)$) based on the previous level ($rel_Q(y)$),
3. we reach the final level of the stakeholders ($\hat{Q} \rightarrow \hat{S}$) and compute a final relevance ($rel(s)$) based on the previous level ($rel(x)$),

In this work, we investigate only these 3 levels, but one can imagine to iterate the second level by reusing recursively the last relevance values to compute the next ones. Such an approach could, like a PageRank, refine iteratively $\hat{Q}$ until it converges to a consolidated set of nodes, before to reach the last level and compute the relevance of each stakeholder based on this consolidated $\hat{Q}$ and their relevance values. Here, we limit ourselves to a single iteration.

6.5 Discussion

If we focus on the MN inference, we noticed several advantages from applying it to our SRTC graph. The first one is how this structure fits: it can deal with undirected graphs, including graphs having loops (while Bayesian Networks forbid it for instance) and it can exploit our weights without imposing any normalization (while Bayesian Networks require probabilities). This way, we minimize the additional constraints on our data, which means that we have a better flexibility to investigate our approach. Another advantage is that we can consider the query at the computation level, meaning that we can build the whole network progressively and compute the query on demand: no adaptation of the network is required. As we shown, MNs also has a global scale independence which gives us some freedom for choosing the scale of the data.

However, there is also limitations, starting from this scale independence which is global: we would have preferred to have a scale independence allowing us to choose the scale for each relation type, like the GA. Another limitation is that the exact computation of a MN grows with the number of relations which, in our case, explodes with the number of nodes. Although the implementation we use allows to make an approximative computation, the results obtained in the evaluations (Part III) are far to be satisfying. It adds to the fact that it is the only implementation we found able to deal with our MN, most of the existing tools focusing on Markov Logic Networks or Bayesian Networks. Maybe some investigations on parallelism could be worth, as MNs have been shown to be particularly interesting for image processing [Szirányi et al., 2000], especially because our cliques are highly local structures (variable pairs).

By choosing to use a GA, we have focused on its ability to converge towards a best value: although it is not guaranteed to reach the best one, we know that it cannot be trapped in a cycle, which is one of the issues we had with the approximative computation of MNs. Another advantage is that we could design relatively simple functions (based on average computation) and apply it to our 4-partite graph while preserving a scale independence at the level of the relation types, at the opposite of the MN. Other techniques could have been used, like centrality measures used in social networks [Freeman, 1978, Borgatti, 2005], but it seems that when we go for a 2-mode graph (i.e. 2 different kinds of nodes, here we have 4), centrality measures already gain a significant complexity [Everett and Borgatti, 2005]. Our GA also allows us to specify how many stakeholders to consider, which is of interest because expert recommenders rarely need to recommend more than a few top experts. Lastly, the fact that we have $\hat{Q}$ in the description of the individuals can be used to build an explanation of the selection of stakeholders $\hat{S}$, which is a feature that we did not have with the whole computation made by the MN.

Still, this approach has drawbacks, in particular the need to specify *a priori* the 4 limits N_S, N_R, N_T and N_C, although nothing so far justifies to use any specific value. The fact that we use a multi-objective optimization is also arguable given that, at the end, we give priority to the relevance of $\hat{Q}$ over the relevance of $\hat{S}$, which is a single-objective strategy (we bet on the storage of less good individuals for having more diversity in the population, which is an important aspect of GAs). The issue is actually more complex: we could also optimize each node separately, leading to manage many objectives at the same time, but it would need to use a many-objectives algorithm [Ishibuchi et al., 2008]. Our main objective was to investigate an algorithm ensuring some convergence towards the best individuals and able to deal with a reduced amount of information, so we did not investigate further the different GA alternatives, but we definitely think that it is an interesting future work. A last limitation we can spot is that we had to implement our own reproduction and mutation operators, which have of course significant impact on the efficacy and efficiency of the GA, and further investigation to improve them could be of interest too.

Our building of the query is also rather simplistic by relying solely on textual similarity and by giving as a priority to select topics, then terms, then roles, while we could imagine for instance to give the possibility to the user to explicit what he means, for instance through annotated terms or smarter parsing strategies. This issue relates first of all to the ease of use, while here we focus more on correctness and consistency, but it nevertheless remains an important aspect to consider for a complete design ready to use.

Looking further at the SRTC graph, one might argue on our idea of using a common mechanism for every nodes, although different types of nodes involve different types of information. Our intuition here relies on the general interpretation of the weights as correlation evidences, while the scale independence provides the flexibility for dealing with weighting strategies able to represent properly each type of information. We can also imagine to use generic sources to strengthen the robustness of our graph by decreasing the noise or enriching it through well-established datasets, like DBPedia[3] or WordNet[4].

Nevertheless, we show through all this chapter that a proper design of EF system can be done based on the indicators used in existing RE works. Consequently, we can start to provide an affirmative answer to our RQ 1: *Can we design an EF process able to consider the core artefacts (topics, terms,*

[3] DBPedia: http://wiki.dbpedia.org/

[4] WordNet: https://wordnet.princeton.edu/

and roles) of the two RE approaches? However, we still need to confirm that such a design allows to provide proper expert recommendations, which is the purpose of the Part III dedicated to the evaluation of our approaches.

Part III

Evaluation

Chapter 7

Evaluation Process

Through the previous chapters, we have designed an approach to recommend experts (Chapter 6) and a formalisation of the expert rankings with associated measures (Chapter 5). In this chapter, we describe a systematic evaluation process which uses these measures to evaluate our approach, and which contains three phases: (i) generate the rankings, (ii) identify the exploitable settings, and (iii) evaluate their correctness and consistency. In Section 7.1, we first describe the measures used to evaluate the stability of our approach, which is our main criteria to identify exploitable settings. Then, Section 7.2 provides the assumptions to fulfil with the associated compliance measures to establish the correctness and consistency of the produced rankings. Finally, Section 7.3 gives the full picture of the whole process to relate the technical details of the generation phase to the measures of the second and third phases.

7.1 Stability of the Approach

The aim here is to focus on the ability for the approach to converge to a *stable* result, independently of its *validity*. Consequently, rather than analysing the distance between the rankings produced by the approach and a gold standard, we aim at comparing the rankings produced between each other to evaluate internal properties. Such an evaluation has several advantages: (i) one can identify intrinsic issues, like cyclic behaviours hurting the reliability of the approach, (ii) validation settings can be selected based on such evaluation, for instance by identifying when the results stabilize to choose the right time-out, (iii) no gold standard is needed. In other words, although these measures do not validate the approach, they can show if the approach need refinements before to have to build any gold standard, which is of particular interest for real case studies in which it is hard and costly to build.

To design these measures, we use the notion of *multiset* [Simovici and Djeraba, 2008] (p. 33), also called collection or list or bag, which is a set where the unicity criteria is discarded (i.e. each element can appear multiple times) and is usually written with square brackets [...] instead of curly ones {...}. This notion is useful to speak about the rankings (or any other things) generated through different runs of our approach, because it is possible to obtain the same ranking from several runs (which is what we want to observe). By using multisets, we keep all the instances, independently of their equalities, and we can use the *multiset sum* operator $\uplus$ to add new elements, such that $[a, b] \uplus [a, c] = [a, a, b, c]$. Notice that we order the elements for a convenient reading, but it does not affect the multiset, so $[a, a, b, c] = [a, b, a, c]$.

We designed two measures to evaluate the stability of our approach based on the multiset of rankings it generates. The first one is the *re-run variability*, described in Algorithm 3: given the multiset V_t of all the rankings produced with the time-out t, we compute the distance for each pair of rankings in $V_t \times V_t$. With this measure, smaller are the distances, better is the guarantee that the time-out t produces a stable ranking, making it more deterministic. The second one is the *extra-run variability*, described in Algorithm 4: given all the rankings V_t produced with the time-out t and all the rankings V_{t+1} produced with $t + 1$, we compute the distance for each pair of rankings in $V_t \times V_{t+1}$. In this case, smaller are the distances, lower is the effect of giving extra computation time, so we can save it by computing t rather than $t + 1$. For the distance, we can use any distance suited for comparing rankings, in particular the measures described in Section 5.2.2 (DD, ODD, or PDD depending on what we want to highlight).

Algorithm 3 Re-run variability computation.

```
Input  $V_t$: Multiset of rankings produced at a given time-out $t$
Input  $d$: ranking distance
Output  $D$: Multiset of re-run distances
D ← ∅
for each $v_1 \in V_t$ do
    for each $v_2 \in V_t \backslash \{v_1\}$ do
        $D \leftarrow D \uplus [d(v_1, v_2)]$
    end for
end for
```

Algorithm 4 Extra-run variability computation.

```
Input  $V_t$: Multiset of rankings produced at a given time-out $t$
Input  $V_{t+1}$: Multiset of rankings produced at the next time-out $t + 1$
Input  $d$: ranking distance
Output  $D$: Multiset of extra-run distances
$D \leftarrow \emptyset$
for each $v_t \in V_t$ do
    for each $v_{t+1} \in V_{t+1}$ do
        $D \leftarrow D \uplus [d(v_t, v_{t+1})]$
    end for
end for
```

Through measures of these kinds, it is possible to evaluate the approach in a more progressive manner by identifying problems before to check any correctness, and so before to have to build a (costly) gold standard. A high re-run variability means for instance that the approach maintains some randomness, generating always different rankings by running it again with the same setting. A re-run variability close to zero but with a high extra-run variability means that the approach is rather deterministic but continuously changes its result over time, which can happen for instance when we enter into a loop. A low re-run associated with a low extra-run means not only that the approach is deterministic, but also that it provides the same ranking over time, so we can consider the produced ranking to be *the* ranking of the approach to evaluate. Once a setting is found to provide such a stable ranking, it is then worth to compare it to a gold standard to validate it.

We had to design this variability procedure because we faced situations where we did not observe a convergence to the gold standard, but we were not able to identify reliably the cause of it. In particular, it was not clear whether the result it provides is simply wrong or whether the algorithm itself did not converge yet. With this procedure, we can check that the algorithm converges, and identify when it does so, before to compare it to the gold standard. This procedure has shown us that some applications of our MN-based approach (Section 6.4.1) did not converge, motivating us to consider the approach based on a Genetic Algorithm (Section 6.4.2) which enforces the convergence.

However, although these algorithms provide interesting values, they require a lot of computation time: Algorithm 3 has a complexity of $|V_t| \times (|V_t| - 1) \rightarrow O(n^2)$, while Algorithm 4 has a complexity of $|V_t| \times |V_{t+1}| \rightarrow O(n.m)$ which in our case is similar to $O(n^2)$ because the settings are selected randomly, so $|V_t| \approx |V_{t+1}|$. In other words, if we have a huge amount of rankings to evaluate, adding just one more ranking increases the computation time in a significant way. Some of our datasets have more than a million of rankings because of the plurality of settings involved, which makes the use of such a measure unreasonable on the full dataset.

Consequently, we inspired from other measures to obtain statistical values which are faster to compute and act as summaries, while the previous algorithms can be used when we need to have a more detailed analysis of specific cases. For these summaries, we inspired from the notion of *variance* and *bias*, often used in statistics and machine learning to measure two different kinds of errors [Hastie et al., 2009]. For the interested reader, we enter in more details in Appendix B.1, where we present the original formula and how we adapt them to our case. We summarise here only the most interesting alternatives we identified, which are a variance version of the re-run measure (Equation 7.1) and a bias version for the extra-run measure (Equation 7.2). Both of them build on our centroid computation $c(V_t)$, described in Section 5.1.2.

$$var_{\text{re-run}}(V_t) = \frac{\sum\limits_{v \in V_t} d(v, c(V_t))}{|V_t|} \tag{7.1}$$

$$bias_{\text{extra-run}}(V_t, V_{t+1}) = d(c(V_t), c(V_{t+1})) \tag{7.2}$$

Regarding their complexity, the centroid computation is a simple average, so it is linear ($O(n)$), and needs to be computed only once. Equation 7.1 then compare it to each ranking, so we add $O(n)$ which means it remains linear. Equation 7.2 has the same complexity, because it simply computes two centroids and a single comparison. These measures are consequently a lot more interesting from a computational point of view.

However, another limitation is introduced because of the use of the centroid, which is the disappearance of *Disagreement*. Indeed, in case of a balanced amount of *Inferior* and *Superior* among the rankings, the centroid order atom chosen is *Unordered*, which is also the case when *Unordered* is the most present. Due to this, a comparison between such a centroid and a ranking (for Equation 7.1) or another centroid (for Equation 7.2) necessarily leads to an increase of *Indifference*. In other words, when a significant amount of *Indifference* is observed with these measures, we cannot say whether it is due to an actual lack of ordered pairs (which is fine) or a balanced *Disagreement* (which is a source of variability). This is in this kind of situation that the algorithms 3 and 4 should be used to obtain a more detailed information.

Another limitation is intrinsic to the extra-run evaluation, independently of the measure used: the period between each sample (t and $t+1$) limits the amount of information we can retrieve from our analysis. This is a general phenomenon which affects any discrete evaluation, so we do not detail it here, but it means that the set of time-outs we consider can reduce our ability to distinguish some behaviours, which is something that should be kept in mind. The interested reader can refer to Section B.3 for a description of this phenomenon.

7.2 Consistency and Correctness

In this section, we focus on the *validity* of the approach, so what are the properties that the produced rankings should have when we give it specific networks and queries. For this purpose, we specify general assumptions that should hold independently of the context, and for which some should be applied to the specific network and queries to identify the corresponding, concrete gold standards. As a reminder, our approach extracts information from sources to represent it through a weighted graph of stakeholders, roles, topics, and terms, which consequently have for sets of nodes (S, R, T, and C) related with weighted links. For the notations, we use $w(a, b)$ to speak about the weight of the relation between two nodes a and b in the network, Q for a specific query so $Q \in 2^{R \cup T \cup C}$, the usual $s \in S$ for a stakeholder, and $v(Q)$ for the ranking built from the query Q. Moreover, when we say that $v(Q) = s_1 {>} s_2 {>} s_3$, we mean as in Section 5.1.1 that the ranking produced from the query Q assigns the ranks 1 to s_1, 2 to s_2, 3 to s_3, and $\varnothing$ (no rank) to any other stakeholder ($v(Q) = \emptyset$ to say that no element has a rank). We also add the $\ni$ (contains) operator to describe partial constraints, such that

$v(Q) \ni s_1 > s_2$ means that the ranking produced from the query Q provides, among others, an order atom $s_1 > s_2$.

Assumption 1. *An absence of stakeholder evidence should lead to have only* Unordered *order atoms:*

$$\forall s \in S, \forall n \in R \cup T \cup C, w(s,n) = 0 \Rightarrow v(Q) = \emptyset$$

This assumption leads to consider a modified network, where we set the weights of all the relations involving a stakeholder to zero. One might also consider a generalisation of this assumption by considering homogeneous weights (all stakeholders are linked with the exact same weight) rather than zero weights only, but we didn't see further advantage to such an assumption so we restrict ourselves to this initial version. With such a network, the gold standard should be a ranking with all the stakeholders having no rank, otherwise it means that the function used to rank the stakeholders may introduce some *query-specific* bias. For instance, an algorithm designed specifically for a given context may have some hard coded rules giving well-known stakeholders as most experts for well-known queries, and use the actual data to revise the ranking. We consider such effect as unwanted because (i) new stakeholders are assumed to be less expert although no evidence shows it and (ii) it is a community-specific assumption so good results based on it are not generalizable.

In order to check that the produced ranking v is actually empty, we can use the compliance measures defined in Section 5.2.3, in particular Equation 5.6, which looks for a strict compliance to the reference: $TotalComp(\emptyset, v(Q))$. Closer we are to one, better is the compliance of the ranking, while a value of zero would mean that it is completely ordered, thus introducing a strong bias.

Assumption 2. *An empty query should lead to have only* Unordered *order atoms:*

$$Q = \emptyset \Rightarrow v(Q) = \emptyset$$

This assumption must be applied to the actual network (not a modified version), and the gold standard is also an empty ranking to which we should strictly comply ($TotalComp(\emptyset, v(Q)) = 1$), otherwise it means that the function used to rank the stakeholders introduces some *network-specific* bias. For example, if several stakeholders aggregate more evidence than others (e.g. because they are involved since a long time), one could think that they are better to recommend in general, before to consider any domain of expertise, which leads to a natural bias. Indeed, if we ask for a topic which is completely unrelated, this bias would be the only information, leading to recommend these people while we actually have no evidence at all. Moreover, we could argue that because we have a lot of data about them, if we have no data at all related to the searched expertise, then these stakeholders have a higher probability to not have it, compared to other stakeholders for who we have less information about.

Assumption 3. *If a ranking produced from a query Q_1 orders (s_1, s_2) and another ranking produced from a query Q_2 optimistically agrees, then the ranking produced from the composition of Q_1 and Q_2 should provide the same order atom:*

$$\left.\begin{array}{l} s_1, s_2 \in S \\ Q_1, Q_2 \subset R \cup T \cup C \\ v(Q_1) \ni s_1 {>} s_2 \\ v(Q_2) \ni s_1 {>} s_2 \vee s_1 ? s_2 \end{array}\right\} \Rightarrow v(Q_1 \cup Q_2) \ni s_1 {>} s_2$$

The optimistic agreement of this assumption refers to the definitions of Section 5.2.1, so an optimistic agreement occurs when Q_2 gives the same order atom than Q_1 or let it free from any constraint. This assumption targets the *consistency* of the rankings rather than their *correctness*, so the produced rankings should not contradict themselves by combining queries providing similar results. Here, each combined query has its own gold standard, $gs_{composition}$, which can be retrieved by looking at the rankings produced for sub-queries.

To establish these gold standards, we designed a systematic procedure through Algorithm 5. First, it retrieves the order atoms of each sub-query (lines 1–15) which implies to find a representative ranking because we generate several rankings for each query, what we do by computing the centroid of the set (line 5). Then, we remove the conflictual pairs to have the final gold standard (lines 17–23). Once $gs_{composition}$ is established, we want all its retrieved pairs to be complied with, so we need to check that $OrderComp(gs_{composition}, v(Q)) = 1$ (Equation 5.8). Equation 5.7, which is similar but includes the *Indifference*s, would show an artificially high compliance if only few pairs remain in the gold standard.

With the previous assumptions, we focus on assumptions that we think to be generalizable (i.e. not limited to our own approach) and which should come as requirements for any EF system. Yet, although we cover the consistency of the approach and its correctness for extreme cases (no data or no query), we still miss the correctness for normal situations (with data and query). We noticed from the literature that gold standards for EF systems are usually built based on (i) the feedback obtained from people within the studied community or (ii) from resources not used in the developed approach, so in summary based on another EF system, usually not validated itself [Vergne and Susi, 2015]. While in some domains we can easily build objective rankings by looking at who performs the best (e.g. which chess player

Algorithm 5 Building of the gold standard $gs_{composition}$ for a query Q.

Input Q: combined query ($|Q| \geq 2$)

Input $V(.)$: Function providing the rankings generated for a specific query

Output $gs_{composition}$: Gold standard for Q

```
// Retrieve pairs from sub-queries of $Q$
$gs_{composition} \leftarrow \emptyset$
for each $Q' \in 2^Q \backslash Q$ do
    // Build a representative ranking for the sub-query
    $v \leftarrow c(V(Q'))$
    // Retrieve all its $Superior$ pairs
    $S \leftarrow stakeholdersOf(v)$
    for each $(s_1, s_2) \in S \times S$ do
        if $v(s_1, s_2) = Superior$ then
            $gs_{composition} \leftarrow gs_{composition} \cup \{(s_1, s_2)\}$
        else if $v(s_1, s_2) = Inferior$ then
            $gs_{composition} \leftarrow gs_{composition} \cup \{(s_2, s_1)\}$
        end if
    end for
end for

// Remove conflictual pairs
$S \leftarrow stakeholdersOf(gs_{composition})$
for each $(s_1, s_2) \in S \times S$ do
    if $\{(s_1, s_2), (s_2, s_1)\} \subset gs_{composition}$ then
        $gs_{composition} \leftarrow gs_{composition} \backslash \{(s_1, s_2), (s_2, s_1)\}$
    end if
end for
```

wins the most), it is not the case for every domains [Simonton, 2006]. In more complex domains involving many tasks, like medicine or development, it becomes hard to establish what should be considered and how to do it in order to draw a valid evaluation of the performance level of someone. As mentioned by [Ackerman and Beier, 2006], such a task is achieved through judgements of *subject-matter experts*, making expertise evaluation a recursive problem: you need to identify experts in order to identify experts. This is one of the main challenges for evaluating an EF system, and why it is hard to build an objective gold standard like we did with the previous assumptions.

This leads us to our next assumption, which is more a statement of the actual state of the practice rather than an insightful requirement.

Assumption 4. *If we can reliably confirm that s_1 is more expert than s_2 for a given query, then s_1 should be ranked higher than s_2 when querying on that query.*

Although it may look like a tautology, the point here is to state that if a (set of) source(s) show a high level of trust for expertise evaluation, then the rankings produced by the new EF system should be compliant with the ranking provided by this (set of) source(s). A major drawback in this assumption is that it lets the interpretation of *reliable* completely open, and it does not tell for instance how we should deal with several reliable sources providing conflicting order atoms (e.g. absolute majority, unanimity, prioritized sources). As mentioned above, the literature supports that each domain might rely on different sources (e.g. objective rankings, subject-matter experts) so the reliability of the source(s) should be based on domain-specific evidences. Nevertheless, it is important to state this assumption at least to (i) be complete in regard to current practices and (ii) highlight the fact that, if one uses some sources to build a gold standard, a particular attention should be given in showing their reliability.

Another element making this assumption arguable is that it explicitly focuses on the *query*, what goes against usual conventions in IR: when building a document retrieval system (from which EF systems are mostly inspired), we focus on the *information need* of the user, not the query in which it is translated and which is provided to the system [Manning et al., 2008]. For instance, in our approach, when we provide a topic as query, we interpret it as the information need "I want to know who is the most expert in this topic", while we could also interpret it as "I want to know who has worked the most in this topic". In our view, if we focus on the information need, then an additional validation phase should be considered, to show that the query given to the system provides an accurate model of this information need. Otherwise, a system could for instance simply represent both the previous information needs with the same query (thus considering them as equivalent) and have acceptable results for both in some case studies, although it is not generalizable because we know that years of work (considered by the second need) are a poor indicator for high performance levels (considered by the first need) as reminded by [Ericsson, 2006c]. This is a reason why we think that the information need and its modelling into a query should be clearly separated, and if the same query happened to be considered for information needs which are not equivalent then it should be stated as a limitation of the EF system. Additionally, if a source provides a ranking by assuming a different information need than we do, then we should reject this ranking as a valid gold standard for evaluating our approach, because this is not what we expect to be represented by our queries. Going further, separating the statement of the information need and its modelling into a formal query opens opportunities: the field of RE is in great part about eliciting and formalizing the needs of the stakeholders, which corresponds well to this translation task from a need to a query, so it makes sense to consider them separately as RE tasks rather than EF ones.

In summary, Assumption 4 says that, given a query corresponding to a well-identified information need, if a reliable source provides a ranking for this specific query/information need, then the EF system must comply with it. In this situation, if the gold standard provides few constraints (few ordered pairs), then having even a single disagreement might have a dramatic effect on the compliance if we consider only the ordered pairs. Because we think that a gold standard with few constraints is proportionally less important than a gold standard with a lot of constraints, the disagreement should be proportional to the amount of constraints provided by the gold standard, so all pairs should be considered, with a poor gold standard leading to a naturally high compliance. More formally, given a query Q and its gold standard $gs_{expected}$ provided by some reliable sources, the compliance to this gold standard is achieved when $OptimComp(gs_{expected}, v(Q)) = 1$ (Equation 5.7). With such a compliance, the data representation used (our network described in Section 6.2.2), combined with the computation made on it (the process proposed in Section 6.4), should be considered as suitable to obtain outputs as expected by our reliable sources.

7.3 Full Process

In order to evaluate our approach in different contexts, we first apply a generation process which builds the network based on the data available for this context, then run the implemented EF system with different settings depending on the inference engine used (MNs or Genetic Algorithm). Each setting corresponds to a specific combination of the parameters involved, and each parameter is restricted to a pre-defined set of values to have a finite (and tractable) set of settings to explore. Each run is made by selecting a random setting, running the algorithm with this setting, storing the setting and the produced ranking in the dataset, and restart for another run. Because there

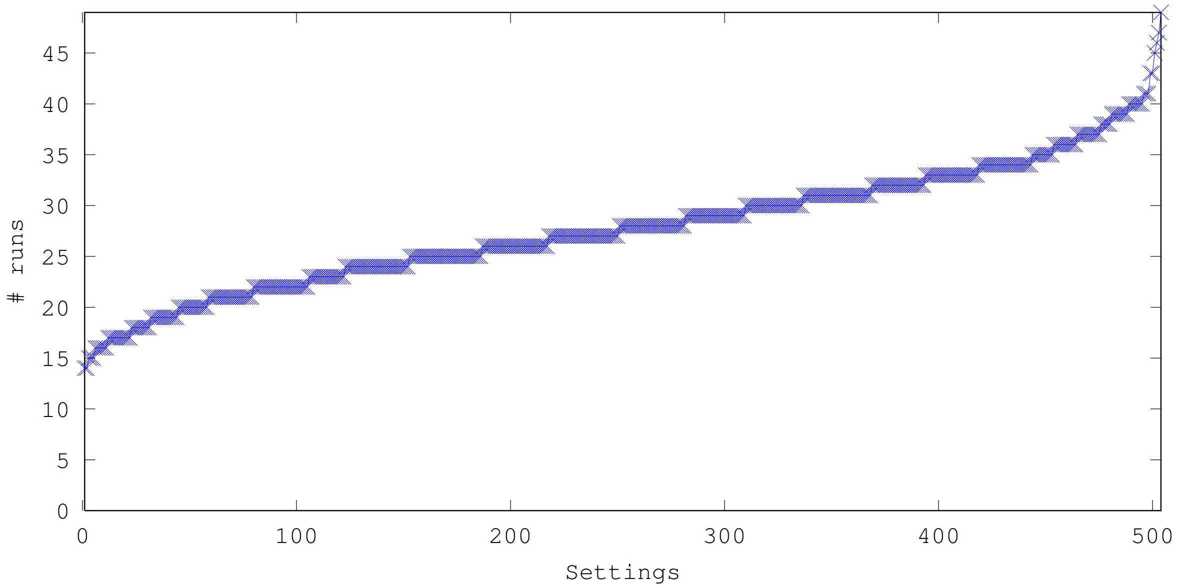


Figure 7.1: Example of distribution of the runs among the different combinations of parameters. The x axis represents all the possible combinations, sorted by number of runs. Each combination being selected randomly, most of them have a number of runs close to the average, while few can have some deficit (left) or excess (right) of runs.

is random processes in our algorithms, it is important to generate several rankings for the same setting in order to have a reliable evaluation, so the random selection is always made on the full set of settings. The resulting dataset provides a list of independent runs with each its setting and ranking, and with most of the settings having a balanced number of runs as shown in Figure 7.1, allowing us to apply statistical evaluations.

The second phase aims at identifying the relevant settings for establishing a reliable evaluation, in particular the time-out to use to ensure that the approach provides a stable ranking. For this, we apply the re-run and extra-run variability procedures described in Section 7.1, using the variance version of the re-run procedure (Equation 7.1) and the bias version of the extra-run procedure (Equation 7.2). Because we want to have an informative evaluation, we use the two distances described in Section 5.2.2, ODD and PDD, for each procedure. It allows us to draw, for re-run or for extra-run, a graph with

2 curves showing how the variability evolves with the time-out, as shown in Figure 5.3 (reproduced below), with the aim of finding a small time-out with small enough variability for each. It is worth to note that these variability measures are setting-specific: if several settings are considered to show a general tendency, then the variability of each setting is computed separately and the resulting set of variability values are then averaged to obtain a single point for the curve. In case we need more detailed results, we can use the basic procedures (algorithms 3 and 4) to replace the single values by sets of points.

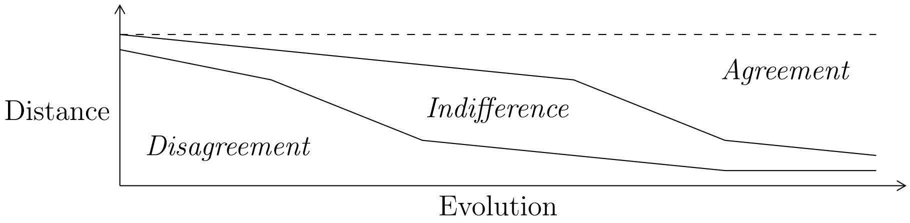


Figure 7.2: Example of graph showing the evolution of ODD (bottom curve) and PDD (top curve) when the agreement increases. We expect this kind of curve when the rankings produced by an automated technique converge to a stable, unique ranking.

The previous description of the second phase focuses on the time-out because our techniques provide approximative results refined through additional computation time. However, in the case of MNs, we can compute exact results if the graph is small enough, which means that the time-out has –in theory– little to do on the result. Actually, if the exact computation fails, the remaining time is used for computing an approximative result, which means that we have to pay attention regarding which settings can actually provide an exact result. These settings return a result by consuming only the time required to compute the exact result, so we can identify them by looking at the settings requiring a constant time which is significantly inferior to the highest time-out. Indeed, if the time required is close to or higher than the maximum time available, it is still possible that giving even more time would

Assumption	Description	Compliance
1	No data	$TotalComp(\emptyset, v(Q)) = 1$
2	No query	$TotalComp(\emptyset, v(Q)) = 1$
3	Composition	$OrderComp(gs_{composition}, v(Q)) = 1$
4	Expected	$OptimComp(gs_{expected}, v(Q)) = 1$

Table 7.1: List of assumptions to check for validating the approach.

lead to exploit the approximative computation. Thus, a variant of this second phase for exact MN computation is to look for settings able to provide exact results to identify relevant settings.

Once an interesting setting is identified, the last phase aims at evaluating the consistency and validity of the approach by checking its compliance to the assumptions described in Section 7.2, as summarized in Table 7.1. This compliance is checked in a systematic way by drawing two compliance graphs of the settings considered, as shown in Figure 7.3. Each graph shows the level of compliance achieved (in $[0; 1]$) for the dimension considered (the time-out in this example), the left one showing a cloud of points to see which values are reached, the right one showing where these points concentrate. Usually, when we consider exact computation the x axis shows the different functions considered, while if we consider an approximative computation it shows how the compliance evolves with (logarithmic) time. Once the compliance to each assumption is checked, the validity of the approach is inferred and discussed for the given context.

Unfortunately, building the datasets for each experiment in order to check all the assumptions is costly, and any fix leading to an update of the computations leads to recompute all of it to not miss anything. Consequently, we were not able to compute all the datasets, but we tried to maximise the coverage of our analysis by having some experiments more exhaustive than others. Thus, the first experiment is the most controlled and has the

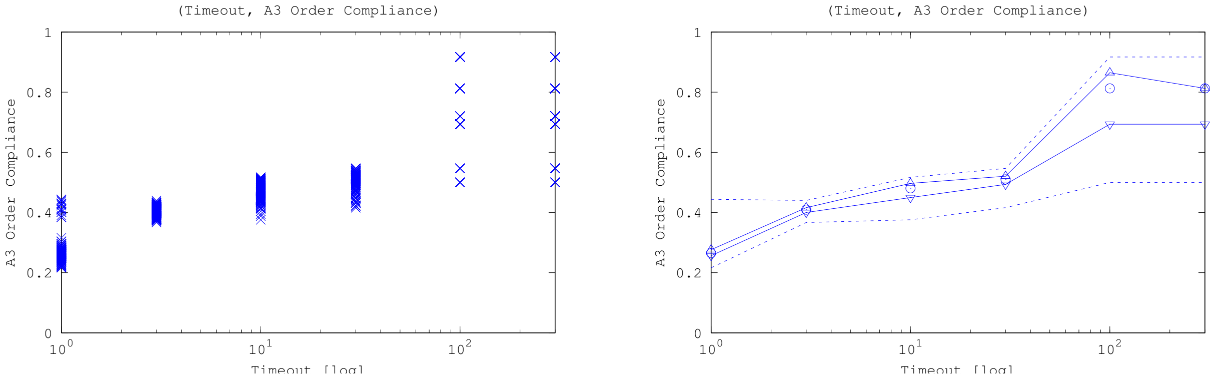


Figure 7.3: Example of compliance graphs of an assumption. The left graph shows the compliance of each generated ranking. The right graph shows the distribution of the points: circle for median, lower and upper triangles for 1$^{\text{st}}$ and 3$^{\text{rd}}$ quartiles, dashed lines for min/max. Here the left graph shows that nothing seems to evolve from 100s (10^2) while the right graph shows that it concentrates more on the highest values.

most exhaustive analysis, while the last experiment focuses only on specific assumptions that we consider the worthiest to stress.

Chapter 8

Evaluation Studies

In this chapter, we apply our evaluation process designed in Chapter 7 to different contexts. The first context, described in Section 8.1, provides a fully controlled environment, where we attempt to stress the ability for our approaches to obtain the right outputs when expected inputs are provided. Section 8.2 applies it to a context which introduces some noise, although it was generated in lab and so is still controlled to some extents. The last application, described in Section 8.3, considers a completely uncontrolled source of data taken from an OSS forum. Finally, we close this chapter by summarising and discussing the results in Section 8.4.

8.1 Evaluation 1: Controlled Data

For this first evaluation, we intend to validate that all the assumptions previously described hold on the output by providing a fully controlled input, thus ensuring that specific input properties lead to expected results. For that purpose, we first set up a fully controlled graph in Section 8.1.1, highlighting the properties of the inputs we plan to work on. Then, we describe the different datasets involved in Section 8.1.2, including the gold standards to satisfy based on the assumptions previously defined. This is followed by the application of our systematic evaluation process in Section 8.1.3.

8.1.1 Synthetic Data

We want a fully controlled input able to stress the relevant properties to check. Because our approach does not precise in which way the information should be extracted from sources, it makes no sense to design specific sources from which we extract the nodes and relations we want. Instead, we directly design the whole graph based on the properties we want to exploit through the processing of this graph with the MN and GA. This graph should not only contain data, but ensures that we represent data prone to be extracted, which means that the relations should make sense regarding what is usually observed in practice. In other words, we need to define a set of stakeholders S, a set of roles R, a set of topics T, a set of terms C, and the weighted relations L to build this graph. The detailed design is provided in Appendix C, and we summarize here the main characteristics of this design.

We start by establishing n topics, which are the topics we expect to be queried, and we relate them to a set of m terms in such a way that relevant terms for a given topic have high weights with this topic while irrelevant terms have low weights. To do so, we use the Zipf's law [Ullah and Giles, 2011] (p. 139), which starts from a high weight for a central term and decreases quickly, with a long tail of low-weight terms, as shown in Figure 8.1. We use this law because it is particularly representative of natural language behaviours, in particular for describing the frequency of words in a corpus made of natural sentences [Manning et al., 2008]. With such a law, we are able to design *term profiles*, which means templates of relations to link the terms to other nodes in the graph. This is for instance what is exploited by [Castro-Herrera and Cleland-Huang, 2009], from which we inspire, when they compute the similarity between vectors of terms, each vector representing a "profile" over the whole set of terms.

Then, we add a role for each topic, which allows us to represent some

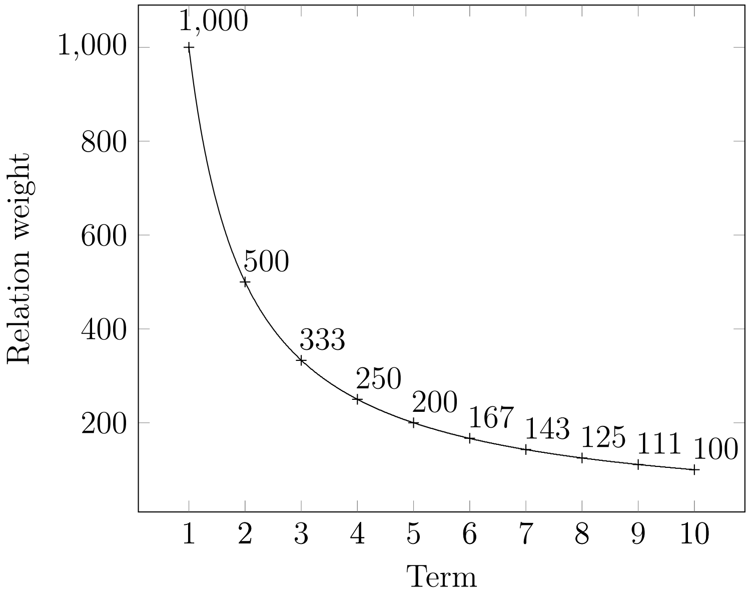
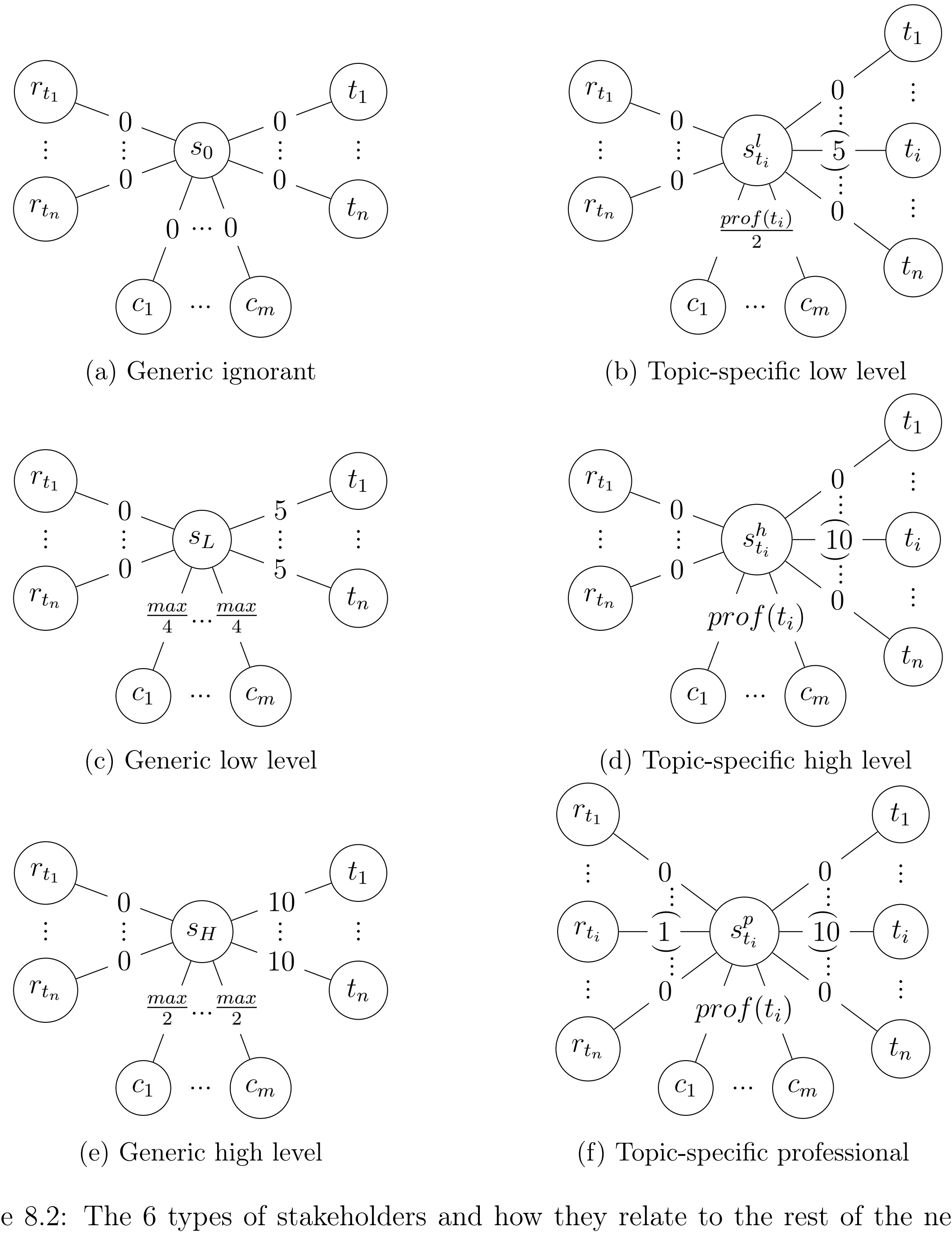


Figure 8.1: Weights for the relations with terms for building a term profile for a given topic. These weights follow a Zipf's law to align with common observations on natural languages.

Social Recognition, and for these roles to be representative of their topic we relate both with a weight of 1. We also add indirect links by applying the term profile of the topic to the role, which enforces this aspect of representativeness. Then, we need to introduce specific profiles of stakeholders, illustrated in Figure 8.2, in order to establish obvious rankings that we can use as gold standards for our evaluation. For each topic, we introduce three topic-specific profiles: a stakeholder having low expertise s^l which relates to the term profile by using a fraction of each weight, a stakeholder with high expertise s^h which relates to the full term profile, and a professional expert s^p which relates to the term profile and to the role, thus giving it an additional evidence making it appear as more expert than s^h. We also introduce generic stakeholders, which relate to the terms in a balanced way rather than through a specific term profile: an ignorant s_0 with zero weights, a generic "amateur" s_L with low weights, and a generic expert s_H with high weights.



(a) Generic ignorant

(b) Topic-specific low level

(c) Generic low level

(d) Topic-specific high level

(e) Generic high level

(f) Topic-specific professional

Figure 8.2: The 6 types of stakeholders and how they relate to the rest of the network (topics, roles, and terms). for each sub-figure, the stakeholder (center) is related to roles (left), topics (right), and terms (bottom) with specific weights.

8.1.2 Datasets

In this section, we describe the different datasets produced for this experiment. These datasets can be accessed online[1].

Source Graph

For our experiment, we built a full weighted graph based on 5 topics and 10 terms, leading to have 5 roles (one per topic) and 18 stakeholders (3 for each topic + 3 generic stakeholders), for a total of 305 weighted relations. For the term profiles, we used a start value $max = 1000$. For the sake of Assumption 1 (no data), an altered version of the graph was produced, where the links relating the stakeholders to other nodes have a weight of zero.

Rankings of the Markov Network Approach

We ran our MN approach on the full graph in order to obtain rankings for queries on 0 topic for Assumption 2 (no query), 1 topic for Assumption 4 (expected), and 2 topics for Assumption 3 (composition). However, because it was among the first datasets that we produced, we first generated the rankings for the queries on which we already know the gold standard (1 topic) before to generate the ones required for other assumptions, which is a threat towards the random selection of the settings to run. In order to evaluate how the rankings evolve within a reasonable time, we considered a maximum time-out of 300s. In order to minimize the number of time-outs to consider, we have used a logarithmic scale, so we can observe early, quick improvements (1s, 3s, and 10s) as well as late, slow improvements (30s, 100s, 300s). All the potential functions listed in Table 6.1 have been investigated, and we have also used the ability of the MN library to run exact and approximative computation to make it a parameter as well. As a result,

[1] Datasets: http://selab.fbk.eu/vergne/Thesis-2016

although the dataset has been generated in different phases, we ensured that the queries remain balanced (average runs per setting between 22 and 27), and in its final state we have an average of 24.25 runs per setting, ranging from 10 to 44, which took a total of 393.26h to generate.

For the sake of Assumption 1 (no data), rankings have been produced based on the altered version of the graph, where the links relating the stakeholders to other nodes have a weight of zero. We only generated it for queries of 0 and 1 topic, but because this is the only assumption to check with this dataset we assume that queries of 2 topics have a low chance to provide additional information. As a result, although it also suffers a generation in different phases (average runs per setting between 21 and 29), we obtained a dataset with an average of 27.79 runs per setting, ranging from 14 to 49, which took a total of 169.77h to generate.

Rankings of the Genetic Algorithm Approach

The dataset of the GA approach has been generated also for queries of 0, 1, and 2 topics, but all the other parameters are different, especially regarding the number of nodes to consider for each category. Because topics were only 5, we considered 1, 3, and 5 of them, and we did the same for the roles, which have the same numbers. For the terms, we took as a general rule to use a logarithmic scale with 1, 3, 10, and 30 terms, because they are usually the most numerous nodes. However, we made a mistake by considering 30 terms also in this experiment, while there is only 10 terms in the graph. Although it consumes more time by adding extra cases to run, it does not reduce the quality of the dataset, and we took it as an opportunity to check that 10 and 30 terms provide similar results, otherwise the representativeness of the whole dataset might have been strongly argued. For the stakeholders, we also took a logarithmic scale with 1, 3, 10, and 18 stakeholders: while 18 stakeholders is interesting to check our assumptions, lower values might be interesting to see if they properly cover only the highest stakeholders, although we did not go that far in our investigation. The 3 type-specific functions, each combined with the 3 overall functions, have been considered. Regarding the time-out, we considered a logarithmic scale like the MN but, because we deal with numbers of rounds rather than seconds, we extended it to consider 1, 3, 10, 30, 100, 300, and 1000 rounds. As a result, the dataset has an average of 7.28 runs per setting, ranging from 0 to 22 (0.58% have 0 or 1 ranking), with a total generation time of 838.69h.

For the sake of Assumption 1 (no data), rankings have been produced based on the altered version of the graph, where the links relating the stakeholders to other nodes have a weight of zero. The settings considered are the same, and the dataset has an average of 10.28 runs per setting, ranging from 0 to 27 (0.03% have 0 or 1 ranking) for a total generation time of 1034.92h.

Gold Standard

The assumptions 1 and 2 both have a fixed gold standard (empty ranking), while Assumption 3 is based on a fixed procedure (Algorithm 5). Only the assumption 4 needs to design a gold standard based on the specific network considered, so we focus on this assumption here. All the gold standards are summarized in Table 8.1.

The gold standard of Assumption 4 is rich but straightforward, because the synthetic data has been designed precisely on this purpose. First, each topic has its own stakeholders organized by expertise level: s_t^l is the least expert while s_t^h and s_t^p are the most expert, but with s_t^p having an additional evidence through its role r_t, thus the constraint $s_t^p{>}s_t^h{>}s_t^l$ if we query for a topic t. For the generic stakeholders, a similar organization has been used in order to have, for *any* topic t queried, the constraint $s_H{>}s_L{>}s_0$. Additionally, we have the topic-specific stakeholders which are assumed to be experts only on their own topics, so for a query on topic $t \neq t'$ we have

Assumption	Gold Standard
1	$\emptyset$
2	$\emptyset$
3	Output-dependent
4	For $Q = \{t\}, t \in T$: $s_t^p{>}s_t^h{>}s_t^l{>}s_{t'}^p?s_{t'}^h?s_{t'}^l{>}s_0$ and $s_H{>}s_L{>}s_0$

Table 8.1: Gold standard rankings for synthetic data.

$s_t^p?s_t^h?s_t^l{>}s_{t'}^p?s_{t'}^h?s_{t'}^l$. We might consider to add a constraint between specific and generic stakeholders, for instance ensuring that the specific stakeholders are higher ranked than the generic ones because they better fit the term profiles $prof(t)$, but we might also argue that because a generic stakeholder has mastered other topics he might have a richer experience leading to an expertise of "better quality". So excepted for s_0, which is by definition the least expert in any topic, we prefer to avoid adding such a constraint and remain with the ones defined above, which implies to consider a gold standard, for a topic t, of two rankings ($s_t^p{>}s_t^h{>}s_t^l{>}s_{t'}^p?s_{t'}^h?s_{t'}^l{>}s_0$ and $s_H{>}s_L{>}s_0$) which can be merged into a single ordering because no ordered pair is conflicting.

8.1.3 Results

We describe here the main results of the analysis of the datasets, which are detailed in Appendix D.

Markov Network (exact)

From the analysis of the execution time, we obtained that most of the potential functions are able to compute exact results: only Id, which is the function using the raw weights, fails by consuming all the time-out. Id+5 (add 5 as a prior to each weight), Norm (normalise the weights), S-Norm

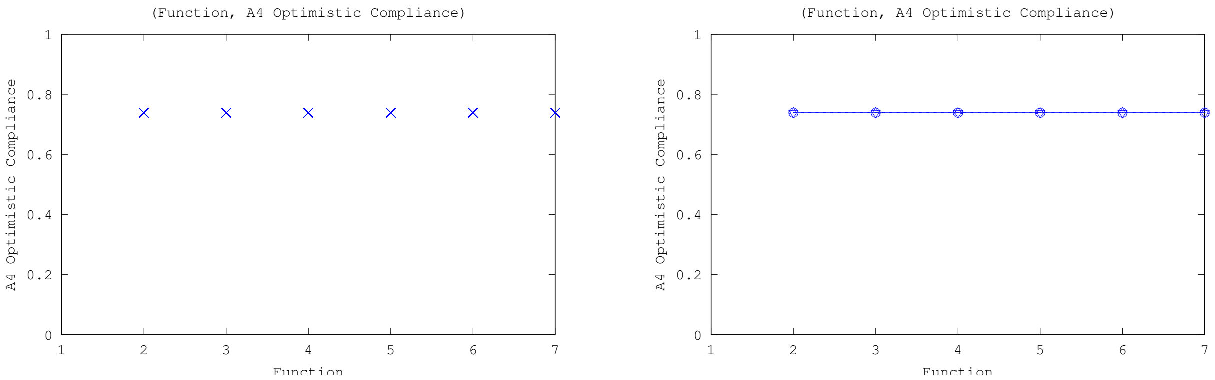


Figure 8.3: Evolution of the assumption compliance of the MN technique for synthetic data with exact computation. Assumption 4 (expected) shows only partial compliance (73.9%). By focusing on ordered pairs, poor compliance is achieved (29.8%).

(semi-normalisation), Norm+5 (normalisation with prior values), S-Norm+5 (semi-normalisation with prior), and WoE (based on weight of evidence) all show equivalent performance from a computation time perspective, consuming between 4s and 18s depending on the parameters, including queries.

Regarding the compliance to gold standards, the assumptions 1 (no data), 2 (no query), and 3 (composition) all show a full compliance. In other words, the exact computation of the MN provides correct results in extreme cases as well as consistent results for what has been investigated. However, Assumption 4 (expected) shows only partial compliance, with 73.9% of the pairs being compliant with the gold standard, which seems particularly low for noise-free data. We show it through Figure 8.3 because we consider this assumption as the most important one to satisfy, because this is the one used in usual EF works. If we restrict to ordered pairs, the compliance drops to 29.8% because most of it is due to unconstrained pairs. This low compliance occurs because s_0 is ranked below all the other stakeholders and nothing more, so only pairs comparing s_0 are compliant while all the others are missing. In brief, the rankings produced are rather uninformative by ranking only one stakeholder, so although it is correct, such a ranking would be useless in practice.

Markov Network (approximative)

From the analysis of the re-run variance, only Id, Norm, S-Norm, and WoE provide rather stable rankings, although some variability occurs ($ODD \approx 8.3\%$). The other functions show a high level of *Indifference*, which is because s_0 is the only one ranked differently, which is not interesting. Putting aside the uninformative functions, the extra-run bias shows a rather constant instability of the rankings with ODD around 25%. Because it is way higher than the variability observed with the re-run variance, it means that a significant addition of variability is added by the evolution of the rankings over time.

Regarding the compliance, no function satisfy Assumptions 1 (no data) because they provide almost totally ordered rankings. This is due to the approximative computation which, although it provides values close to 0.5, fails to provide strictly equal values, leading to rank the stakeholders. We observe the same phenomenon for Assumption 2 (no query) with Id, Norm, and S-Norm, while Id+5, Norm+5, and S-Norm+5 achieve a good level of compliance because only s_0 has a different rank. WoE is the only one to show significant differences between the probabilities of the stakeholders, thus motivating the order (and the lack of compliance). For Assumption 3 (composition), we observe somehow the opposite: Id, Norm, and S-Norm achieve some compliance (as well as WoE) while Id+5, Norm+5, and S-Norm+5 remain totally non-compliant. But even if we achieve some compliance with some functions, it remains below 60% in general. Finally, Assumption 4 (expected) offers in Figure 8.4 the most unexpected results by having Id, Norm, S-Norm, and WoE achieving an average of 81.6% of compliance (85.6% at the highest time-out), which is even higher than the 73.9% of the exact computation. Id+5, Norm+5, and S-Norm+5 however achieve the minimal level of compliance (62.7%) because of the unconstrained pairs which are optimisti-

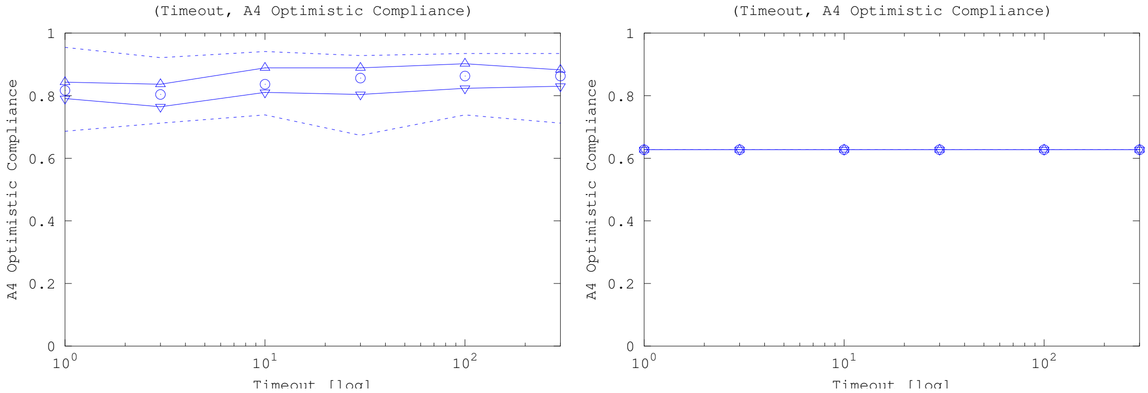


Figure 8.4: Evolution of the assumption compliance of the MN technique for synthetic data with approximative computation. Only the distributions are shown for readability. Assumption 4 (expected) shows two tendencies: Id/Norm/S-Norm/WoE (left) are globally high but not perfect, while Id+5/Norm+5/S-Norm+5 (right) are stuck at 62.7%, which is the worst possible value.

cally considered as compliant. In short, if we focus on Id, Norm, S-Norm, and WoE, they might provide some good results as long as we can fix the approximation issue (values close to 0.5 instead of equal).

Genetic Algorithm

From the re-run variance and extra-run bias, we can observe almost no stability issue with around 90% of *Agreement* and 10% of *Indifference* (because of actual *Unordered* pairs). This observation occur when choosing a setting which computes the full graph, which enforces such a situation, but only few differences are observed when we compute a minimal sub-graph of 1 role, 1 topic and 1 term. ST1 (type-specific function which computes a local average) and ST2 (which uses smarter weights for the average) are particularly interesting by showing almost the same results in both situations, while ST3 (simplification of ST2) looses more stability with less nodes. This good results also with few nodes probably come from the fact that the GA generates 100 random individuals from the start: because we have only 250 possible minimal sub-graphs, good individuals appear fast.

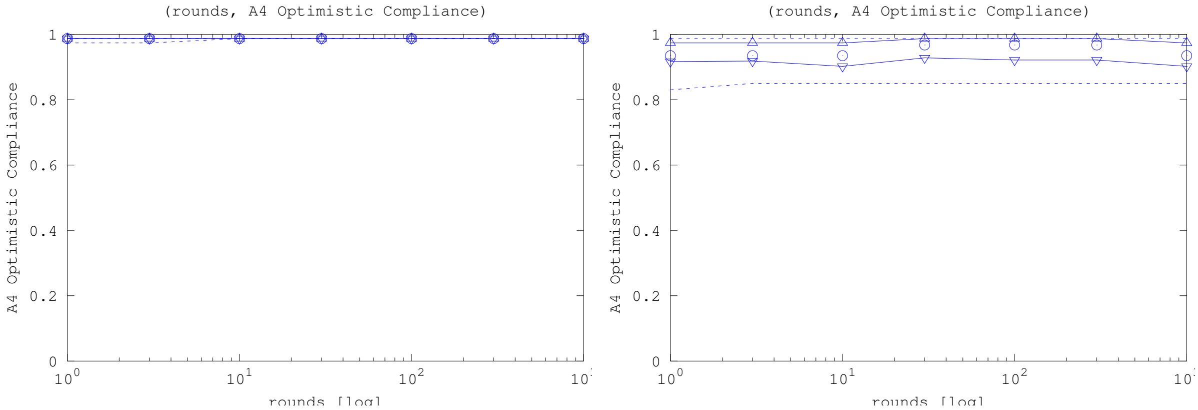


Figure 8.5: Evolution of the assumption compliance of the GA technique for synthetic data with min data. Assumption 4 (expected) shows almost perfect compliance for ST1 (left), which is good if we consider that ST1 is a naive computation, while ST3 (right) does not achieve perfect compliance due to its simplification compared to ST2 (perfect so not shown here), although it remains high in average (94.2%).

Regarding the compliance, ST3 provides rather poor results compared to ST1 and ST2, which achieve full compliance for the assumptions 1 (no data) and 2 (no query), a high level of compliance for Assumption 3 even with a minimal sub-graph (94.5% in average), and close to full compliance for Assumption 4 (expected) as shown in Figure 8.5. Actually, ST2 properly achieves full compliance on the latter, but this functions also consumes a significant amount of time (between 30s and 35s with a minimal sub-graph). ST1, in the other hand, looses only a negligible piece of compliance but can be computed in 5s only. Despite these really good results, more stress should be given through a bigger graph to avoid the immediate convergence.

8.2 Evaluation 2: Semi-Controlled Data

In this evaluation, we intend to validate that our assumptions still hold with an input involving some natural language aspects. Firstly, we describe in Section 8.2.1 the procedure we used to build natural discussions to use as sources. Secondly, we show how we generated each dataset in Section 8.2.2, in particular how we extracted the nodes and weighted relations from our sources to build the corresponding graph. Finally, we provide the analysis of these datasets by applying our systematic evaluation process in Section 8.2.3.

8.2.1 Cuisine Discussions

In order to stress our approach in a controlled situation based on non-synthetic data, we have executed a procedure involving 3 participants, Alice, Bob and Carla, in order to generate a set of e-mails that we could use as sources of data. To preserve anonymity we have renamed the stakeholders of this experiment, and the datasets have been anonymised correspondingly. The original sources (e-mails) are not provided because of the need to deeply alter them in order to enforce anonymity, what we solve in the next experiment by using public data. We asked them to discuss via e-mail about 2 cooking-related threads, Mongolian food and Tiramisu, which correspond to specific dishes of their native countries, thus making obvious who is the most expert in which topic. While presenting them the project, we have hidden the EF aspect of it to avoid biases, presenting only the topics to discuss and the constraints to follow to maintain the quality of the discussion.

The experiment has started just after its presentation to the participants and has lasted 2 days during which 30 messages were exchanged. Alice has provided 8 contributions (4 for Mongolian food and 4 for Tiramisu), Bob 9 contributions (4 and 5) and Carla 13 (6 and 7). The participants were free to reply when they wanted, so they could participate during their free time, increasing the chance to have natural discussions. They respected the constraint to exchange by e-mail only and using the reply-all functionality to be sure that the messages were sent to everyone and the discussions did not split. Although we were in copy to follow the discussions and monitor them if necessary, we did not have to intervene because they respected the constraints, remained close to the initial topics, and the contributions were quite balanced.

8.2.2 Datasets

In this section, we describe the different datasets produced for this experiment. These datasets can be accessed online[2].

Source Graph

At the opposite of the synthetic data presented previously, for which we designed directly the input graph, here we needed to build it from the e-mails of the participants. In order to build our graph, we have designed a *node extractor* and a *relation extractor* in order to parse the e-mails and extract the relevant nodes (stakeholders, roles, topics, and terms) and their weighted relations. The node extractor uses Algorithm 6 to consider an author as a *stakeholder* (S), the nouns in the subject of the e-mails as *topics* (T), and the nouns in the body of the e-mail as *terms* (C). We did not consider *roles* (R) in this experiment because we did not define ones at that time and did not consider any source of data to extract relations between roles and other nodes. The relation extractor uses Algorithm 7 to relate the terms and topics to the author, based on who wrote what, as well as the terms and topics together, based on which term is used in which discussion.

Algorithm 6 Node extractor for a single e-mail.

```
Input  $mail$: Natural language e-mail
Output $S, R, T, C$: Extracted stakeholders, roles, topics and terms
$S \leftarrow \{authorOf(mail)\}$
$R \leftarrow \emptyset$
$T \leftarrow nounsOf(subjectOf(mail))$
$C \leftarrow nounsOf(bodyOf(mail))$
```

[2] Datasets: http://selab.fbk.eu/vergne/Thesis-2016

Algorithm 7 Relation extractor for e-mails.

```
Input $mail$: Natural language e-mail
Input $S, R, T, C$: Stakeholders, roles, topics and terms
Output $L$: Weighted relations
$L \leftarrow \emptyset$
$a \leftarrow authorOf(mail)$
if $stakeholder(a) \in S$ then
    for each $n \in nounsOf(bodyOf(mail))$ do
        if $term(n) \in C$ then
            $L \leftarrow L \uplus \{\langle stakeholder(a), term(n), 1\rangle\}$
            // The symbol $\uplus$ (multiset sum) acts as a union
            // symbol which sums the weights of similar relations,
            // so $\{\langle a, b, 2\rangle, \langle a, c, 1\rangle\} \uplus \{\langle a, b, 3\rangle\} = \{\langle a, b, 5\rangle, \langle a, c, 1\rangle\}$.
        end if
    end for
end if
for each $t \in T$ do
    if $t \in nounsOf(subjectOf(mail))$ then
        $L \leftarrow L \uplus \{\langle stakeholder(a), t, 1\rangle\}$
        for each $n \in nounsOf(bodyOf(mail))$ do
            if $term(n) \in C$ then
                $L \leftarrow L \uplus \{\langle t, term(n), 1\rangle\}$
            end if
        end for
    end if
end for
```

The authors of the e-mails were identified by looking at the sender field of the e-mail. We retrieved the nouns to identify the terms and topics by using the software GATE [Cunningham et al., 2011], a free and open source Java software to manage text processing with natural languages. It was chosen because it appears as a reference regarding natural language processing, aggregating well known tools like Lucene and WordNet and providing a complete extraction process. A custom merging process was made to put together similar words, like singular and plurals, or lower and upper case versions.

The resulting dataset contains the 3 stakeholders, Alice, Bob, and Carla, and 3 topics, *Tiramisu*, *Mongolian*, and *food*, which means that the queries we want to have specific gold standards for are the 1-topic query *Tiramisu* and the 2-topic query {*Mongolian*, *food*}. Additionally, no roles are used but we extracted 293 terms, and 865 weighted relations were generated. No altered version of the graph was produced for Assumption 1 (no data), because we consider that the investigation made on the experiment with synthetic data provides strong enough results to motivate that we allocate more time to other, less robust aspects.

Rankings of the Markov Network Approach

We ran the MN approach for queries of 0-3 topics, which allows us to cover the empty query for Assumption 2 (no query), the two queries we need for Assumption 4 (expected), and all the possible combinations of topics for Assumption 3 (composition). Like the experiment on synthetic data, we investigated all the functions listed in Table 6.1, with a logarithmic time-out of 1, 3, 10, 30, 100, and 300s, and with exact and approximative computation. The resulting dataset provides an average of 20.28 runs per setting, ranging from 9 to 35, and generated in 168.32h. No dataset have been generated for Assumption 1 (no data).

Rankings of the Genetic Algorithm Approach

Similarly, we also covered all the topics for the GA approach, from 0 to 3 topics. For the nodes limits, we considered 1, 2, and 3 stakeholders, as well as 1, 2, and 3 topics, so we cover all the possible cases. For the terms, we covered 1, 3, 10, and 30 nodes, so we can investigate how the number of nodes affect the results at low and large scales, without going until the full set of terms (293), because this algorithm is designed to deal with few nodes only and some functions are particularly long to compute already with few nodes. For the rest of the parameters, we did like for the synthetic data, so trying all the type-specific and overall functions with 1, 3, 10, 30, 100, 300, and 1000 rounds. As a result, the dataset provides an average of 15.87 runs per setting, ranging from 3 to 36, and has been generated in 955.31h. No dataset have been generated for Assumption 1 (no data).

Gold Standard

During the discussions of the participants, several questions and suggestions were provided and we could easily assess who was the expert for each topic. However, to confirm our claims and build our gold standard, we asked the participants to fill a form for each discussion after the experiment, where we asked them what is, from their own point of view, their level of knowledge (newbie, advanced, expert) and the most knowledgeable participant for each discussion. Alice and Bob were consistently confirmed as the most knowledgeable ones on how to prepare Tiramisu while Carla was the expert for Mongolian food. This is what allows us to build the gold standard for Assumption 4, which is listed with the other assumptions in Table 8.2. Like for synthetic data, the assumptions 1 and 2 both have a fixed gold standard (empty ranking) and Assumption 3 builds on outputs of sub-queries, so no specific decision needs to be made for them.

Assumption	Gold Standard
1	(not considered)
2	∅
3	Output-dependent
4	Mongolian food: Carla>Alice?Bob Tiramisu: Alice?Bob>Carla

Table 8.2: Gold standard rankings for cuisine discussions.

8.2.3 Results

We describe here the main results of the analysis of the datasets, which are detailed in Appendix E. We do not consider Assumption 1 (no data) because no dataset has been generated for it.

Markov Network (exact)

By analysing the computation time, we observe that Id is unable to perform an exact computation, while all the other functions can be used as long as the query is not empty, while for synthetic data the empty query was also fine. If we restrict to the remaining cases, no difference is observed between the functions regarding their computation time, which is between 1s and 2s.

Regarding the compliance, Assumption 2 (no query) is considered as not satisfied because no function is able to compute exactly an empty query. For Assumption 3 (composition), while Id+5, Norm+5, and S-Norm+5 are fully compliant, the remaining function are fully non-compliant. And if we focus on the compliant functions, Assumption 4 (expected) shows in Figure 8.6 an arguable result: two queries are covered by this assumption for this dataset (Mongolian food and Tiramisu), but only one is fully compliant (Tiramisu), while the other is not. Indeed, independently of the query, the rankings provide the same orders, which means that people have the same rank independently of what is queried. A deeper investigation shows that

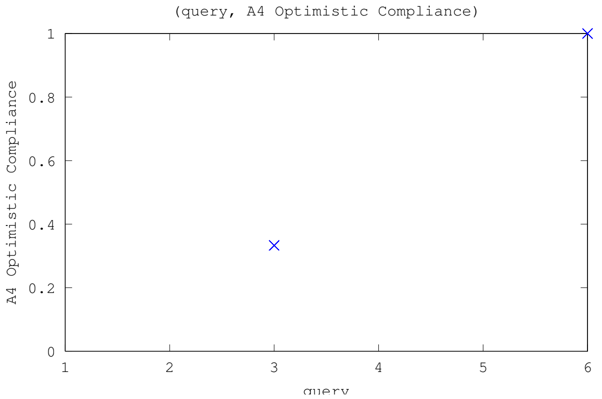


Figure 8.6: Evolution of the assumption compliance of the MN technique for cuisine data with exact computation. A focus on the 2 queries of the gold standard shows that only 1 is compliant and the other almost not at all (only for the unconstrained pairs), so Assumption 4 (expected) cannot be considered as fulfilled.

not only the orders are the same for both queries, but (i) the probabilities computed are the very same, and (ii) it happens for any non-empty query, not just Mongolian food and Tiramisu. In brief, Assumption 4 (expected) is only partially satisfied, and this partial satisfaction appears to be more a matter of luck given the constance over the queries.

Markov Network (approximative)

By looking at the re-run variance, we can observe three interesting patterns of stability. First, Id remains stable although the variability grows slightly with the time-out to reach 10.4%. Second, Id+5, Norm+5, and S-Norm+5 start mainly with uninformative rankings (75% of *Indifference*) but progressively increase their informativeness (decreases until 15%) with negligible amounts of *Disagreement* during the whole process (0.1%). Third, WoE shows no *Disagreement* at all and starts from more informative rankings (57.5% of *Indifference*) but gain only few informativeness at high time-outs (decreases to 52.8%). By looking at the extra-run bias, Id looses its interest by showing a constant evolution of its rankings, while the others show similar curves

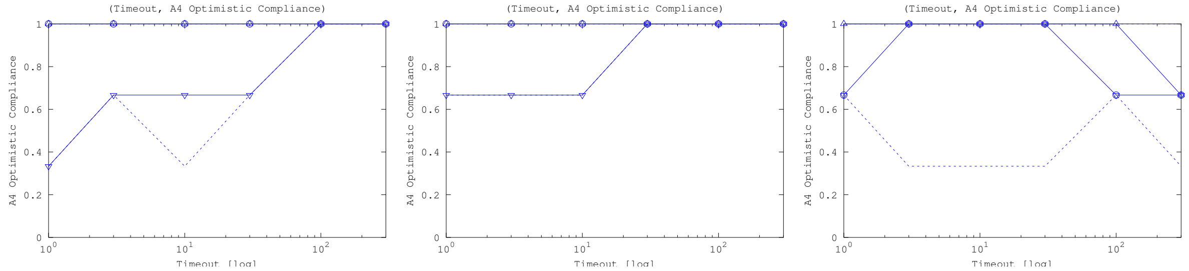


Figure 8.7: Evolution of the assumption compliance of the MN technique for cuisine data with approximative computation. Assumption 4 (expected) shows increasing compliance for the priorised functions (left). WoE (middle) is even better, while Id (right) seems always high but subject to instability.

than for the re-run variance. Id+5, Norm+5, and S-Norm+5 show again a progressive gain of *Agreement* with negligible *Disagreement*, meaning that the rankings converge towards an informative and stable ranking, although our time-outs are too low to see how far it can go. WoE shows also that its rankings agree between time-outs, but no much gain of informativeness occur, leading to rankings ordering less than half of the possible pairs.

If we check the compliance, Assumption 2 (no query) shows a broad tendency to not comply which, like for synthetic data, is due to the approximative computation leading to close probabilities but not strictly equal ones. This phenomenon is particularly straightforward when looking at the compliance of Id+5, Norm+5, and S-Norm+5, which start fully compliant but progressively loose it until reaching zero. Assumption 3 (composition) shows a general lack of compliance, although we observe a slow increase with Id+5, Norm+5, and S-Norm+5. Finally, Assumption 4 (expected) unexpectedly provides the best results in Figure 8.7, with Id being globally compliant although it is subject to some instability, while Id+5, Norm+5, S-Norm+5, and WoE converge properly to a perfect compliance with more computation time. In brief, we observe similar results than with synthetic data, where we face an approximation issue but where Assumption 4 tends to be better satisfied than with the exact computation.

Genetic Algorithm

Regarding the re-run variance, ST2 appears to be the most interesting when combined with MT1 (basic average of the type-specific values) and MT3 (weighted average based on the number of nodes in the setting) by offering low PDD and ODD, and they are the only ones able to reach properly $ODD = 0$ with some settings and queries, so the rankings tend to be highly similar at a given time-out with these functions. If we look at their extra-run bias, it appears that the rankings remain with some stable *Disagreement*, although a significant part of *Agreement* is maintained, more or less high depending on the queries but above 75%. This stable *Disagreement* might be explained by having different sub-graphs having all high relevance based on the query, and providing different rankings because of the different nodes involved in these sub-graphs.

For the considered functions, the assumptions 2 (no query) and 3 (composition) are fully satisfied, while Assumption 4 (expected) offers mitigated results. Figure 8.8 shows in particular that ST1 only partially complies, ST2 always has only one query satisfied, like for the exact MN computation, and ST3 tends to have the same behaviour although some improvements are observed at high time-outs.

8.3 Evaluation 3: Public Data

In this last evaluation, we investigate further our techniques by exploiting them on a bigger context, involving more stakeholders and more distant topics. Consequently, we first describe in Section 8.3.1 the public discussions we have used as sources, which provide question-answer discussions and requests for supports, thus being usual information we can use to discover and refine requirements. Then, we show how we generated each related dataset in Section 8.3.2 before to provide the results of our systematic evaluation process

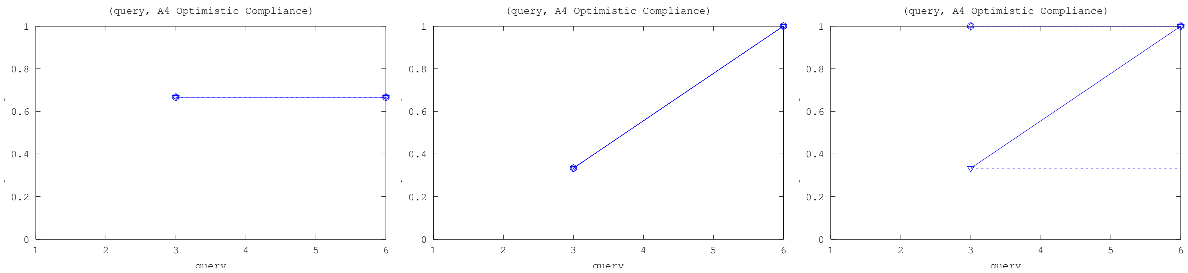


Figure 8.8: Evolution of the assumption compliance of the GA technique for cuisine data on min data. Assumption 4 (expected) can only be reasonably satisfied with MT1, so we do not consider the others here. ST1 (left) partially complies to each query, ST2 (middle) complies only with one of them, and ST3 (right) shows a better compliance if we focus on the highest time-out.

through Section 8.3.3.

8.3.1 XWiki OSS Forum

While the previous experiments maintained some control on the data, this experiment has been run on a software project involving a huge community of people. XWiki[3] is an Open Source Software (OSS) which takes the form of a platform for managing wikis. It has a community of contributors, including a company managing the development of the OSS and selling support and training on it. This community interact through different media, in particular a mailing list for support and discussions about the software. We have used the archives of this mailing list, which are freely available online[4], to retrieve the e-mails exchanged and re-build the discussion threads.

We have restricted ourselves to e-mails of the year 2012, and we removed the ones related to discussions started before 2012 to avoid having inconsistent threads. Consequently, we retrieved 2728 e-mails organized in 713 threads, having each between 1 and 37 e-mails. All of them have been organized and formatted in order to present them to human subjects through a

[3] XWiki platform: http://dev.xwiki.org

[4] XWiki archives: http://lists.xwiki.org/pipermail/users/

survey for establishing a gold standard on several topics. The details of the survey are presented in [Vergne, 2016a] and the related data can be accessed online[5], but as a summary we could obtain gold standards for two topics: *Debian*, which relates to 34 e-mails in 6 threads, and *Hibernate*, with 37 e-mails in 8 threads, involving a total of 18 XWiki contributors. 10 subjects independent from the XWiki community were involved in the survey, so the gold standards could be built by avoiding the bias due to personal relationships between the contributors.

8.3.2 Datasets

In this section, we describe the different datasets produced for this experiment. These datasets can be accessed online[6].

Source Graph

In order to build our graph, we have used the Algorithm 6 and Algorithm 7 from the previous experiment to extract the nodes and relations from the e-mails. By applying these algorithms on the 14 discussions related to the two topics, we retrieved the 18 contributors (S), 42 topics (T) and 969 terms (C), related by 7536 weighted relations. Once again, no role were considered, but it might be interesting to exploit some additional information to label the stakeholders, like *comitter*, *contributor*, *translator* and other status provided in the Hall of Fame[7] of XWiki. An additional effort has been made to clean the data, especially to identify unique authors by aggregating different e-mail addresses for similar names of author, and to remove noise in the body of the e-mails like quotations. However, this process still need to be improved because some noise, like huge source code excerpts, is removed manually by

[5] Survey data: http://selab.fbk.eu/vergne/Experiment-2014-02-19/

[6] Datasets: http://selab.fbk.eu/vergne/Thesis-2016

[7] XWiki HoF: http://dev.xwiki.org/xwiki/bin/view/Community/HallOfFame

forbidding special terms.

No altered version of the graph was produced for Assumption 1 (no data), because we consider that the investigation made on the experiment with synthetic data provide strong enough results to motivate that we allocate more time to other aspects.

Rankings of the Markov Network Approach

We focused the generation of our rankings on the empty query, for Assumption 2 (no query), and the two queries *Debian* and *Hibernate*, for Assumption 4 (expected). Assumption 3 (composition) requires a significant addition of queries to cover different combinations, what we considered to be too costly because we wanted to generate an equivalent dataset for the GA approach, which takes a significant time as we could observe from the previous experiments. Like the previous experiments, the time-outs were 1, 3, 10, 30, 100, and 300s, and all the potential functions were considered, but only with the approximative computation, the graph being too big for the exact computation. The generated dataset provides an average of 15.95 runs per setting, ranging from 5 to 28, and generated in 42.11h. No dataset have been generated for Assumption 1 (no data).

Rankings of the Genetic Algorithm Approach

The dataset generation of the GA approach being really costly, due to the many parameters it involves and some functions computing a significant part of the whole graph, we had to be particularly restrictive on the parameters. Although we considered the same queries (empty + *Debian* + *Hibernate*), we did not consider all the stakeholders, which involved only 1, 3, and 10 nodes over 18. Also the topic limits were small, with 1, 3, and 10 topics, far from the 42. The terms got the same limits than for other experiments, with 1, 3, 10, and 30 terms. For the rounds, with an initial distribution of 1, 3, 10, 30, 100, 300, and 1000 rounds, we saw that it would lead us to have a really poor dataset in terms of runs per setting due to the time required to generate it, thus we decided to sacrifice the 1000 rounds to save a significant amount of time (around 70%). We did not sacrifice 300 rounds because we expected that more time was required to converge, motivating to increase the number of rounds rather than decreasing it. We also did not sacrifice the small values because of the small difference it makes on the overall time, which is proportional to the value, and the information loss it implies. These sacrifices were made so that we could still investigate all the functions combinations, which are the main interest for us. In its final state, the dataset provides an average of 3.59 runs per setting, ranging from 0 to 12 (2.2% have 0 rankings, 10.8% have 1 ranking only), generated in 986.4h. The dataset was generated on two different machines to maximize the amount of information and, although it implies to have different execution times, they are still compatible because the time-outs used are numbers of rounds. No dataset have been generated for Assumption 1 (no data).

Gold Standard

From the survey involving 10 subjects, we obtained 10 rankings for each topic (*Debian* and *Hibernate*), and different centroids were built for each of them to establish gold standards, with a procedure similar to the one described in the sections 5.1.2 and 5.1.3. The main difference with the procedure described in this thesis is that the survey centroids maintain as much information as possible, so *Unordered* pairs happen only when there is a strictly equal amount of rankings providing *Superior* and *Inferior* for the corresponding pair of stakeholders. In this survey, each topic has 3 centroids: one based on the subjects who worked on that topic first, one based on the subjects who worked on that topic last, and a centroid on all of them. From the analysis of the survey, we concluded that the overall one is the most reliable, so this is the one we

Assumption	Gold Standard
1	(not considered)
2	∅
3	(not considered)
4	Survey-based: - one for *Debian* (13 stakeholders) - one for *Hibernate* (10 stakeholders)

Table 8.3: Gold standard rankings for XWiki discussions.

use here. Consequently, *Debian* has a gold standard ranking 13 stakeholders into 13 ranks (i.e. a total order) and *Hibernate* has a gold standard ranking 10 stakeholders into 7 ranks (the last 3 ranks contain 2 people each). We summarize all the gold standards for each assumption in Table 8.3.

8.3.3 Results

We describe here the main results of the analysis of the datasets, which are detailed in Appendix F. We do not consider Assumption 1 (no data), because no dataset has been generated for it, nor Assumption 3, because no composed query has been generated. The graph for XWiki being too big to compute exact MNs, only the approximative computation is analysed.

Markov Network (approximative)

From the re-run variance, Norm and WoE appear to be the most interesting functions, with an increasing *Agreement* reaching more than 83% of pairs at the highest time-out and, although Norm also shows a significant *Disagreement* (14.2%) it remains low for WoE (4.6%). Id, Norm+5, and S-Norm might also be of interest because they show a similar increase of *Agreement*, but they loose some of it at high time-out. The extra-run bias comforts WoE as being the most stable function: 77.5% of *Agreement* is achieved between the highest time-outs with only 2.9% of *Disagreement*. The other interesting functions show some mirroring: the *Disagreement* tends to increase with the *Agreement*, although there is a bit more of the latter, which shows that rankings are still significantly evolving even at high time-outs.

The compliance to Assumption 2 (no query) is never met, which is explained with the same reason than for the other datasets: although really close, the generated probabilities are not strictly equal, leading to order the stakeholders instead of keeping them at the same rank. Assumption 4 (expected) also shows a poor compliance in Figure 8.9, showing no more than 40% of compliance for any function, with some of them reaching a clear palier, like WoE around 20%.

Genetic Algorithm

The dataset being particularly costly to generate, the data generated so far provides only 3.59 runs per setting in average, with 13.0% having 0 or 1 ranking only, which is a part for which we cannot investigate the variability at all. Consequently, we have focused our evaluation on settings which minimise this threat while remaining realistic, which lead us to consider the settings for 10 stakeholders, 3 topics, and 10 terms, which has 10.5% of settings of 0 or 1 rankings. The fact that only 10 stakeholders are ranked over 18 means that the rankings are incomplete, which has an effect on the interpretation of the results. Indeed, by taking ranking which have 10 random stakeholders while ensuring that they all fully agree, the average *Agreement* is 33.3% (details are provided in the appendix). In other words, we should obtain around this value for showing great *Agreement*, while reaching higher values would show that the rankings focus on a specific subset of stakeholders.

By looking at the re-run variance, we obtain 43.7% of *Agreement* in average and almost no *Disagreement* (1.2% in average). As described above, these are “good” values from an *Agreement* perspective because we do a bit

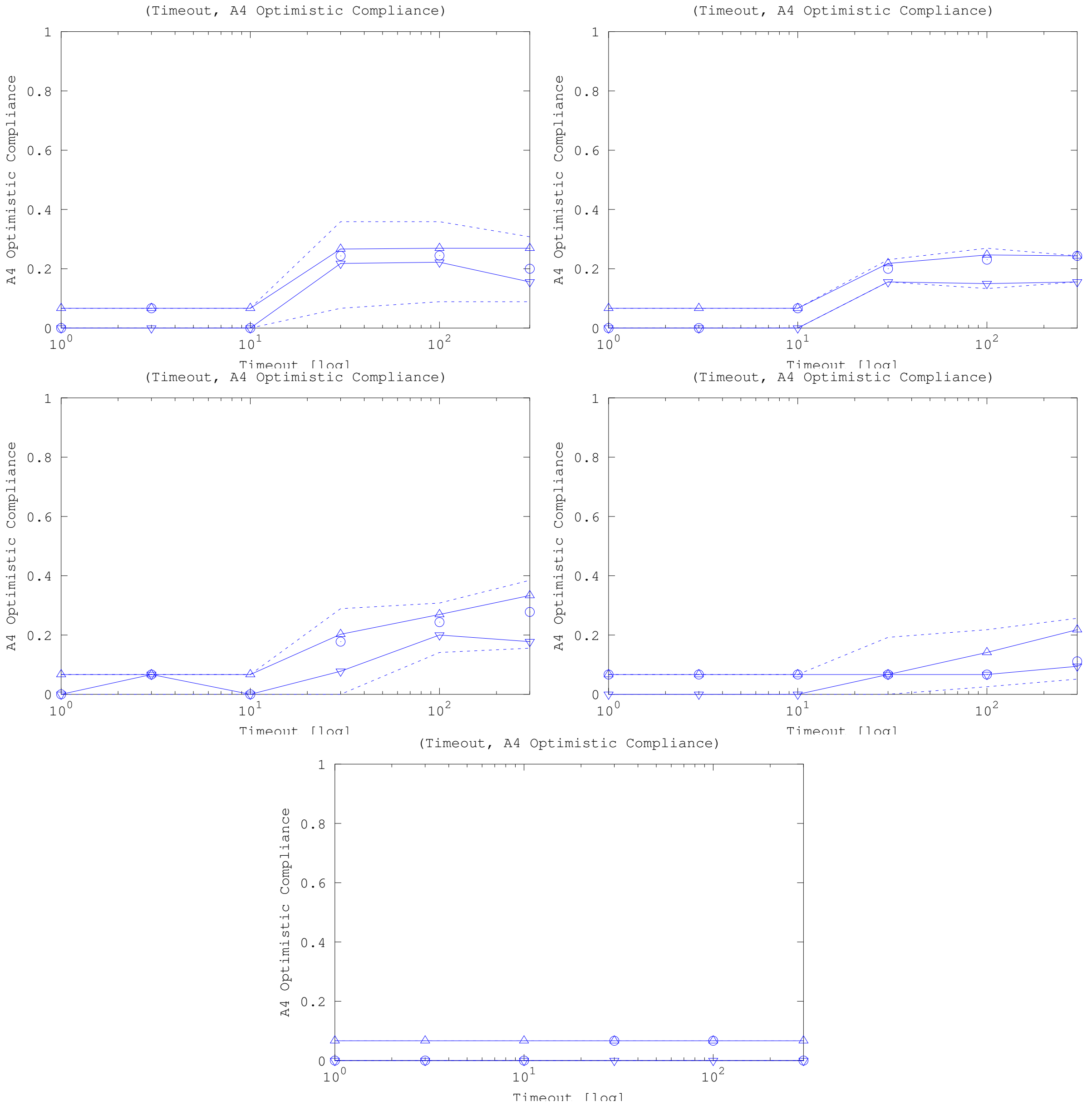


Figure 8.9: Evolution of Assumption 4 (expected) compliance of the MN technique for XWiki data with approximative computation. It shows different behaviours: Id/Norm/S-Norm (top-left) gain in compliance by becoming informative but seem to loose some of it at the highest time-out, WoE (top-right) clearly reaches a palier, Norm+5 (middle-left) seems to increase constantly but the logarithmic scale shows that it becomes costly, S-Norm+5 (middle-right) does not show much because of its late gain of informativeness, and Id+5 (bottom) shows to which extent the lack of orders for half of the gold standards provide some free compliance.

better than random stakeholders, but the most interesting functions are the one able to go significantly beyond. (ST1, MT1) and (ST2, MT2) are such functions: they offer the greatest increase of *Agreement*, particularly at the highest time-out (resp. 74.9% and 98.9%). However, if these results show a high level of *Agreement* between rankings of a same time-out, the extra-run bias is not as high. (ST1, MT1) remain around 40% of *Agreement* between two time-outs, while (ST2, MT2) is even worse by remaining below 20%. The fact that their extra-run *Agreement* is lower than their re-run *Agreement* means that the rankings are still evolving in terms of stakeholders (the *Disagreement* remaining close to zero, this is more probable than an evolution of the orders of the pairs). Other functions might be more interesting from an extra-run perspective, but we rarely go higher than 40% of *Agreement*.

Finally, by looking at the compliance aspect, Assumption 2 (no query) is always satisfied but Assumption 4 (expected) varies between 0% and 31.1%, with all the functions providing similar results, as shown in Figure 8.10. If the maximal level of compliance can also be affected by the incompleteness of the rankings, we only have two queries in our gold standard, and one should still target 100% while the other should target 57.7% (details in appendix). If lower values are achieved, which is the case here, this can be due to wrong ordered pairs, but also to wrong choices of stakeholders to rank. The important point here is that the best ranking achieves 31.1% of compliance, which means that not a single generated ranking properly satisfies Assumption 4.

8.4 Discussion

With the previous sections providing a context-driven perspective of each evaluation, we start this section with an approach-driven summary of the results in Section 8.4.1, thus highlighting the specificities of each approach across the different evaluations. Then, Section 8.4.2 highlights the threats

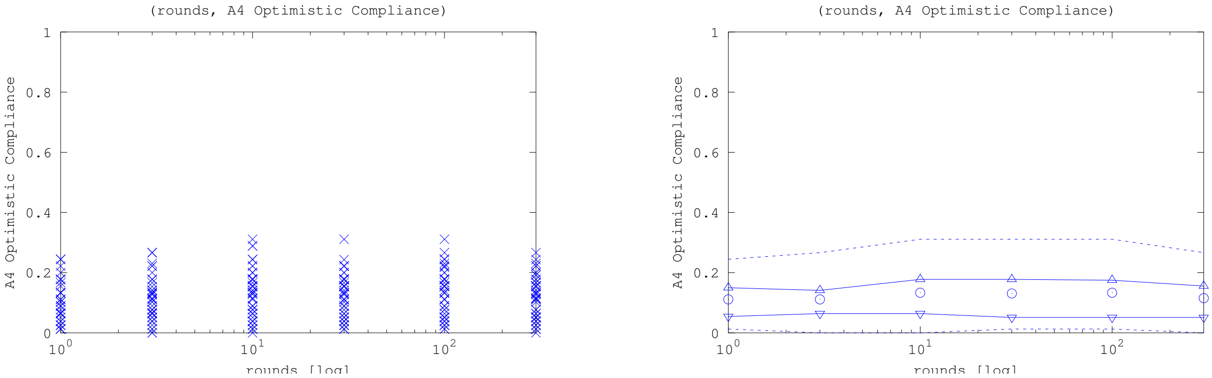


Figure 8.10: Evolution of the assumption compliance of the GA technique for XWiki data. Assumption 4 (expected) is only poorly satisfied. Independently of the functions, we achieve at most 31.1% of compliance.

to validity over all our evaluations, before Section 8.4.3 lists some points to further investigate for improving our approach.

8.4.1 Summary of the Results

Starting from the MN, its exact computation has shown to be of arguable interest across the datasets. First, the approach has been shown to be correct in the extreme case of Assumption 1 (no data), but this validation has been done only through the synthetic data. Similarly, it satisfies Assumption 2 (no query) with the synthetic data, but it fails for the cuisine discussions because of its inability to provide a query without using the approximative computation. A better result has been achieved for Assumption 3 (composition): full compliance has been achieved for the synthetic data, and if we focus on the functions Id+5, Norm+5, and S-Norm+5 we can also claim full compliance for the cuisine discussions. Although this assumption provides the best results, Assumption 4 (expected) seems to be a lot more difficult to satisfy. On the synthetic data, only 73.9% of the pairs are compliant, which seems particularly low for noise-free data and is mainly due to the small coverage of the gold standard. If we restrict to ordered pairs, the compliance drops to 29.8%, which occurs because only s_0 is ranked differently to all the other stakeholders, which makes it rather useless. For the cuisine discussions, although we obtain full compliance for one query, the compliance drops to the minimum for the other, giving a partial compliance too.

If we look at the approximative computation of the MN, which allows to cover also the XWiki evaluation, we observe somehow reversed results: the compliance hardly occurs for the three first assumptions while it reaches good levels for the last one. More precisely, Assumptions 1 (no data) and Assumption 2 (no query) are not satisfied because of the approximative computation which, although it provides values close to 0.5, fails to provide strictly equal values, leading to rank the stakeholders while it should not. For Assumption 3 (composition), we observe some compliance only for some functions with the synthetic data and cuisine discussions (XWiki did not cover this assumption). Assumption 4 (expected) offers the best results by reaching 85.6% of compliance at the highest time-out for the synthetic data and almost if not full compliance for the cuisine discussions. However, XWiki shows that it is not always the case, with no more than 40% of compliance. We can also add that the lack of convergence of the approximative computation towards the results of the exact computation (especially regarding Assumption 4) makes its results more questionable.

Finally, the GA offers the best results in terms of stability as well as compliance. In particular, Assumptions 1 (no data) is fully satisfied, although covered only for synthetic data. We obtain a better result with Assumption 2 (no query), which is fully satisfied in every context (synthetic data, cuisine discussions, and XWiki). Assumption 3 (composition) is also fully satisfied where investigated (XWiki does not cover it). Only Assumption 4 (expected) varies, with fully compliant rankings only for synthetic data. The cuisine discussions offers mitigated results, showing most of the time that only one query is satisfied like for the exact MN computation, while XWiki shows a globally low level of compliance.

In brief, the approximative MN computation and the GA are the most interesting, with the MN providing better results for Assumption 4 and the GA for the other assumptions. But both cases need further investigation: the approximative MN to improve its ability to consider approximative equality and the GA to improve its compliance.

In terms of computation time, the exact MN provide a ranking within 4-18s for synthetic data and 1-2s for the cuisine discussions, while we were not able to determine the time required for XWiki. The approximative computation has the time-out has a parameter, so it is upon a priori decision, but the highest time-outs investigated is 300s (5min). Regarding the GA, it achieves various performances depending on the parameters, in particular the type-specific functions. If we fix the parameters to 3 stakeholders, 0 or 1 role, 3 topics, 10 terms, and 300 rounds to have comparable settings for each context, we obtain the following average results: ST1 runs in 1s with synthetic data, 4s for cuisine discussions, and 14s for XWiki ; ST2 runs in 5s with synthetic data, 36s for cuisine discussions, and 1515s (25min) for XWiki ; ST3 runs in 2s with synthetic data, 14s for cuisine discussions, and 27s for XWiki. This shows that, as expected, ST2 explodes with the size of the graph, motivating to use a simplified version like ST1 or ST3. Based on the results of our evaluations, it appears that ST1 is the most interesting, not only because it is faster, but also because it better complies to our assumptions than ST3.

These results allow us to answer to some extents to our research questions, starting from RQ 1: *Can we design an EF process able to consider the core artefacts (topics, terms, and roles) of the two RE approaches?* Although we obtained some good results, the literature usually focuses on the compliance towards Assumption 4 which, for our approaches, still need to be improved. So we may consider to be on the right path because of the high compliance we achieve with the other assumptions (especially with the GA) and the few good compliance we achieve with Assumption 4, but we cannot provide a definitive answer to this question based on our current results. Regarding RQ 2: *How can we compare incomplete and partially ordered rankings of experts?* Our evaluations showed that we can investigate rather deeply the comparison between such rankings, although we faced some limitation regarding the compliance assessment with incomplete rankings. Some improvement might be needed on this aspect but, from a general perspective, we consider that our formalisation offers a relevant way to compare incomplete and partially ordered rankings.

8.4.2 Threats to Validity

In the following, we try to identify the threats to validity for our evaluation based on the classification of [Wohlin et al., 2012] (chap. 8.8–8.9).

Threats to internal validity We can mention such a threat for the building of the XWiki gold standard [Vergne, 2016a] for which we observed a difference between rankings built on a given set of discussions as the first task or as the second task. This might be due to a learning effect, in particular because the discussions involve common participants, leading the second task to build on some preliminary knowledge acquired from the first task. There was also comments about the difficulty to build the rankings by writing the name of the participants on paper, which takes time and space, and thus might influence the subjects towards not changing their initial ranking even if they think it should be fixed. The fact that the subjects were involved in an experiment rather than facing a real situation requiring them to identify proper experts might have also lead them to be less rigorous on their criteria.

Threats to external validity We only considered three cases, including one completely synthetic and one less controlled but still made in the lab, which might affect the representativeness of our evaluation towards evaluating our approaches. Although the synthetic data was designed with the intent to mimic natural situations, like with the use of the Zipf's law, it remains biased towards what we think to be natural. This dataset involves for instance a lot of redundancy by having stakeholders, topics, and roles related through the same term profiles in a consistent way, which is an aspect for which we have no support. The other contexts show that a more natural dataset may involve a graph without roles and more generally with partial information, but also with noisy data, what was not considered with synthetic data. The cuisine discussions has also been produced completely out of a RE context, which makes them less representative of a source of data used in this context.

Threats to construct validity It might be that the lack of compliance to Assumption 4 for cuisine and XWiki data for the GA comes from some inadequacy of our approach to properly represent the quantity of expertise of the stakeholders. In particular, we inspired from existing approaches in RE but we miss for instance the PERCEIVED DOMAIN SKILL dimension which is represented in our meta-model of expertise. The incomplete rankings faced with XWiki (10 stakeholders over 18) stress the suitability of our measures, for which we needed to revise the compliance target. The settings investigated are also rather limited, in particular regarding the time-outs investigated through a logarithmic distribution which covers only few instants of the whole life cycle of the ranking generation. Moreover, we have only considered topic-based queries, while roles and terms can also be queried, leading to reduce the investigation of the query. There was also only three stakeholders to rank for the cuisine discussions, leading to a rather limited case with only 10 possible rankings (6 totally ordered, 3 partially ordered, 1 uninformative), thus making it easier to satisfy. The topic nodes for the cuisine discussions were also only 3, which is extremely limited to stress side effects due to unrelated topics, and roles were absent from the cuisine discussions and XWiki.

Threats to conclusion validity These threats are more numerous, but some of them are mitigated because they only apply to some evaluations. Regarding the compliance aspect, Assumptions 1 (no data) has been covered only with the synthetic data. The compliance to Assumption 3 is also not fully evaluated: we only considered the combinations of 2 topics although we can go until 5 for synthetic data, and it was ignored for XWiki. The compliance to Assumption 4 for synthetic data does not refer to proper expertise, but to expected results from a formalization perspective, which makes it arguably suited to draw conclusions on the ability of our approach to rank experts. Regarding XWiki, we can also mention that the GA analysis is reduced to few settings (10 stakeholders, 3 topics, 10 terms) and faces an issue regarding its small amount of runs per setting, which hurts its ability to provide robust results. The gold standard for Assumption 4 has also been built through a survey which does not remove all doubts: expertise levels have been evaluated through a limited knowledge based on few discussions, and the validation of these gold standards builds mainly on the consistency between the subject rankings (i.e. social agreement) and self-assessment of the subject expertise, which we know from the literature to be among the good but not best indicators [Ericsson, 2006b].

8.4.3 Sources of Improvements

First, we might highlight interesting improvements to do at the evaluation level. On aspect is on stressing the approach robustness, in particular by introducing noise in the data, like adding/removing nodes or altering the weights of the relations. It would be interesting to establish some kinds of signal-noise ratio from which the rankings start to be impacted depending on the size of the graph, the scales of the weights, and other properties. We can also consider the correctness of the approach, which is stressed through its compliance to gold standards, especially the assumptions 1, 2, and 4. While the two firsts build on theoretical gold standards common to any context, Assumption 4 (expected) build on context-specific rankings, which means that we should ensure validate them. In our evaluations, we used simple measures, like self-assessment and agreement between subjects, but more robust validation methods need to be used to provide proper guarantees, especially by looking at performance in representative and authentic tasks of the domain [Ericsson, 2006d, Ericsson, 2006c]. We may recommend for instance to look at [Ericsson, 2006a], especially the chapters 8 to 14 which focus on several processes for evaluating expertise.

Regarding our approaches, we can highlight specific issues worth to investigate. In particular the need to manage better close values when computing approximative MNs to properly have *Unordered* pairs when the values are close enough to be considered equal. The extraction process for the cuisine discussions and XWiki could also be improved, and not only by improving the cleaning of the noise. We used for instance these cuisine discussions in [Morales-Ramirez et al., 2014], were we considered the intentional aspect (e.g. responses, suggestions, questions) in the e-mails to refine the weights of the relations. No deep investigation were run with these intentional components, which is why we did not include them in this thesis, but we definitely think that it is an important aspect to consider with natural language sources to better represent the actual knowledge of the participants. We also saw through our evaluations that the approximative MN does not provide the same results than the exact one, and further investigation on this aspect may highlight some fundamental differences between the exact and approximative computation. These differences may be related to actual properties of the expertise evaluation, thus motivating to process the data in a progressive manner rather than relying on a “one-shot” technique like the exact computation. If we consider the GA approach, by confirming the perfect compliance with synthetic data through ST2 we show evidence of consistency between (i) our interpretation of the data represented through the synthetic case, (ii) our modelling of it as a weighted graph of correlated S/R/T/C nodes, and (iii) the computation which maximizes evidences. Yet, this high compliance is not achieved with actual expertise in non-synthetic evaluations through Assumption 4, thus it is probable that the investigation should focus more on (i) the reliability of the gold standards built and (ii) on the very initial interpretation of how to model expertise. For example, is the lack of relation between two nodes of the same type in our model justified? Are the profiles (e.g. term and stakeholder profiles in the synthetic data) represented well enough to be properly considered? etc.

Finally, we can also stress our formalisation of experts rankings. We saw that our compliance measures do not fit well when the reference ranking is incomplete, so further revisions of these measures would be of interest, for instance by choosing a more suited normalisation factor. It might also be that, like ODD and PDD offers complementary perspectives worth to keep together, compliance measures require to evaluate complementary properties to provide sounding values. We also saw that Assumption 4 might be evaluated not only through $OptimComp(v, gs)$, but also through $OrderComp(v, gs)$, which further support the need to find complementary measures. We might also consider different kinds of *Unordered* pairs: the ones for which we *cannot* tell by lack of evidences, and the ones for which we *should not* tell because of the presence of contradictory evidences. This differences might support the need to introduce the equality in our formalisation, not as a default when no order can be given, but as a way to express the irrelevance of ordering one stakeholder above another for the given query. For example, for a query about *cryptography*, one stakeholder could be better on teaching the field while another could be better on designing novel techniques, making the decision to use a given order (*Superior* or *Inferior*) a biased evaluation towards specific dimensions of the domain. Such a difference could be made when querying for *cryptography teaching* or *cryptography design*, but not at the more general level of *cryptography*. A last relevant improvement we see is the design of measures focusing on the top experts, like in IR measures but adapted to the consider *Unordered* pairs, for instance by considering the number of pairs making a stakeholder *Inferior* or *Superior* to another.

Part IV

Conclusion

Chapter 9

Summary and Future Works

This thesis is interdisciplinary by the way it bridges different fields in order to come out with several kinds of contributions, whether they are conceptual by building on Psychology literature, formal by adapting well known measures to the specific context of EF, practical by using different approaches to process our data, or methodological by enriching usual processes with more measures and control. We summarize these contributions in Section 9.1 before to tell how they help us to answer our research questions in Section 9.2. We then conclude by highlighting in Section 9.3 potential extensions and improvements that can be done as future works.

9.1 Contribution Summary

Our first contribution is our meta-model of expertise described in Chapter 4, which maps expertise-related concepts from literature in Psychology with concepts in EF and RE. In particular, by modelling the Evaluator who evaluate some Performers, we consider the role of an EF system towards the people it will recommend. Furthermore, we map the Perceived Domain Knowledge and the Social Recognition to the notions of topics/terms and roles that we retrieve in RE works, and which provide the inputs of this EF system. By going as far as mapping the concept of Performance Evaluation, and all its specialisations, to the usual concept of expert ranking, we also manage the output of this EF system, allowing us to close the loop. Although limited, our model allows us to analyse existing EF systems and shows some interests in helping to design new ones.

Building on this first contribution, we go further in Chapter 6 by inspiring from those works in RE which are close to existing EF systems due to the techniques they use. We establish an approach which considers the same kinds of indicators of accessible knowledge and social recognition, but in a more comprehensive way by considering also their inter-relations. We design one version based on MNs, which focuses on computing probabilities, as well as a version based on a GA, which exploits the advantages of optimization techniques to fix some issues observed with the first version.

Not satisfied with applying usual evaluation procedures, which focus on the correctness of some rankings based often on gold standards which are themselves hard to guarantee as correct, we extend this procedure in Chapter 7 by considering two phases. The first phase focuses on identifying settings which provide stable results, a phase which diminishes the arbitrariness of selecting relevant settings, but also offers preliminary measures to identify instability problems earlier, without having to pass through a costly gold standard building. The second phase, which reuses the usual gold standard validation, extends it with three other assumptions to satisfy, which cover the correctness as well as the consistency of the produced rankings, and which are based on predefined gold standards. This evaluation procedure has been applied to three different contexts in the chapters 8.1 to 8.3, each highlighting specific aspects from the fully controlled data to the fully uncontrolled one.

This revision and extension of the evaluation procedure has been possible because of a last contribution of this thesis, which is our revision of the formalisation of a ranking of experts, which was initially based on IR assumptions through the analogy with a ranking of documents. We revised this formalisation in Chapter 5 by (i) introducing the notion of *Unordered* pair and (ii) by allowing the discard of the transitivity principle of usual orders to build orderings, what we consider to be a requirement to properly deal with rankings which are partially ordered and incomplete. With this formalisation, we provided a procedure to easily build centroids, enriched with a procedure to transform an ordering into a ranking, but also new measures able to compute the distance between two orderings, or to evaluate their compliance to a reference. All these measures have been shown to build on simple and broadly recognised measures already used in IR (precision and recall), but used in a novel way on pairs of items to make them able to (i) consider orders, (ii) deal with partial orders, and (iii) be more robust to incomplete rankings.

9.2 RQs Answers

Based on our contributions, we can finally attempt to answer the research questions presented in Section 3.2.

RQ 1. *Can we design an EF process able to consider the core artefacts (topics, terms, and roles) of the two RE approaches?*

By designing our techniques based on a MN or a GA, we intended to establish a novel EF system, thus a system that we could use to obtain relevant recommendations of expert. As shown by our evaluations, MNs can provide some good results with approximative computation but are unable to satisfy some assumptions due to these approximations, while the GA has shown a complementary performance by satisfying these assumptions affected by approximations. So far, we still fail to achieve high levels of compliance for the gold standard used in usual evaluation procedures, unless we consider completely synthetic data. We might argue that these gold standards are hard to validate themselves, but we believe that in contexts where a high agreement is achieved among all the actors involved, like the evaluation on cuisine discussions in Section 8.2, it is unreasonable to call for a potential wrong gold standard. As such, our techniques still need improvement, which means that we have no strong support to provide an affirmative answer to RQ 1, but the fact that we could obtain a GA which is entirely compliant for synthetic data in Section 8.1, added to the ability for this approach to greatly comply with all the assumptions but the usual gold standard in other cases, which still delivers a partial compliance, let us think that we are on the right path. In particular, some more investigation might be needed for extracting the right weights, so focusing on the extraction algorithms, but also on using sources providing evidences about skills, which has not been considered so far but is part of the relevant indicators present in our meta-model of expertise.

RQ 2. *How can we compare incomplete and partially ordered rankings of experts?*

This question came from fundamental issues that we had while trying to build our gold standards. One is that it is hazardous to enforce people to provide totally ordered rankings, because a lack of information justifies the use of partial orders, but also because we faced a case where an increasing feeling of self-expertise lead to provide rankings which are more partially ordered than people who feel less experts. Another issue is the very practical observation that human-made rankings cannot be as exhaustive than computer-made rankings, which lead us to think that the usual requirements behind rankings comparison do not fit for EF. We intended to solve this problem through our novel formalisation, described in Chapter 5, and to show its ability to deal with incomplete and partially ordered rankings through our evaluations. This was particularly illustrated with the third evaluation in Chapter 8.3, for which we had incomplete and partially ordered rankings (computer-made as well as human-made), yet we were able to properly evaluate their compliance, although some improvements might be required to compute values more robust to the incompleteness (the highest compliance achievable was lower because of this incompleteness). Consequently, we consider that we have provided a clear evidence for providing an affirmative answer to RQ 2, and that measures based on our formalisation are good candidates to support such comparisons, whether we speak about symmetric agreement or asymmetric compliance.

RQ 3. *How can we support the correction of an existing EF system?*

For this question, we have built a meta-model of expertise in Chapter 4 which has provided a good support for us to identify potential lacks in existing EF systems. However, we miss the empirical confirmation that these lacks *need* to be fixed for achieving better expertise evaluation. But although this absence of confirmation makes us unable to offer a definitive answer to RQ 3, our meta-model also seems to offer support in designing new EF systems. Indeed, it helps to drive the identification of relevant elements to consider, like evidences of skills, knowledge, and social recognition, which should not only lead to observe lengthy, domain-related experience but also reproducibly superior performance. We think that this kind of support is of particular help for people not familiar with the core notion of expertise and how to evaluate it, and thus makes a good starter for people working in the RE field.

In brief, until a fully compliant EF system is made, we can hardly provide a definitive answer to RQ 1 and RQ 3. Moreover, although we can provide a more robust answer to RQ 2, it is nevertheless confirmed only by some theoretical grounding and few uses in our evaluation. More empirical evidences is needed to properly assess the suitability of our formalism to deal with EF contexts. This thesis might appear as too dispersed through its attempt to bring, in addition to a novel EF system, some insights from literature in Psychology formalised into a meta-model of expertise, as well as a deep revision of the formalisation and evaluation of expert rankings which involves logics and mathematics and goes against traditional uses of IR measures. Yet, we think that a comprehensive work which investigates several aspects of EF was a required attempt, and we find further motivation in our position from [Cheng and Atlee, 2009]:

> *Researchers need to think beyond current RE and SE knowledge and capabilities, in order to make significant headway in addressing the challenges posed by emerging systems. They need to be willing to search for new solutions that may lead to paradigm shifts in RE practices, at the risk of possible failures.*
>
> Betty H.C. Cheng and Joanne M. Atlee

9.3 Going Further

Through this thesis, we discussed each of our contributions in many ways, trying to identify not only interesting advantages but also limitations requiring additional work. In this last section, we summarize the elements that we think to be of highest interest for future works.

Regarding the expertise meta-model presented in Chapter 4, an obvious future work is related to its incompleteness. We rely exclusively on literature in Psychology to identify the main concepts, which gives us a top-down approach, while it could be complemented with a bottom-up approach, like [Yimam-Seid and Kobsa, 2003], which analyses existing EF techniques to identify relevant low level concepts. Furthermore, because it is based on literature about expertise, our meta-model can be considered as a relevant basis for building an ontology (i.e. a specification of a shared conceptualization [Guarino et al., 2009]). With our top-down approach, such an ontology would be categorised as an *upper ontology* of expertise evaluation, which means an ontology offering domain-generic concepts and relations to support the description of expertise evaluators like EF systems. By completing it with bottom-up approaches, we could design *domain ontologies*, which means ontologies focusing on a specific domain, like expertise evaluation in research, or in software development, etc. Researchers in RE are also currently interested in the notion of creativity, which is also a way to identify some of the highest experts in a domain [Ericsson, 1999]. Other perspectives like this one could consequently be added to this meta-model or ontology.

For our formalisation of experts ranking, described in Chapter 5, we can also highlight future works that we think to be of first importance. One of the most important is probably to find some equivalence with current measures able to give priority to top stakeholders, like CGk and derived measures [Vergne, 2016b]. An interesting direction could be towards counting the number of ordered pairs which make an item inferior, what we see as a direct equivalent of the notion of rank (no pair for the first rank, one pair for the second, and so on). Additionally, counting the number of pairs which make it superior provides a complementary measure, which allows to consider different ranking strategies used to deal with ties. For instance, for a ranking $a > (b, c) > d$, a standard ranking would assign the ranks $(a = 1, b = c = 2, d = 4)$, which corresponds to the number of pairs making them inferior to which we add 1. Another standard ranking would assign them the maximal ranks, so $(a = 1, b = c = 3, d = 4)$, which can be retrieved by computing the number of pairs making the item superior and removing it from the total number of items. A third standard is the fractional ranking, so $(a = 1, b = c = 2.5, d = 4)$, which is the average of both. Probably we can extend our formalisation with these concepts to give priorities to top experts. A second important aspect that we consider is the weight of the *Unordered* pairs when building centroids: maybe more investigation on the use of an equal order may help to give a more systematic procedure.

Finally, through Chapter 6 and Part III, which relate to our EF approach and its evaluation, we saw that some additional work is required before to obtain something usable. In particular, we think that the information we extract from our sources is rather limited, in part because we did not use roles in any of evaluation involving non-synthetic data. Additionally, a quick look at our meta-model shows that we also totally ignore the skill dimension, which makes it also a relevant addition to consider for future works. With today's "public lifes" (i.e. open source repositories like GitHub, open source measures like Sonar, etc.) it should be also easier to find indicators of reproducibly superior performance, which is required to find the highest levels of expertise, so it might be interesting to give a look to these sources. Finally, if our meta-model of expertise is extended to become a fully featured ontology, it might be interesting to design an EF system relying on the formal reasoning of ontologies to infer expert rankings.

What we hope is that the contributions we have presented in this thesis give some reliable basis to build on, in order to investigate all these aspects from a solid starting block.

Bibliography

[Ackerman and Beier, 2006] Ackerman, P. L. and Beier, M. E. (2006). Methods for Studying the Structure of Expertise: Psychometric Approaches. In *The Cambridge Handbook of Expertise and Expert Performance*, Cambridge Handbooks in Psychology. Cambridge University Press, ISBN: 978-0-511-81679-6, http://dx.doi.org/10.1017/CBO9780511816796.009.

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). 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, ISSN: 1041-4347, DOI: 10.1109/TKDE.2005.99.

[Balog, 2008] Balog, K. (2008). *People search in the enterprise*. PhD thesis, University of Amsterdam, http://hdl.handle.net/11245/1.296653. ISBN: 978-90-90-23247-8.

[Balog, 2012] Balog, K. (2012). Expertise Retrieval. *Foundations and Trends® in Information Retrieval*, 6(2-3):127–256, ISSN: 1554-0669, 1554-0677, DOI: 10.1561/1500000024, http://www.nowpublishers.com/article/Details/INR-024.

[Bédard and Chi, 1992] Bédard, J. and Chi, M. T. H. (1992). Expertise. *Current Directions in Psychological Science*, 1(4):135–139, ISSN: 0963-7214, http://www.jstor.org/stable/20182156.

[Berkson, 1944] Berkson, J. (1944). Application to the Logistic Function to Bio-Assay. *Journal of the American Statistical Association*, 39(227):357, ISSN: 01621459, DOI: 10.2307/2280041, http://www.jstor.org/stable/2280041?origin=crossref.

[Boehm and Turner, 2008] Boehm, B. and Turner, R. (2008). *Balancing agility and discipline: a guide for the perplexed*. Addison-Wesley, Boston, Mass., 6. print edition, ISBN: 978-0-321-18612-6, https://books.google.com/books?id=C6DDzaAuI48C.

[Borgatti, 2005] Borgatti, S. P. (2005). Centrality and network flow. *Social Networks*, 27(1):55–71, ISSN: 0378-8733, DOI: 10.1016/j.socnet.2004.11.008, http://www.sciencedirect.com/science/article/pii/S0378873304000693.

[Bozzon et al., 2013] Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., and Vesci, G. (2013). Choosing the Right Crowd: Expert Finding in Social Networks. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT '13, pages 637–648, New York, NY, USA. ACM, ISBN: 978-1-4503-1597-5, DOI: 10.1145/2452376.2452451, http://doi.acm.org/10.1145/2452376.2452451.

[Castro-Herrera and Cleland-Huang, 2009] Castro-Herrera, C. and Cleland-Huang, J. (2009). A Machine Learning Approach for Identifying Expert Stakeholders. In *2009 Second International Workshop on Managing Requirements Knowledge (MARK)*, pages 45–49. DOI: 10.1109/MARK.2009.1.

[Castro-Herrera and Cleland-Huang, 2010] Castro-Herrera, C. and Cleland-Huang, J. (2010). Utilizing recommender systems to support software requirements elicitation. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineer-*

ing, RSSE '10, pages 6–10, New York, NY, USA. ACM, ISBN: 978-1-60558-974-9, DOI: 10.1145/1808920.1808922, http://doi.acm.org/10.1145/1808920.1808922.

[Cheng and Atlee, 2009] Cheng, B. H. C. and Atlee, J. M. (2009). Current and Future Research Directions in Requirements Engineering. In Lyytinen, K., Loucopoulos, P., Mylopoulos, J., Robinson, B., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M. J., and Szyperski, C., editors, *Design Requirements Engineering: A Ten-Year Perspective*, volume 14 of *Lecture Notes in Business Information Processing*, pages 11–43. Springer Berlin Heidelberg, ISBN: 978-3-540-92966-6, http://www.springerlink.com/content/p478k58932626j2m/abstract/.

[Chi, 2006] Chi, M. T. H. (2006). Two Approaches to the Study of Experts' Characteristics. In Ericsson, K. A., Charness, N., Feltovich, P. J., and Hoffman, R. R., editors, *The Cambridge handbook of expertise and expert performance*, pages 21–30. Cambridge University Press, New York, NY, US, ISBN: 0-521-60081-2 (Paperback); 0-521-84097-X (Hardcover); 978-0-521-60081-1 (Paperback); 978-0-521-84097-2 (Hardcover).

[Cleland-Huang and Laurent, 2014] Cleland-Huang, J. and Laurent, P. (2014). Requirements in a Global World. *IEEE Software*, 31(6):34–37, ISSN: 0740-7459, DOI: 10.1109/MS.2014.144.

[Cunningham et al., 2011] Cunningham, H., Bončeva, K., and Maynard, D. (2011). *Text Processing with GATE*. University of Sheffield Dept. of Computer Science, Sheffield, ISBN: 978-0-9565993-1-5 0-9565993-1-1.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, ISSN: 1089-778X, DOI: 10.1109/4235.996017.

[Durillo and Nebro, 2011] Durillo, J. J. and Nebro, A. J. (2011). jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, ISSN: 0965-9978, DOI: 10.1016/j.advengsoft.2011.05.014, http://www.sciencedirect.com/science/article/pii/S0965997811001219.

[Dutoit and Paech, 2003] Dutoit, A. H. and Paech, B. (2003). Eliciting and Maintaining Knowledge for Requirements Evolution. In Aurum, A., Jeffery, R., Wohlin, C., and Handzic, M., editors, *Managing Software Engineering Knowledge*, pages 135–155. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-05573-7 978-3-662-05129-0, http://link.springer.com/10.1007/978-3-662-05129-0_7.

[Ericsson, 1999] Ericsson, K. A. (1999). Creative expertise as superior reproducible performance: Innovative and flexible aspects of expert performance. *Psychological Inquiry*, 10(4):329–333, http://www.tandfonline.com/doi/pdf/10.1207/S15327965PLI1004_5.

[Ericsson, 2006a] Ericsson, K. A., editor (2006a). *The Cambridge Handbook of Expertise and Expert Performance.* Cambridge University Press, Cambridge ; New York, ISBN: 978-0-521-84097-2 978-0-521-60081-1, http://www.cambridge.org/us/academic/subjects/psychology/cognition/cambridge-handbook-expertise-and-expert-performance.

[Ericsson, 2006b] Ericsson, K. A. (2006b). The Influence of Experience and Deliberate Practice on the Development of Superior Expert Performance. In Ericsson, K. A., Charness, N., Feltovich, P. J., and Hoffman, R. R., editors, *The Cambridge handbook of expertise and expert performance*, pages 683–703. Cambridge University Press, New York, NY, US, ISBN: 0-521-60081-2 (Paperback); 0-521-84097-X (Hardcover); 978-0-521-60081-1 (Paperback); 978-0-521-84097-2 (Hardcover).

[Ericsson, 2006c] Ericsson, K. A. (2006c). An Introduction to Cambridge Handbook of Expertise and Expert Performance: Its Development, Organization, and Content. In Ericsson, K. A., Charness, N., Feltovich, P. J., and Hoffman, R. R., editors, *The Cambridge handbook of expertise and expert performance*, pages 3–19. Cambridge University Press, New York, NY, US, ISBN: 0-521-60081-2 (Paperback); 0-521-84097-X (Hardcover); 978-0-521-60081-1 (Paperback); 978-0-521-84097-2 (Hardcover).

[Ericsson, 2006d] Ericsson, K. A. (2006d). Protocol Analysis and Expert Thought: Concurrent Verbalizations of Thinking during Experts' Performance on Representative Tasks. In Ericsson, K. A., Charness, N., Feltovich, P. J., and Hoffman, R. R., editors, *The Cambridge handbook of expertise and expert performance*, pages 223–241. Cambridge University Press, New York, NY, US, ISBN: 978-0-521-60081-1 978-0-521-84097-2.

[Ericsson et al., 1993] Ericsson, K. A., Krampe, R. T., and Teschromer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3):363–406, ISSN: 1939-1471 (Electronic); 0033-295X (Print), DOI: 10.1037/0033-295X.100.3.363.

[Everett and Borgatti, 2005] Everett, M. G. and Borgatti, S. P. (2005). Extending centrality. *Models and methods in social network analysis*, 35(1):57–76, https://books.google.it/books?id=4Ty5xP_KcpAC&oi=fnd&pg=PA57&ots=9MEIuevaB0&sig=FBijK_K54Qm0Ei54IMr6SKzd6nA.

[Felfernig and Burke, 2008] Felfernig, A. and Burke, R. (2008). Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce*, ICEC '08, pages 3:1–3:10, New York, NY, USA. ACM, ISBN:

978-1-60558-075-3, DOI: 10.1145/1409540.1409544, http://doi.acm.org/10.1145/1409540.1409544.

[Freeman, 1978] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, ISSN: 0378-8733, DOI: 10.1016/0378-8733(78)90021-7, http://www.sciencedirect.com/science/article/pii/0378873378900217.

[Good, 1960] Good, I. J. (1960). Weight of Evidence, Corroboration, Explanatory Power, Information and the Utility of Experiments. *Journal of the Royal Statistical Society. Series B (Methodological)*, 22(2):319–331, ISSN: 0035-9246, DOI: 10.2307/2984102, http://www.jstor.org/stable/2984102. ArticleType: research-article / Full publication date: 1960 / Copyright © 1960 Royal Statistical Society.

[Groff and Jones, 2012] Groff, T. and Jones, T. (2012). *Introduction to Knowledge Management*. Routledge, ISBN: 978-1-136-39240-5.

[Guarino et al., 2009] Guarino, N., Oberle, D., and Staab, S. (2009). What Is an Ontology? In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-70999-2 978-3-540-92673-3, http://link.springer.com/10.1007/978-3-540-92673-3_0.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY, ISBN: 978-0-387-84857-0 978-0-387-84858-7, http://link.springer.com/10.1007/978-0-387-84858-7.

[Hofmann et al., 2010] Hofmann, K., Balog, K., Bogers, T., and de Rijke, M. (2010). Contextual factors for finding similar experts. *Journal of* *the American Society for Information Science and Technology*, 61(5):994–1014, ISSN: 15322882, DOI: 10.1002/asi.21292, http://doi.wiley.com/10.1002/asi.21292.

[IEEE Standards Board, 1990] IEEE Standards Board (1990). *IEEE standard glossary of software engineering terminology*. Institute of Electrical and Electronics Engineers, New York, N.Y, ISBN: 978-1-55937-067-7.

[Ishibuchi et al., 2008] Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. pages 2419–2426. IEEE, ISBN: 978-1-4244-1822-0, DOI: 10.1109/CEC.2008.4631121, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4631121.

[Karimzadehgan et al., 2009] Karimzadehgan, M., White, R. W., and Richardson, M. (2009). Enhancing Expert Finding Using Organizational Hierarchies. In Boughanem, M., Berrut, C., Mothe, J., and Soule-Dupuy, C., editors, *Advances in Information Retrieval*, number 5478 in Lecture Notes in Computer Science, pages 177–188. Springer Berlin Heidelberg, ISBN: 978-3-642-00957-0 978-3-642-00958-7, http://link.springer.com/chapter/10.1007/978-3-642-00958-7_18.

[Kendall, 1938] Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81, ISSN: 00063444, DOI: 10.2307/2332226, http://www.jstor.org/stable/2332226.

[Kindermann and Snell, 1980] Kindermann, R. and Snell, American Mathematical Society, J. L. (1980). *Markov random fields and their applications*. American Mathematical Society, Providence, R.I., ISBN: 0-8218-3381-2 978-0-8218-3381-0, http://catalog.hathitrust.org/api/volumes/oclc/6762242.html.

[Lim et al., 2010] Lim, S. L., Quercia, D., and Finkelstein, A. (2010). StakeNet: using social networks to analyse the stakeholders of large-scale software projects. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ICSE '10, pages 295–304, New York, NY, USA. ACM, ISBN: 978-1-60558-719-6, DOI: 10.1145/1806799.1806844, http://doi.acm.org/10.1145/1806799.1806844.

[Liu et al., 2013] Liu, D.-R., Chen, Y.-H., Kao, W.-C., and Wang, H.-W. (2013). Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management*, 49(1):312–329, ISSN: 0306-4573, DOI: 10.1016/j.ipm.2012.07.002, http://www.sciencedirect.com/science/article/pii/S0306457312000891.

[Loucopoulos and Karakostas, 1995] Loucopoulos, P. and Karakostas, V. (1995). *System requirements engineering*. McGraw-Hill international series in software engineering. McGraw-Hill Book Co, London ; New York, ISBN: 978-0-07-707843-0.

[Maalej and Thurimella, 2009] Maalej, W. and Thurimella, A. (2009). Towards a Research Agenda for Recommendation Systems in Requirements Engineering. In *2009 Second International Workshop on Managing Requirements Knowledge (MARK)*, pages 32 –39. DOI: 10.1109/MARK.2009.12.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press, New York, ISBN: 978-0-521-86571-5.

[Marwick, 2001] Marwick, A. (2001). Knowledge management technology. *IBM Systems Journal*, 40(4):814–830, ISSN: 0018-8670, DOI: 10.1147/sj.404.0814.

[Maybury, 2006] Maybury, M. T. (2006). Expert finding systems. Technical report, MITRE Center for Integrated Intelligence Systems Bedford, Massachusetts, USA, http://infoautoclassification.org/public/articles/Maybury_MITRE-Technical-Report-Expert-Finding-Systems.pdf.

[McDonald and Ackerman, 2000] McDonald, D. W. and Ackerman, M. S. (2000). Expertise recommender: a flexible recommendation system and architecture. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 231–240. ACM Press, ISBN: 978-1-58113-222-9, DOI: 10.1145/358916.358994, http://portal.acm.org/citation.cfm?doid=358916.358994.

[Michalewicz and Fogel, 2004] Michalewicz, Z. and Fogel, D. B. (2004). *How to Solve It: Modern Heuristics*. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-662-07807-5, http://dx.doi.org/10.1007/978-3-662-07807-5.

[Mockus and Herbsleb, 2002] Mockus, A. and Herbsleb, J. D. (2002). Expertise browser: a quantitative approach to identifying expertise. In *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, pages 503–512, New York, NY, USA. ACM, ISBN: 1-58113-472-X, DOI: 10.1145/581339.581401, http://doi.acm.org/10.1145/581339.581401.

[Mohebzada et al., 2012] Mohebzada, J., Ruhe, G., and Eberlein, A. (2012). Systematic mapping of recommendation systems for requirements engineering. In *2012 International Conference on Software and System Process (ICSSP)*, pages 200 –209. DOI: 10.1109/ICSSP.2012.6225965.

[Mooij, 2010] Mooij, J. M. (2010). libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models. *Journal of Machine Learning Research*, 11:2169–2173, ISSN: 1532-4435, http://dl.acm.org/citation.cfm?id=1859890.1859925.

[Morales-Ramirez et al., 2012a] Morales-Ramirez, I., Vergne, M., Morandini, M., Sabatucci, L., Perini, A., and Susi, A. (2012a). Revealing the obvious?: A retrospective artefact analysis for an ambient assisted-living project. In *2012 IEEE Second International Workshop on Empirical Requirements Engineering (EmpiRE)*, pages 41 –48. DOI: 10.1109/EmpiRE.2012.6347681.

[Morales-Ramirez et al., 2012b] Morales-Ramirez, I., Vergne, M., Morandini, M., Sabatucci, L., Perini, A., and Susi, A. (2012b). Where Did the Requirements Come from? A Retrospective Case Study. In Castano, S., Vassiliadis, P., Lakshmanan, L. V., and Lee, M. L., editors, *Advances in Conceptual Modeling*, number 7518 in Lecture Notes in Computer Science, pages 185–194. Springer Berlin Heidelberg, ISBN: 978-3-642-33998-1 978-3-642-33999-8, http://link.springer.com/chapter/10.1007/978-3-642-33999-8_23.

[Morales-Ramirez et al., 2014] Morales-Ramirez, I., Vergne, M., Morandini, M., Siena, A., Perini, A., and Susi, A. (2014). Who is the Expert? Combining Intention and Knowledge of Online Discussants in Collaborative RE Tasks. In *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, pages 452–455, New York, NY, USA. ACM, ISBN:

978-1-4503-2768-8, DOI: 10.1145/2591062.2591103, http://doi.acm.org/10.1145/2591062.2591103.

[Nebro et al., 2015] Nebro, A. J., Durillo, J. J., and Vergne, M. (2015). Redesigning the jMetal Multi-Objective Optimization Framework. In *Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1093–1100. ACM Press, ISBN: 978-1-4503-3488-4, DOI: 10.1145/2739482.2768462, http://dl.acm.org/citation.cfm?doid=2739482.2768462.

[Nuseibeh and Easterbrook, 2000] Nuseibeh, B. and Easterbrook, S. (2000). Requirements engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA. ACM, ISBN: 1-58113-253-0, DOI: 10.1145/336512.336523, http://doi.acm.org/10.1145/336512.336523.

[Omidvar et al., 2014] Omidvar, A., Garakani, M., and Safarpour, H. R. (2014). Context based user ranking in forums for expert finding using WordNet dictionary and social network analysis. *Inf Technol Manag*, 15(1):51–63, ISSN: 1385-951X, 1573-7667, DOI: 10.1007/s10799-013-0173-x, http://link.springer.com/article/10.1007/s10799-013-0173-x.

[Pohl, 1994] Pohl, K. (1994). The three dimensions of requirements engineering: A framework and its applications. *Information Systems*, 19(3):243–258, ISSN: 03064379, DOI: 10.1016/0306-4379(94)90044-2, http://linkinghub.elsevier.com/retrieve/pii/0306437994900442.

[Ricci et al., 2011] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer US, Boston, MA, ISBN: 978-0-387-85819-7 978-0-387-85820-3, http://link.springer.com/10.1007/978-0-387-85820-3.

[Richardson and Domingos, 2006] Richardson, M. and Domingos, P. (2006). Markov logic networks. *Mach Learn*, 62(1-2):107–136, ISSN: 0885-6125, 1573-0565, DOI: 10.1007/s10994-006-5833-1, http://link.springer.com/article/10.1007/s10994-006-5833-1.

[Robillard et al., 2010] Robillard, M., Walker, R., and Zimmermann, T. (2010). Recommendation Systems for Software Engineering. *IEEE Software*, 27(4):80 –86, ISSN: 0740-7459, DOI: 10.1109/MS.2009.161.

[Serdyukov and Hiemstra, 2008] Serdyukov, P. and Hiemstra, D. (2008). Modeling Documents As Mixtures of Persons for Expert Finding. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR'08, pages 309–320, Berlin, Heidelberg. Springer-Verlag, ISBN: 3-540-78645-7 978-3-540-78645-0, http://dl.acm.org/citation.cfm?id=1793274.1793313.

[Simonton, 2006] Simonton, D. K. (2006). Historiometric Methods. In Ericsson, K. A., Charness, N., Feltovich, P. J., and Hoffman, R. R., editors, *The Cambridge handbook of expertise and expert performance*, pages 319–335. Cambridge University Press, New York, NY, US, ISBN: 978-0-521-60081-1 978-0-521-84097-2.

[Simovici and Djeraba, 2008] Simovici, D. A. and Djeraba, C. (2008). *Mathematical tools for data mining: set theory, partial orders, combinatorics*. Advanced information and knowledge processing. Springer, London, ISBN: 978-1-84800-200-5 978-1-84800-201-2.

[Sonnentag et al., 2006] Sonnentag, S., Niessen, C., and Volmer, J. (2006). Expertise in Software Design. In *The Cambridge handbook of expertise and expert performance*. Cambridge University Press, New York, NY, US, ISBN: 0-521-60081-2 (Paperback); 0-521-84097-X (Hardcover); 978-0-521-60081-1 (Paperback); 978-0-521-84097-2 (Hardcover), http://kops.uni-konstanz.de/handle/123456789/10597.

[Spaeth and Desmarais, 2013] Spaeth, A. and Desmarais, M. C. (2013). Combining Collaborative Filtering and Text Similarity for Expert Profile Recommendations in Social Websites. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, *User Modeling, Adaptation, and Personalization*, number 7899 in Lecture Notes in Computer Science, pages 178–189. Springer Berlin Heidelberg, ISBN: 978-3-642-38843-9 978-3-642-38844-6, http://link.springer.com/chapter/10.1007/978-3-642-38844-6_15.

[Szirányi et al., 2000] Szirányi, T., Zerubia, J., Czúni, L., Geldreich, D., and Kato, Z. (2000). Image Segmentation Using Markov Random Field Model in Fully Parallel Cellular Network Architectures. *Real-Time Imaging*, 6(3):195–211, ISSN: 1077-2014, DOI: 10.1006/rtim.1998.0159, http://www.sciencedirect.com/science/article/pii/S1077201498901590.

[Ullah and Giles, 2011] Ullah, A. and Giles, D. E. A. (2011). *Handbook of empirical economics and finance*. Chapman & Hall/CRC, Boca Raton, FL, ISBN: 978-1-4200-7036-1, http://www.crcnetbase.com/isbn/9781420070354.

[Vergne, 2016a] Vergne, M. (2016a). Gold Standard for Expert Ranking: A Survey on the XWiki Dataset. Technical Report arXiv:1603.03809 [cs.SE], http://arxiv.org/abs/1603.03809.

[Vergne, 2016b] Vergne, M. (2016b). Mitigation Procedures to Rank Experts through Information Retrieval Measures. Technical Report arXiv:1603.04953 [cs.IR], http://arxiv.org/abs/1603.04953.

[Vergne et al., 2013] Vergne, M., Morales-Ramirez, I., Morandini, M., Susi, A., and Perini, A. (2013). Analysing User Feedback and Finding Experts: Can Goal-Orientation Help? In *6th International i* Workshop*, volume 978, pages 49–54, Valencia, Spain. CEUR Workshop Proceedings, http://ceur-ws.org/Vol-978/paper_9.pdf.

[Vergne and Susi, 2014] Vergne, M. and Susi, A. (2014). Expert Finding Using Markov Networks in Open Source Communities. In Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., and Horkoff, J., editors, *Advanced Information Systems Engineering*, number 8484 in Lecture Notes in Computer Science, pages 196–210. Springer International Publishing, ISBN: 978-3-319-07880-9 978-3-319-07881-6, http://link.springer.com/chapter/10.1007/978-3-319-07881-6_14. DOI: 10.1007/978-3-319-07881-6_14.

[Vergne and Susi, 2015] Vergne, M. and Susi, A. (2015). Breaking the Recursivity: Towards a Model to Analyse Expert Finders. In Johannesson, P., Lee, M. L., Liddle, S. W., Opdahl, A. L., and López, Ó. P., editors, *Conceptual Modeling*, volume 9381, pages 539–547. Springer International Publishing, Cham, ISBN: 978-3-319-25263-6 978-3-319-25264-3, http://link.springer.com/10.1007/978-3-319-25264-3_40.

[Vivacqua, 1999] Vivacqua, A. (1999). Agents for expertise location. In *Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 9–13. http://www.aaai.org/Papers/Symposia/Spring/1999/SS-99-03/SS99-03-003.pdf.

[Wohlin, 2014] Wohlin, C. (2014). Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 38:1–38:10, New York, NY, USA.

ACM, ISBN: 978-1-4503-2476-2, DOI: 10.1145/2601248.2601268, http://doi.acm.org/10.1145/2601248.2601268.

[Wohlin et al., 2012] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-29043-5 978-3-642-29044-2, http://link.springer.com/10.1007/978-3-642-29044-2.

[Yimam-Seid and Kobsa, 2003] Yimam-Seid, D. and Kobsa, A. (2003). Expert-Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1):1–24, ISSN: 1091-9392, DOI: 10.1207/S15327744JOCE1301_1, http://dx.doi.org/10.1207/S15327744JOCE1301_1.

[Zave, 1997] Zave, P. (1997). Classification of research efforts in requirements engineering. *ACM Computing Surveys*, 29(4):315–321, ISSN: 0360-0300, DOI: 10.1145/267580.267581, http://doi.acm.org/10.1145/267580.267581.

[Zhang et al., 2007] Zhang, J., Ackerman, M. S., and Adamic, L. (2007). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 221–230, New York, NY, USA. ACM, ISBN: 978-1-59593-654-7, DOI: 10.1145/1242572.1242603, http://doi.acm.org/10.1145/1242572.1242603.

Appendices

Appendix A

Formalisation Details

This appendix centralises the proofs and various aspects investigated for the formalisation described in Chapter 5.

A.1 Relations Between DD, ODD, and PDD

ODD as a lower bound of DD.

$$\begin{aligned}
A + I + D &\geq A + D && \text{(because } I \geq 0\text{)} \\
\frac{A + I + D}{D} &\geq \frac{A + D}{D} && \text{(if } D > 0\text{)} \\
\frac{D}{A + I + D} &\leq \frac{D}{A + D} \\
ODD &\leq DD
\end{aligned}$$

By definition, $D \geq 0$. In the case where $D = 0$, we can see that $ODD = DD = 0$, thus the result $ODD \leq DD$ holds also for this case. □

PDD as an upper bound of DD.

$$\begin{aligned}
I + D &\geq D && \text{(because } I \geq 0\text{)} \\
\frac{I + D}{A} &\geq \frac{D}{A} && \text{(if } A > 0\text{)} \\
\frac{A}{I + D} &\leq \frac{A}{D} && \text{(if } D > 0\text{)}
\end{aligned}$$

$$\begin{aligned}
\frac{A}{I+D}+1 &\leq \frac{A}{D}+1 \\
\frac{A}{I+D}+\frac{I+D}{I+D} &\leq \frac{A}{D}+\frac{D}{D} \\
\frac{A+I+D}{I+D} &\leq \frac{A+D}{D} \\
\frac{I+D}{A+I+D} &\geq \frac{D}{A+D} \\
PDD &\geq DD
\end{aligned}$$

By definition, $A \geq 0$ and $D \geq 0$. In the case where $A = 0$, we can see that $PDD = DD = 1$, thus the result $PDD \geq DD$ holds also for this case. In the case where $D = 0$, we can see that $PDD = \frac{I}{A+I} \geq 0$ and $DD = 0$, thus the result $PDD \geq DD$ holds also for this case. □

$PDD - ODD$ as an evaluation of the ratio of Indifference *(I).*

$$\begin{aligned}
PDD - ODD &= \frac{I+D}{A+I+D} - \frac{D}{A+I+D} \\
&= \frac{(I+D)-(D)}{A+I+D} \\
&= \frac{I}{A+I+D}
\end{aligned}$$
□

A.2 *Agreement*-based Distances

One could be interested in exploiting distances based on the number of *Agreements* directly, let say AD instead of DD, OAD instead of ODD, and PAD instead of PDD (replacing "Disagreement" by "Agreement"). However, to make it a proper distance (and not a similarity measure), one should take its complement, leading to these measures:

$$AD = 1 - \frac{A}{A+D} = \frac{(A+D)-(A)}{A+D} = \frac{D}{A+D} = DD \tag{A.1}$$

$$OAD = 1 - \frac{A+I}{A+I+D} = ... = \frac{D}{A+I+D} = ODD \tag{A.2}$$

$$PAD = 1 - \frac{A}{A+I+D} = ... = \frac{I+D}{A+I+D} = PDD \tag{A.3}$$

Consequently, looking at the *Agreement*s does not provide more information and we can restrict ourselves to ODD and PDD, because a disagreement relates more naturally to a notion of distance, while an agreement refers more to a notion of similarity.

A.3 Comparison to Usual IR Measures

Additionally, the formulae above provides trivial similarity measures based on *Agreement*s:

$$AS = \frac{A}{A+D} \tag{A.4}$$

$$OAS = \frac{A+I}{A+I+D} \tag{A.5}$$

$$PAS = \frac{A}{A+I+D} \tag{A.6}$$

This is particularly interesting for comparing them to usual IR measures, like the ones analysed in [Vergne, 2016a], in which the measures based on precision and recall show the biggest interest. In this report, precision and recall are particularly interesting if we represent a ranking as a set of ordered pairs, which is precisely what we do here by considering orderings, so sets of order atoms. For bridging our measures to usual ones, let's focus on recall:

$$Recall = \frac{|\{\text{relevant items}\} \cap \{\text{retrieved items}\}|}{|\{\text{relevant items}\}|}$$

For comparing two rankings v_1 and v_2, the relevant items are the ordered pairs of one ranking, let say v_1, and the retrieved items are the ordered pairs of the other, v_2. No *Unordered* pair is considered in the initial definitions (i.e. rankings are complete and totally ordered), so no *Indifference* occurs, only *Agreement*s and *Disagreement*s. This situations makes a direct parallel between the recall formula above and AS, because A corresponds to the common pairs between v_1 and v_2, so the intersection between the relevant and retrieved items in recall, and $A + D$ corresponds to the total number of pairs which, for two rankings on the same items, is equivalent to the number of relevant items in recall.

As mentioned in our previous work, by having incomplete and partially ordered rankings we might have pairs which are absent, leading to *Indifferences*, so adding I at the denominator, which shows then the equivalence between recall and PAS. The main issue of this recall is its inability to differentiate absent and reversed pairs because it computes only the intersection, which is the problem we identified with PDD and which remains in its equivalent similarity PAS. In the report, we mentioned that we can reverse this logics by enriching the computing with a mitigation procedure, while here we just need to use OAS which maximises this similarity by considering any *Indifference* as an *Agreement*.

In brief, we see that the similarity measures OAS and PAS are equivalent to the best measures we found from our analysis of existing IR measures, but specifically designed for comparing incomplete and partially ordered rankings. This shows that ODD and PDD, their equivalent distances, build on the same interpretation than recall, thus providing a good theoretical grounding.

A.4 Comparison to Kendall's τ Coefficient

Kendall's τ coefficient [Kendall, 1938] measures the correlation of two rankings over the same set of elements by computing a value in $[-1; 1]$. A value of 1 means that the rankings provide the items in the very same order, a value of -1 occurs when they are completely reversed, and a value close to 0 happens in balanced cases, usually with random rankings. In his original work, Kendall makes a parallel with Spearman's rank correlation coefficient ρ, showing that it provides similar value. At the opposite of ρ, τ is of particular interest for us because it builds on equivalent concepts, as we show below.

The definition provided on Wikipedia[1] is rather simple, so we reproduce it here. Let (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) be a set of observations of the joint random variables X and Y respectively, such that all the values of (x_i) and (y_i) are unique. Any pair of observations (x_i, y_i) and (x_j, y_j), where $i \neq j$, are said to be *concordant* if the ranks for both elements agree: that is, if both $x_i > x_j$ and $y_i > y_j$ or if both $x_i < x_j$ and $y_i < y_j$. They are said to be *discordant*, if $x_i > x_j$ and $y_i < y_j$ or if $x_i < x_j$ and $y_i > y_j$. If $x_i = x_j$ or $y_i = y_j$, the pair is neither concordant nor discordant. The Kendall τ coefficient is defined as:

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{n(n-1)/2}$$

If we reformulate it with our own concepts, we may say that for two rankings X and Y, we can look at all the pairs in *Agreement* (concordant) and in *Disagreement* (discordant) to establish a correlation coefficient, normalized over the total number of pairs (*Agreement* + *Disagreement* + *Indifference*). Or more formally:

$$\begin{aligned}
\tau &= \frac{A - D}{A + I + D} \\
&= \frac{A}{A + I + D} - \frac{D}{A + I + D} \\
&= \left(1 - \frac{I + D}{A + I + D}\right) - \frac{D}{A + I + D} \\
\tau &= 1 - PDD - ODD
\end{aligned}$$

[1] Kendall's τ on Wikipedia: https://en.wikipedia.org/w/index.php?title=Kendall_rank_correlation_coefficient&oldid=707755936

Once again, the overlapping with more conventional measures supports our own formalisation and the measures we designed based on it. It is however worth to notice that, because τ does not consider I in the numerator, it cannot differentiate between the case of two highly disagreeing rankings (balanced A and D) and the case of two rankings having many *Indifference*s (A and D close to zero) which both gives a τ close to zero. Consequently, we think that having both PDD and ODD provides a better information than τ alone if we deal with partially ordered or incomplete rankings.

Appendix B

Evaluation Measures

This appendix centralises some details of the evaluation measures designed in Chapter 7.

B.1 Variance-based Measures

We can consider the notion of *variance* over a set $X = \{x\}$, which is defined as $var(X) = E[x - E[X]]^2$ ([Hastie et al., 2009] p. 223), where $E[X]$ is the expected (or average) value over a set X. In other words, the variance compares each item of the set to an average item, and returns a squared average of these comparisons (if we remove the square, it is a *standard deviation*). We could imagine a proper variance formula for the re-run case by computing an average ranking, like the centroid $c(V_t)$ (Section 5.1.2), and making it a proper ranking if required (Section 5.1.3). Then, the difference between rankings can be computed by using the same distance $d(v_1, v_2)$ than the basic measure (e.g. DD, ODD, or PDD), leading to Equation B.1:

$$var_{\text{re-run}}(V_t) = \left(\frac{\sum\limits_{v \in V_t} d(v, c(V_t))}{|V_t|} \right)^2 \tag{B.1}$$

For the extra-run case, we compare elements from two different multisets (V_t and V_{t+1}), so we could imagine to compare the *rankings* of t to the *centroid ranking* of $t+1$ (Equation B.2), or the reverse (Equation B.3):

$$var_{\text{extra-run } t}(V_t, V_{t+1}) = \left(\frac{\sum\limits_{v \in V_t} d(v, c(V_{t+1}))}{|V_t|} \right)^2 \tag{B.2}$$

$$var_{\text{extra-run } t+1}(V_t, V_{t+1}) = \left(\frac{\sum\limits_{v \in V_{t+1}} d(v, c(V_t))}{|V_{t+1}|} \right)^2 \tag{B.3}$$

A problem with both these extra-run measures is that we loose the link between the specific ranking v and the average ranking, so such a variance computation would be arguably meaningful. However, independently of the measure used, our objective is to know when we obtain a representative ranking for our approach, thus when any ranking we could produce would remain close to each other, leading all these measures to converge to zero.

From a practical perspective, we also have to consider that our distances (DD, ODD, and PDD) express ratios of ordered pairs, which give them an concrete interpretation. With the square of the variance formula however, we loose this ability, which makes it less interesting than the standard deviation form. Thus, if we consider Equation B.1 to be the most justified, then an interesting measure is the following, which removes its square to maintain its interpretation as a ratio of ordered pairs:

$$var_{\text{re-run}}(V_t) = \frac{\sum\limits_{v \in V_t} d(v, c(V_t))}{|V_t|} \tag{B.4}$$

B.2 Bias-based Measures

Another relevant similar measure is the *bias* towards an ideal value $\hat{x}$, which is defined as $bias(X, \hat{x}) = E[X] - \hat{x}$ ([Hastie et al., 2009] p. 223), thus making a single comparison between an expected or average value and a reference or gold standard. In the case where a gold standard ranking $\hat{v}$ is available, one can take this gold standard as $\hat{x}$ and compute the bias for a given time-out t, as shown by Equation B.5:

$$bias_{\text{gold}}(V_t, \hat{v}) = d(c(V_t), \hat{v}) \tag{B.5}$$

By generalizing to a multiset of gold standard rankings $\hat{V}$, as we faced in some experiments, we could take its centroid $c(\hat{V})$ as the ideal value $\hat{x}$ (notice that $c([\hat{v}]) = \hat{v}$, so it is a proper generalization) to compute the bias for the time-out t, as shown in Equation B.6:

$$bias_{\text{gold}}(V_t, \hat{V}) = d(c(V_t), c(\hat{V})) \tag{B.6}$$

Additionally, we could compute the variance of $\hat{V}$ by using Equation B.1, to know how reliable it is: lower is the variance, higher is the agreement among the different rankings, and thus the representativeness of the centroid as an ideal value.

Assuming the absence of such a gold standard, which is the purpose of this section, it is clear that we cannot adapt the bias definition to the re-run case, because it does not provide an ideal value. However, we can do it for the extra-run case by considering $c(V_{t+1})$ as our ideal value, as shown in Equation B.7:

$$bias_{\text{extra-run}}(V_t, V_{t+1}) = d(c(V_t), c(V_{t+1})) \tag{B.7}$$

Indeed, the rankings at $t + 1$ have more computation time than the rankings at t, thus they are probably closer to the final value returned by the approach, and so the centroid for $t + 1$ is a good candidate to represent this “ideal” (final) value. A high bias would mean that it is preferable to have a time-out of $t + 1$, to be closer to the final value, while a low bias would mean that it is better to have a time-out of t, to preserve computation time. Considering that it is arguable to consider $c(V_{t+1})$ as closer to the final value (there can have cyclic behaviours), we can also smooth this interpretation and simply consider a high value as a hint that assigning more computation time would have a significant effect on the final result, while a low value would mean that computing t or $t+1$ does not change significantly the result, so we can take the most interesting one (i.e. the cheapest). In both cases, what we are interested about is when to stop computing, so from which t this measure become close to zero and remains there.

B.3 Extra-run Limitations

For the extra-run measure, which compares rankings from consecutive time-outs, a particular limitation needs to be highlighted. We consider a discrete time (time-outs t and $t+1$), and if the approach can generate a ranking for a time-out with a finer granularity (i.e. between t and $t+1$), then we are potentially losing information. From a simple perspective, if our approach can be run for any time-out $t \in \{1,2,3,...\}$ but we sample it on the time-outs $t \in \{100,200,...\}$ for our analysis, then we do not know the behaviour of our approach for the time-outs 1-99, 101-199, etc. From a more advanced perspective, we should refer to the Nyquist-Shannon sampling theorem, which states which information we loose and why.

Indeed, any continuous signal can be represented as a sum of sinusoids, usually through its Fourier transform which translates a function of time into a function of frequencies. Each sinusoid has an amplitude, a frequency, and a phase or delay, and the Fourier transform shows in particular which frequencies are involved to build the observed signal. By sampling such a signal with a discrete time, we obtain samples at a given frequency f, and the Nyquist-Shannon sampling theorem states that only the sinusoids having a frequency lower than $\frac{f}{2}$ can be retrieved. In other words, the function of frequencies observed through these samples is bounded by the sampling frequency, so the original signal retrieved from them can be altered if it requires frequencies above this boundary. So not only we are unable to see isolated phenomenon between samples, but we can also miss general behaviours like cycles with a frequency close to the sampling frequency. For example, with a pure sinusoidal signal having a fixed frequency, if we sample it at the same frequency, then we would observe a straight line without any variation (a constant function), while the amplitude and delay determine which value is observed to be "constant".

In our case, the signal is not continuous but discrete, because we deal with iterations, but if the sampling period is significantly larger than the period of a single iteration, then we face a similar situation. It can be avoided by having the finest sampling period than possible, but it increases the time required to generate our evaluation data because we generate each sample independently. Another matter is that we do not use a periodic sampling (e.g. $t \in \{10, 20, 30, ...\}$), but a logarithmic one ($t \in \{1, 3, 10, 30, ...\}$), which means that we have yet another kind of observation, with a finer granularity at the beginning but a larger one at the end. However, the effect is generally the same: we can miss some information, so a trade-off needs to be made and this limitation must be kept in mind.

Appendix C

Design of Synthetic Data

We designed synthetic data for the purpose of having a fully controlled evaluation of our approach in Section 8.1, able to stress specific properties like how stakeholders are related to other nodes and how it affects their relevance as experts. Consequently, with the aim of building a full graph of stakeholders (S), roles (R), topics (T), and terms (C) related with weighted links (L), we start by building a set of n topics $T = \{t_1, ..., t_n\}$. These are the topics we plan to query, and thus each of them will have their own gold standard based on how they relate to the other nodes. Our objective then is to build specific "profiles" of relations, such that the nodes are related in ways which make their relevance obvious when we query a given topic.

Starting from the terms, we consider a set of m terms $C = \{c_1, ..., c_m\}$ which are related to the topics through a term profile $prof(t)$ for each topic $t \in T$. This notion of term profile aims at representing a natural "jargon" distribution, for instance a profile about the topic *programming* should be more related to the term *code* than *ingredient*. This is for instance what is exploited by [Castro-Herrera and Cleland-Huang, 2009], from which we inspire, when they compute the similarity between vectors of terms, each vector representing a "profile" over the whole set of terms. Consequently, each term profile corresponds to a set of term nodes associated to specific weights to represent a given topic, so $prof(t) = \{(c, w) | c \in C, \langle t, c, w\rangle \in L\}$. Then, applying a term profile to a node x means to relate x to all these terms with their corresponding weights.

To build our term profiles, each topic t is assigned to an ordered set of terms $OC^t = \{oc_1^t, ..., oc_m^t\}$ which contains each term of C in a random order. Then, we build the term profile $prof(t) = \{(oc_i^t, w(oc_i^t))\}$ by using a specific weighting function. In our case, we use the Zipf's law [Ullah and Giles, 2011] (p. 139), which starts from a high weight for a central term and decreases quickly, with a long tail of low-weight terms, as shown in Figure C.1. We use this law because it is particularly representative of natural language behaviours, in particular for describing the frequency of words in a corpus made of natural sentences [Manning et al., 2008]. Consequently, once the terms are randomised, we make them a term profile by computing $w(oc_i^t) = round\left(\frac{max}{i}\right)$, with max being the weight of the central term.

For the roles, we consider a role r_t for each topic t, like we could have the role *programmer* for the topic *programming*, *cook* for *cooking*, and so on. Consequently, we have the roles $R = \{r_{t_1}, ..., r_{t_n}\}$ with the weighted relations $\langle t, r_t, 1\rangle$, meaning that knowing about the topic t usually correlates to having the role r_t and vice-versa. We use a simple weight of 1 because each role is related to a single topic and vice versa, so no particular priority need to be implemented. To remain consistent, we relate the terms C to each role r_t by using the term profile of the topic t, such that $(c, w) \in prof(t) \Rightarrow \langle r_t, c, w\rangle \in L$. So we have not only a direct relation between the role and the topic, but we also have an indirect relation through the same term profile. The relations between t, r_t and the terms C is summarized in Figure C.2.

Regarding the stakeholders, summarized in Figure C.3, we consider several profiles which relate differently to each topic depending on their level of expertise. Half of the profiles are topic-specific, which means that they are intended to have some expertise in a single topic only, while the other half is

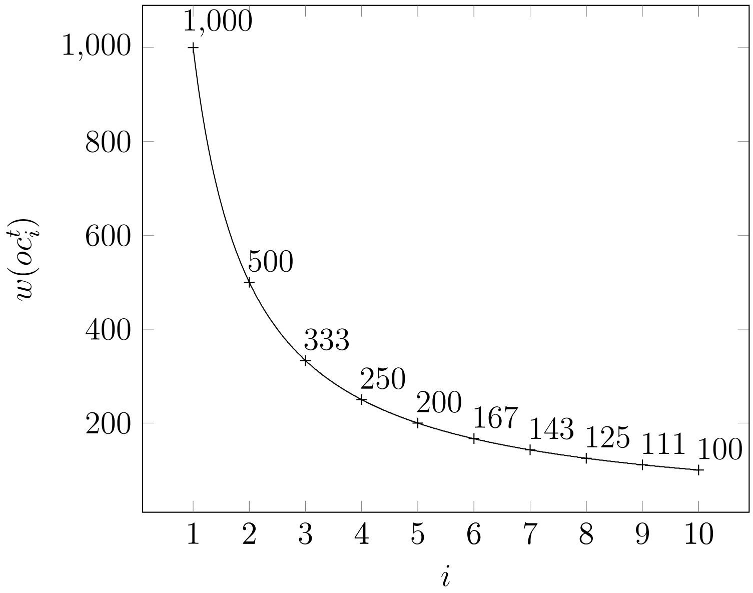


Figure C.1: Distribution of weights for the ordered terms $OC^t = \{oc_i^t\}$ of the term profile $prof(t) = \{(oc_i^t, w(oc_i^t))\}$ for the topic t. The central term (rank 1) has a weight of 1000and the following terms follow a Zipf's law to align with common observations on natural languages.

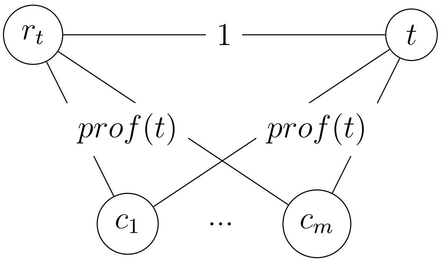


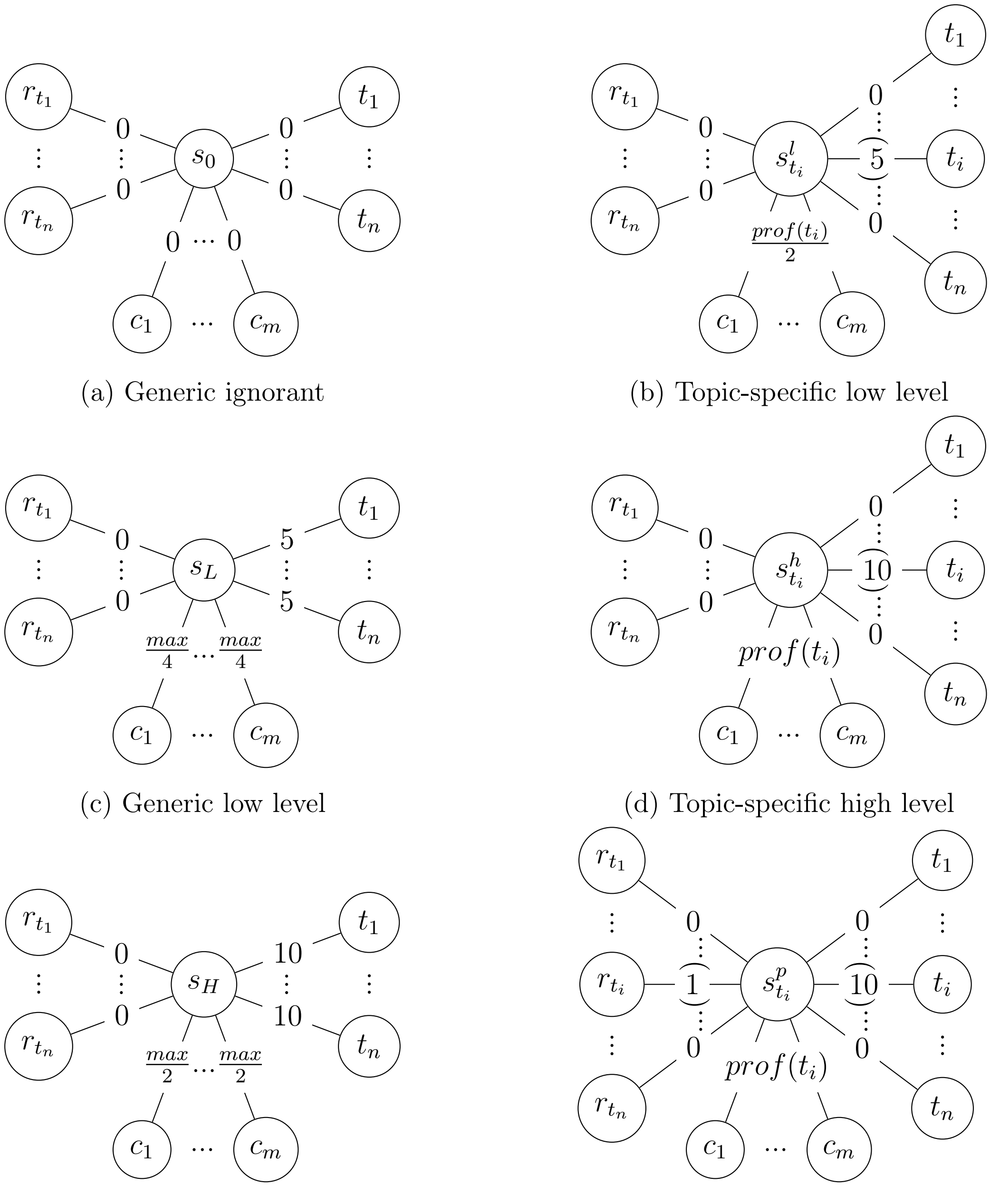
Figure C.2: Graphical representation of how topics, roles and terms are related, given a chosen topic t. Each role is related to its corresponding topic with a weight of 1 and to any other topic (not shown here) with a weight of zero. The relations with terms build on the term profile $prof(t)$, with the same weights for both t and r_t.

generic, meaning that they spread their expertise over the whole set of topics with equal levels.

For the topic-specific profiles, we design a stakeholder with a low level of expertise, another one with a high level of expertise, and a third one with also a high level of expertise but an additional role, to add some Social Recognition. Consequently, we first have low level stakeholders for each topic, represented by $S^l = \{s^l_{t_1}, ..., s^l_{t_n}\}$ with the relations $\langle s^l_t, t, 5\rangle$. Similarly, we have high level stakeholders for each topic, represented by $S^h = \{s^h_{t_1}, ..., s^h_{t_n}\}$ with the relations $\langle s^h_t, t, 10\rangle$. We also have professional stakeholders for each topic, represented by $S^p = \{s^p_{t_1}, ..., s^p_{t_n}\}$, with the same relations $\langle s^p_t, t, 10\rangle$ but with an additional $\langle s^p_t, r_t, 1\rangle$, so they have the corresponding role. Each of these topic-specific stakeholders (s^l_t, s^h_t, and s^p_t) is related to the terms C based on the corresponding term profile $prof(t)$, but while high level stakeholders (S^h and S^p) use the same weights, the low level stakeholders (S^l) use half weights to show less expertise. These topic-specific stakeholders are illustrated in the figures C.3b, C.3d, and C.3f.

Finally, we consider three generic stakeholders: s_0 who is completely ignorant, s_L who has a low level on every topics, and s_H who is an expert everywhere. We could illustrate these three cases by speaking about a baby for s_0, a really curious person for s_L, and some kind of "God of knowledge" for s_H (we take no religious stance here, we only set up an extreme case for our synthetic data). More precisely, we add relations such that $\forall t \in T, \{\langle s_0, t, 0\rangle, \langle s_L, t, 5\rangle, \langle s_H, t, 10\rangle\} \subset L$, and stakeholder-term relations such that $\forall c \in C, \{\langle s_0, c, 0\rangle, \langle s_L, c, \frac{max}{4}\rangle, \langle s_H, c, \frac{max}{2}\rangle\} \subset L$, but we set up no stakeholder-role relations (zero-weight). We use weights of $\frac{max}{2}$ or $\frac{max}{4}$ for terms to reflect some "effort distribution", so the generic stakeholders have spent less time in each specific topic (compared to topic-specific stakeholders) but more time in others. For 10 terms with $max = 1000$, which are the parameters used for our evaluation, the weights for the term profile $prof(t)$

sum up to 2929, while the weights of s_L sum up at a lower level of 2500, and the weights of s_H sum up at a higher level at 5000. These generic stakeholders are illustrated in the figures C.3a, C.3c, and C.3e.



(a) Generic ignorant

(b) Topic-specific low level

(c) Generic low level

(d) Topic-specific high level

(e) Generic high level

(f) Topic-specific professional

Figure C.3: The 6 types of stakeholders and how they relate to the rest of the network (topics, roles, and terms). In particular, we can see how the topic-specific stakeholders (right) relate to their topic t_i and to terms with specific weights $prof(i)$, with the professional (f) adding the role r_i compared to the high level (d), while generic stakeholders (left) relate with equal weights, showing no specializations.

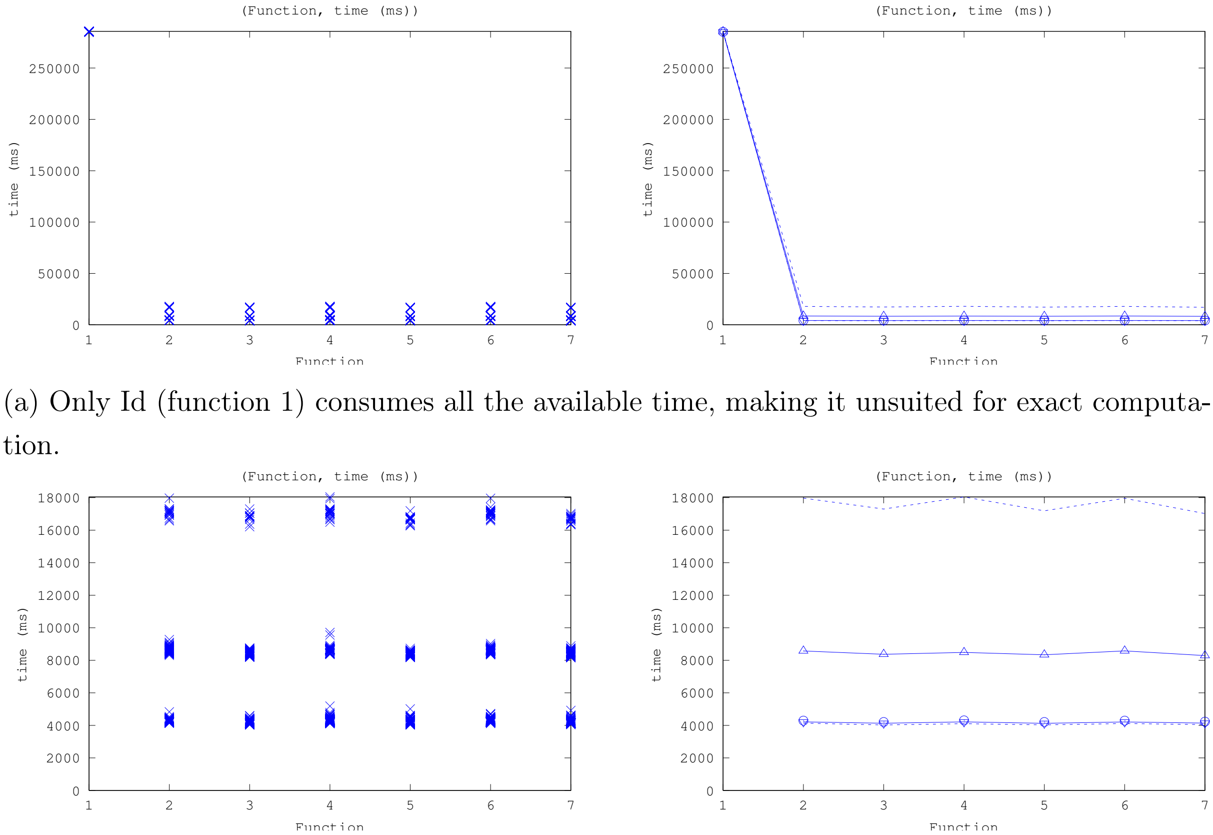
Appendix D

Detailed Analysis for Synthetic Data

The graph for synthetic data being small enough, we can compute exact values for the MN approach, which is what we start from. Through this analysis, we should be able to establish whether the MN approach provides meaningful results. After the analysis of the exact computation, we compare it to the approximative computation, which is supposed to converge towards the results of the exact one. Because the approximative computation is supposed to be used in big graphs, when the exact computation is not reasonable, this comparison should provide some insights from a performance perspective. Then, a third analysis is made for the GA approach, which should support at the same time the correctness and performance of the approach.

D.1 Markov Network (exact)

By looking at the time consumed by the different functions when providing them the highest time-out, Figure D.1a shows that only Id consumes all of it, making it unsuited for exact computation. If we ignore this function, Figure D.1b shows that the other functions (Id+5, Norm, S-Norm, Norm+5, S-Norm+5, and WoE) are equivalent regarding their time of computation, with local differences due to other parameters, like the queries. Although the graphs do not show the results for the altered dataset having zero weights



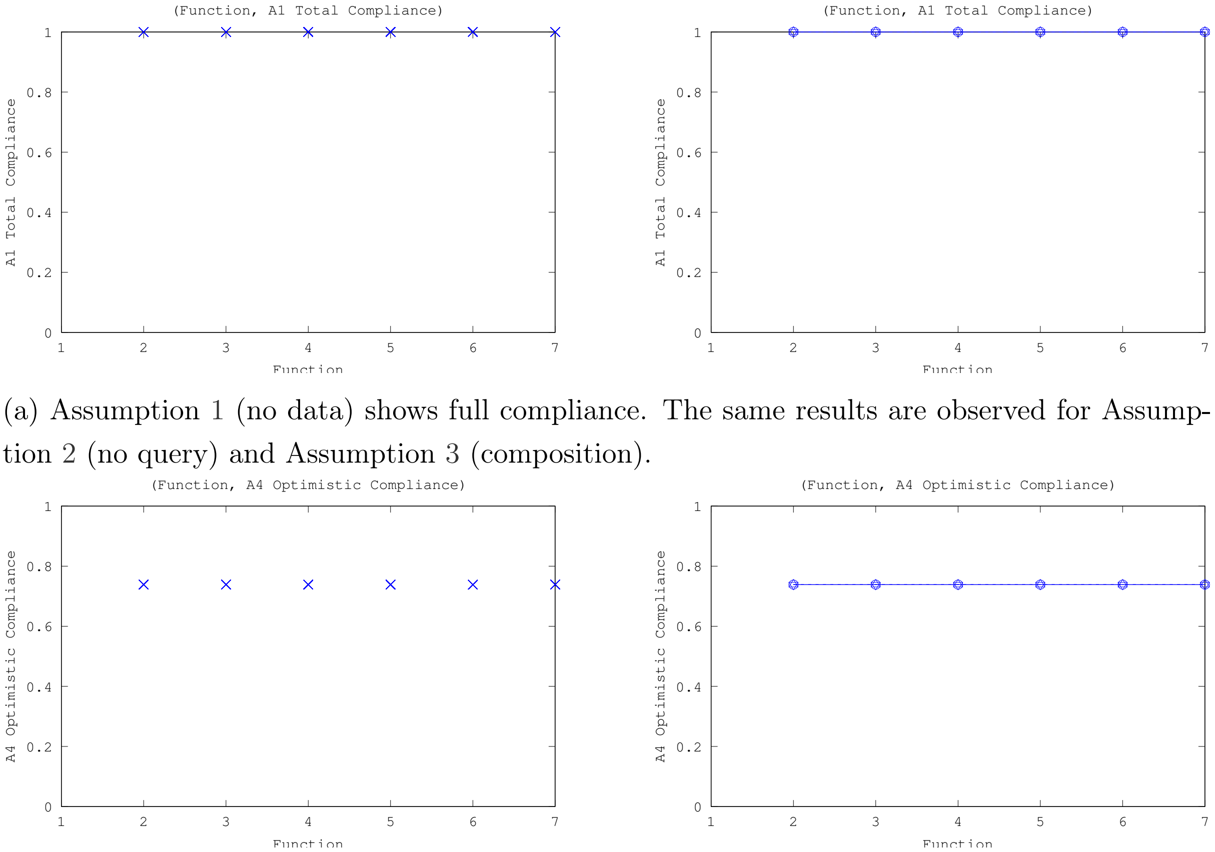
(a) Only Id (function 1) consumes all the available time, making it unsuited for exact computation.

(b) For the other functions, no clear difference is shown.

Figure D.1: Time required for exact computation at highest time-out (300s) of the MN technique for synthetic data.

for stakeholders (Assumption 1), we observed similar results on both.

When looking at how the different functions (excepted Id) comply with the gold standards, we can see that perfect compliance is achieved for all the formal assumptions (no data, no query, and composition), as shown in the figures D.2a. However, Assumption 4 (expected) is not as successful: Figure D.2b shows that, if all the functions provide equivalently good results, only 73.9% of pairs comply with the gold standard, which we consider to be low for noise-free data. And still, a significant part is due to the limited coverage of our gold standards, which provide only few order atoms: by focusing on ordered pairs only (i.e. using $OrderComp(gs, v)$ instead of $OptimComp(gs, v)$), we find that only 29.8% of them are satisfied, which



(a) Assumption 1 (no data) shows full compliance. The same results are observed for Assumption 2 (no query) and Assumption 3 (composition).

(b) Assumption 4 (expected) shows only partial compliance (73.9%). By focusing on ordered pairs, poor compliance is achieved (29.8%).

Figure D.2: Evolution of the assumption compliance of the MN technique for synthetic data with exact computation.

shows a clear lack of compliance. Actually, s_0 is ranked below all the other stakeholders and nothing more, so only pairs comparing s_0 are compliant while all the others are missing, justifying that 70.2% of the ordered pairs are not compliant. The fact that our gold standards only provide few pairs push then $OptimComp(gs, v)$ to increase the compliance, reducing the total amount of non-compliant pairs to 26.1% only. Additionally, this lack of ordering is also why it achieves full compliance for Assumption 3: whether we query one or two topics, the ranking is the same.

In brief, although being correct for extreme cases and generally consistent (assumptions 1 to 3 are satisfied), the exact computation provides a rather uninformative ranking, independently of the function used.

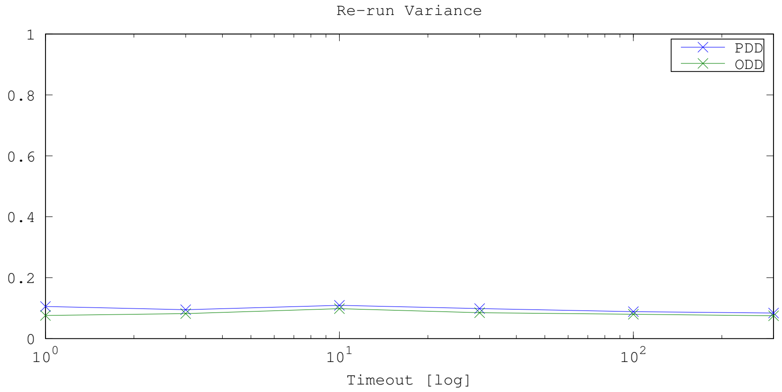
D.2 Markov Network (approximative)

What could be interesting is to see how the approximative computation converges toward this result: Does it actually converge towards the same result than the exact computation? Does it produce some more informative rankings before to loose them with more computation? Does it pass through more *Agreement* than *Disagreement*? We will see in the following that, actually, the approximative computation does not necessarily converge towards the exact one. Moreover, if additional issues can arise due to the approximations, better results can also be achieved.

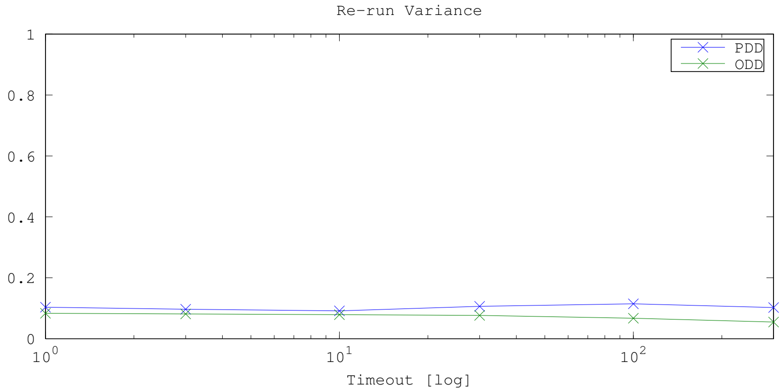
To do so, we first try to identify what are the relevant settings, especially the time-outs from which some functions may provide stable results. For this, we ignore the case of empty queries, because they are assumed to provide uninformative rankings, which means that they artificially increase the amount of *Indifference* ($PDD - ODD$) on the graphs.

If we check the re-run variance of our approach, with Figure D.3, we can observe that 8.3% of *Disagreement* is preserved with Id, Norm, and S-Norm, while almost all the rest is in *Agreement*. WoE is slightly better because it looses some *Disagreement* with further computation, although it is not transformed in *Agreement*. For the remaining functions Id+5, Norm+5, and S-Norm+5, although no *Disagreement* is present, a lot of *Indifference* can be observed, which is because we obtain similar results than the exact computation: excepted s_0, all the stakeholders have the same probability, making the functions mainly uninformative. However, as we will see later, s_0 is not always ranked last.

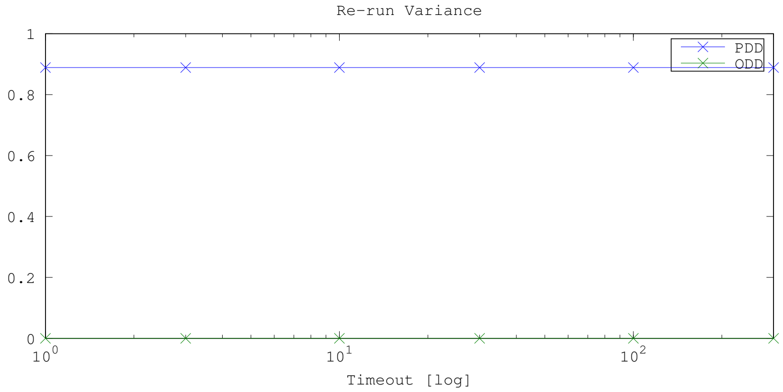
If we check the extra-run bias for all functions but WoE, as shown in



(a) Id, Norm, and S-Norm are informative with almost no *Indifference* ($PDD - ODD \approx 1.4\%$), but still have some *Disagreement* ($ODD \approx 8.3\%$).



(b) WoE is similar but shows a minor decreasing of *Disagreement* with more computation (from 8.3% to 5.5%).

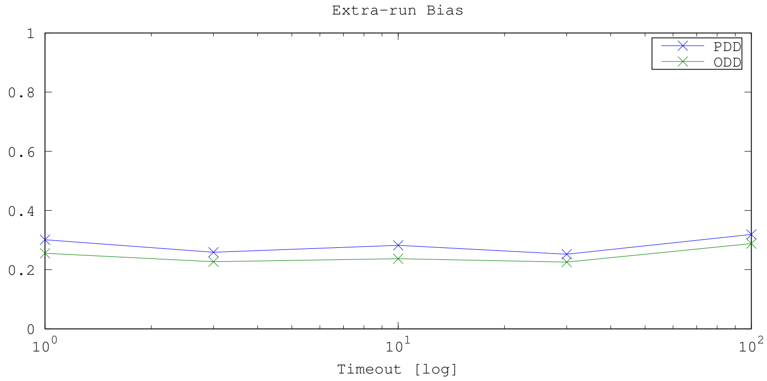


(c) Id+5, Norm+5, and S-Norm+5 have no *Disagreement* ($ODD = 0\%$) but are mainly uninformative ($PDD - ODD = 88.9\%$) and so have low *Agreement* too ($1 - PDD = 11.1\%$).

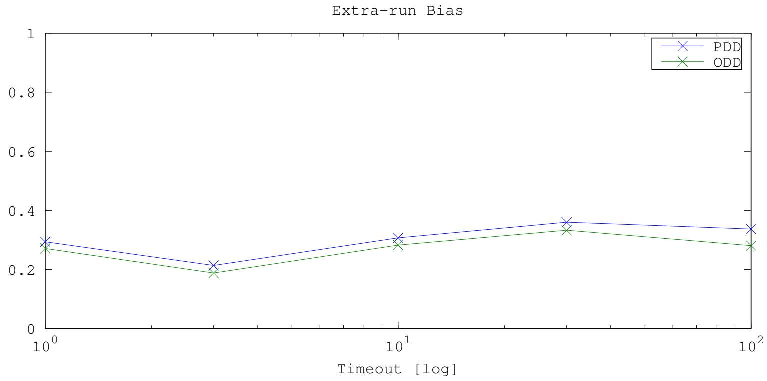
Figure D.3: Evolution of the re-run variance of the MN technique for synthetic data with approximative computation.

Figure D.4, Id remains unstable by having always around 24.7% of *Disagreement* between centroids of different time-outs. Norm and S-Norm gain some stability at the beginning but loose it on the long term, showing that they do not converge either. Id+5, Norm+5, and S-Norm+5 provide the same graphs than their re-run variance, which means that the few pairs in *Agreement* at a given time-out do not reverse at the next time-out. Indeed, the rankings might be different depending on the query, but they remain the same for a given query.

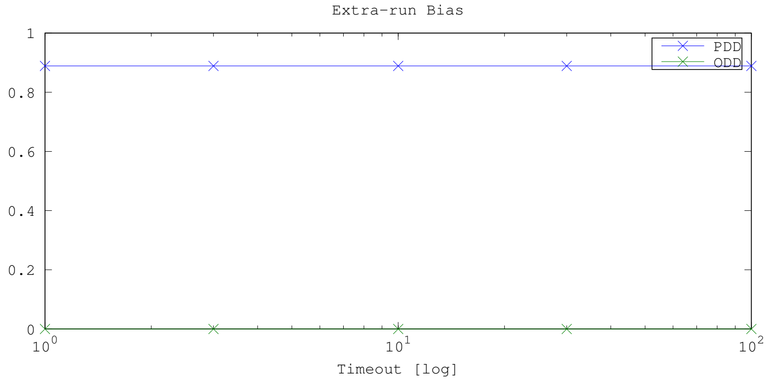
Only WoE shows a potential convergence: if we look at the overall behaviour of WoE in Figure D.5a, it appears that some improvement is achieved through ODD, but the key point here is that what is increasing is not the *Agreement* (i.e. a decrease of PDD) but the *Indifference* (i.e. PDD remains high). This is where the limitations of the variance and bias measures need to be considered: because we are dealing with centroids, we need to pay attention whether the *Indifference*–resulting from *Unordered* pairs in the centroids– is due to actual *Unordered* pairs of the generated rankings or to a balanced *Disagreement* between them. To know this, we need to check the distances between each ranking rather than relying on centroids, which is done by using Algorithm 4, described in Section 7.1. By computing all the ODD values, which are higher than 0 only when there is an explicit *Disagreement*, we can see as illustrated by Figure D.5b that, over the 750 values comparing rankings at 100s to rankings at 300s, (i) 423, so 56.4%, are higher than zero, (ii) with a minimum value of 29.4% of pairs in *Disagreement*. This means that the *Indifference* observed with the bias measure (i) is mainly due to a *Disagreement* (ii) on almost one third of the pairs. The fact that similar observations are made for each different setting (i.e. each query) explains why the bias-based ODD reaches a proper zero: the centroid of each setting is concerned, so computing ODD between them provides zero values only, resulting in an equivalent mean over all the settings. Additionally, by



(a) Id remains with a lot of *Disagreement* ($ODD \approx 24.7\%$) so the rankings tend to always change with more runs.



(b) Norm and S-Norm gain some stability at the beginning but soon loose it.



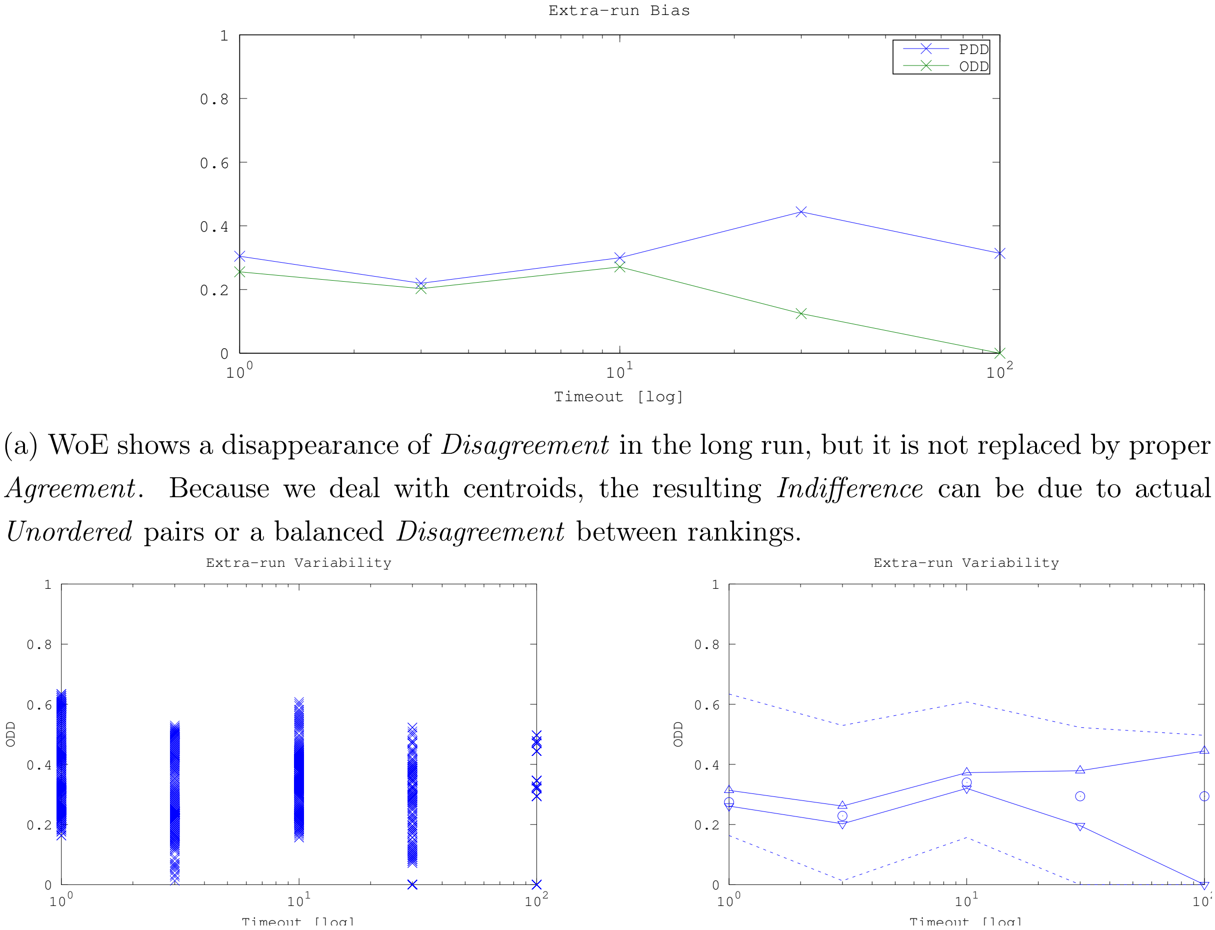
(c) The few informative pairs of Id+5, Norm+5, and S-Norm+5 tend to remain the same, because no *Disagreement* occur between different time-outs.

Figure D.4: Evolution of the extra-run bias of the MN technique for synthetic data with approximative computation without WoE.

computing the detailed PDD values, we observed only a small increase of the distance (e.g. for the 72 additional cases higher than zero, we reach only 0.7% of pairs not in *Agreement*), so the centroids lack in ordered pairs mainly due to a balanced *Disagreement* of the rankings produced. In other words, if we wanted to use WoE, we would need to generate several rankings and compute their centroid to obtain a "stable" ranking. If the *Indifference* of the centroid was due mainly to intrinsic *Unordered* pairs (i.e. ODD concentrate around zero on the detailed graph), we could have considered the rankings to be stable, because generating only one would have been equivalent to generate a centroid. Of course, this need to generate a centroid is the same for the other functions, because they show clear instability already from their bias-based measures.

Through the analysis of each function, we saw that Id+5, Norm+5, and S-Norm+5 provide similar results than the exact computation, for which we already know how poorly it complies to the different assumptions. However, we will see in the following that having an approximative computation introduces specific effects leading to significant differences. Similarly, although the other functions achieve no proper convergence, leading to our inability to identify stable settings to exploit, we will see by analysing assumption compliances that they might still have some interest.

Regarding Assumption 1 (no data), we can see on Figure D.6a that no function complies, meaning that all of them provide rankings close to be totally ordered if not so. This phenomenon occurs because the probabilities remain close to 0.5 (in [0.480; 0.523]) but not strictly equal, leading to ordered pairs instead of *Unordered* ones. This observation is true also for Id+5, Norm+5, and S-Norm+5, which have shown to provide similar rankings than the fully compliant, exact computation: because of the approximation, no *Unordered* pair is properly generated, leading to an almost complete deletion of compliance.



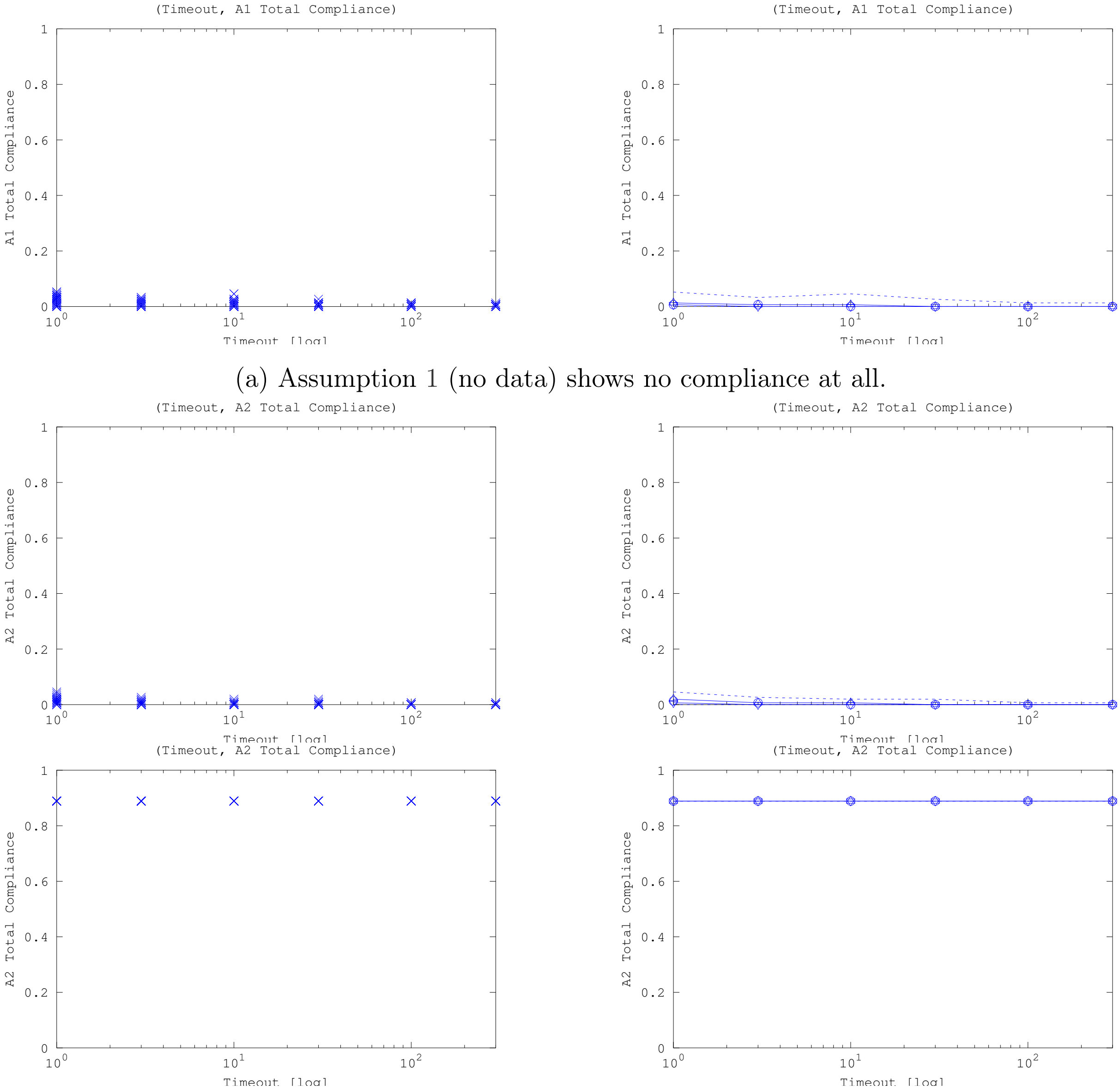
(a) WoE shows a disappearance of *Disagreement* in the long run, but it is not replaced by proper *Agreement*. Because we deal with centroids, the resulting *Indifference* can be due to actual *Unordered* pairs or a balanced *Disagreement* between rankings.

(b) The detailed ODD distances show that, for the highest timeout, more than half of the distances are high ($ODD \geq 29.4\%$), thus the previous *Indifference* is due to a major *Disagreement* between the generated rankings.

Figure D.5: Evolution of the extra-run variability of the MN technique for synthetic data with approximative computation and WoE function.

For Assumption 2 (no query), Figure D.6b shows that only the functions Id+5, Norm+5, and S-Norm+5 achieve an interesting level of compliance. However, this compliance is not perfect: indeed, we saw that s_0 is the only one having a different probability, and here s_0 is ranked *higher* than anyone else. This is because the probability computed for s_0 is always 0.5, independently of the query, while it is 0 for the other nodes in the case of an empty query. If we look at the other functions, which are not compliant at all, we can see that Id, Norm, and S-Norm are subject to the same phenomenon than before: all the values are in $[0.480; 0.515]$, so really close to 0.5, but never strictly equal, leading to enforced orders. Thus, this lack of compliance is again a matter of precision due to the approximative computation. For WoE, however, we observe a clear tendency towards being non-compliant: if we look at the probabilities of each ranking, to see when the orders are justified, we can see that the gaps increase with additional computation time. In average, the rankings start with a small max-min difference (0.028 at 1s, 0.021 at 3s) which then increases significantly (0.233 at 10s, 0.416 at 30s, 0.474 at 100s, and 0.492 at 300s), thus motivating the orders and –consequently– the lack of compliance for this assumption. This is probably due to the design of this function, which "zooms" on 0.5, so the small differences which may occur in other functions are exacerbated by WoE, making them grow progressively.

For Assumption 3 (composition), however, the compliance tendency is reversed. Indeed, Figure D.7a shows that Id+5, Norm+5, and S-Norm+5 are fully non-compliant, while all the other functions provide a strictly positive compliance value, although not exactly in the same way. In particular, WoE starts from an average of 36.9% of compliant pairs and monotonously increases until 58.8%. The remaining functions, while they show a similar increase with similar boundaries, are not monotonous: they show some loss of compliance at 30s or 100s. This intermediate loss is already a limitation, because we never know when to stop the algorithm to obtain a good level



(a) Assumption 1 (no data) shows no compliance at all.

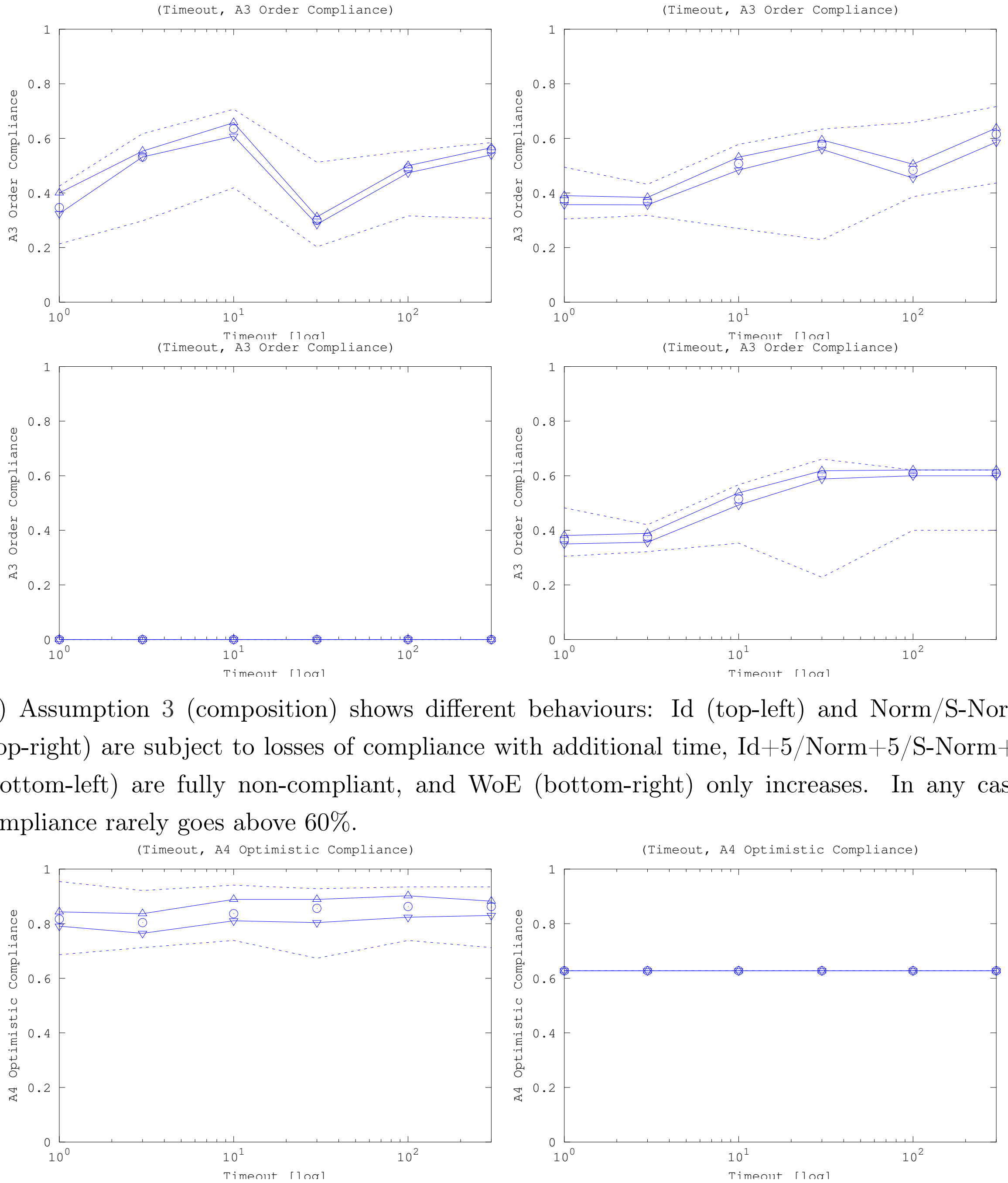
(b) Assumption 2 (no query) shows two tendencies: Id/Norm/S-Norm/WoE (top) are not compliant at all, while Id+5/Norm+5/S-Norm+5 (bottom) are highly compliant.

Figure D.6: Evolution of the assumption compliance (1 and 2) of the MN technique for synthetic data with approximative computation.

of compliance, but even with WoE which constantly increases we face the limitation of having around 60% of compliance only, which is really poor for noise-free data.

Finally, Assumption 4 (expected) offers the most unexpected results. Once again, we can see two tendencies depending on whether we use the prior-based functions or not, as shown in Figure D.7b. In the case of prior-based functions, although we have the same phenomenon of having only s_0 having a different probability, the result is not at all the one expected. Indeed, while for the exact computation the rankings produced always place s_0 lower than the others, the approximative computation provides also rankings where s_0 is higher. In particular, s_0 is ranked lower for queries of 2 topics, but it is ranked higher for queries of 1 topic, which is precisely the case covered by our gold standard. In other words, the prior-based functions achieve the minimal level of compliance possible with 62.7%, which is offered by the unconstrained pairs. At the opposite, the other functions not only achieve a better result, but they achieve generally a better result than the exact computation itself. Indeed, the exact computation was able to achieve 73.9% of compliance, while here even with the lowest time-out we achieve an average of 81.6% of compliance. Additionally, this average reaches 85.6% at the highest time-out, with a difference between Id which remains around the same level while the other functions are the ones increasing with additional time.

In brief, with the current approach, the approximative computation of the MN seems to be significantly affected by the minor differences in the probabilities, leading to unwanted ordered pairs. However, if this issue can be fixed, it might be that the approximative computation shows more interesting results than the exact one through the functions Id, Norm, S-Norm, and WoE. However, it is in itself an issue, because it is hard to motivate the use of an “approximative” approach when the exact one, towards which it is supposed to converge, shows poor results. Consequently, the results so far



(a) Assumption 3 (composition) shows different behaviours: Id (top-left) and Norm/S-Norm (top-right) are subject to losses of compliance with additional time, Id+5/Norm+5/S-Norm+5 (bottom-left) are fully non-compliant, and WoE (bottom-right) only increases. In any case, compliance rarely goes above 60%.

(b) Assumption 4 (expected) shows two tendencies: Id/Norm/S-Norm/WoE (left) are globally high but not perfect, while Id+5/Norm+5/S-Norm+5 (right) are stuck at 62.7%, which is the worst possible value.

Figure D.7: Evolution of the assumption compliance (3 and 4) of the MN technique for synthetic data with approximative computation. Only the distributions are shown for readability.

of the MN-based approach do not show sufficient support to use it out of an experimental environment, but they also show that further investigation might provide improvements which provide better results than the exact computation.

D.3 Genetic Algorithm

Although there is no exact and approximative computation to consider for the GA approach, there is a somehow equivalent dichotomy that we can do. Indeed, the aim of the GA is to compute a small part of the whole graph, part which is tuned based on several parameters. These parameters, which are the number of nodes of each type to consider, can be set up to consider the whole graph, thus 18 stakeholder nodes, 5 role nodes, 5 topic nodes, and 10 term nodes. This is what we call the *max data* perspective. At the opposite, the *min data* perspective considers still 18 stakeholder nodes, but only 1 role node, 1 topic node, and 1 term node. The reason why we consider 18 stakeholders also for min data is that they are the ones to be validated: by considering only 1 stakeholder, we could only validate whether or not the selected stakeholder is the most relevant one. We prefer to keep the whole set of stakeholders to see broadly how the ranking is affected. As a matter of fact, the dataset has been generated also with a limit of 30 term nodes although there is only 10, and we confirm that the results are highly similar whether we choose 10 or 30 nodes.

We focus now on the interesting settings for the GA approach, especially the time-outs from which some functions may provide stable results. Like for MN, we ignore the case of empty queries, because they are assumed to provide uninformative rankings, which means that they artificially increase the amount of *Indifference* ($PDD - ODD$) on the graphs.

By looking at the re-run variance in Figure D.8, we always obtain $ODD =$

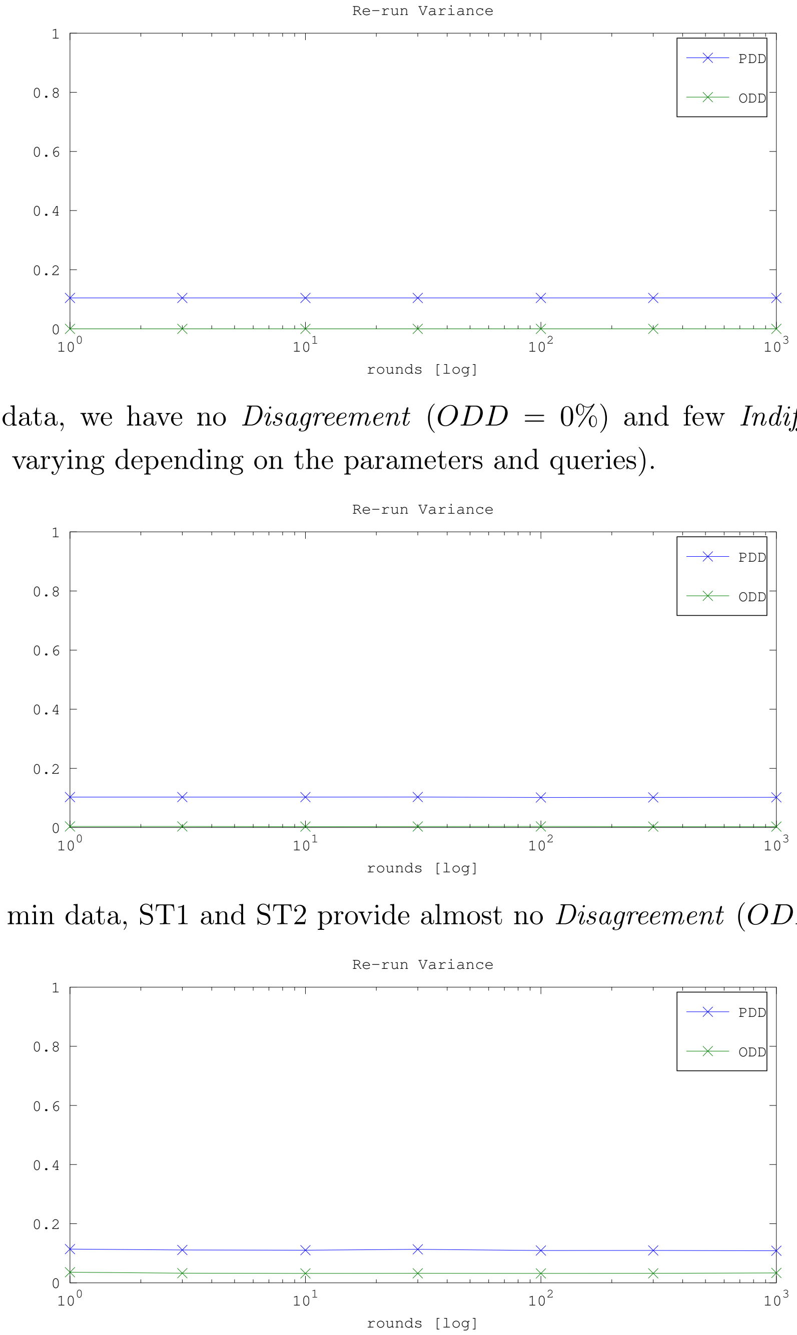
0 with max data, so we have no *Disagreement* between the rankings, which is good. PDD is not really high, but it introduces an *Indifference* zone, for which we need to confirm whether it happens because of proper *Unordered* pairs or a balanced *Disagreement* between the rankings. By looking at each function combination (STx, MTx), we could confirm that we always have $ODD = 0$, which means that it is a proper representation of the *Unordered* pairs of the rankings produced. Further analysis shows that different parameters and different queries provide different values of PDD, for instance ST1 has 8.3%, ST2 has 3.5%, and ST3 has 19.6%. These observations are in fact consistent with the max data perspective: because we compute the whole graph, the result should be always the same, so if a lack of *Agreement* is observed, it should be due to *Unordered* pairs. This max data perspective, like the exact computation of the MN, gives us an idea of what the min data perspective should look like.

If we consider min data, ST1 and ST2 have almost no *Disagreement* ($ODD \approx 0.3\%$), which is further confirmed by detailed ODD which are, although not constant, often below 2.0% (3rd quartile). With ST3, however, the *Disagreement* starts to be noticeable ($ODD \approx 3.3\%$), and a detailed ODD tends to double it in average, often triple it (3rd quartile). Another observation is that the detailed ODD values are the ones showing the best a convergence effect: the maximal values appear at low time-outs while the high time-outs tend to have only values close to the average. Yet, even in this situation we do not observe much difference between different time-outs, which is probably due to the implementation design: the GA has a generation of 100 individuals, but because there is in total 5 roles, 5 topics, and 10 terms, only $5 \times 5 \times 10 = 250$ individuals are possible. If we take 100 random individuals at the first round, there is a high chance to have one of the best individuals from the start, leading to these results (33% chance if there is only 1 good individual, 98% if there is 10). In other words, the best level of stability should be globally expected from the very start with this synthetic data, and because the GA tends to keep the best individuals, this stability should be preserved and reflect also in the extra-run measures.

Figure D.9 confirms this expectation by showing highly similar results for the extra-run bias, including with the analysis of the detailed ODD measures. In brief, because of the nature of the algorithm used and the small data to compute, the best individuals should be found after few rounds on the synthetic data, even with min data. The small *Disagreement* observed with min data –which is preserved with more rounds– probably reflect the fact that we consider only 1 topic, 1 role, and 1 term to compute the values, which is so poor in information than several combinations, leading to different rankings, can provide the best relevance values. It is worth mentioning that no specific parameter (amount of roles, topics, or terms) appears to have more influence than the others: increasing one of them decreases the *Disagreement* but never to the point of max data. Similarly, adding a bit of each does not seem to provide much improvements, and only maximizing them lead to obtain a proper convergence. These observations are important because being able to produce reliable rankings based on a small part of the whole graph is the main assumption behind the GA approach. If it is hard to draw any conclusion now with such a small graph, it is something to keep in mind when dealing with bigger ones.

Based on the re-run and extra-run analysis, any time-out seems fine because the best individuals seem to be generated quickly, although it is better to consider at least more than 1 round for min data. Selecting a relevant time-out is for sure more important for other experiments, which deal with more data.

For Assumption 1 (no data), Figure D.10a shows a perfect compliance. For a matter of integrity, we inform the reader that the figure has been generated based on a reduced set of settings because of material limitations making us

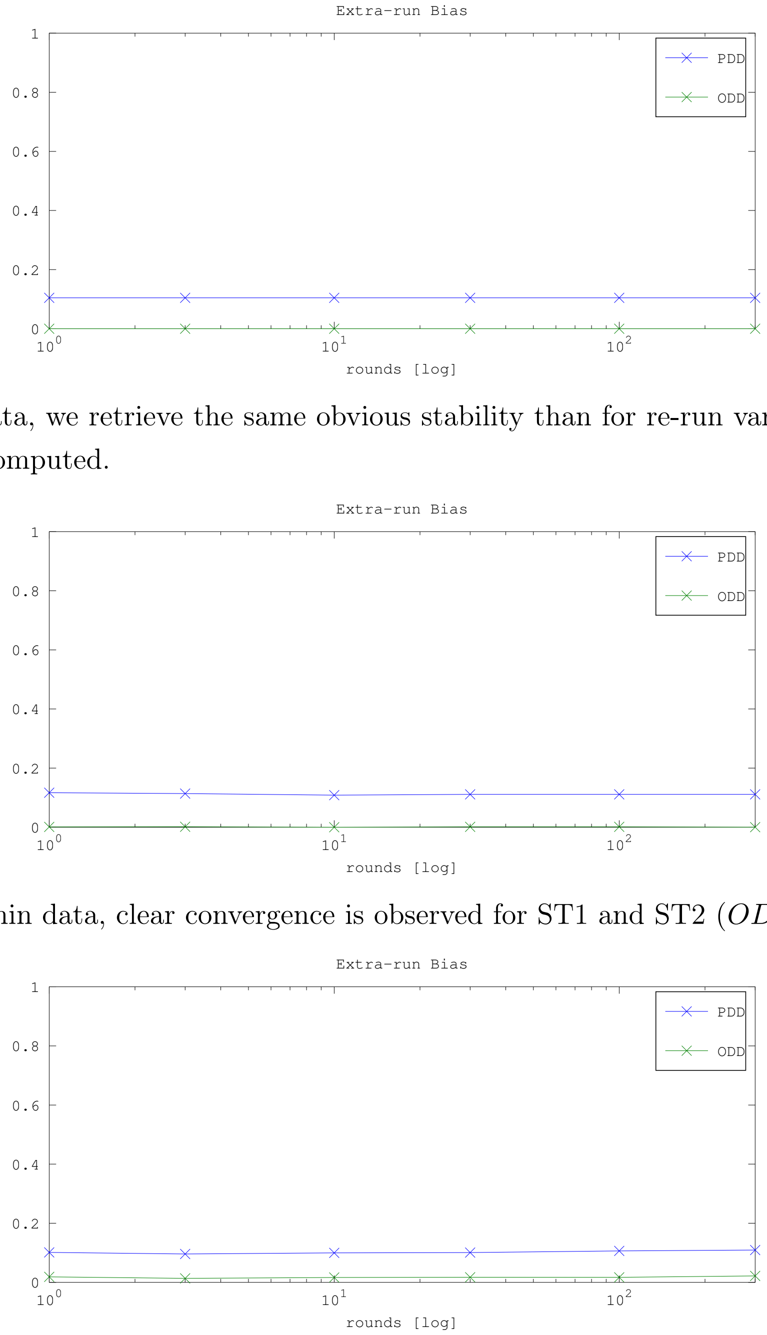


(a) With max data, we have no *Disagreement* ($ODD = 0\%$) and few *Indifference* ($PDD - ODD = 10.5\%$, varying depending on the parameters and queries).

(b) With min data, ST1 and ST2 provide almost no *Disagreement* ($ODD \approx 0.3\%$).

(c) With min data, ST3 is the least interesting with more *Disagreement* ($ODD \approx 3.3\%$).

Figure D.8: Evolution of the re-run variance of the GA technique for synthetic data.



(a) With max data, we retrieve the same obvious stability than for re-run variance because the whole graph is computed.

(b) With min data, clear convergence is observed for ST1 and ST2 ($ODD \approx 0.1\%$).

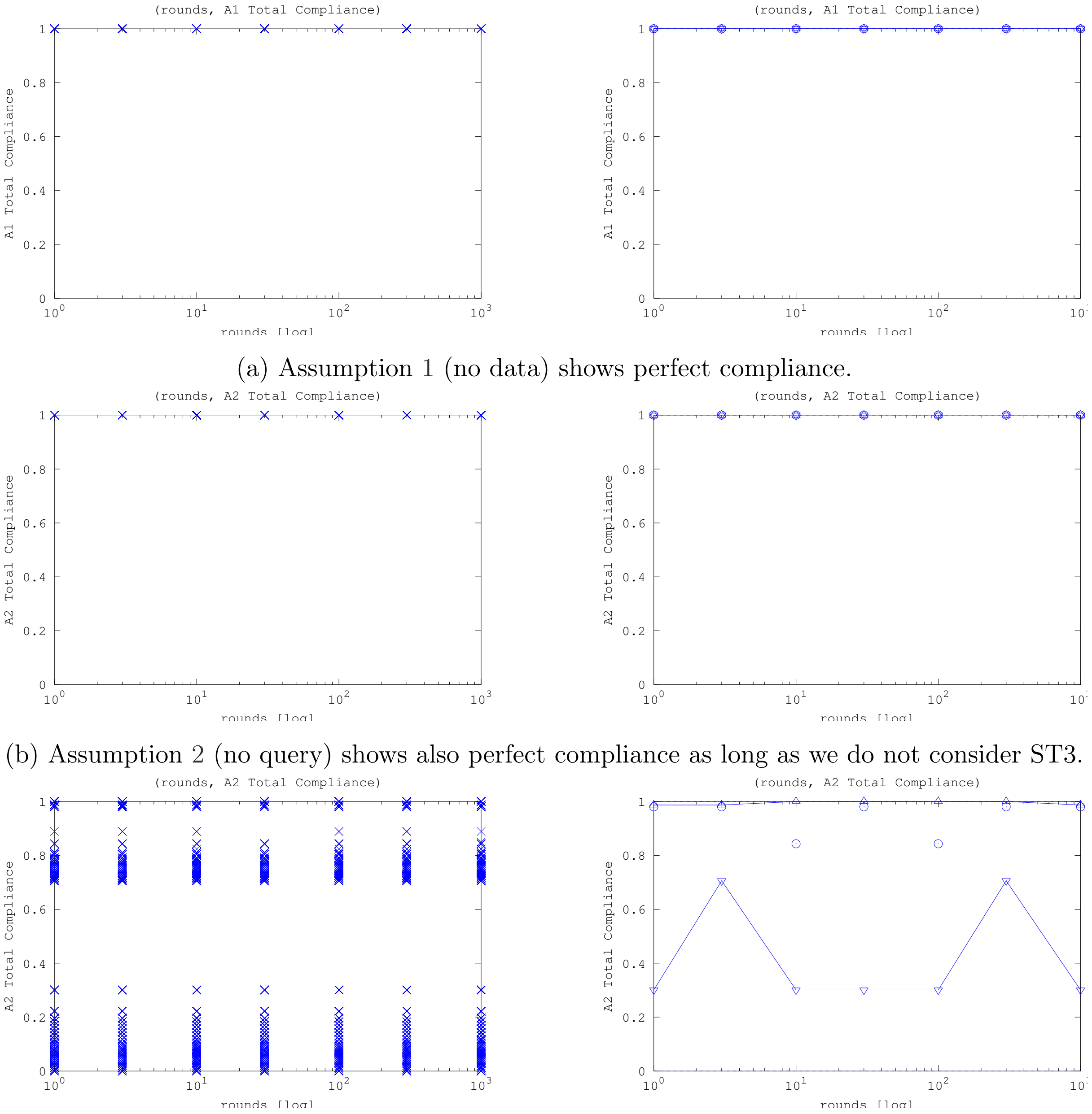
(c) With min data, ST3 shows some *Disagreement* ($ODD \approx 1.8\%$).

Figure D.9: Evolution of the extra-run bias of the GA technique for synthetic data.

unable to compute all the dataset. A different procedure has been used to exhaustively check that the compliance is perfect everywhere.

For Assumption 2 (no query), ST1 and ST2 provide a perfect compliance, as shown by Figure D.10b. With ST3, although a significant part achieves high if not perfect compliance, Figure D.10c shows that there is still a significant amount of rankings which have low levels of compliance, thus showing that ST3 tends to order the pairs. As such, ST3 is subject to a *network-specific* bias, so the simplification it provides compared to ST2 results in an additional inconvenient. This has to be balanced with the computation time saved, which varies significantly depending on the parameters, as shown by Figure D.11. Moreover, it is hard to classify it as good or poor by considering how small is the full graph. However, the ratio of time saving is expected to increase with larger networks, because ST2 should be subject to a combinatorial explosion, so it is not a definitive result and ST3 might still be interesting if its compliance is satisfying, although the network-bias remains unwanted. Additionally, ST1 is a lot more efficient in saving computation time, but we consider it as a naive computation of the network, so we expect it to loose in compliance to Assumption 4 with bigger graphs. If this expectation is not confirmed, then it might be a good alternative to ST2 due to this gain in performance.

Assumption 3 (composition), summarized in Figure D.12, is perfectly fulfilled with max data. Because we compute the full graph, the compliance is expected to properly reach 1 rather than being only close to it, which is what is observed here. The min data case is not perfect, but it is expected due to the minimal data which can lead to different results, and the vast majority still remains broadly compliant. Once again, the compliance from the very start even for min data can be explained by the best individuals having a high chance to be generated from the first round. An interesting observation is how the minimum remains rather low, close to 60% of compli-

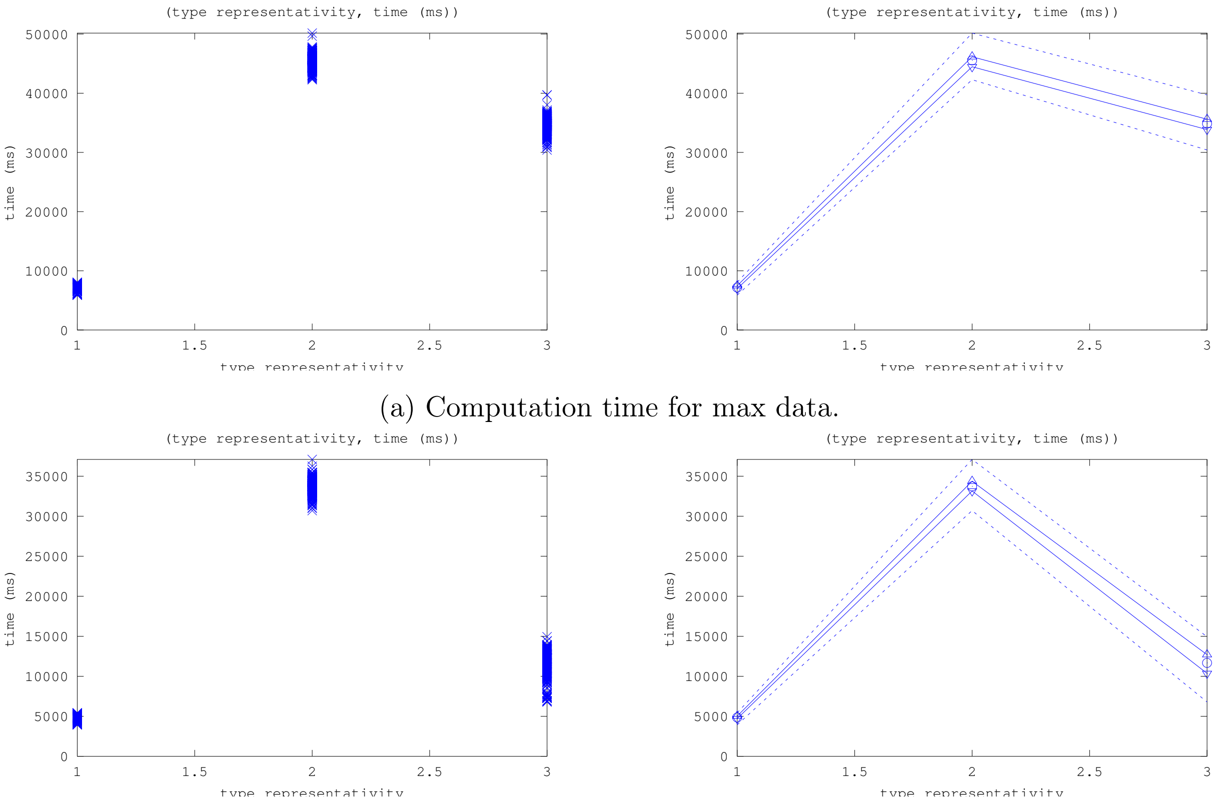


(a) Assumption 1 (no data) shows perfect compliance.

(b) Assumption 2 (no query) shows also perfect compliance as long as we do not consider ST3.

(c) ST3 implies a significant drop of compliance to Assumption 2, which needs to be balanced with the gain of time it is assumed to provide compared to ST2.

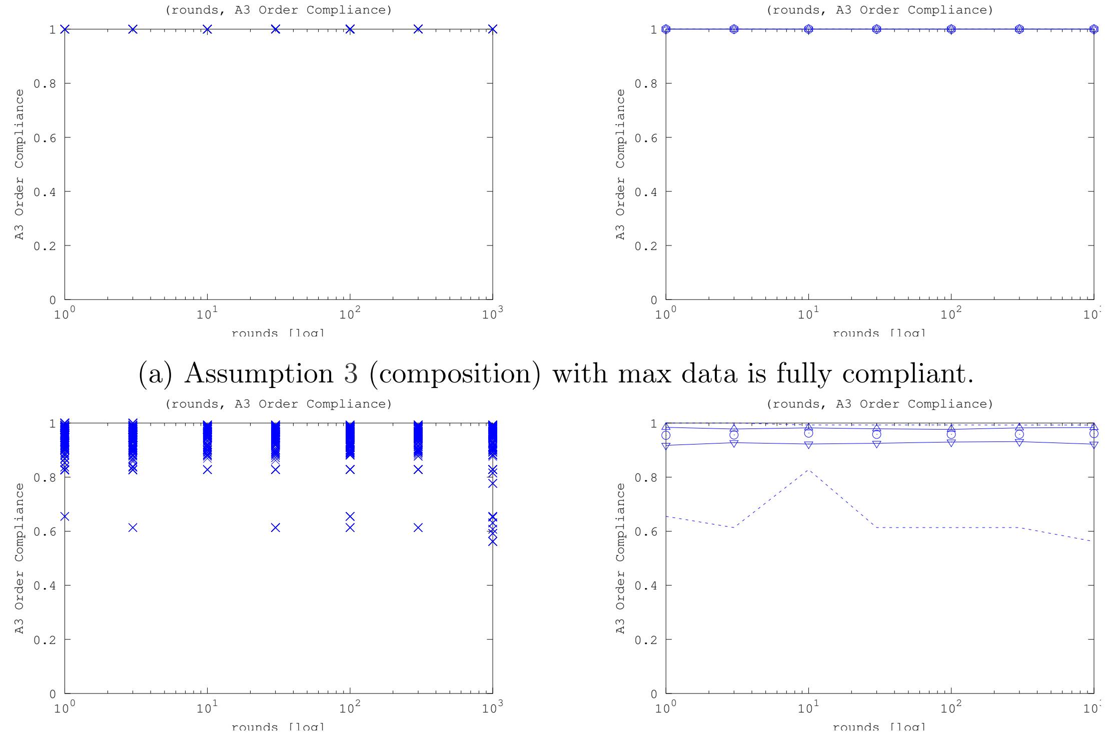
Figure D.10: Evolution of the assumption compliance (1 and 2) of the GA technique for synthetic data.



(a) Computation time for max data.

(b) Computation time for min data.

Figure D.11: Computation time of ST1, ST2, and ST3 (resp. functions 1, 2, and 3) with 1000 rounds. ST3 consumes always more than ST1 and becomes less interesting with an increasing sub-graph to compute.



(a) Assumption 3 (composition) with max data is fully compliant.

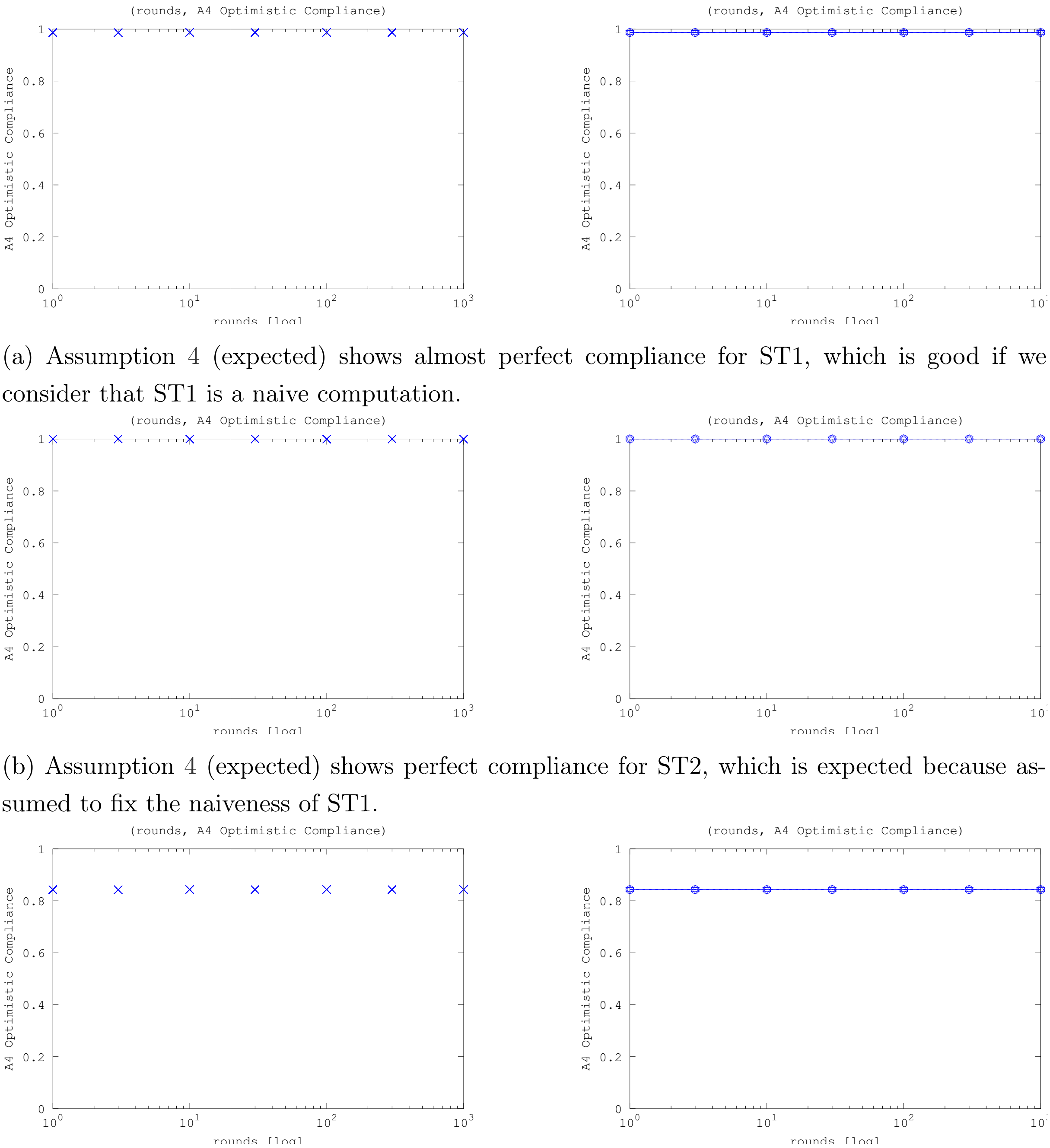
(b) Assumption 3 (composition) with min data shows a generally high compliance (94.5% in average), although it can reach significantly lower levels ($min = 56.2\%$) and is rarely perfect.

Figure D.12: Evolution of the assumption compliance (3) of the GA technique for synthetic data.

ance. The only explanation we see is the potential presence of a Pareto front, so we might have several apparent but incorrect "best solutions" due to the poor information provided by min data. Further investigation is needed to properly explain this observation.

For Assumption 4 (expected) with max data on Figure D.13, ST1 provides high compliance, although it is not perfect. ST2 however provides a perfect fit, which is consistent with the idea that ST2 is an improved version of ST1. ST3, which was designed to improve the performance of ST2, is unfortunately the least compliant, and looking at the order compliance shows that almost half of the ordered pairs are not compliant. If we look at min data with Figure D.14, ST1 and ST2 show similar results if we ignore the small loss of compliance at the start, which is expected due to the less information available. The compliance of ST3 is globally improved, which was unexpected, but it still remains far below the others with a *Disagreement* generally between 5% and 20% of the ordered pairs.

As a summary, if we consider max data for which we compute the full graph, then ST2 seems to perfectly fit all the assumptions, and although ST1 does not perfectly fit Assumption 4 it still has a rather high level of compliance and provides a significant save in computation time with almost an order of magnitude less. Consequently, ST2 seems to be the right function to use, and if performance issues arise, ST1 appears to be an interesting alternative. If we consider min data, we can draw the same conclusions, which is unexpected because a lot less information is provided and, yet, we achieve the highest levels of compliance with enough rounds. This is a good evidence to show that the idea of computing a smaller part of the graph can still give reliable results, although it should be stressed with other experiments, which build on non-synthetic data.

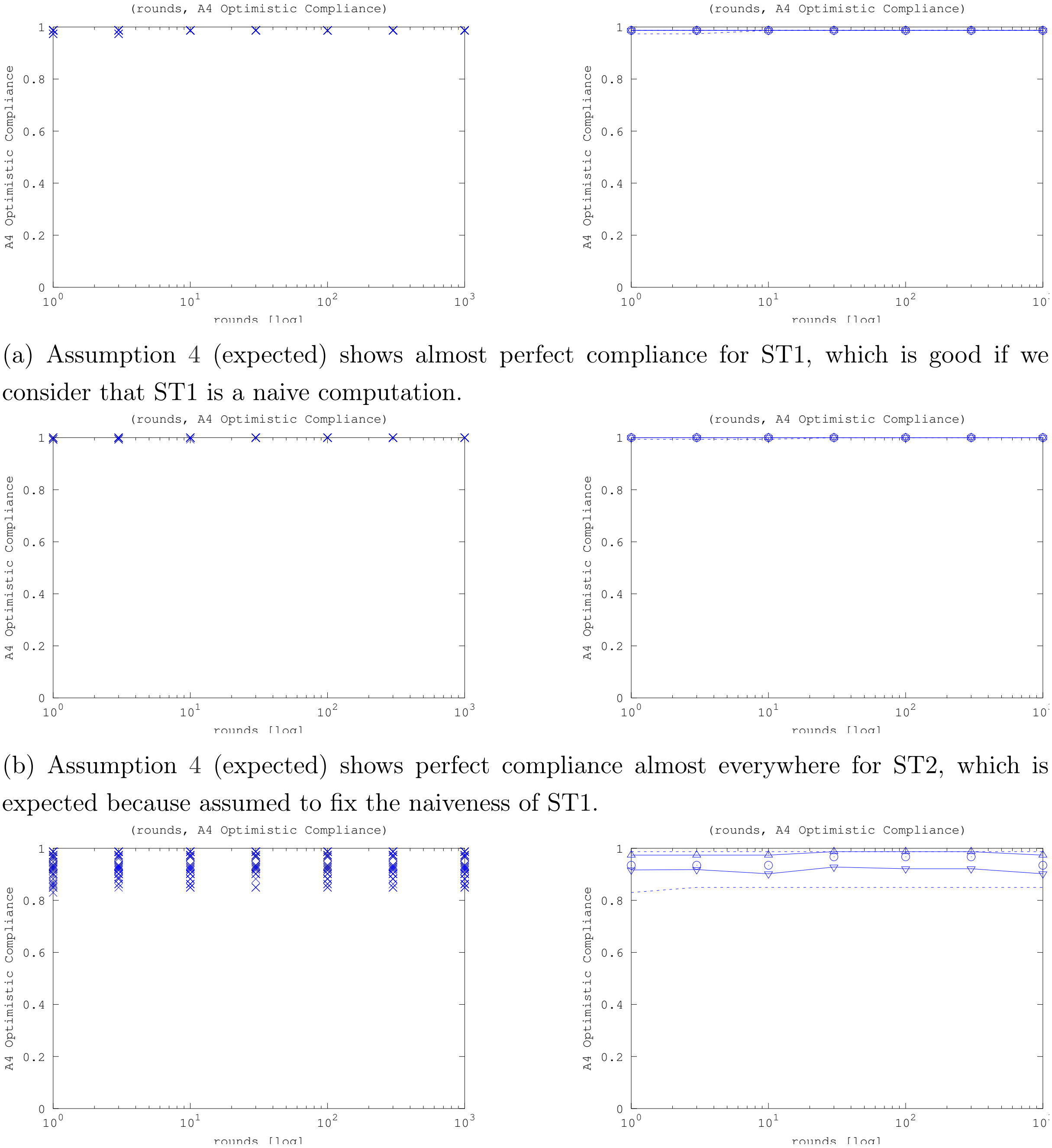


(a) Assumption 4 (expected) shows almost perfect compliance for ST1, which is good if we consider that ST1 is a naive computation.

(b) Assumption 4 (expected) shows perfect compliance for ST2, which is expected because assumed to fix the naiveness of ST1.

(c) Assumption 4 (expected) shows ST3 does not achieve proper compliance (84.3%) due to its simplification compared to ST2. Focusing on ordered pairs reduces to 57.9%.

Figure D.13: Evolution of the assumption compliance (4) of the GA technique for synthetic data with max data.



(a) Assumption 4 (expected) shows almost perfect compliance for ST1, which is good if we consider that ST1 is a naive computation.

(b) Assumption 4 (expected) shows perfect compliance almost everywhere for ST2, which is expected because assumed to fix the naiveness of ST1.

(c) Assumption 4 (expected) shows ST3 does not achieve perfect compliance due to its simplification compared to ST2, although it remains high in average (94.2%).

Figure D.14: Evolution of the assumption compliance (4) of the GA technique for synthetic data with min data.

Appendix E

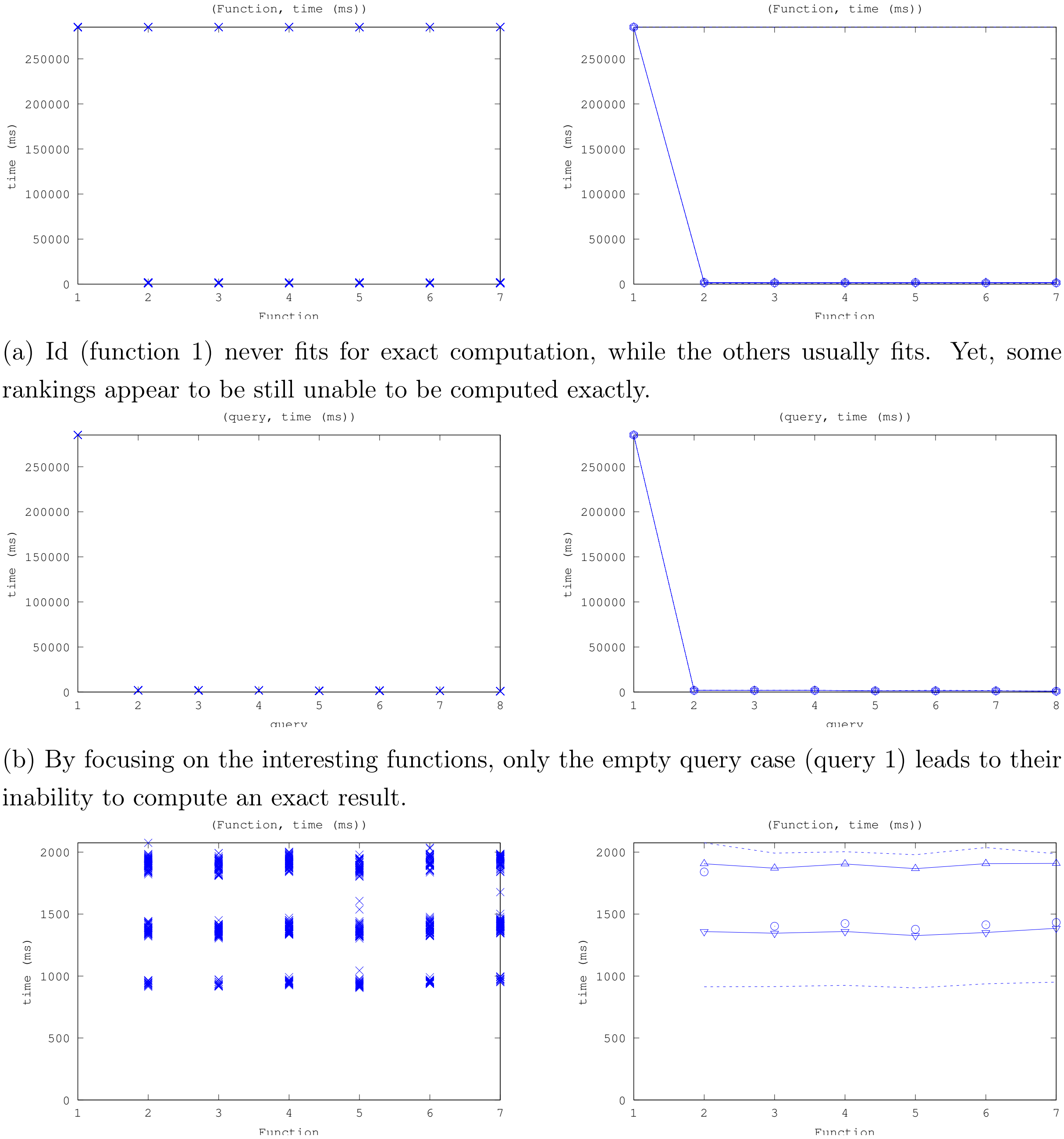
Detailed Analysis for Cuisine Data

The graph for cuisine discussion being small enough, we can compute exact values for the MN approach, which is what we start from. Through this analysis, we should be able to establish whether the MN approach provides meaningful results. After the analysis of the exact computation, we compare it to the approximative computation, which is supposed to converge towards the results of the exact one. Because the approximative computation is supposed to be used in big graphs, when the exact computation is not reasonable, this comparison should provide some insights from a performance perspective. Then, a third analysis is made for the GA approach, which should support at the same time the correctness and performance of the approach.

E.1 Markov Network (exact)

For the exact computation of the MN approach, we identify the relevant settings by looking at the ones able to consume less than the available time. Figure E.1a shows that only Id consumes all the time, making it unsuited for exact computation, while the other functions are usually able to compute exactly. Yet, it is not always the case, and Figure E.1b shows that, for all these functions, only the empty query leads to the inability to produce an exact result. If we remove this last case to see what are the most interesting functions, Figure E.1c shows that no one stands out, all of them being equivalent regarding their time of computation.

Now that all the functions but Id are shown to be able to deal with exact computation, as long as we query something, we need to know how well they comply with our gold standards. Regarding Assumption 1 (no data), we cannot check it because no dataset has been generated for this purpose. In the absolute, Assumption 2 (no query) cannot be evaluated either because the remaining functions are not able to compute exactly with empty queries, so it does not make sense to evaluate the resulting rankings for the purpose of exact computation. Rather, we consider that their inability to provide rankings based on an exact computation corresponds to a failure in satisfying the assumption. For Assumption 3 (composition), Figure E.2a shows that full compliance is achieved for the priorised functions, Id+5, Norm+5, and S-Norm+5, while the others are fully non-compliant. For these latter, this is because they only provide uninformative rankings by putting everyone at the same rank, so no ordered pair needs to be checked, which leads to a zero value. If we focus on the priorised functions, then Assumption 4 (expected) shows a mixed compliance, sometimes full, sometimes poor (not compliant at all if we consider only ordered pairs). Figure E.2b, which focuses on the queries of the gold standard, shows that only 1 is compliant (Tiramisu) and the other not at all (Mongolian food). This is because, independently of the query, the rankings provide the same orders, which means that people have the same rank independently of what is queried. A deeper investigation shows that not only the orders are the same for both queries, but that the probabilities computed are the very same, which happens for any non-empty query and for any of the three priorised functions. In brief, Assumption 4 (expected) cannot be considered as fulfilled because the same ranking is provided independently of the query.

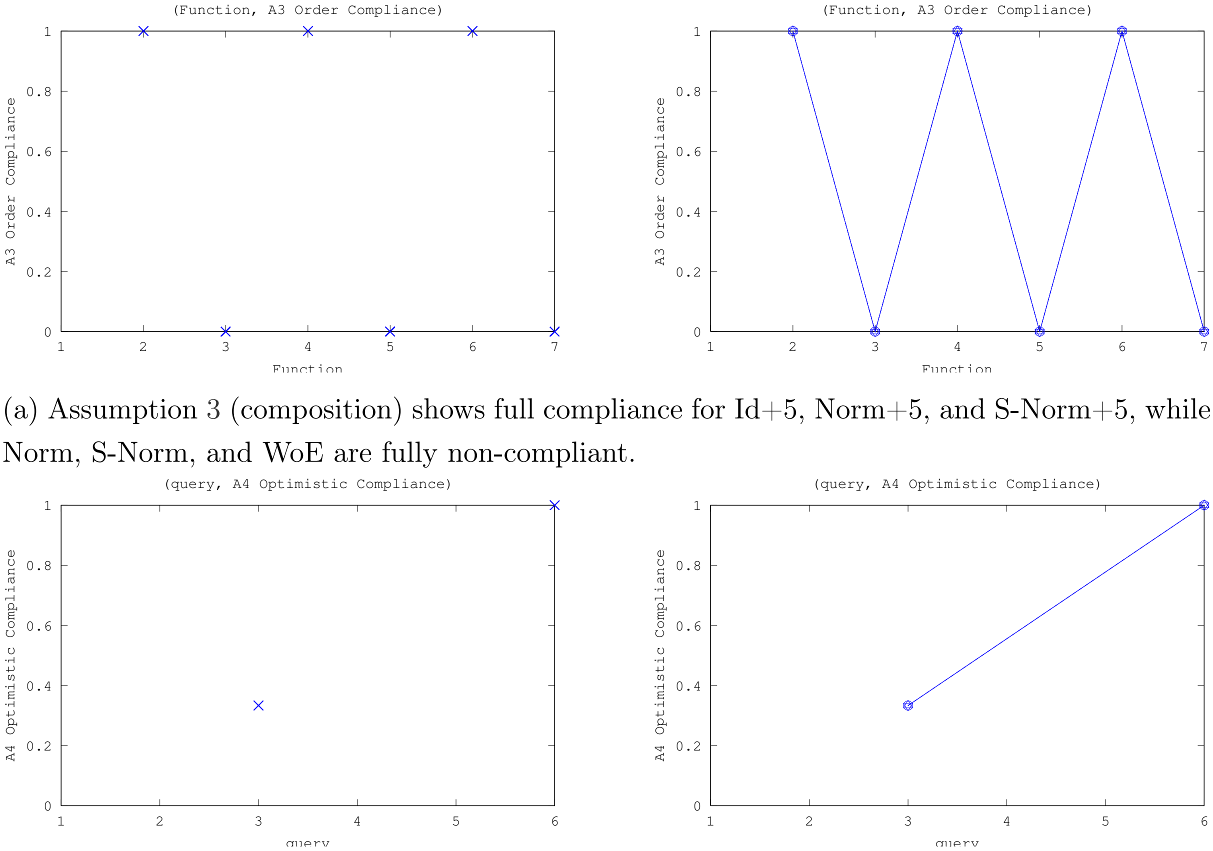


(a) Id (function 1) never fits for exact computation, while the others usually fits. Yet, some rankings appear to be still unable to be computed exactly.

(b) By focusing on the interesting functions, only the empty query case (query 1) leads to their inability to compute an exact result.

(c) By focusing on the non-empty queries, no time advantage is shown between each function.

Figure E.1: Time required for exact computation at highest time-out (300s) of the MN technique for cuisine data.



(a) Assumption 3 (composition) shows full compliance for Id+5, Norm+5, and S-Norm+5, while Norm, S-Norm, and WoE are fully non-compliant.

(b) A focus on the 2 queries of the gold standard shows that only 1 is compliant and the other almost not at all (only for the unconstrained pairs), so Assumption 4 (expected) cannot be considered as fulfilled.

Figure E.2: Evolution of the assumption compliance of the MN technique for cuisine data with exact computation.

As a summary, even if we have functions able to provide informative rankings, they produce the same ranking independently of the query and are unable to compute empty queries. Thus, it does not seem that the exact MN approach can be used on these cuisine discussions to obtain relevant results.

E.2 Markov Network (approximative)

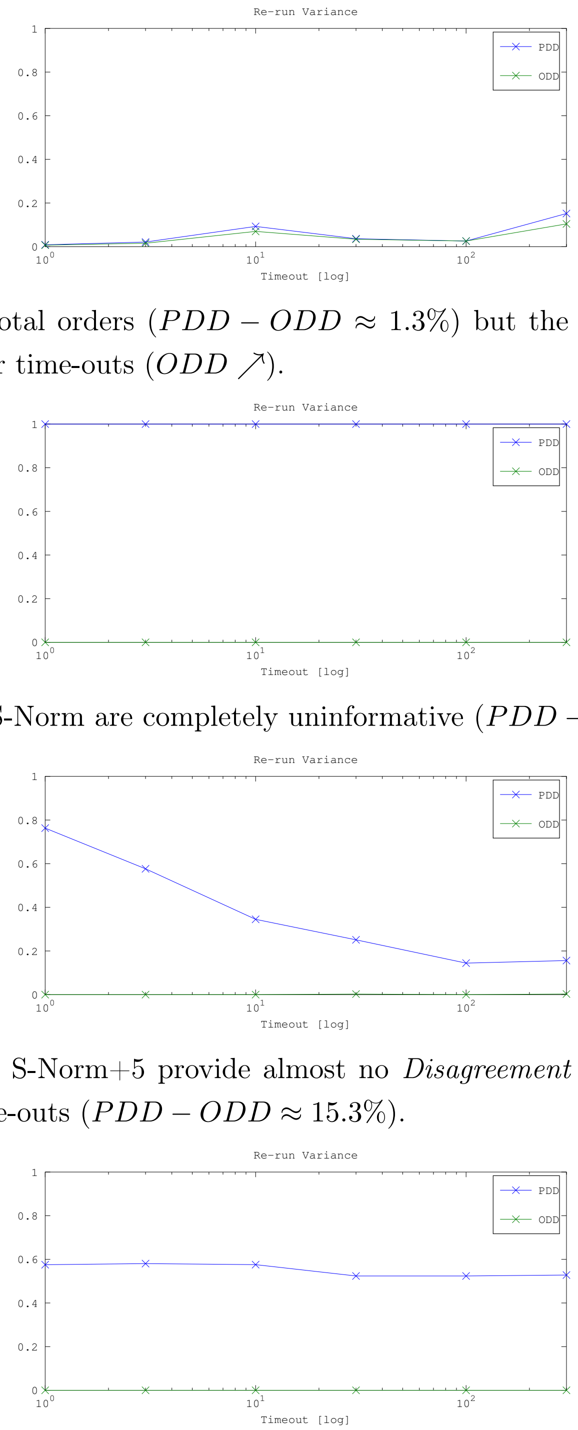
As the experiment with synthetic data shows us that the approximative computation can do better than the exact one, we also analyse the approximative computation for this cuisine dataset to see if we obtain the same observations. We first try to identify what are the relevant settings, especially the time-outs from which some functions may provide stable results. For this, we ignore the case of empty queries, because they are assumed to provide uninformative rankings, which means that they artificially increase the amount of *Indifference* ($PDD - ODD$) on the graphs.

By analysing the re-run variance, shown in Figure E.3, we can see that Id is highly informative by having really close PDD and ODD, but it shows some tendency to generate more diverse rankings with higher time-outs, although the *Disagreement* only reaches 10.4% of pairs. At the opposite, Norm and S-Norm remain completely uninformative: all stakeholders are at the same rank because of having the same probability of 0 for empty query, 1 otherwise. Id+5, Norm+5, S-Norm+5 on the other hand are interesting: ODD is close to 0 and a significant decrease of PDD from 76.3% to 15.6% is observed. Further investigation shows that the resulting *Indifference* area is mainly due to actual *Unordered* pairs: the detailed ODD values are null for a vast majority of rankings. Finally, WoE shows a different behaviour: ODD is always perfect (0%) and PDD starts from a lower level than the priorised functions (57.5%), but only a small decrease of PDD is observed with higher time-outs (52.8%). Once again, the *Indifference* area is due to proper *Unordered* pairs.

Figure E.4 shows again different behaviours through the extra-run bias (Norm and S-Norm are not analysed because completely uninformative). Id does not appear to converge to a specific ranking, as shown by the significant *Disagreement* ($ODD \approx 18.1\%$). This observation is worsen by the huge *Indifference* ($PDD - ODD \approx 46.7\%$) which happens to be caused by a significant amount of pairs in *Disagreement* between the individual rankings, if we look at detailed ODD values. The priorised functions Id+5, Norm+5, S-Norm+5 do not show much *Disagreement* between time-outs ($ODD \approx 1.3$) and a progressive gain of *Agreement* ($1 - PDD \nearrow$) thus mimicking the re-run variance. The analysis of detailed ODD values confirm that the rankings start mainly unordered and progressively gain in information. These functions offer consequently one of the most interesting behaviour, because the ordered pairs are found progressively, and additional time leads to more ordered pairs. This can be particularly interesting for instance as an anytime algorithm[1], as long as correct rankings are provided. Finally, WoE shows a similar behaviour, but the lack of decrease of *Indifference* at high time-outs shows that it gets stuck to only few ordered pairs.

Consequently, no specific time-out seems to stand out: whether the function remains unstable, whether it still need more time to stabilize. Thus, we investigate the assumption compliance independently of any time-out, looking simply at how compliance evolves with more computation time in Figure E.5. Assumption 1 (no data) has not been checked because of the absence of altered dataset for this purpose. However, we can see that Assumption 2 (no query) tends to be never fulfilled, although the priorised functions start with a good level of compliance. Like for the experiment on synthetic data, this lack of compliance happens mainly due to approximative

[1] Anytime algorithms are algorithms we can stop at any arbitrary time rather than by fixing an a priori time-out.



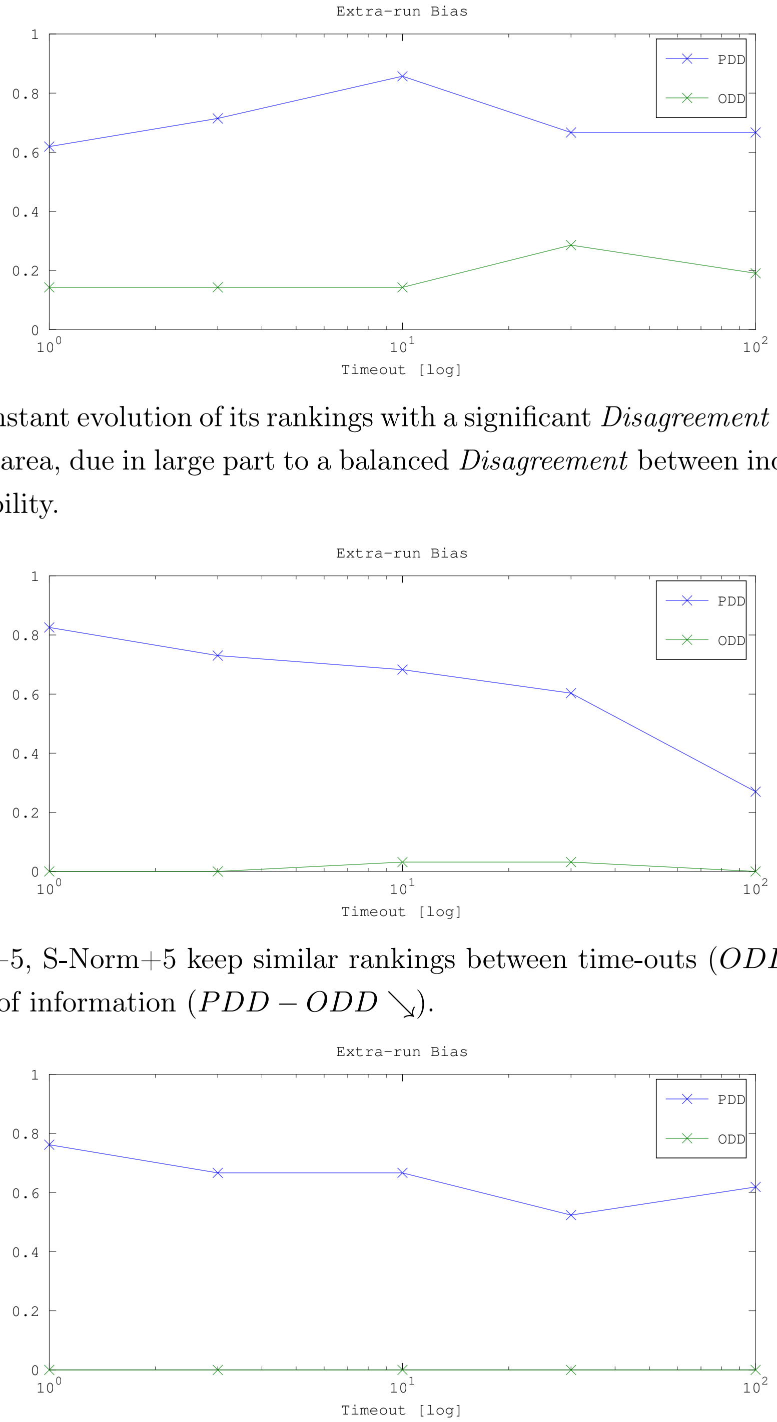
(a) Id provides nearly total orders ($PDD - ODD \approx 1.3\%$) but the final rankings tend to be more diverse with higher time-outs ($ODD \nearrow$).

(b) Norm and S-Norm are completely uninformative ($PDD - ODD = 100\%$).

(c) Id+5, Norm+5, and S-Norm+5 provide almost no *Disagreement* ($ODD \approx 0.1\%$) and few *Indifference* at high time-outs ($PDD - ODD \approx 15.3\%$).

(d) WoE shows no *Disagreement* at all ($ODD = 0\%$) but remains at a rather high amount of *Indifference* even with high time-out ($PDD - ODD \approx 55.1\%$).

Figure E.3: Evolution of the re-run variance of the MN technique for cuisine data with approximative computation.



(a) Id shows a constant evolution of its rankings with a significant *Disagreement* ($ODD \approx 18.1\%$). The *Indifference* area, due in large part to a balanced *Disagreement* between individual rankings, add to the instability.

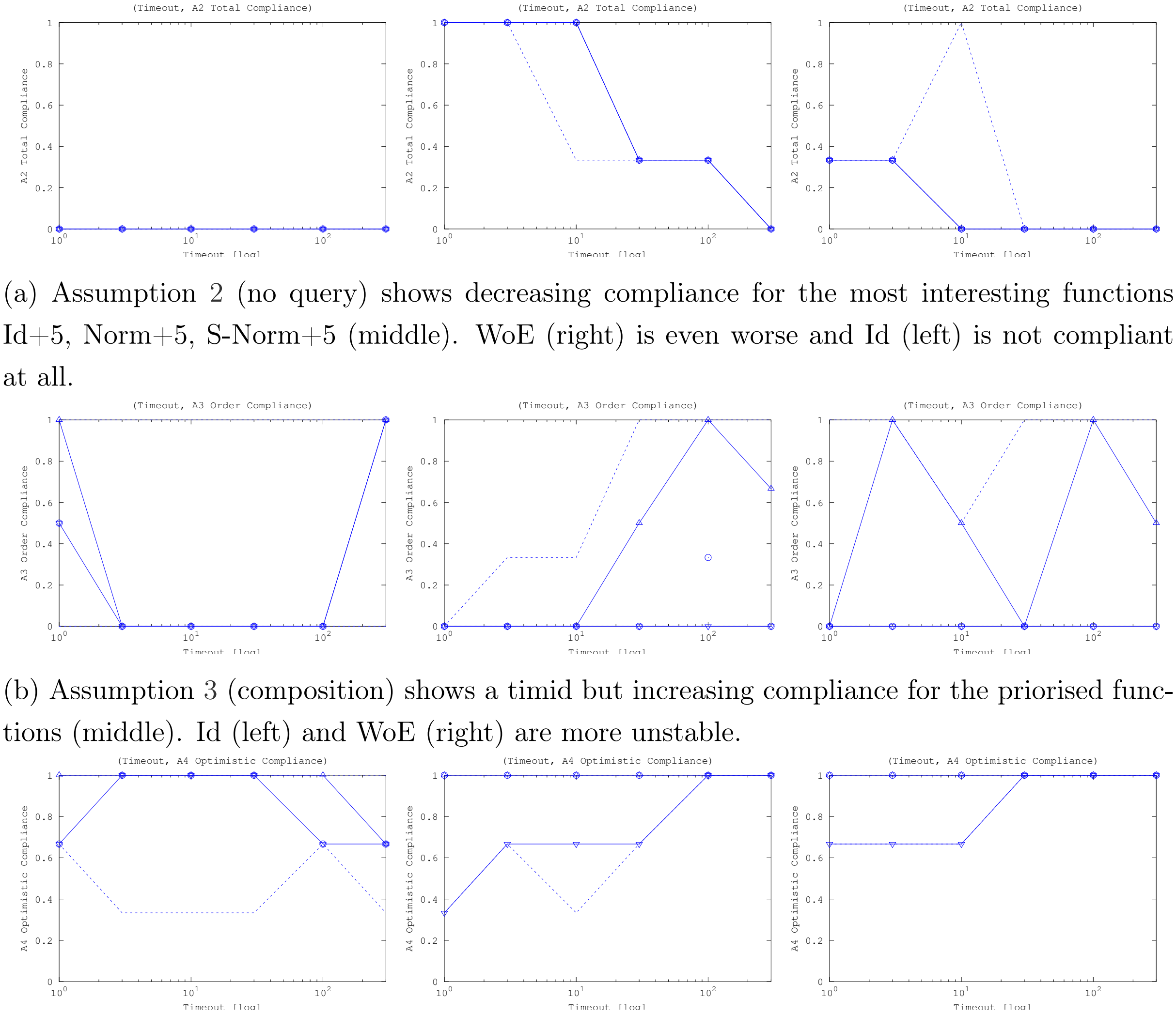
(b) Id+5, Norm+5, S-Norm+5 keep similar rankings between time-outs ($ODD \approx 1.3\%$) with a progressive gain of information ($PDD - ODD \searrow$).

(c) WoE shows a similar behaviour, but the lack of decrease of *Indifference* at high time-outs shows that it gets stuck to only few ordered pairs.

Figure E.4: Evolution of the extra-run bias of the MN technique for cuisine data with approximative computation.

values which are really close but not strictly equal: Id produces probabilities close to 0.5, while Id+5/Norm+5/S-Norm+5 are close to 1. Only WoE provides significantly different values, with Bob equal or close to 0, Alice close to 1, and Carla equal to 1, which make it the only “confirmed” non-compliant function. Assumption 3 (composition) shows slightly better results with an increase of compliance for priorised functions at high time-outs, although it does not seem robust if we consider the lack of compliance at the highest time-out. The instability of the other functions make them unreliable, which we could consider as a failure to satisfy the assumption. Finally, Assumption 4 (expected) unexpectedly provide the best results, with Id being globally compliant although it is subject to some instability, and the other functions converging properly to a perfect compliance with more computation time.

The good levels of compliance achieved for Assumption 4 should be balanced with the fact that (i) the rankings are short, with only 3 items to rank, (ii) the GS is partially ordered, with only 2 pairs constrained over 3, and (iii) gold standards are provided only for 2 queries. Moreover, because Assumption 3 is not supposed to be impacted by the approximative computation (it is based on centroids) and because it relates to the consistency of the approach, we expect it to be among the easiest to satisfy, but it is not the case here. We take all these observations as a good evidence that the approach probably does not fit, and that the gold standards for Assumption 4 do not stress enough the correctness of this approach by checking only 2 queries. Nevertheless, it is only a support for further investigation, and if the approximation issue can be fixed, the priorised functions appear to be among the most interesting ones.



(a) Assumption 2 (no query) shows decreasing compliance for the most interesting functions Id+5, Norm+5, S-Norm+5 (middle). WoE (right) is even worse and Id (left) is not compliant at all.

(b) Assumption 3 (composition) shows a timid but increasing compliance for the priorised functions (middle). Id (left) and WoE (right) are more unstable.

(c) Assumption 4 (expected) shows increasing compliance for the priorised functions (middle). WoE (right) is even better, while Id (left) seems always high but subject to instability.

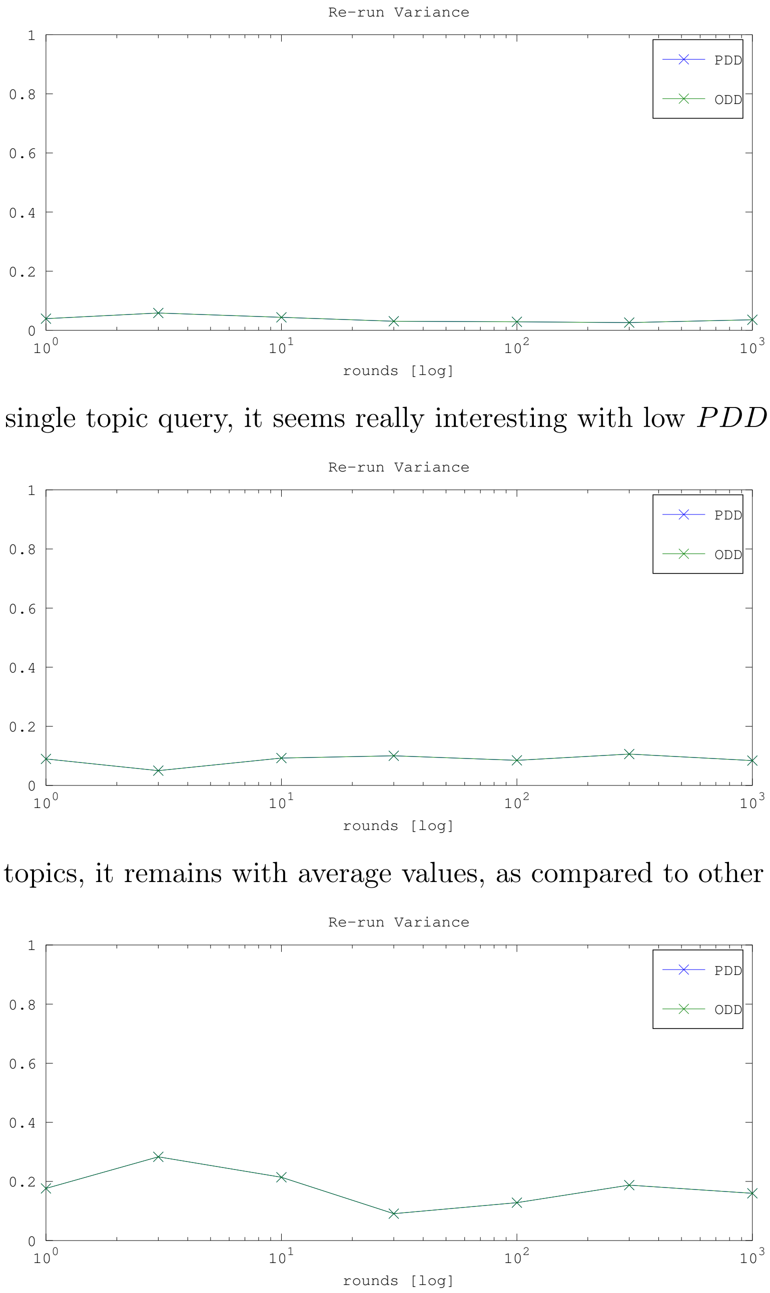
Figure E.5: Evolution of the assumption compliance of the MN technique for cuisine data with approximative computation. Only the distributions are shown for readability.

E.3 Genetic Algorithm

Like for the experiment on synthetic data, although the GA approach does not have an exact and an approximative computation, we can define two similar perspectives. *Max data* focuses on settings which compute a significant part of the graph, which here corresponds to 3 stakeholders, 3 topics, and 30 terms, while *min data* focuses on the smallest sub-graphs, thus 3 stakeholders, 1 topic, and 1 term. We keep 3 stakeholders because we want to see how the whole ranking is impacted by these different settings. At the opposite of the synthetic data, max data does not correspond to a full graph (293 terms are available, an order of magnitude more than the limit of 30), so we can expect some variability to occur. As a reminder, we identify the relevant settings by ignoring the case of empty queries, because they are assumed to artificially increase the amount of *Indifference* ($PDD - ODD$).

For the re-run variance with max data, a usual analysis which tries to establish which functions are the most interesting does not work well. The general tendency is rather constant, and varying the functions (STx or MTx) does not allow to identify significant differences in the evolution of the re-run variance, thus forcing to focus on query-specific differences. The behaviours can vary significantly between the queries depending on the functions used, and if PDD and ODD remain usually within $[0; 0.3]$, many shapes are observed: low or high PDD/ODD, increasing or decreasing, no to many gaps between PDD and ODD. We highlight here only some combinations that we consider to be illustrative for our explanations. Figure E.6 shows well how (ST1, MT1) is interesting for single topic queries, by having low PDD and ODD, but less interesting with richer queries. Figure E.7 shows the reverse tendency for (ST2, MT3), which starts relatively low too, although a bit higher than (ST1, MT1), and decreases even more with richer queries, until to have almost perfect values (PDD and ODD close to zero if not equal). Careful readers may have noticed in the presentation of the datasets that some settings are particularly poor in rankings (with a minimum of 3 rankings), explaining why so few variance could be observed with the richest query. As a matter of fact, each setting here has between 10 and 15 rankings, so the stability is properly established. A last interesting case to show for illustration would be ST3: the recurrent observation is that it provides always average distances, comparable to Figure E.6b, which is why we do not make a dedicated figure. After these three illustrations, we can summarize our full analysis: it appears that (ST2, MT1) and (ST2, MT3) are the most interesting combinations. Indeed, although (ST2, MT1) is less interesting than (ST1, MT1) for single-topic queries, it remains close to it and ST1 usually provides the worst ODD values. Additionally, both (ST2, MT1) and (ST2, MT3) offer lower PDD and ODD values with richer queries, and they are the only ones able to reach properly $ODD = 0$ (with the richest queries).

If we take the min data perspective and redo the analysis, we observe similar interests, as shown in Figure E.8. (ST1, MT1) thus appears as rather interesting with a really low *Disagreement* ($ODD \approx 2.8\%$). Similarly, ST2 is once again interesting with MT1 or MT3 with really low *Disagreement* ($ODD \approx 1.4\%$) but some noticeable *Indifference* ($PDD - ODD \approx 8.5\%$ with a decreasing tendency). ST3, as usual, is the least interesting by providing a significant amount of *Disagreement* ($ODD \approx 14.3\%$). At the opposite of previous analyses, we do not observe clear differences between the timeouts, which can be partly explained with the small graph we are dealing with. With no roles, the 3 topics and 293 terms lead to $3 \times 293 = 879$ possibilities, and with 100 random individuals generated at the first round, assuming that 10 individuals appear as good ones among the 879, we already have a probability of 68.2% to obtain at least one of them. Thus, we might expect to have already some good individuals early, leading to almost immediate convergence.

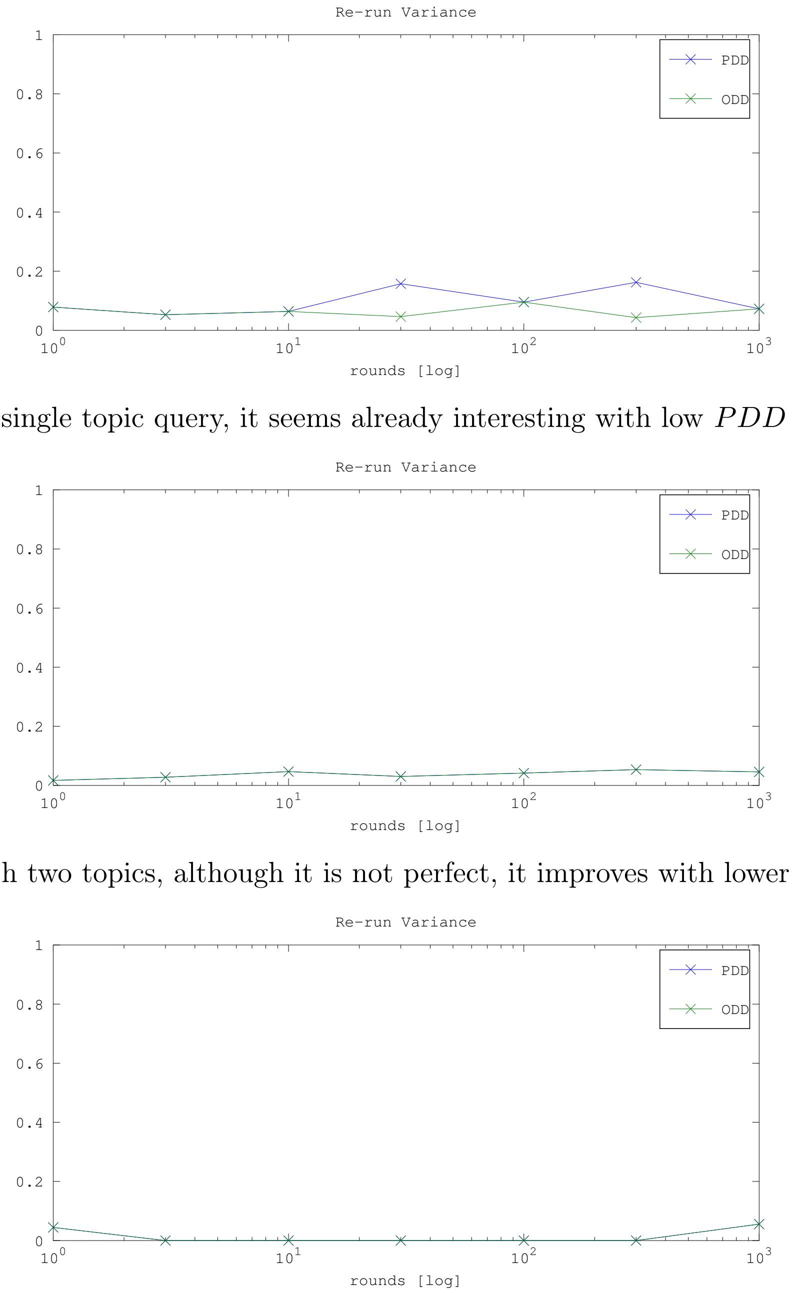


(a) With a single topic query, it seems really interesting with low PDD and ODD.

(b) With two topics, it remains with average values, as compared to other combinations.

(c) With three topics, it has among the worst values.

Figure E.6: Evolution of the re-run variance of the GA technique for cuisine data with (ST1, MT1) on max data.

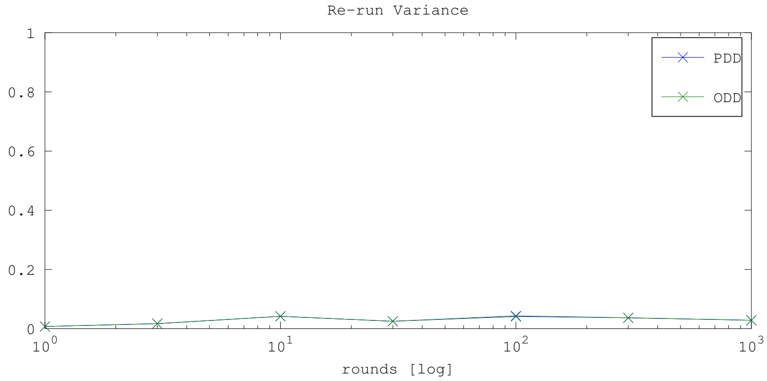


(a) With a single topic query, it seems already interesting with low PDD and ODD.

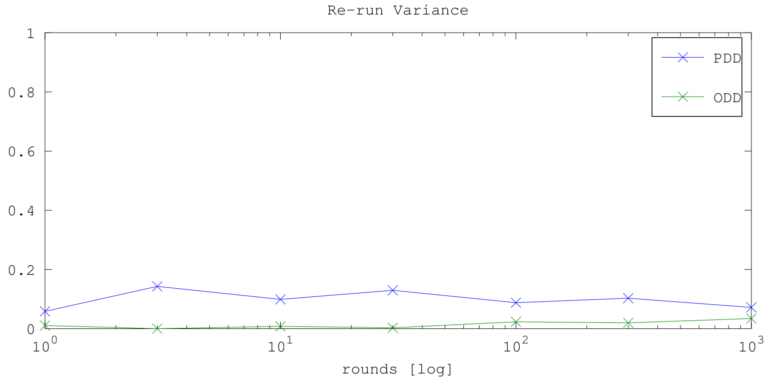
(b) With two topics, although it is not perfect, it improves with lower values.

(c) With three topics, it is almost perfect with $ODD = 0$ almost everywhere.

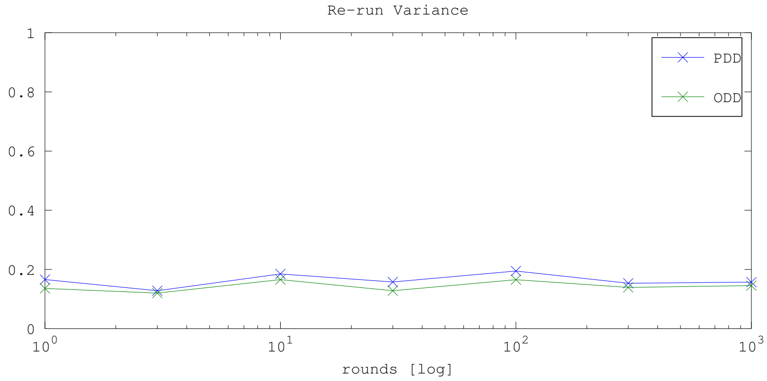
Figure E.7: Evolution of the re-run variance of the GA technique for cuisine data with (ST2, MT3) on max data.



(a) (ST1, MT1) provides almost always totally ordered rankings ($PDD - ODD \approx 0\%$) with a low *Disagreement* ($ODD \approx 2.8\%$).



(b) ST2 provides a really low *Disagreement* if combined with MT1 or MT3 ($ODD \approx 1.4\%$) but more *Indifference*, yet it decreases with more time.



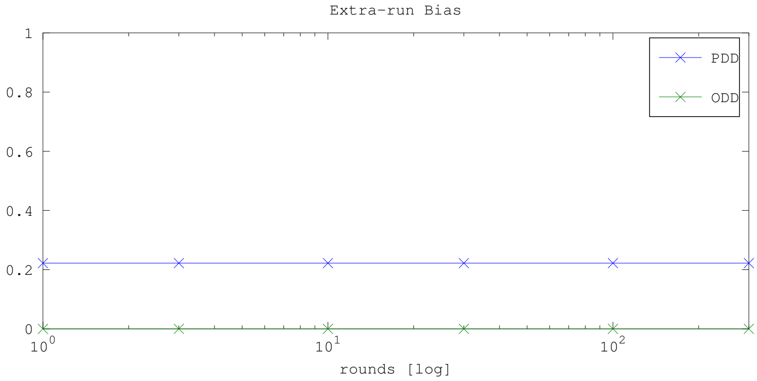
(c) ST3 remains in general with a high *Disagreement* ($ODD \approx 14.3\%$), which makes it uninteresting compared to other cases.

Figure E.8: Evolution of the re-run variance of the GA technique for cuisine data on min data.

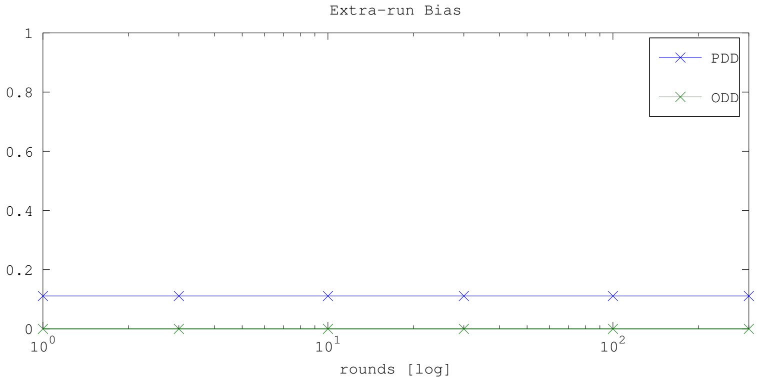
For the extra-run bias, rather than reproducing a complete analysis of the different settings, we focus only on the most interesting combinations found so far: (ST2, MT1) and (ST2, MT3). With max data, Figure E.9 shows that the two have almost equivalent behaviours, which are highly similar to what have been observed with the re-run variance. The highest *Disagreement* occur with single topic queries, decreases then for two topics, until it almost completely disappears for the three topic query. The same observations occur with min data, excepted that it does completely disappear with the richest query for both MT1 and MT3.

From these analyses, a common observation can be made: not only it is hard to establish interesting functions, because we have to go until differentiating the queries to see differences, but the re-run and extra-run variability tend to remain rather constant in time. Rather than a lack of convergence, it might be that a lot of individuals happen to provide the best results, leading not only to obtain the "best" sub-graphs from the start (as a matter of relevance value), but also that these various but equivalent sub-graphs produce different rankings, which would explain why the variability is preserved among the different time-outs.

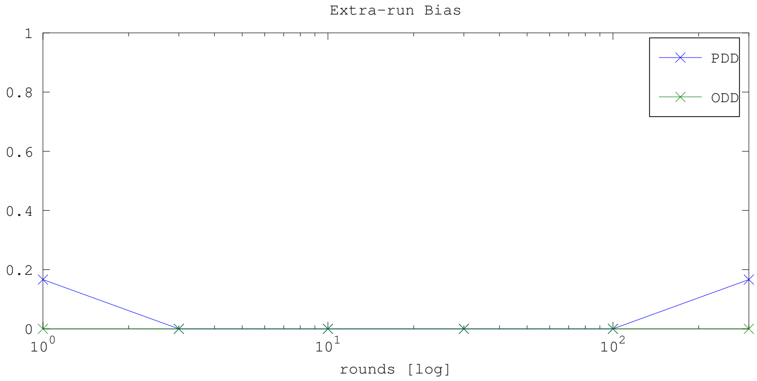
The validation of the assumptions, at the opposite of the variability analysis, is more straightforward and allow us to consider all the functions combinations again. Figure E.10 shows only the results of the min data perspective, which is the most interesting for us because the GA approach aims at focusing on a small part of the whole graph. Assumption 1 (no data) is not checked because no altered dataset has been generated for it. Assumption 2 (no query) is always satisfied, independently of the settings chosen. Assumption 3 (composition), however, shows different tendencies depending on the type-specific function used. With min data, ST1 is rather random while ST3 is mainly compliant and ST2 fully compliant, while with max data ST1 increases with time and ST3 is almost fully compliant too. Assumption 4



(a) Single topic queries show the most *Indifference*, due to a significant amount of *Disagreement* between individual rankings.



(b) With two topic queries, the *Indifference* significantly decreases because of a lot less *Disagreement* at individual levels.

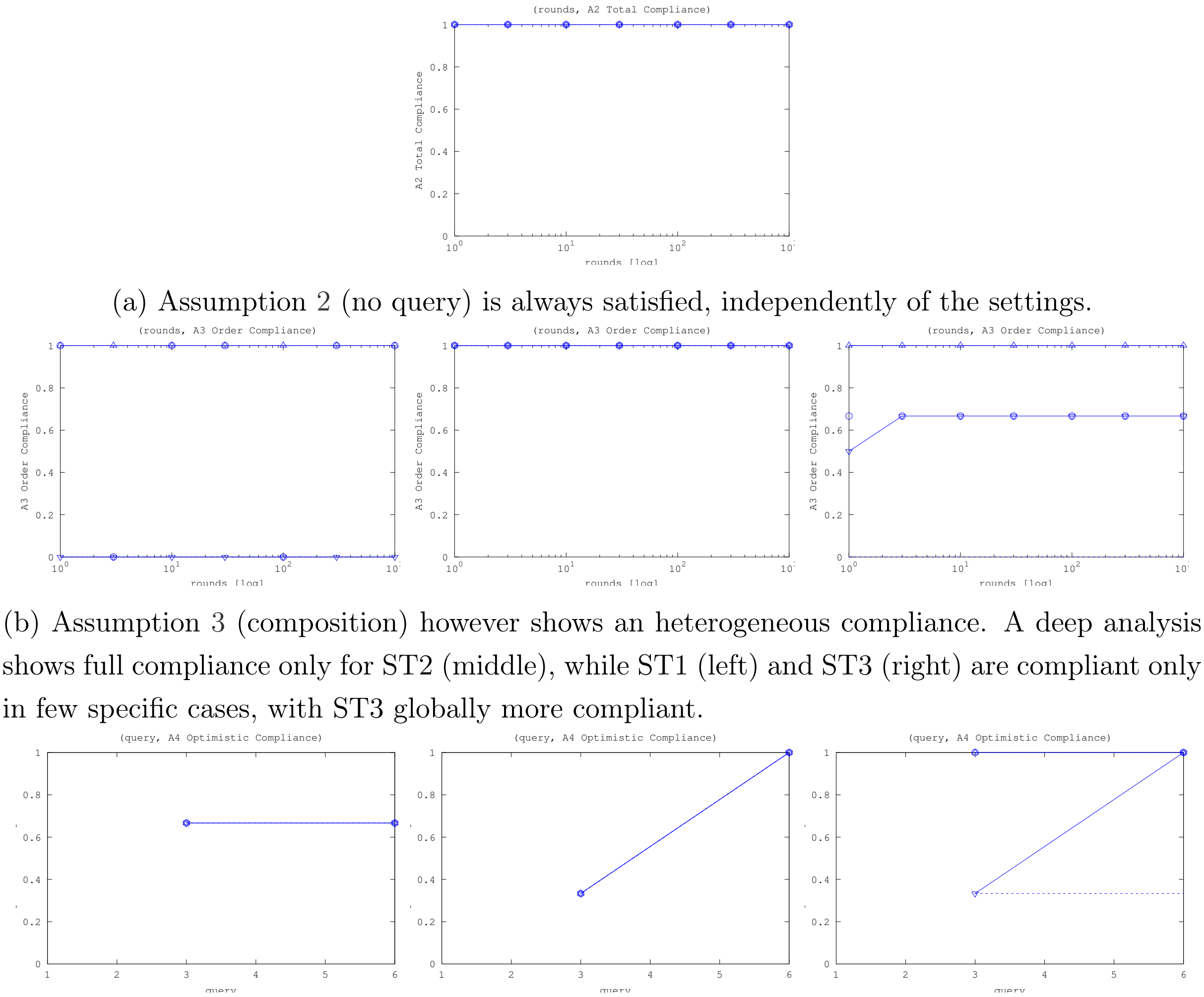


(c) With three topics, it completely disappears with MT1, almost completely with MT3.

Figure E.9: Evolution of the extra-run bias of the GA technique for cuisine data on max data with (ST2, MT1) and (ST2, MT3).

(expected) is the hardest to satisfy, with MT1 which seems to be a requirement to obtain some reasonable compliance levels with min data. In this case, ST1 shows a partial compliance for each query, while ST2 focuses on a single query, and if ST3 appears to be more compliant than non-compliant with min data, max data makes it equivalent to ST2.

Consequently, it appears that although ST2 shows the best results in general, it falls short on Assumption 4, where ST1 appears to be better. Indeed, ST2 provides the same ordered pairs for two queries supposed to provide reversed ones, making it compliant with one but not the other. Only (ST1, MT2) appears to be able to provide correct rankings when it has enough information (it works for max data but not for min data). Further analysis shows that decreasing only the number of topics (from 3 to 2 to 1) tends to decrease the compliance at high time-outs, but it still remains high in value and low time-outs are not affected. The compliance suffers a lot more when decreasing the amount of terms (from 30 to 10 to 3 to 1), although the highest time-out still maintains a good level of compliance in most of the cases. With this, we see that we have on one side a “consistent” approach (ST2) and on the other a “correct” one (ST1 with MT2) but not both.



(a) Assumption 2 (no query) is always satisfied, independently of the settings.

(b) Assumption 3 (composition) however shows an heterogeneous compliance. A deep analysis shows full compliance only for ST2 (middle), while ST1 (left) and ST3 (right) are compliant only in few specific cases, with ST3 globally more compliant.

(c) Assumption 4 (expected) can only be reasonably satisfied with MT1, so we do not consider the others here. ST1 (left) partially complies to each query, ST2 (middle) complies only with one of them, and ST3 (right) shows a better compliance if we focus on the highest time-out.

Figure E.10: Evolution of the assumption compliance of the GA technique for cuisine data on min data. Only the distributions are shown for readability.

Appendix F

Detailed Analysis for XWiki Data

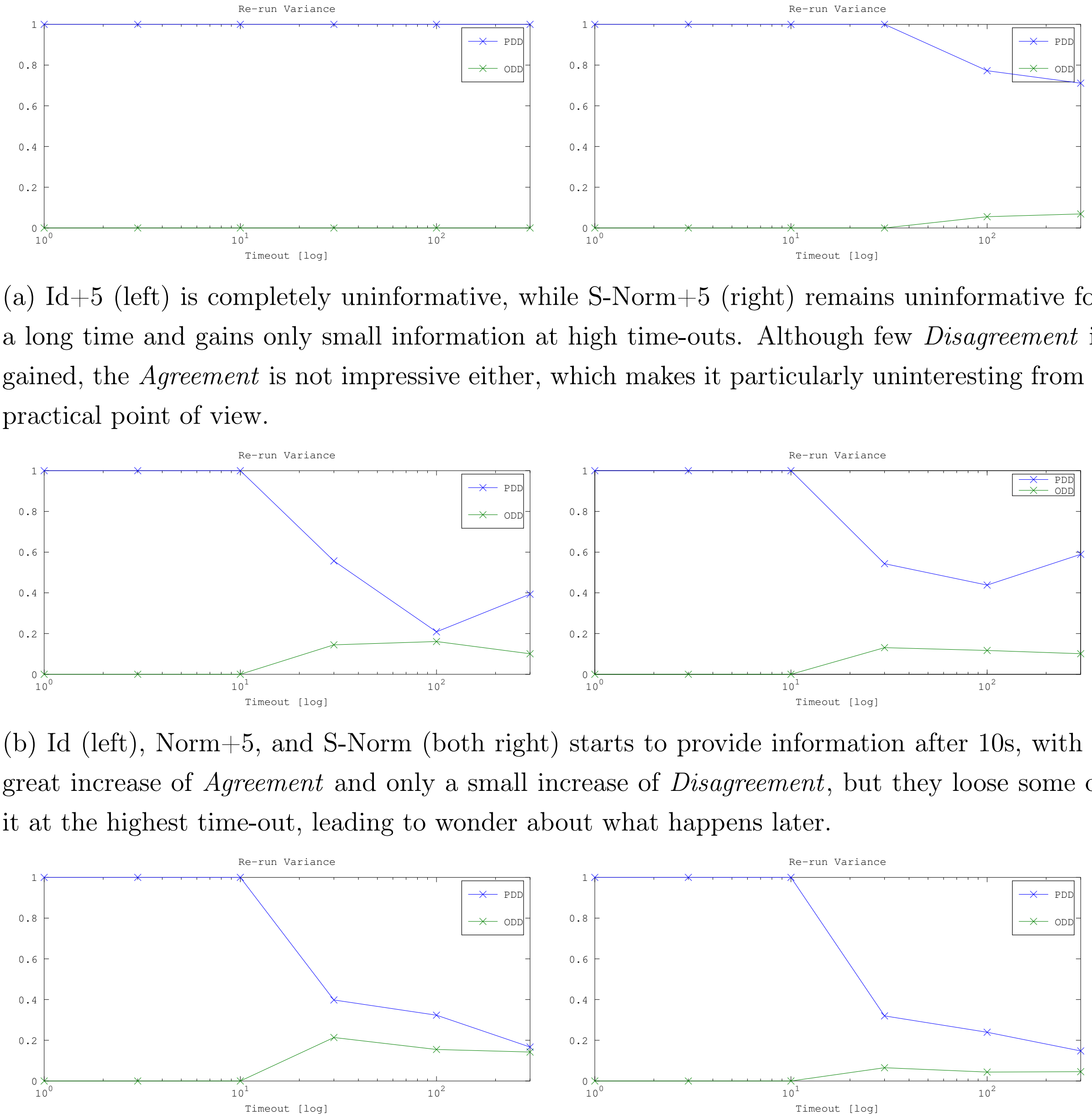
The graph for XWiki being too big to compute exact MNs, we directly analyse the approximative computation, in order to see if it provides sounding results. Then, we analyse the results of the GA approach to see its own ability to provide consistent and correct results.

F.1 Markov Network (approximative)

We first want to know what are the relevant settings, especially the timeouts from which some functions may provide stable results. For this, we ignore the case of empty queries, because they are assumed to provide uninformative rankings, which means that they artificially increase the amount of *Indifference* ($PDD - ODD$) on the graphs.

By analysing the re-run variance, we can see different behaviours. The least interesting one is a broad lack of informativeness: Figure F.1a shows that Id+5 remains completely uninformative and S-Norm+5 remains uninformative for a long time before to gain only a bit of *Agreement* ($1 - PDD = 28.9\%$). If we look at the detailed ODD values to confirm whether the high level of *Indifference* is due to proper *Unordered* pairs or a balanced *Disagreement*, we can see that indeed the *Disagreement* occurs only after 10s and finishes on an increasing slope. We can imagine that more time would lead to further increase. Other functions show more interesting results, as illustrated in Figure F.1b. Id starts to have informative results after 10s with a great increase of *Agreement* until 100s ($1 - PDD = 79.1\%$) and only a small increase of *Disagreement* ($ODD = 16.1\%$) with only few *Indifference* ($PDD - ODD = 4.8\%$). However, further computation leads to a loss of *Agreement* ($1 - PDD = 60.7\%$) which introduces some doubts on the ability for the function to converge, and thus require to see what happens at higher time-outs. Norm+5 and S-Norm have a similar behaviour, but with significantly less gain in *Agreement* (resp. $1 - PDD = 52.7\%$ and 59.7% at 100s). Finally, the remaining functions are the most interesting because of their constant increase in *Agreement*, as shown in Figure F.1c. Similarly to the previous functions, Norm and WoE provide information only after 10s, but shows then a great improvement of *Agreement*. However, the continue this improvement until reaching the highest level at the highest time-out ($1 - PDD = 83.3\%$ for Norm, 85.2% for WoE). Although the *Disagreement* remains high for Norm ($ODD = 14.2\%$ at the end), WoE shows a particularly low level ($ODD = 4.6\%$) which is further confirmed by looking at the detailed ODD values ($ODD \approx 9.2\%$ at the highest time-out), making it the function providing the most stable rankings.

When looking at the extra-run bias, we also observe three types of behaviours which more or less reproduce the observations of the re-run variance. Figure F.2a shows again that Id+5 is totally uninformative while S-Norm+5 gain only some *Agreement* at the highest time-out ($1 - PDD = 27.1\%$). If bias ODD remains null, the detailed ODD values show that some *Disagreement* occurs after 10s, although it remains indeed really low, proving that the *Indifference* is due to proper *Unordered* pairs. The second type of functions, represented in Figure F.2b, concerns the functions showing an increasing *Agreement*, but balanced with an increasing *Disagreement*. In these interesting but arguable functions, we retrieve again Id, Norm+5, and



(a) Id+5 (left) is completely uninformative, while S-Norm+5 (right) remains uninformative for a long time and gains only small information at high time-outs. Although few *Disagreement* is gained, the *Agreement* is not impressive either, which makes it particularly uninteresting from a practical point of view.

(b) Id (left), Norm+5, and S-Norm (both right) starts to provide information after 10s, with a great increase of *Agreement* and only a small increase of *Disagreement*, but they loose some of it at the highest time-out, leading to wonder about what happens later.

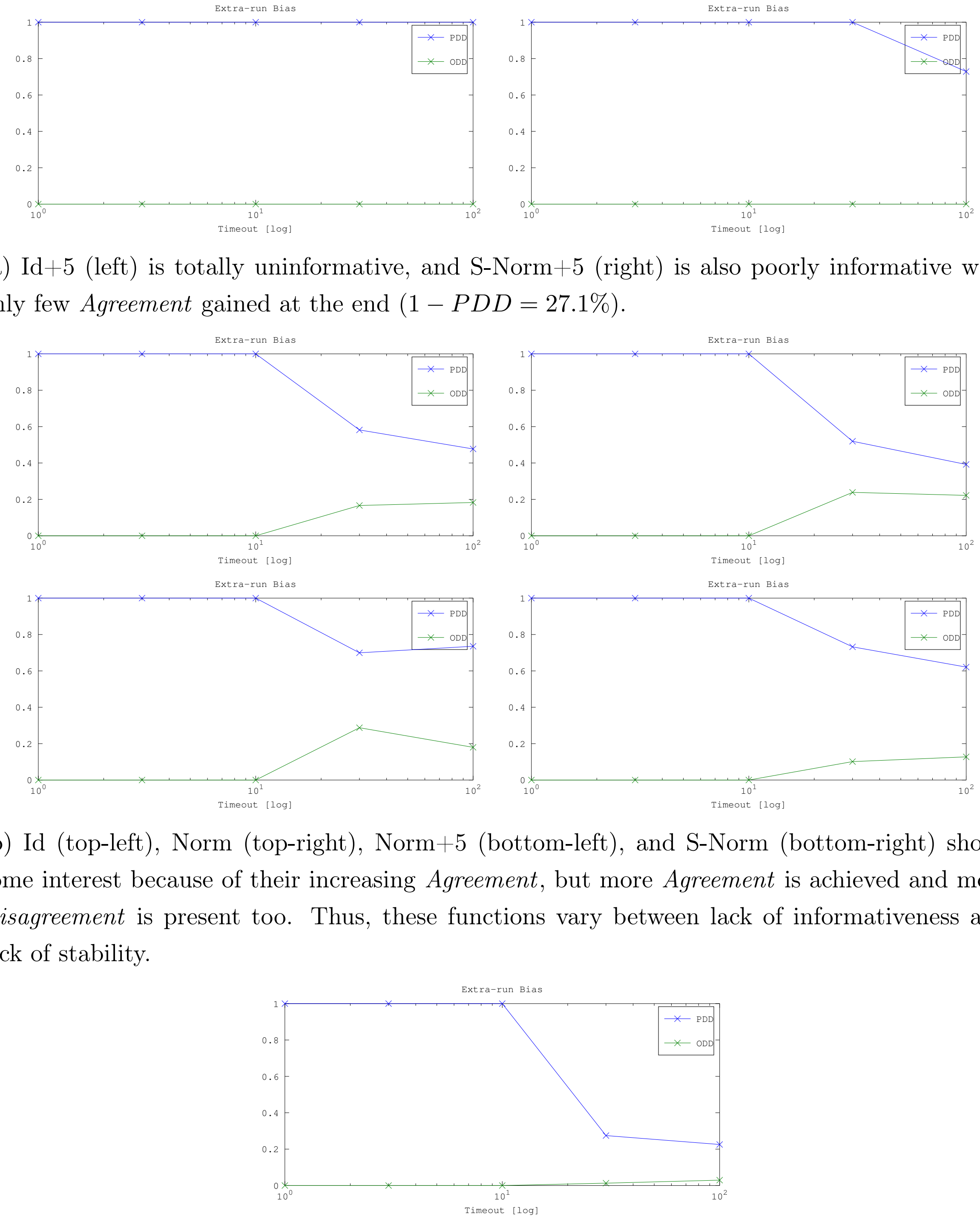
(c) Norm (left) and WoE (right) show the best behaviours by constantly increasing *Agreement*. WoE especially is the most interesting with its low level of *Disagreement* ($ODD \leq 6.5\%$).

Figure F.1: Evolution of the re-run variance of the MN technique for XWiki data with approximative computation.

S-Norm, like for the re-run variance, but we also find Norm, which has among the best re-run variance but still with a high *Disagreement*. We could further separate these functions into two sub-types: Id and Norm are the ones who finishes with the highest *Agreement* (resp. $1 - PDD = 52.3\%$ and 60.8%) but also with the highest *Disagreement* (resp. $ODD = 18.3\%$ and 22.2%), while Norm+5 and S-Norm provide the least informative rankings (resp. $PDD - ODD = 55.5\%$ and 49.3%). Additionally, the fact that the detailed ODD values of the least informative functions remain high in average (resp. 46.9% and 36.8% at the highest time-out) let think that the low informativeness of their centroid is due for a significant part to a balanced *Disagreement* of the individual rankings. Finally, Figure F.2c shows, similarly to the re-run variance, how WoE provides the most stable rankings by finishing on a fairly good level of *Agreement* ($1 - PDD = 77.5\%$) with almost no *Disagreement* ($ODD = 2.9\%$). If the detailed ODD values show that some balanced *Disagreement* might be involved too ($ODD = 9.1\%$ in average at the highest time-out), it remains the most stable function if we do not count the uninformative ones. Moreover, the fact that it shows similar values for the 30s/100s 100s/300s comparisons lets think that no significant change occurs after 30s.

In summary, we saw that WoE offers the best stability, although it is not perfect, with a great amount of *Agreement* between its rankings. Moreover, it seems that it is not required to process the data more than 30s, although the small *Disagreement* still present at the end could justify to run the algorithm several times and take the centroid of the generated rankings.

Now that we can identify the interesting functions in term of stability, we need to evaluate their consistency and correctness through their compliance to our gold standards. Assumption 1 (no data) cannot be checked because we did not generate rankings on an altered version of the graph, and Assumption 3 (composition) cannot be checked either because we have no composed



(a) Id+5 (left) is totally uninformative, and S-Norm+5 (right) is also poorly informative with only few *Agreement* gained at the end ($1 - PDD = 27.1\%$).

(b) Id (top-left), Norm (top-right), Norm+5 (bottom-left), and S-Norm (bottom-right) shows some interest because of their increasing *Agreement*, but more *Agreement* is achieved and more *Disagreement* is present too. Thus, these functions vary between lack of informativeness and lack of stability.

(c) WoE is the most interesting function, reaching high *Agreement* ($1 - PDD = 77.5\%$) with low *Disagreement* ($ODD = 2.9\%$).

Figure F.2: Evolution of the extra-run bias of the MN technique for XWiki data with approximative computation.

query to apply it on. Regarding Assumption 2 (no query), Figure F.3 shows that the informative functions (Id, Norm, Norm+5, S-Norm and WoE) are not compliant at all if we look after 10s, which is the time required to provide informative rankings with non-empty queries. As observed in previous experiments, excepted for WoE we find that the probabilities generated are all close to 0.5, with an average difference between the min/max values of 0.027, so we might consider that they *could* be compliant. Only WoE actually provides probabilities which are far away, thus justifying the orders and the corresponding lack of compliance. Another difference is that WoE is the only one which generates a compliant ranking at the highest time-out, but it is only 1 run over 17, so it could be due to some noise and we do not have enough data to investigate it further. Although not shown in the figure, Id+5 is totally compliant, but it is because of its general lack of informativeness, and S-Norm+5 looses some compliance at high time-outs, when it starts to provide ordered pairs for non-empty queries, so we can expect it to follow the same path than the other functions and become even less compliant with more time. For Assumption 4 (expected), however, we can see different behaviours, as shown in Figure F.4. Most of the functions react similarly to Id, Norm, and S-Norm, which gain in compliance after 10s, when they start to gain in informativeness. But these functions in particular seem to loose some of their compliance at the highest time-out, although the small difference could be considered also as a palier. WoE clearly reaches a palier, and the high stability achieved after 30s ensures that it cannot improve further. Norm+5, however, is one of the least informative, and yet it is able not only to achieve similar levels of compliance, but it shows that it can still increase its compliance with more time. However, the logarithmic scale of the time axis shows that this increase becomes costly. The other functions are the least informative, and their graphs do not provide much information but the minimal compliance achievable because of the few pairs not constrained in

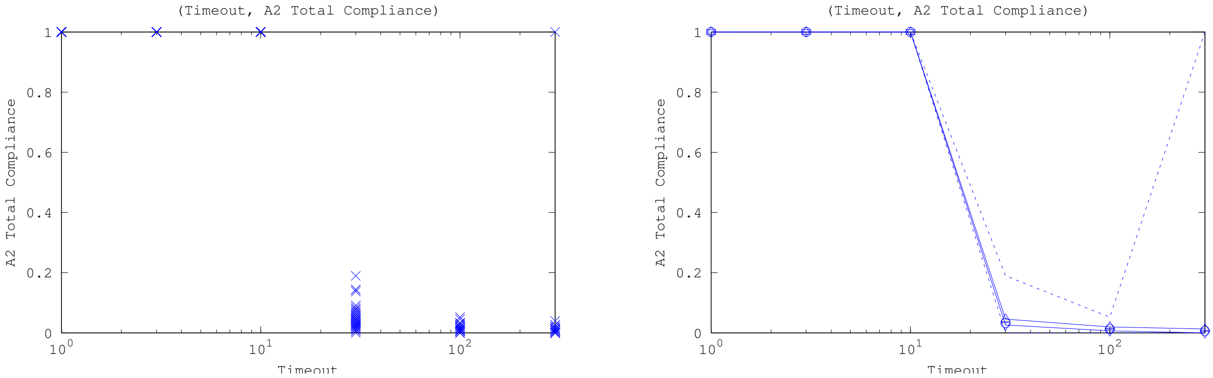


Figure F.3: Evolution of Assumption 2 (no query) compliance of the MN technique for XWiki data with approximative computation. Only the distributions are shown for readability. It is never met by any function if we exclude the ones broadly uninformative (Id+5 and S-Norm+5).

the gold standards. Added to the already low level of compliance of the previous functions, which always remain below 40%, we can only conclude that Assumption 4 is never satisfied with this dataset.

As a summary, it appears that the MN approach is globally unsuited to perform an EF task on this dataset, not only because of the general lack of stability of the generated rankings, but also because they remain fairly low in terms of compliance to our assumptions. Indeed, even if we might solve some compliance issues with Assumption 2 by fixing the approximative equalities, not a single ranking achieves better than 40% of compliance for Assumption 4.

F.2 Genetic Algorithm

This dataset was particularly costly to generate, and yet is far to provide the same amount of rankings than for the other evaluations. In particular, the whole dataset only has an average of 3.59 runs per setting, with 13.0% having 0 or 1 ranking only, which is a part for which we cannot investigate the variability at all. Even with 2 or 3 rankings, we can argue that it is

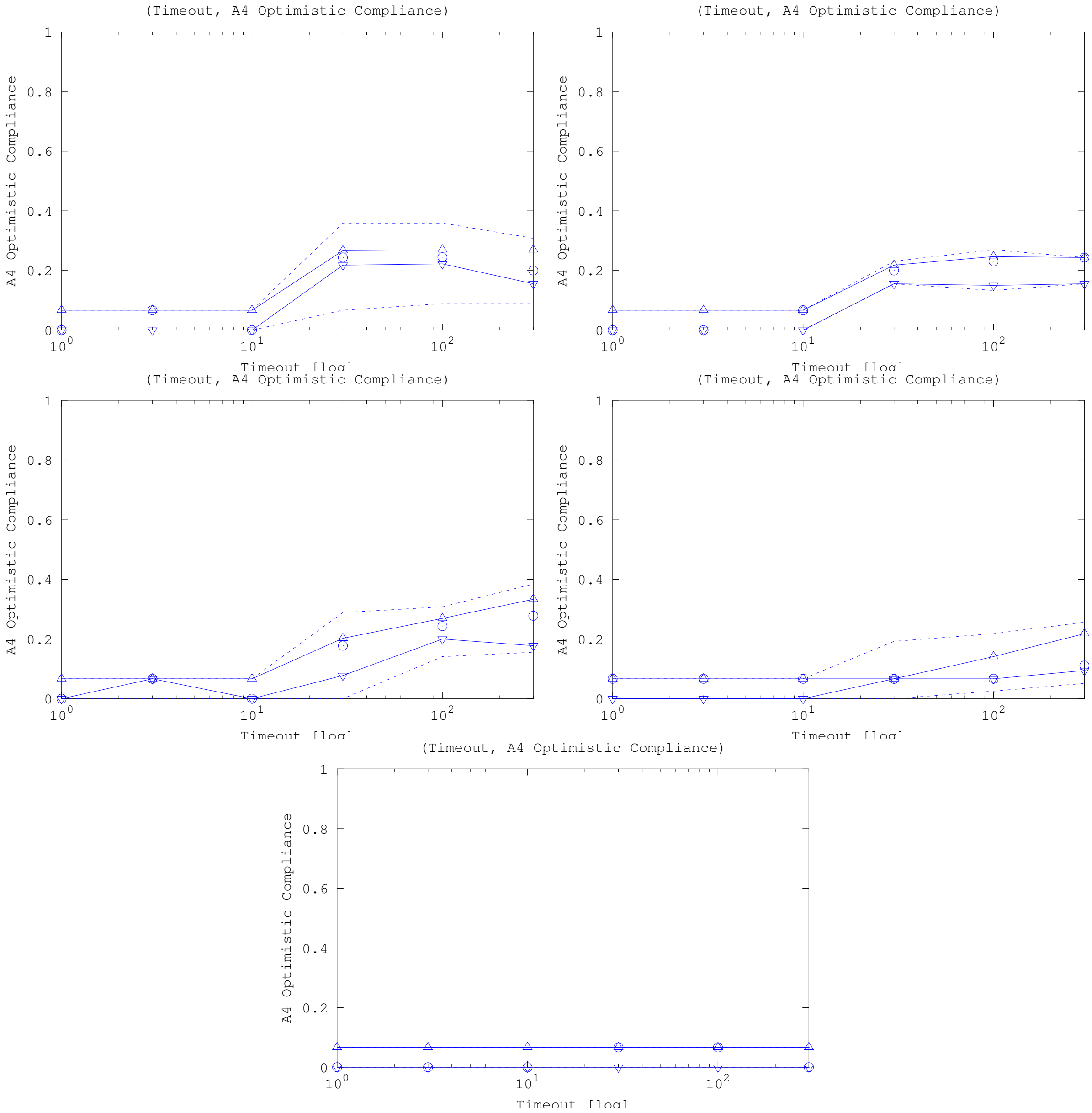


Figure F.4: Evolution of Assumption 4 (expected) compliance of the MN technique for XWiki data with approximative computation. Only the distributions are shown for readability. It shows different behaviours: Id/Norm/S-Norm (top-left) gain in compliance by becoming informative but seem to loose some of it at the highest time-out, WoE (top-right) clearly reaches a palier, Norm+5 (middle-left) seems to increase constantly but the logarithmic scale shows that it becomes costly, S-Norm+5 (middle-right) does not show much because of its late gain of informativeness, and Id+5 (bottom) shows to which extent the lack of orders for half of the gold standards provide some free compliance.

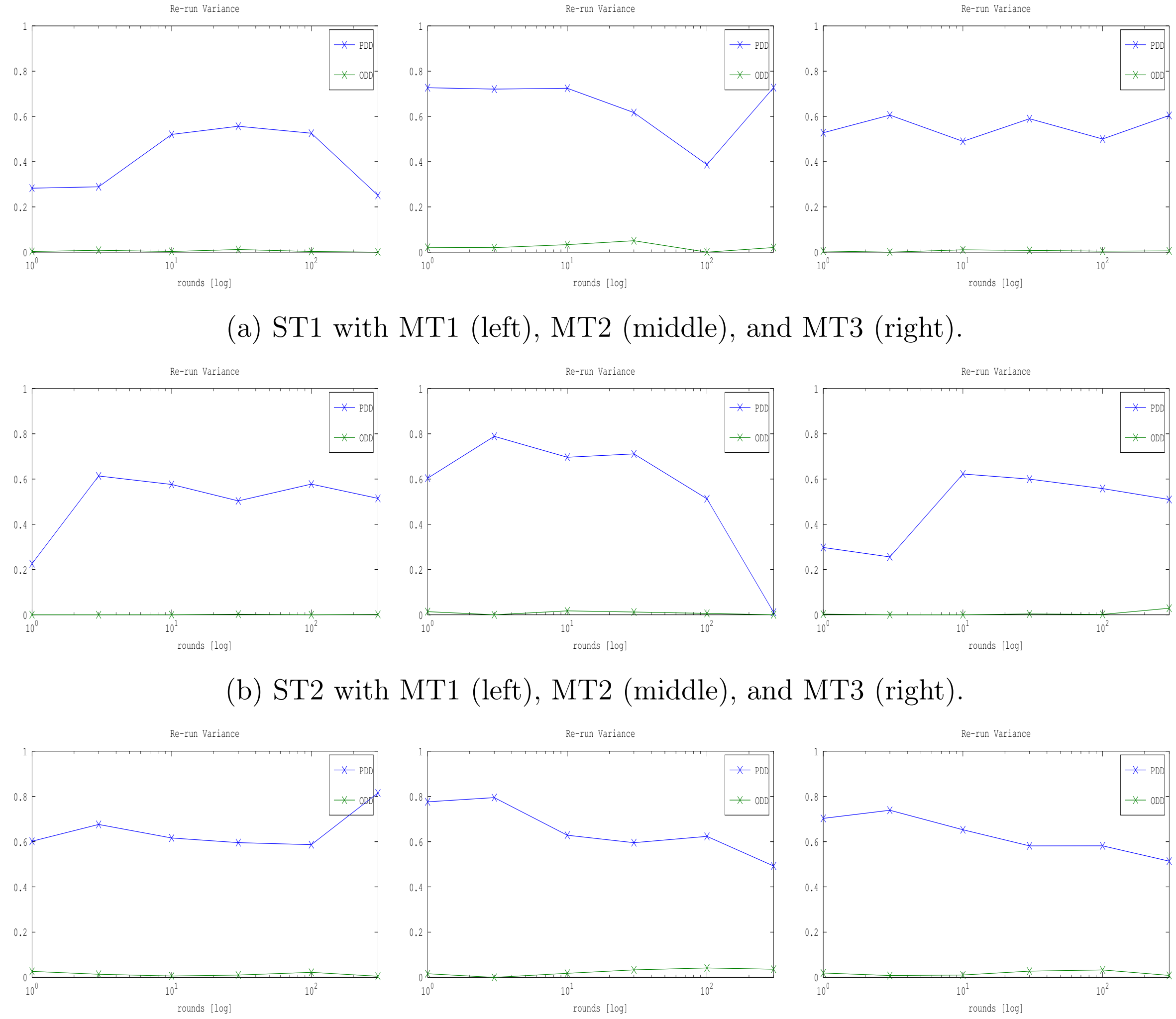
too small to ensure the representativeness of our dataset, but then we reach 51.9% of it. Rather than throwing it away, we prefer to analyse what we can while minimizing the threats.

One threat to minimize in priority is the number of settings having less than 2 rankings, because they are the ones from which we cannot, by definition, have a variability analysis. Because our analysis should cover all the rounds for the variability analysis, all the queries for the assumptions, and all the function combinations because this is what we are interested in, we filter only the other parameters, thus the number of stakeholder nodes, topic nodes, and term nodes. Moreover, because our gold standards are based on 13 and 10 stakeholders and we want to have rankings that we can properly validate, the stakeholders nodes are fixed to 10, which is the maximal value. The remaining settings are the number of nodes for topics (1, 3, or 10) and terms (1, 3, 10, or 30). The most interesting combination is 10 topics and 1 term (10/1), which has only 9.8% of settings having 0 or 1 rankings, but this combination is far to be realistic, because it selects a lot of topics and only one term, while we would expect to see the reverse. The next most interesting combination is 3 topics and 10 terms (3/10), which seems way more realistic to us and has 10.5% of settings of 0 or 1 rankings, which is highly similar to the previous one.

Another threat to minimize is the representativeness of the dataset, which means that each setting should have enough rankings to give a good idea of what to expect from it. This threat is more a matter of preference than filtering, because the overall dataset is poor anyway. To minimize this threat, we need to maximize the number of rankings in each setting, what can be measured in several ways. If we choose the minimum, it is always 0 or 1, so it makes no sense, and the maximum only covers the few best settings, which is not interesting. If we choose the first quartile, it is always 2, and the third quartile almost always 5 (4 for one case), so it makes no sense either.

The median identify few best cases with 4 rankings, but none of the cases identified before (10/1 and 3/10) are concerned because they both have the minimal value of 3 rankings. The average goes from 3.46 to 3.83, which offers only small differences, so even if the case 10/1 is higher than 3/10 (resp. 3.69 and 3.46), we consider that the realism of 3/10 takes priority, even if it has the worst average. Consequently, for our analysis of the GA approach, we will focus on the settings having 10 stakeholders, 3 topics, and 10 terms.

By analysing the re-run variance in Figure F.5, we can see various behaviours. Some cases show a decrease of *Agreement*, like (ST2, MT1) or (ST2, MT3), while others increase, like (ST2, MT2) and (ST3, MT2). If we accept the idea that the variety of rankings first increase before to reach a consensus, then we might be interested in (ST1, MT1) and (ST2, MT2), which are also the only ones offering more than 50% of *Agreement* at the highest time-out. More generally, there is a common tendency to have between a third and a half of *Agreement* between the rankings (43.7% in average) and almost no *Disagreement* (1.2% in average). The huge resulting *Indifference* (55.1% in average) is confirmed to be mainly due to *Unordered* pairs in individual rankings by looking at the detailed ODD values.



(a) ST1 with MT1 (left), MT2 (middle), and MT3 (right).

(b) ST2 with MT1 (left), MT2 (middle), and MT3 (right).

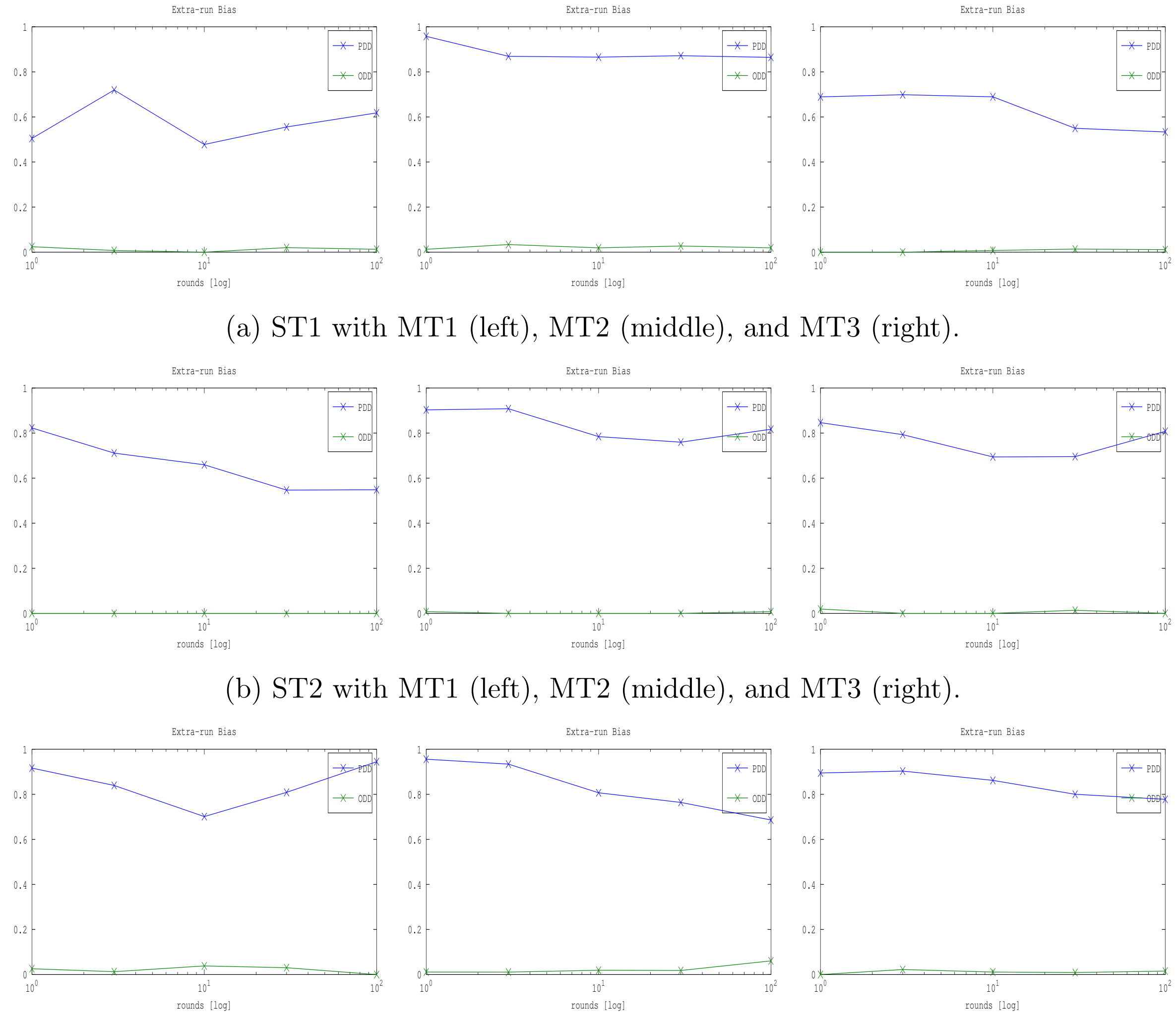
(c) ST3 with MT1 (left), MT2 (middle), and MT3 (right).

Figure F.5: Evolution of the re-run variance of the GA technique for XWiki data.

By looking in details at the rankings, we can actually see that they tend to agree, but also to rank different stakeholders, which is why there is a huge *Indifference* area. Indeed, because each ranking provides 10 stakeholders, they provide $\frac{10\times9}{2} = 45$ ordered pairs (or less if it is partially ordered). But because there is 18 stakeholders in total, two rankings can have in common from 2 to 10 stakeholders, and only the ordered pairs in common will be considered. The other pairs lead to an *Indifference* because at least one of the two rankings does not provide it. If we consider the average of 6 stakeholders in common, then only $\frac{6\times5}{2} = 15$ ordered pairs are considered (even less if they are partially ordered), which leads to an *Agreement* of $\frac{15}{45} = 33.3\%$ if they all agree. What we observe here is a bit better: almost no *Disagreement* is observed with detailed ODD, and the remaining pairs –as long as they are provided by the two rankings– are in *Agreement*, leading to our average of 43.7%, which is a bit above the random average.

In other words, the *Agreement* ratio is naturally decreased because of the incomplete set of stakeholders considered, but the rankings are, actually, broadly agreeing on how to order them. What is interesting then is to identify the functions able to provide a higher *Agreement* than this average, because in order to do so, they need to focus on a subset of stakeholders, hopefully the most relevant ones. In other words, (ST1, MT1) and (ST2, MT2) appear to be the most interesting because they finishes on an *Agreement* significantly higher than the average (resp. 74.9% and 98.9%).

When we compare consecutive time-outs through the extra-run bias, as shown in Figure F.6, we can see again a great variety of behaviours. Once again, the *Disagreement* remains close to zero, although by looking at the detailed ODD values we can see that ST3 provides the worst distances (independently of the MTx) and that MT2 tend to make it increase (independently of the STx). If we focus on the *Agreement*, we can see some really poor combinations, like (ST1, MT2) which remains low, or (ST3, MT1) which gains a bit before to loose it all. Although several provides an increasing slope, the most interesting appear to be (ST1, MT3) and (ST2, MT1), which are the only ones able to finish with more than 40% of *Agreement*, which is still low.



(a) ST1 with MT1 (left), MT2 (middle), and MT3 (right).

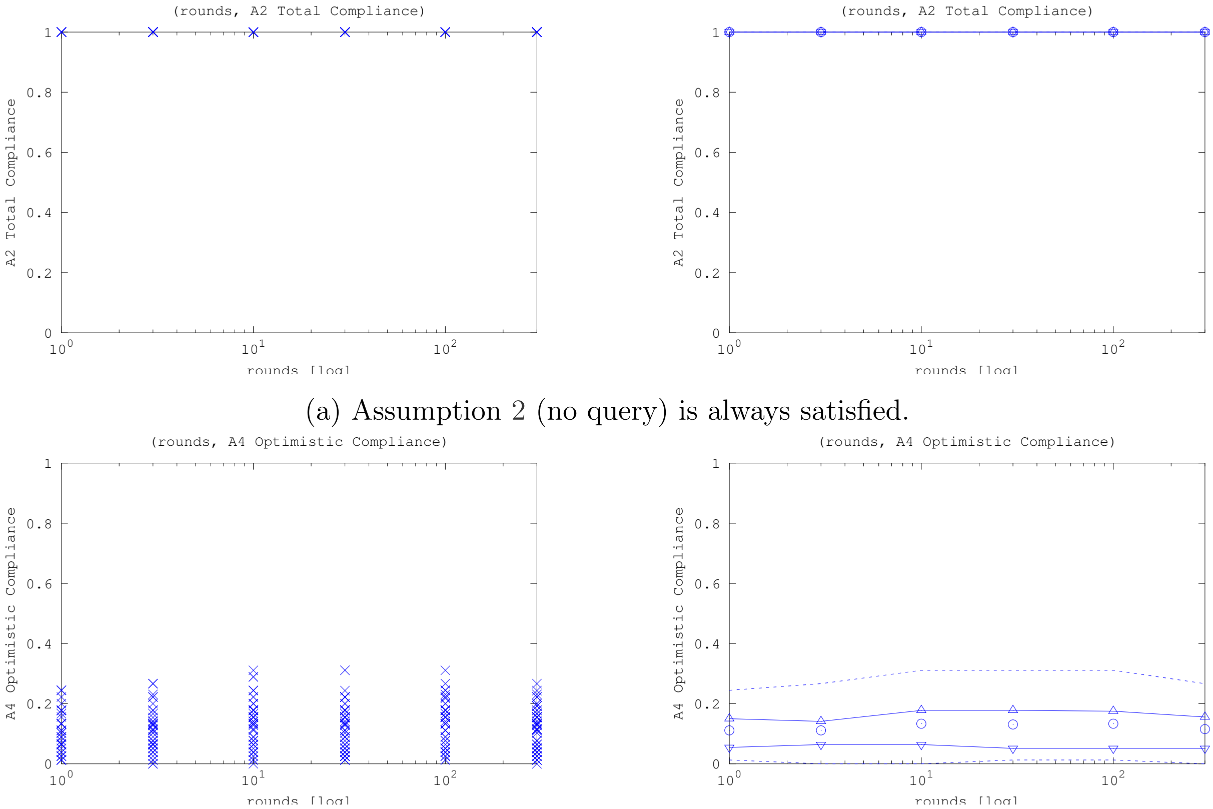
(b) ST2 with MT1 (left), MT2 (middle), and MT3 (right).

(c) ST3 with MT1 (left), MT2 (middle), and MT3 (right).

Figure F.6: Evolution of the extra-run bias of the GA technique for XWiki data.

Consequently, we can identify some interesting functions, but we cannot assess a proper convergence: although the rankings generated globally agree, they maintain some variability on the stakeholders they rank, which means that the recommendations can vary greatly regarding who is recommended. Maybe this issue can be solved with more time, or maybe we need to revise the fitness function of the GA. However, even if we achieve a greater stability, we first of all need to obtain correct rankings.

For this dataset, no altered graph has been produced for checking Assumption 1 (no data), and no composed query has been considered for checking Assumption 3 (composition). By focusing on Assumption 2 (no query), as observed in the previous experiments, full compliance is achieved. However, Assumption 4 (expected) never reaches a reasonable level of compliance. Like for the *Agreement*, there is some effects due to the limited amount of stakeholders considered in the rankings, whether they are generated GA rankings or gold standard rankings. The GA rankings have 10 stakeholders, so $\frac{10\times9}{2} = 45$ pairs, while one gold standard has also 10 stakeholders, so also 45 pairs, and the other has 13 stakeholders, so $\frac{13\times12}{2} = 78$ pairs. In other words, the maximal compliance achievable for the smallest gold standard is $\frac{45}{45} = 100\%$, while for the biggest gold standard it is $\frac{45}{78} = 57.7\%$. These limits should be considered when evaluating the correctness of the GA rankings, but as we see none of them go beyond 30% of compliance, which is only half of 57.7% and a third of 100%. Thus, without forgetting the poor representativeness of our dataset, reasonable compliance seems to be far away, leading to consider the current GA technique as unsuited for supporting an EF task on the XWiki dataset.



(a) Assumption 2 (no query) is always satisfied.

(b) Assumption 4 (expected) however is only poorly satisfied. Independently of the functions, we achieve at most 31.1% of compliance.

Figure F.7: Evolution of the assumption compliance of the GA technique for XWiki data.