From databases to artificial intelligence

Zoltan Miklos

To cite this version:

Zoltan Miklos. From databases to artificial intelligence. Databases [cs.DB]. Université de Rennes 1 [UR1], 2020. tel-02501285

HAL Id: tel-02501285

https://hal.inria.fr/tel-02501285

Submitted on 9 Mar 2020

From databases to artificial intelligence

Zoltán Miklós

Habilitation thesis
University of Rennes 1

Defense date: 6th March 2020

Sihem Amer-Yahia	researcher at CNRS, France (referee)
Angela Bonifati	professor at University of Lyon 1, France (referee)
Philippe Cudré-Mauroux	professor at University of Fribourg, Switzerland (referee)
Bernd Amann	professor at Pierre and Marie Curie University, France (examiner)
François Taïani	professor at University of Rennes 1, France (examiner)

Abstract

This habilitation thesis synthesizes our research efforts in the area of relational databases and artificial intelligence. In particular, we present our work on various data matching problems, where we need to establish connections between different pieces of information, such that these correspondences reflect our human understanding of the relation among them.

Entity resolution aims to identify entity references (person or company names, geographic locations) such that they refer to the same real word entity. We summarize here our work on this problem in the context of Web documents. Our methods rely on supervised machine learning techniques.

We discuss our work on database schema matching. Our work addressed a specific setting of this problem, where we need to match a set of schemas based on a network of their pairwise interactions. Even if available schema matching tools can obtain a set of good quality attribute correspondences, if we would like to use them for data integration, we need to eliminate the remaining errors. This phase still requires the involvement of human experts. We model this post-matching phase and we propose new techniques to reduce the necessary human efforts and to guide the work of the experts. Our methods rely on (probabilistic) reasoning methods that exploit the relevant consistency constraints.

Crowdsourcing is a model where one can request a service that is realized by human workers form from a large crowd of users, in exchange for a small payment. Crowdsourcing platforms enable human-in-the-loop algorithms and they were also used to construct large labelled datasets. A central issue in this setting is the quality of the obtained results, in particular in the context of knowledge-intensive crowdsourcing, where workers need specific skills to complete the requested task. Affecting the tasks to workers who do not have these skills could lead to poor quality results. We propose new methods which can improve the expected quality of the results through the use of simple forms of reasoning on human skills.

We also discuss our perspectives for future research directions.

Contents

Acknowledgements

I would like to thank my referees Sihem Amer-Yahia, Angela Bonifati, Philippe Cudré-Mauroux for their time and efforts. I also would like to thank the further members of my HDR jury Bernd Amann and François Taïani.

I would like to thank my PhD supervisor at University of Oxford, Gerog Gottlob, who guided my PhD work. He continues to encourage me, even many years after graduation. I would like to thank Karl Aberer, who hosted me as a postdoctoral researcher in his lab at EPFL and who also continues to encourage me many years after leaving his lab. The postdoctoral years opened for me completely new perspectives. Karl also involved me in research supervision that was a very valuable experience. I also would like to thank David Gross-Amblard and Arnaud Martin for their constant support and for the number of suggestions on this HDR thesis. I would like to thank my students, in particular, the PhD students whom co-supervised or worked with closely: Surender Reddy Yerva, Hung Quoc Viet Nguyen, Nguyen Thranh Tam, Panagiotis Mavridis. I also would like to thank my current PhD students: Ian Jeantet, Rituraj Singh, Maria Massri, François Mentec. I also would like to thank Olivier Ridoux and Patrice Quinton for the interesting discussions and for their valuable comments on this HDR thesis.

I also would like to thank my parents, who constantly supported me. A special thank to my wife Erika and to my children David and Mélina.

Chapter 1

Introduction

1.1 Humans, computers, databases and artificial intelligence

1.1.1 Computers and humans: changing roles

The term *computer* meant in the early 17th century usage a profession: a person who had to perform mathematical calculations[1]. This occupation was known even until the mid 20th century, until electronic computers become available. We can even find references to this metier in the works of Alan Turing [123]: "The human computer is supposed to be following fixed rules; he has no authority to deviate from them in any detail." For difficult computations, the same task was performed in parallel by multiple teams, to assure correctness [53]. The arrival of electronic computers has of course completely eliminated this profession and today we can realize calculations with the help of our modern computers at a scale that would be impossible through human computers. The frequency of errors has also radically improved: humans make accidental errors in computations and these are completely eliminated through the use of electronic devices.

In fact, humans do not make accidental errors in mathematical computations only, but in all the activities they do [59], [103], [109]: (software) engineering, writing text following the orthographic rules, decision making, playing the piano or driving a car. According to the Latin saying "Errare humanum est, perseverare autem diabolicum." that is "To err is human but to persist in error is diabolical", suggesting that we should avoid errors, even if we cannot avoid them completely. This prevalent presence of error in human activities was not only observed by ancient Romans, but this is also supported by modern neuroscience [7][2]. As one could replace human computers through machines, the quest has started in various other domains, whether we could build machines to realize the same tasks as humans, but without the human errors.

[1] `https://en.wikipedia.org/wiki/Computer_(job_description)`

[2] We tried to avoid the orthographic and other errors in this HDR thesis, as much as we could. We insist on the omnipresence of errors in human activities as an argument and not as an apology for the eventual remaining errors in this thesis.

1.1.2 Building intelligent machines, but without the human errors

Computing devices are useful to execute mathematical calculations, but already the key figures of early computer science such as Alain Turing or John von Neumann identified that their devices and models could do much more than simple arithmetic and they made important first steps towards constructing "intelligent" computers[3]. Despite the earlier efforts, one attributes the beginning of artificial intelligence research to the Dartmouth college summer research project that took place in 1956. Participants aimed to simulate every aspect of human intelligence through machines. The summer workshop could of course not achieve this ambitious goal, but a number of methods that have been proposed since these early days turned out to be useful in various ways [113]. These include the symbolic methods for reasoning and problem solving, their probabilistic versions, and other forms of uncertainty management, as well as statistical machine learning based techniques or methods based on artificial neural networks and deep learning [50], but also a number of other techniques. High profile achievements of artificial intelligence research have demonstrated that computers can go beyond human capabilities in specific areas (1997: IBM Deep Blue defeated Garry Kasparov, the word chess champion, 2018: google's DeepMind defeats human go champion, etc.). Artificial intelligence methods have completely changed certain areas such as natural language processing, computer vision, or others. These artificial intelligence methods are increasingly used in practically all domains of our life and this trend is likely to continue [56], [77].

While the goal of artificial intelligence research is to build intelligent machines, it is not very clear still today what intelligence is. There are a number of definitions and approaches that were considered, but none of the definitions is widely accepted and each approach has shortcomings. For example, Gardner [46] proposed the theory of multiple intelligences, that is a widely-debated concept. The most widely used textbook on artificial intelligence by Russell and Norvig [113] reviews different directions: 1) think like a human, 2) act like a human, 3) think rationally, 4) act rationally. It is however not clear how humans think and most likely, even if we build intelligent machines, they will not think as humans. The approach "act like a human" is the approach proposed by Turing [123], that is referred often as a Turing test. The test played an important role over the years, but it has also a number of problems [19]. Rationality is an appealing concept, as it can be translated to a mathematical model and utilities, however psychologists [5] and cognitive scientists [93] argue that humans do not act or think rationally. There is no precise definition, based on which one could decide which tasks or achievements one should consider as efforts in artificial intelligence.

Realizing arithmetic computations is hardly considered as a task in artificial intelligence as of 2020, even if it certainly requires some forms of intelligence and even if humans need specialized education and training for realizing this task[4]. AI effect [91] is a situation where we do not consider a task any more to require intelligence, as we understand how to do it. From

[3] Ada Lovelace (1815-1852) also realized that her Analytical Engine is suitable for a wide range of tasks, including music composition, as we are able to formulate the basic rules of harmony and composition as an abstract set of operations.

[4] For example, Samuel Pepys FRS (1633-1703) reports about his difficulties learning multiplcation in his diary `http://www.pepys.info/1662/1662jul.html`.

this perspective, artificial intelligence is essentially whatever hasn't been done yet[5]. We can list a number of tasks that we cannot realize with computers alone. These include commonsense reasoning, autonomous driving or software development or natural language translation (from source language to target language). Even in these areas there are a number of methods and tools that can complete the task but the quality is still far from perfect (as of 2020). For example, in the context of automatic language translation, the output of automated translation tools requires human proof reading, if one would like to use the output text in a professional context.

With the help of the historical perspective of this introduction we intended to point out that the perimeters of artificial intelligence research are not very clear. The problems we discuss in this habilitation thesis also require human intervention and our works aim to reduce the need of human involvement or to develop tools that can complement the human efforts.

1.1.3 Managing data in an increasingly interconnected world

Database systems today are considered basic and often the most robust components of information systems. If we take a similar historical perspective that we followed in this introduction, we can identify very similar types of motivations for the conception and realization of database systems, as we had for the computer or intelligent machines. Without database systems, developers had to deal with the management of their data for each application separately. This was of course an error-prone part of the code, as developers had to deal with both the physical access and logical structure of their data. Edgar F. Codd, the inventor of the relational database model emphasizes in his Turing award lecture [23] the productivity gains we could obtain through the separation of physical and logical representation of data[6]. The model of Codd enabled to realize database systems, that eliminated the accidental errors of data access that application developers accidentally added. The real productivity gain could only be realized once the database management systems became available. These systems implement the relational model, in very efficient ways. Many of these efficient implementations are based on the ideas of Michael Stonebraker, who is the recipient of the 2014 Turing award. Database systems have completely changed how we store and manage data.

The clean logical structure of database schemas and the access to data through well-defined queries -in the presence of transactional and concurrent access services of database systems- enables also to use the same database from multiple applications. Based on the database schema, application developers can design their specific queries, corresponding to the application requirements. There is however another use case, where one would like to integrate data from multiple databases. Recent technological developments result in an increasingly interconnected world [77] and in this way we face more and more often this situation where we need to integrate data from multiple sources. These might not only be relational databases, but also data in

[5] The original quote of Lary Tesler was more precisely: “Intelligence is whatever machines haven't done yet” http://www.nomodes.com/Larry_Tesler_Consulting/Adages_and_Coinages.html

[6] Software developers certainly improved their productivity through database systems. It is not clear whether the arrival of electronic computers resulted a productivity gain in the economic sense, that is known as the productivity paradox [14]. We note here that the productivity paradox (of computers) is a highly debated issue among economists.

various formats, including semi-structured or even unstructured data. If these sources of data were developed independently, we need to understand what is the relation between the different models of the data. Establishing the connections between different attributes of the models and also between the constants of different data sources is challenging and requires human involvement. As this is the case, researchers have proposed methods and tools to realize these tasks automatically or support the work of experts who are involved in these efforts.

1.2 Research questions and methods

The work that we synthesize in this habilitation thesis focuses on various data matching problems. In these problems we need to establish connections between various elements of the involved data, based on our human understanding. The specific goals for establishing these connections are different in these problems, but in all cases this should reflect our human understanding of the situation. In particular, we address the following problems:

- *Entity resolution for Web documents.* One would like to identify the entities (such as persons, companies, geographic locations, etc.) in Web documents that refer to the same real-world entity. For example, determine whether the term *Paris* refers to the capital city of France, to a small city with the same name in Texas, in United States, or to a first name of a person.

- *Schema matching for a network of schemas.* Schema matching is process of establishing connections between the attributes of different database schemas. We worked on this problem in a specific setting where one would like to match a network of schemas, in a pairwise fashion.

- *Worker and task assignment for knowledge-intensive crowdsourcing.* Crowdsourcing platforms enable to realize tasks through a crowd workers. In this problem the goal is to match workers to tasks based on the competences, which are listed in task requests and in user profiles.

The research challenges in these settings are specific to the particular problem, but the common ultimate goal in these problems is to realize these tasks without human intervention. As we are far from this goal, more pragmatically researchers try to develop methods that can support and complement the work of human experts who are involved in these tasks.

For the entity resolution problem we rely on supervised machine learning techniques. In our methods we try to compare the particular available pieces of information in the documents and to develop classification methods that can predict the correct entity reference. Unfortunately, the Web documents contain only a partial set of information. For example, if we would like to understand whether a page that contains the word “Orange” is related to the company Orange, then the presence of the company’s URL is a strong evidence for the connection, while if the URL is not present, a classifier that is based on the similarity of the URLs would perform poorly. In our work we tried to design a suitable way of combining classifiers in this setting such that we take into account the absence of specific features in the documents.

For database schema matching one would like to establish a connection between the schema attributes, based on their meaning [55]. Understanding the nature of the semantics of terms in a natural language is a central question in a number of fields, including philosophy [128], linguistics [100], cognitive science [83]. Modelling the semantics of natural language is also highly relevant for computer science, in particular for natural language processing [37] and information retrieval [87]. Word embeddings (word2vec [97], BERT [32], ELMO [97], GPT-2 [107]) that are essentially vector representations of the word semantics [74], are also essential for artificial intelligence. Understanding the terms in the schemas as words in a natural language can guide the experts to understand the intended meaning of the schema attributes. To avoid to deal with the semantics of terms, schema designers can rely on industry-backed standards, such as `http://schema.org/`. Semantic web technologies [4], [58] can also be very useful, in particular if one can link the schema attributes to an ontology, expressed for example in the OWL[7] ontology language. Such formal representations are not always available or easy to obtain. In our work we assumed that no such representation is available and we need to establish the relevant connections.

While schema matching tools can help to obtain a set of correspondences, but clearly these methods cannot achieve perfect results. In the case of database schema matching, obtaining a relatively high accuracy matching might not be sufficient and one needs a completely perfect matching for the purposes of data integration. This post-matching phase is often costly as it involves human experts. We model this phase and based on our model, we would like to reduce the necessary human efforts and guide the work human experts. Our methods rely on a range of techniques, including reasoning using Answer Set Programming (ASP), probabilistic reasoning.

Another line of work that we discuss in this habilitation thesis is the problem of matching workers and task on crowdsourcing platforms. Such platforms enable people to request specific tasks that are then executed by human workers, from a large crowd of participants, usually in exchange of a small payment. They offer a specific way to exploit human intelligence. They also played a crucial role in the recent explosion of artificial intelligence: as the state-of-the-art algorithms are supervised, they require high amounts of labelled data to work well. ImageNet [31], for example is a large [8] labelled image dataset that was crucial to demonstrate the efficiency of convolutional neural networks. The basic problem that one faces when using data from these platforms is that humans tend to make errors (accidentally or deliberately). Addressing this problem is even more important for knowledge-intensive crowdsourcing, where workers need to have specific skills to complete the tasks. If one assigns a task to a person who does not have the necessary competences is likely to lead low quality results. We try to exploit a hierarchy of skills to realize basic forms of reasoning about skills to better affect the tasks to workers, and in this way to improve the expected quality of the obtained data.

1.3 Own contributions

We summarize here our contributions that we discuss in this habilitation thesis.

[7] `https://www.w3.org/OWL/`

[8] ImageNet used 120 categories from WordNet taxonomy to categorise objects, present in the collection of ca. 14 million images, constructed through crowdsourcing.

- Entity resolution for Web documents:

  We contributed to the design of similarity functions, in the context of entity resolution in Web document collections. Our entity resolution methods that rely on these functions could obtain good result on known benchmarks. The presented work is based on [133], [134], [132], [137], [95], [135].

- Schema matching networks:

  We formalized the concept of schema matching networks that models a complete matching scenario. We demonstrate how to exploit the structure of this network, in particular in the reconciliation phase. We could reduce the necessary effort, in a setting where we used a crowd in this phase. We could not only obtain a global improvement method but also developed methods for guiding the expert users. These results are based on publications [64], [65], [99], [66], [67], [62], [63] and [45].

- Task affectation for knowledge-intensive crowdsourcing:

  We formalized the task assignment problem for knowledge-intensive crowdsourcing and we demonstrate how to use a hierarchical skill model to obtain a better quality task assignment. The presented results are based on [90], which is an extended version of the paper [89].

In my D.Phil thesis I worked on theoretical questions related to Boolean conjunctive evaluation and decomposition in relational databases. Conjunctive queries are one of the most widely used classes of queries in a relational database. Their expressive power corresponds to the Select-Project-Join queries of relational algebra [2]. Evaluating conjunctive queries is a core task in database systems. The combined complexity of conjunctive query evaluation in relational databases is NP-complete [16]. To overcome this difficulty, commercial database system rely on query optimizers that work very well in practical situations. There is also a large literature of academic research on query optimization, see *e.g.* [69]. It is nevertheless important to understand whether we can identify (possibly large) classes of conjunctive queries, for which the evaluation problem has polynomial complexity. To answer this question is also important for other domains or problems. For example the conjunctive query containment problem is (logspace) equivalent to the query evaluation problem, as well as other important problems, for example, the Constraint Satisfaction Problem (CSP) from artificial intelligence [78], [28] or the Homomorphism problem [41] that asks for the existence of a homomorphism from one relational structure to another. Im my D.Phil thesis [94] I analysed the complexity of various conjunctive query decomposition methods. We contributed to the understanding of the complexity of testing bounded generalized hypertreewidth and related problems and we obtained an NP-completeness result. Based on our understanding of the source of the complexity, we developed specific tractable decomposition methods as well as a methodology for defining other tractable classes of query decomposition. These results were published in [52] and in [51]. Even if these papers were published after my D.Phil. graduation, we do not discuss these results here as they are in a large extent based on the work I realized as a doctoral student.

The rest of this habilitation thesis is organized as follows. Chapter 2 presents our results on the entity resolution problem for Web documents. Chapter 3 synthesizes our work on schema matching networks. Chapter 4 gives an overview of our works on the task affectation problem for knowledge-intensive crowdsourcing. Chapter 5 concludes the habilitation thesis and presents perspectives for future work.

Chapter 2

Entity matching in Web documents

Context

I worked on entity matching problems in the context of the OKKAM project at EPFL, where I used to work as a postdoctoral researcher. OKKAM was a European project with several partners (including University of Trento, University of Hannover, University of Malaga, and others). I was the leader of EPFL team, where I coordinated the research of several PhD students. I also supervised the research of Surender Reddy Yerva, with whom I also worked beyond the direct scope of the project. His PhD supervisor was Karl Aberer.

Contributions

Entity matching: I published a series of papers with Surender Reddy Yerva, where I coordinated his work, and also contributed to the research. These papers include the workshop paper [133], the conference paper [134], which also has a more complete journal version [135] that also relies on the results of [132] and another journal publication [137]. We also published a demonstration paper [136]. Besides the work with Surender, I also contributed to the workshop paper [70] and I also coordinated a joint system engineering effort of the OKKAM project that resulted also a conference publication [95].

2.1 Entity resolution in Web data

Entity matching (or entity resolution) is the problem of identifying the entities that refer to the same real-word entity. There are a number of methods to recognize named entities (such as persons, geographic locations, organisations, etc.), but the extracted entity names on different sites might not refer to the same real-word entity. For example, if we identify the term *Paris*, we would like to determine whether it refers to the capital city of France, to a small city with the same name in Texas, in United States, or to a first name of a person. One of the key challenges to realize automated processing of Web data is entity resolution.

Entity matching is a well studied problem in the context of relational databases [42, 57, 17, 18, 92, 35, 11, 17], for a survey see [73]. Even if the papers are dated back quite early, this topic has also regained in importance recently. It is more and more common and easy to combine independent data sources, especially on the Web. The entity resolution in Web documents is very similar to the entity resolution problem studied in relational databases [20], however there are also several differences. Most importantly Web documents (Web pages, social media messages) often only contain partial or incomplete information about the entities. Web pages are also much less structured as database records. Because of these differences, the models which were developed for databases are not directly applicable in the new setting. Despite of the missing or incomplete information in Web documents, they are also sources of additional pieces of information. In our work we proposed ways to compare entities and predict whether they refer to the same real-word entity. Our quality-aware similarity functions are designed to deal with the partial information.

The rest of this section is organized as follows. Section 2.2 presents our quality-aware similarity functions that we designed to realize entity resolution in Web context. Section 2.3 summarizes our work on entity resolution services that one could offer on the Web and related system engineering questions. Section 2.4 discusses some limitations of our approach and gives perspectives on the work.

2.2 Quality-aware similarity assessment

In [135] we studied two specific variants of the general entity matching problem, namely the *person name disambiguation* problem and the *Twitter message classification*. In the person name disambiguation problem we are given a set of Web documents, each containing a given name and the goal is to cluster the documents such that two documents are in the same cluster if and only if they refer to the same real-world person. In the Twitter classification problem, we are given a set of Twitter messages, each containing a particular keyword, which is a company name. The goal is to classify the messages whether they are related to the company or not. For this problem, we develop company profiles and the task is then to match these profiles to the messages. Both of these problems can be seen as entity matching problems.

Similarity functions try to capture the degree of belief about whether two entities refer to the same real-world entity. There is a number of known techniques to derive similarity values. One can observe that the quality of these methods varies and highly depends on the input, and on specific features of the input. The quality-aware similarity assessment technique combines similarity assessments from multiple sources. As opposed to other combination methods, we estimate the accuracy of individual sources for specific regions of the input (i.e. they are not global estimations) and we use this accuracy estimate for combining similarity values. Additionally, as we are dealing with Web data, the lack of information poses an additional difficulty. Conceptually, our quality-aware similarity assessment technique can be seen as a specific ensemble learning method.

In the following we outline our quality-aware similarity assessment method for entity matching, in the case of Twitter messages. The article [135] give a more complete discussion. We used a semi-automatic process to construct company profiles that we used also to define a training set. We defined a feature extraction function, which compares a tweet T_i to the company entity representation E_k and outputs a vector of features.

$$Fn(T_i, E_k) = \{\, \overbrace{G_1, \ldots, G_m}^{profile-features}, \underbrace{F_1, \ldots, F_n}_{tweet-specific}, \overbrace{U_1, \ldots, U_z}^{ad-hoc} \} \tag{2.1}$$

We used then these features to classify whether a specific Twitter message is related to a company. We constructed classifiers for this purpose, base on the Naive Bayes classification method. If we denote the probability that a tweet is related to a given company C based on the features $(f_1, f_2, \ldots, f_n)$ as $P(f_1, f_2, \ldots, f_n \mid C)$ and the probability that they are not related $P(f_1, f_2, \ldots, f_n \mid \overline{C})$ then for an unseen tweet t, the posterior probabilities of whether the tweet is related to the company or not, can be calculated as in equations (2.2, 2.3).

$$P(C \mid t) = \frac{P(C) * P(t \mid C)}{P(t)} = \frac{P(C) * P(f_1, f_2, \ldots, f_n \mid C)}{P(f_1, f_2, \ldots, f_n)} \tag{2.2}$$

$$P(\overline{C} \mid t) = \frac{P(\overline{C}) * P(t \mid \overline{C})}{P(t)} = \frac{P(\overline{C}) * P(f_1, f_2, \ldots, f_n \mid \overline{C})}{P(f_1, f_2, \ldots, f_n)} \tag{2.3}$$

Depending on whether $P(C \mid t)$ is greater than $P(\overline{C} \mid t)$ or not, the naive Bayes classifier decides whether the tweet t is related to the given company or not, respectively. Thus, for a given set of features, we can construct a Naive Classifier. The quality of such a classifier can largely depend on the availability or quality of specific features we used. Thus we constructed not only one, but several classifiers, based on different set of features. We could then estimate how well these individual classifiers work, based on our training set. For an unseen company, we constructed the individual classifiers, and to obtain a final decision (whether the tweet message is related to a company or not), we combine these individual classifiers, such that we give more importance to the classifiers that are based on good quality features (Algorithm 1). For example, if the URL of a company is present in the tweet message, this is a very strong indication that the message is related to the company, thus the relevant classifier should be considered in an important way. While in the absence of the URL, the corresponding classifier should be given lower importance. [135]

Algorithm 1: Twitter classification

```
    compute decisions using multiple individual classifiers
    identify the regions in the feature space for the companies in the test set
    estimate the accuracy, for each classifier
    combine the decisions of the individual classifiers, using the estimates, for unseen
    companies
    decide whether the entities match
    output the decision
```

For combining the results of individual classifiers we first obtained quality estimations, based on our training set and for specific subsets of the training data. Then we experimented with different aggregation techniques to derive a combined estimation (whether the relevant entities are related). We discuss these techniques more in detail in [135].

2.2.1 Experimental evaluation

We evaluated our methods on two different datasets: the "WWW" dataset [9], that was used as a benchmark in a number of entity resolution methods and the WePS dataset [3]. We participated in the Weps evaluation champagne and workshop. Our method obtained the best results in the campaign [3]. Figure 2.1 shows the performance of the individual similarity functions on the entire WWW'05 dataset. The figure shows three metrics, namely F_p-measure, F-measure and *Rand*-index. The final column, depicted as black in the figure, is the combined performance of our quality-aware combination techniqe, which clearly shows improved performance. Similarly, Figure 2.2 shows the experimental results on the WePS-2 dataset. A more complete description of the methods and results of the evaluation is presented in [135].

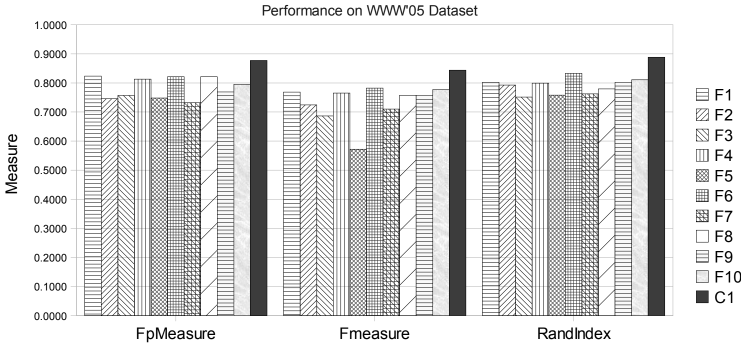


Figure 2.1: WWW results graph.

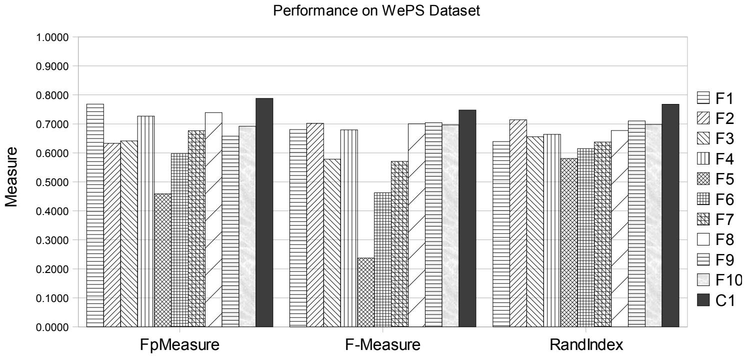


Figure 2.2: WePS-2 results graph.

2.2.2 Dynamic version

In the previous section we summarized our work on entity resolution in Web context. Our method addresses the unreliability of individual similarity functions in this setting, that is caused by the incompleteness of the available information. The presented method however assumed a static set of documents and profiles to construct the classification model. In the case of Twitter messages, one would prefer to classify the messages as they arrive. Moreover, to construct company (*i.e.* entity) profiles, one could exploit the keywords that are present in the messages. We completed our work to address these issues in [137]. We also created a demo [136] of this work, with a more polished user interface.

2.3 Entity Name System

Unique identifiers have an important role in relational databases. If we are able to efficiently match entities on the Web, we could also associate a unique identifier to entities that are references to the same object. For example the term "Paris" could refer to the capital city of France, but could also refer to another city in Texas, United States, or it could also be a first name. Such ambiguities are problematic for human users of Web, who can in some cases eliminate the ambiguity based on the context information. Humans can -in most of the cases- identify the correct meaning, but the problem is rather challenging if we would like to process the documents automatically. There exists some contexts on the Web where people are increasingly using such identifiers, such as for example fo scientific researchers `http://orcid.org` or digital objects `http://www.doi.org/`.

We developed an infrastructure that can serve such unique identifiers at large scale [95]. This requires reflections on access via Web services, storage, scalability, indexing and query processing, evolution of identifiers (*e.g.* merge of two entity descriptions), security and other aspects.

2.4 Limitations and perspectives

We worked on entity resolution problems in the Web context between 2008 and 2012. Our methods offer a simple way to cope with the incomplete available information about the entities in the data. The domain was an active field of research, several researchers worked actively on this problem and its variants. The field remained and continues to be active and a number of benchmark datasets[1] have been proposed [71][2]. Researchers have also proposed a wide-range of new methods that rely on sophisticated blocking techniques [101], [126] or on advanced machine learning methods [106], [61].

[1] `https://dbs.uni-leipzig.de/research/projects/object_matching/benchmark_datasets_for_entity_resolution`

[2] `https://db.disi.unitn.eu/pages/EMBench/`

Chapter 3

Schema matching networks

Context

I worked on the schema matching problem in the context of the NisB project. NisB was a project financed by the European Commission, with partners including SAP, Technion and others. I was involved in the NisB project from the very early phase of preparing the project proposal, to the end of the project and even beyond, as we started a number of collaborations that we continued after the end of the project. I was the leader of the EPFL team in the project. The project financed the PhD thesis of Hung Viet Quoc Nguyen. I followed his research work as a PhD student, but we continued our collaboration after his graduation. His PhD supervisor was Karl Aberer.

Contributions

I published a series of papers with Hung Viet Quoc Nguyen. These papers include the publication at DASFAA'2013 [64], where we collaborated with Tam Thranh Nguyen, who was a master student at the time of the publication. This paper received the "best student paper" award of the conference DASFAA'2013. We also prepared a more complete journal version of this work [65]. As Tam later became a PhD student, we continued collaborating with him. With Hung and another PhD student of the EPFL team Tri Kurniawan Wijaya and with the contributions of our project partners we published the conference paper [99]. I initiated this work, I proposed the basic concepts of the model, and I coordinated the research that lead to this publication. This initial work on schema matching networks demonstrated the usefulness of the model to reduce necessary human efforts through logic-based reasoning. We then oriented our work with Hung towards the use of more flexible probabilistic constraints probabilistic reasoning [66]. We completed this work and recently prepared a journal version [67]. Besides our main line of work with Hung, we explored complementary techniques for schema marching reconciliation using argumentation. We published a work [62] together with Xuan Hoai Luong, who was -at the time of redaction- a bachelor student who used to work as an intern at EPFL and worked on the NisB project, under the supervision of Hung and myself. We also prepared a demonstration paper on this work [63]. With Hung, we also contributed to the work on schema covers [45]. This work on schema covers was initiated by our collaborator Avigdor Gal, from Technion.

3.1 Schema matching and data integration

In the context of relational databases, the role of the human expert is essential for designing schemas and queries for the data. Once the design phase is completed, the meaning of the schema attributes or the values in the data tuples do not play any role, the query evaluation is a computational task that processes the symbols of the data. The schema and the queries precisely determine the computational problems. Once we would like to connect data from different, independently developed databases we need to deal with the differences of meaning at the level of data models, schemas, or at the level of constants. This problem is known as the semantic heterogeneity problem [55]. In this setting of course we cannot ignore the meaning of the involved terms, rather we rely on them to establish the correspondences between pairs of terms.

For this task the role of humans (*i.e.* experts who are involved in data integration) is rather different. Experts, based on their understanding of the terms and their context, could establish these connections (they could provide a correct list of correspondences), although in some cases this might even be a challenging task for humans [114]. While human experts could complete this task, this is a rather tedious task, especially at large-scale, so researchers try to design methods to complete this tasks automatically. Ultimately, one would like to have tools that could realize this task without human intervention. As for now, there exists a number of techniques and tools that can reduce the necessary human involvement, but the human involvement is still necessary. The human involvement can have various forms, for example

– the experts can provide input that one can use for supervised learning methods which automate the matching tasks or

– the experts can correct the errors that still remain if we rely on automated tools.

Semantic heterogeneity is only one of the wide range of data heterogeneity problems. For example, in a data integration problem, one need to deal with the heterogeneity of data structures, ranging from relational data, to semi-structured or unstructured data, or with the heterogeneity of data models, such as different database schemas or ontologies. Our work focused on the semantic heterogeneity issues, so our goal was to establish a matching between the constants that appear in different models, such that we identify terms with the same meaning. Establishing correspondences is only a first step for data integration process. A specific data integration task -even in the example of integrating data from two relational databases- might involve further steps, such as dealing with different integrity constraints, specifying the type of the correspondences (*e.g.* identifying inter-schema inclusion dependencies) or other issues.

Schema matching is a process of establishing correspondences between schema attributes of independently developed database schemas [10]. Ontology matching is a similar process of establishing correspondences between the concepts of two ontologies [39]. The proposed available techniques in the literature rely on the complementary available information. For example, various schema matching techniques exploit the string similarities between the schema attributes, structural similarities between the schemas, similarities in the data itself that is stored using the two different schemas, etc. Researchers use various probabilistic techniques [44], or in the presence of multiple possible sources of information, ensemble techniques try to combine the available evidences [117].

A number of commercial and academic tools has been developed using these above mentioned methods and these tools achieve impressive performance on some datasets. Despite the good performance results, the schema matching is inherently uncertain, thus one cannot expect that a tool will be able to create a matching without errors. As schema matching is one of the first steps of a data integration process, one needs to eliminate the remaining errors, in the computed correspondences. This phase of elimination of errors is often the most costly, as it involves human effort.

From this point of view, automated schema matching tools reduce the necessary efforts, by automatically computing the "easier" cases, so the human involvement is only needed to verify the obtained correspondences and to complete the matching of attributed that could not be obtained. Our work focused on this last phase of schema matching. We have formalized this post-matching phase [99] with the goal also to quantify the involved human effort. Our model was largely inspired by the works of Aberer et al. [1], [25] on emergent semantics. In particular, our integrity constraints are similar, in however the setting is completely different, we do not emphasize the autonomy of the involved databases and the emergence of shared understanding through self-organization. In a distributed, peer-to-peer setting [1] relies on message passing techniques and the sum-product algorithm [79] to realize probabilistic reasoning [104], while we use other computational methods, including expectation maximization and other statistical methods. We also assume that we know the interaction network of the involved databases.

The rest of this section is organized as follows. We present this concept and our model of the reconciliation process in Section 3.2, based on our article [99]. In this work we demonstrate that the model can contribute to minimize the necessary involved human efforts through logic based reasoning that we implemented with the help of Answer Set Programming. We present these methods in Section 3.3. Then we discuss our work on assigning the reconciliation task to crowd workers. The effort of human experts is costly, even if we can reduce their involvement, so the use of crowd workers promises to achieve the task with lower cost. However, crowd workers are less reliable so we need to deal with answer aggregation that we realize with the help of the expectation maximization (EM) algorithm. This approach is discussed in Section 3.4. In Section 3.5 we discuss the use of a more flexible set of probabilistic constraints for schema matching networks and reconciliation. This work makes use of probabilistic reasoning to guide human experts to work through the candidate correspondences in a way that could reduce the overall work. Finally Section 3.6 discusses another approach: we exploit an argumentation framework that can support a group of experts to collaboratively reconcile a set of candidate correspondences. The framework can help the participants the consequences of their decisions (accepting or rejecting a correspondence) which are expressed as arguments. In Section 3.7 we reflect on the limitations of our model.

3.2 Schema matching networks and the reconciliation process

Schema matching literature focuses almost exclusively on matching two schemas (that are often referred as source and target schemas). In real application settings however the enterprises need to match several schemas. In such setting one can also apply a top-down matching approach, where one defines a global schema (or ontology) and all involved schemas are matched to this global schema. This approach has also advantages, but our application setting (that were motivated by the industrial needs of the company SAP[1]) involved pairwise matchings. Before we present our model we discuss a declarative programming model, the answer set programming, that we used to formulate our model.

3.2.1 Answer Set Programming

ASP is rooted in logic programming and non-monotonic reasoning; in particular, the stable model (answer set) semantics for logic programs [47, 48] and default logic [110]. In ASP, solving search problems is reduced to computing answer sets, such that answer set solvers (programs for generating answer sets) are used to perform search.

We now give an overview of ASP. Formal semantics for ASP and further details are given in [38]. Let $\mathcal{C}, \mathcal{P}, \mathcal{X}$ be mutually disjoint sets whose elements are called *constant*, *predicate*, and *variable* symbols, respectively. Constant and variable symbols $\mathcal{C} \cup \mathcal{X}$ are jointly referred to as *terms*. An *atom* (or *strongly negated atom*) is defined as a predicate over terms. It is of the form $p(t_1, \ldots, t_n)$ (or $\neg p(t_1, \ldots, t_n)$, respectively) where $p \in \mathcal{P}$ is a predicate symbol and $t_1, \ldots, t_n$ are terms. An atom is called *ground* if $t_1, \ldots, t_n$ are constants, and *non-ground* otherwise. Below, we use lower cases for constants and upper cases for variables in order to distinguish both types of terms.

An answer set program consists of a set of disjunctive rules of form:

$$a_1 \vee \ldots \vee a_k \leftarrow b_1, \ldots, b_m, \ldots, \text{not}\, c_1, \ldots, \text{not}\, c_n \tag{2.1}$$

where $a_1, \ldots, a_k, b_1, \ldots, b_m, c_1, \ldots c_n$ $(k, m, n \geq 0)$ are *atoms* or *strongly negated atoms*. This rule can be interpreted as an *if-then* statement: if $b_1, \ldots, b_m$ are true and $c_1, \ldots, c_n$ are false, then we conclude that at least one of $a_1, \ldots, a_k$ is true. We call $a_1, \ldots, a_k$ the *head* of the rule, whereas $b_1, \ldots, b_m$ and $c_1, \ldots, c_n$ are the *body* of the rule. A rule with an empty body is a *fact*, since the head has to be satisfied in any case. A rule with an empty head is a *constraint*; the body should never be satisfied.

Example 3.1. Π is an answer set program comprising three rules (X being a variable, c being a constant). Program Π defines three predicates p, q, r. The first rule is a *fact* and the third rule denotes a *constraint*. Further, $p(c), r(c)$ are ground atoms, and $p(X), q(X)$, are non-ground atoms:

[1] `https://www.sap.com/`

$$\Pi = \left\{ \begin{array}{lll} p(c) & \leftarrow & \\ q(X) & \leftarrow & p(X). \\ & \leftarrow & r(c). \end{array} \right\} \tag{2.2}$$

Informally, an answer set of a program is a minimal set of ground atoms, *i.e.*, predicates defined only over constants, that satisfies all rules of the program. An example of an answer set of program Π given in Example 3.1 would be $\{p(c), q(c)\}$.

Finally, we recall the notion of *cautious* and *brave* entailment for ASPs [38]. An ASP Π *cautiously entails* a ground atom a, denoted by $\Pi \models_c a$, if a is satisfied by *all* answer sets of Π. For a set of ground atoms A, $\Pi \models_c A$, if for each $a \in A$ it holds $\Pi \models_c a$. An ASP Π *bravely entails* a ground atom a, denoted by $\Pi \models_b a$, if a is satisfied by *some* answer sets of Π. For a set of ground atoms A, $\Pi \models_b A$, if for each $a \in A$ it holds that some answer set M satisfies a.

3.2.2 Our model

We can now present our model. We first describe the matching problem in a network. Then we present our model of the reconciliation phase and finally we formulate our computational problem in this model.

Matching networks

A schema $s = (A_s, \delta_s)$ is a pair, where $A_s = \{a_1, ..., a_n, \}$ is a finite set of *attributes* and $\delta_s \subseteq A_s \times A_s$ is a relation capturing *attribute dependencies*. This model largely abstracts from the peculiarities of schema definition formalisms, such as relational or XML-based models. As such, we do not impose specific assumptions on δ_s, which may capture different kinds of dependencies, *e.g.*, composition or specialization of attributes.

Let $\mathcal{S} = \{s_1, ..., s_n\}$ be a set of schemas that are built of unique attributes ($\forall 1 \leq i \neq j \leq n, A_{s_i} \cap A_{s_j} = \emptyset$) and let $A_{\mathcal{S}}$ denote the set of attributes in $\mathcal{S}$, *i.e.*, $A_{\mathcal{S}} = \bigcup_i A_{s_i}$. The *interaction graph* $G_{\mathcal{S}}$ represents which schemas need to be matched in the network. Therefore, the vertices in $V(G_{\mathcal{S}})$ are labeled by the schemas from $\mathcal{S}$ and there is an edge between two vertices, if the corresponding schemas need to be matched.

An *attribute correspondence* between a pair of schemas $s_1, s_2 \in \mathcal{S}$ is an attribute pair $\{a, b\}$, such that $a \in A_{s_1}$ and $b \in A_{s_2}$. A *valuation function* associates a value in $[0, 1]$ to an attribute correspondence. *Candidate correspondences* $c_{i,j}$ (for a given pair of schemas $s_i, s_j \in \mathcal{S}$) is a set of attribute correspondences, often consisting of correspondences whose associated value is above a given threshold. The set of candidate correspondences C for an interaction graph $G_{\mathcal{S}}$ consists of all candidates for pairs corresponding to its edges, *i.e.* $C = \bigcup_{(s_i, s_j) \in E(G_{\mathcal{S}})} c_{i,j}$. C is typically the outcome of first-line schema matchers [44]. Most such matchers generate simple $1:1$ attribute correspondences, which relate an attribute of one schema to at most one attribute in another schema. In what follows, we restrict ourselves to $1:1$ candidate correspondences for simplicity sake. Extending the proposed framework to more complex correspondences can use tools that were proposed in the literature, *e.g.*, [43].

A *schema matching* for $G_{\mathcal{S}}$ is a set D of attribute correspondences $D \subseteq C$. Such schema matching is typically generated by second-line matchers, combined with human validation, and should adhere to a set of predefined constraints $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$. Such constraints may require, for example, that at least 80% of all attributes are matched. A schema matching D is *valid* if it satisfies all of the constraints in Γ.

Combining the introduced notions, we define a *matching network* to be a quadruple $(\mathcal{S}, G_{\mathcal{S}}, \Gamma, C)$, where $\mathcal{S}$ is a set of schemas (of unique attributes), $G_{\mathcal{S}}$ a corresponding interaction graph, Γ a set of constraints, and C a set of candidate correspondences.

Reconciliation Process

The set of candidate correspondences C aims at serving as a starting point of the matching process and typically violates the matching constraint set Γ. In this section, we model the reconciliation process under a set of predefined constraints Γ as an iterative process, where in each step a user asserts the correctness of a single correspondence. Starting with the result of a matcher, a set of correspondences, called an *active set*, is continuously updated by: (1) selecting an attribute correspondence $c \in C$, (2) eliciting user input (approval or disapproval) on the correspondence c, and (3) computing the consequences of the feedback and updating the active set. Reconciliation halts once the goal of reconciliation (*e.g.*, eliminating all constraint violations) is reached. It is worth noting that in general, a user may add missing correspondences to C during the process. For simplicity, we assume here that all relevant candidate correspondences are already included in C.

Each user interaction step is characterized by a specific index i. Then, D_i denotes the set of correspondences considered to be true in step i dubbed the *active set*. Further, let u_c^+ (u_c^-) denote the user input where u_c^+ denotes approval and u_c^- denotes disapproval of a given correspondence $c \in C$ and $U_C = \{u_c^+, u_c^- \mid c \in C\}$ be the set of all possible user inputs for the set of correspondences C. Further, $u_i \in U_C$ denotes user input at step i and $U_i = \{u_j \mid 0 \leq j \leq i\}$ is the set of user input assertions until step i. The consequences of such user input assertions U_i are modeled as a set $Cons(U_i) \subseteq U_C$ of positive or negative assertions for correspondences. They represent all assertions that can be concluded from the user input assertions.

A generic reconciliation procedure is illustrated in Algorithm 2. It takes a set of candidate correspondences C, a set of constraints Γ, and a reconciliation goal Δ as input and returns a reconciled set of correspondences D_r. Initially (line 1), the active set D_0 is given as the set of candidate correspondences C and the sets of user input U_0 and consequences $Cons(U_0)$ are empty. Then, we proceed as follows: First, there is a function *select*, which selects a correspondence from the set of candidate correspondences (line 3). Here, all correspondences for which we already have information as the consequence of earlier feedback (represented by $Cons(U_i)$) are neglected. Second, we elicit user input for this correspondence (line 4). Then, we integrate the feedback by updating the set of user inputs U_{i+1} (line 5), computing the consequences $Cons(U_{i+1})$ of these inputs with function *conclude* (line 6), and updating the active set D_{i+1} (line 7). A correspondence is added to (removed from) the active set, based on a positive (negative) assertion of the consequence of the feedback. The reconciliation process stops once D_r satisfies the halting condition Δ representing the goal of reconciliation.

Instantiations of Algorithm 2 differ in their implementation of the *select* and *conclude* rou-

```
Algorithm 2: Generic reconciliation procedure
  input : a set of candidate correspondences $C$, a set of constraints $\Gamma$, a reconciliation
          goal $\Delta$.
  output: the reconciled set of correspondences $D_r$.
  // Initialization
  $D_0 \leftarrow C$; $U_0 \leftarrow \emptyset$; $Cons(U_0) \leftarrow \emptyset$; $i \leftarrow 0$;
  while not $\Delta$ do
      // In each user interaction step (1) Select a correspondence
      $c \leftarrow select(C \setminus \{c \mid u_c^+ \in Cons(U_i) \vee u_c^- \in Cons(U_i)\})$;
      // (2) Elicit user input
      Elicit user input $u_i \in \{u_c^+, u_c^-\}$ on $c$;
      // (3) Integrate the feedback
      $U_{i+1} \leftarrow U_i \cup \{u_i\}$;
      $Cons(U_{i+1}) \leftarrow conclude(U_i)$;
      $D_{i+1} \leftarrow D_i \cup \{c \mid u_c^+ \in Cons(U_{i+1})\} \setminus \{c \mid u_c^- \in Cons(U_{i+1})\}$;
      $i \leftarrow i + 1$;
```

tines. For example, by considering one correspondence at a time, Algorithm 2 emulates a manual reconciliation process followed by an expert. As a baseline, we consider an expert working without any tool support. This scenario corresponds to instantiating Algorithm 2 with a selection of a random correspondence from $C \setminus Cons(U_i)$ ($select(C \setminus Cons(U_i))$) and the consequences of user input are given by the input assertions U_i ($conclude(U_i) = U_i$).

Minimal reconciliation problem

Given the iterative model of reconciliation, we would like to minimize the number of necessary user interaction steps for a given reconciliation goal. Given a schema matching network $(\mathcal{S}, G_{\mathcal{S}}, \Gamma, C)$, a reconciliation goal Δ, and a sequence of correspondence sets $\langle D_0, D_1, \ldots, D_n \rangle$ such that $D_0 = C$ (termed a *reconciliation sequence*), we say that $\langle D_0, D_1, \ldots, D_n \rangle$ is *valid* if D_n satisfies Δ. Let $\mathcal{R}_\Delta$ denote a finite set of valid reconciliation sequences that can be created by instantiations of Algorithm 2. Then, a reconciliation sequence represented by $\langle D_0, D_1, \ldots, D_n \rangle \in \mathcal{R}_\Delta$ is *minimal*, if for any reconciliation sequence $\langle D'_0, D'_1, \ldots, D'_m \rangle \in \mathcal{R}_\Delta$ it holds that $n \leq m$.
Our objective is defined in terms of a minimal reconciliation sequence, as follows.

Problem 1. Let $(\mathcal{S}, G_{\mathcal{S}}, \Gamma, C)$ be a schema matching network and $\mathcal{R}_\Delta$ a set of valid reconciliation sequences for a reconciliation goal Δ. The *minimal reconciliation problem* is the identification of a minimal sequence $\langle D_0, D_1, \ldots, D_n \rangle \in \mathcal{R}_\Delta$.

Problem 1 is basically about designing a good instantiation of *select* and *conclude* to minimize the number of iterations to reach Δ. The approach we took in [99] was to chose appropriate heuristics for the selection of correspondences (*select*) and to apply reasoning for computing the consequences (*conclude*).

3.3 Improving reconciliation through reasoning

Representing the reconciliation network and formalizing the reconciliation process enables to apply a simple form of reasoning, We do not need to present all correspondences to an expert user to verify, as we can infer their correctness. In this way we can reduce the necessary effort to reconcile the correspondences. We have chosen to encode the problem in the framework of Answer Set Programming (ASP) and then to use ASP solvers to realize the necessary reasoning. In the following, we explain how we encoded the problem in the ASP framework. With the help of this encoding, we can reason about attribute correspondences (in the presence of the defined constraints), and in this way we can avoid soliciting human experts about the correspondences, where we can deduce whether they should be correct or not.

Representing matching networks

Let $(\mathcal{S}, G_{\mathcal{S}}, \Gamma, C)$ be a matching network. An ASP $\Pi(i)$, corresponding to the i-th step of the reconciliation process, is constructed from a set of smaller programs that represent the schemas and attributes ($\Pi_{\mathcal{S}}$), the candidate correspondences (Π_C), the active set D_i ($\Pi_D(i)$), the basic assumptions about the setting (Π_{basic}), the constraints (Π_Γ), and a special rule that relates the correspondences and constraints Π_{cc}. The program $\Pi(i)$ is the union of the smaller programs $\Pi(i) = \Pi_{\mathcal{S}} \cup \Pi_C \cup \Pi_D(i) \cup \Pi_{basic} \cup \Pi_\Gamma \cup \Pi_{cc}$. We focus in the section on the four first programs.
Schemas and attributes: $\Pi_{\mathcal{S}}$ is a set of ground atoms, one for each attribute and its relation to a schema, and one for each attribute dependency:

$$\Pi_{\mathcal{S}} = \{attr(a, s_i) \mid s_i \in \mathcal{S}, a \in A_{s_i}\} \cup \{dep(a_1, a_2) \mid s_i \in \mathcal{S}, (a_1, a_2) \in \delta_{s_i}\} \tag{3.3}$$

Candidate correspondences: Π_C comprises ground atoms, one for each candidate correspondence in the matching network: $\Pi_C = \{cor(a_1, a_2) \mid (a_1, a_2) \in C\}$
Active set: $\Pi_D(i)$ is a set of ground atoms, corresponding to the active set D_i:
$\Pi_D(i) = \{corD(a_1, a_2) \mid (a_1, a_2) \in D_i\}$
Basic assumptions: rules in Π_{basic}, as follows.

- *An attribute cannot occur in more than one schema.* We encode this knowledge by adding a rule with an empty head, *i.e.*, a constraint, so that no computed answer set will satisfy the rule body. For each attribute $a \in A_{\mathcal{S}}$ and schemas $s_1, s_2 \in \mathcal{S}$, we add the following rule to Π_{basic}: $\leftarrow attr(a, s_1), attr(a, s_2), s_1 \neq s_2$.
- *There should be no correspondence between attributes of the same schema.* We add a rule to for each candidate correspondence $(a_1, a_2) \in C$ and schemas $s_1, s_2 \in \mathcal{S}$ to Π_{basic}: $\leftarrow cor(a_1, a_2), attr(a_1, s_1), attr(a_2, s_2), s_1 = s_2$.
- *The active set is a subset of all matching candidates.* We add a rule to Π_{basic}: $cor(X, Y) \leftarrow corD(X, Y)$.

Constraints (Π_Γ). We express matching constraints as rules in the program Π_Γ, one rule per constraint, such that $\Pi_\Gamma = \Pi_{\gamma_1} \cup \cdots \cup \Pi_{\gamma_n}$ for $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$. In the following, we give examples of three matching constraints.

- $1:1$ *constraint:* Any attribute of one schema has at most one corresponding attribute in another schema. We capture this constraint with the following rule: $\leftarrow match(X, Y), match(X, Z), attr(Y, S), attr(Z, S), Y \neq Z$.

- *Cycle constraint:* Two different attributes of a schema must not be connected by a path of matches. We call a cycle of attribute correspondences *incorrect*, if it connects two different attributes of the same schema, see Figure 3.2. Formally, a solution is valid if it does not contain any incorrect cycles. We encode this constraint based on a reachability relation (represented by $reach(X, Y)$, where X and Y are variables representing attributes) as follows:

$$\begin{array}{rcl} reach(X,Y) & \leftarrow & match(X,Y) \\ reach(X,Z) & \leftarrow & reach(X,Y), match(Y,Z) \\ & \leftarrow & reach(X,Y), attr(X,S), attr(Y,S), X \neq Y. \end{array} \tag{3.4}$$

- *Dependency constraint:* Dependencies between attributes shall be preserved by paths of matches. To encode this type of constraint, we proceed as follows. First, we model (direct or indirect) reachability of two attributes in terms of the dependency relation (represented by *reachDep*(X,Y), where X and Y are both variables representing attributes). Then, we require that reachability based on the *match* relation for two pairs of attributes preserves the reachability in terms of the dependency relation between the attributes of either schema:

$$\begin{array}{rcl} reachDep(X,Y) & \leftarrow & dep(X,Y) \\ reachDep(X,Z) & \leftarrow & reachDep(X,Y), dep(Y,Z) \\ & \leftarrow & reachDep(X,Y), reach(X,B), \\ & & reachDep(A,B), reach(Y,A). \end{array} \tag{3.5}$$

Connecting correspondences and constraints (Π_{cc}). A rule that computes a set of correspondences that satisfy the constraints of the matching network uses a *rule with a disjunctive head*. We encode a *match* relation (represented by $match(X, Y)$) to compute this set. A candidate correspondence $cor(X, Y)$ is either present in or absent from *match*, the latter is denoted as $noMatch(X, Y)$. This is captured by the rule:

$$match(X,Y) \vee noMatch(X,Y) \quad \leftarrow \quad corD(X,Y). \tag{3.6}$$

Detecting Constraint Violations

Adopting the introduced representation enables us to compute violations of constraints automatically, with the help of ASP solvers. In large matching networks, detecting such constraint violations is far from trivial and an automatic support is crucial.

We say that a set of correspondences $C' = \{c_1, \ldots, c_k\} \subseteq C$ violates a constraint $\gamma \in \Gamma$ if $\Pi_{\mathcal{S}} \cup \Pi_{basic} \cup \Pi_\gamma \not\models_b \Pi_{C'}$. In practice, we are not interested in all possible violations, but rather the minimal ones, where a set of violations is minimal w.r.t. γ if none of its subsets violates γ. Given a set of correspondences C', we denote the set of minimal violations as
$Violation(C') = \{C'' \mid C'' \subseteq C', \Pi_{\mathcal{S}} \cup \Pi_{basic} \cup \Pi_\gamma \not\models_b C'', \gamma \in \Gamma, C'' \text{ is minimal}\}$.

The ASP representation also allows for expressing reconciliation goals. A frequent goal of experts is to eliminate all violations: $\Delta_{NoViol} = \{\Pi(i) \models_b \Pi_D(i)\}$, *i.e.*, the joint ASP bravely entails the program of the active set.

3.3.1 Empirical evaluation

For our evaluation, we used five real-world datasets spanning various application domains, from classical Web form integration to enterprise schemas. We used the following datasets for evaluation.

Business Partner (BP): Three enterprise schemas, originally from SAP, which model business partners in SAP ERP, SAP MDM, and SAP CRM systems.

PurchaseOrder (PO): Purchase order e-business documents from various resources.

University Application Form (UAF): Schemas from Web interfaces of American university application forms.

WebForm: Automatically extraction of schemas from Web forms of seven different domains (*e.g.*, betting and book shops) using OntoBuilder.[2]

Thalia: Schemas describing university courses. This dataset has no exact match, and is mainly used in the first experiment concerning constraint violations.

All datasets are publicly available[3]. We used two schema matchers, COMA [33] and Auto Mapping Core (AMC) [105]. Reasoning was conducted with the DLV system,[4], a state-of-the-art ASP interpreter. All experiments ran on an Intel Core i7 system (2.8GHz, 4GB RAM).

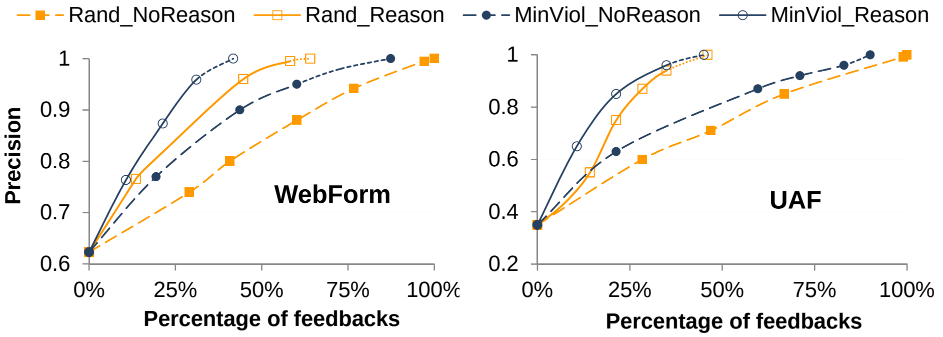


Figure 3.1: User effort needed to achieve 100% precision.

We evaluated our reconciliation framework in different settings. We varied the construction of schema matching networks in terms of dataset, matcher, and network topology. For the reconciliation process, we considered different types of users and reconciliation goals. We measured the quality improvements achieved by reconciliation and the required human efforts as follows:

Precision We measures quality improvement where precision of the active set at step i is defined as $P_i = (|D_i \cap G|)/|D_i|$, with G being the exact match.

User effort is measured in terms of feedback steps relative to the size of the matcher output C, *i.e.*, $E_i = i/|C|$ (where a user examines one correspondence at a time).

We studied the extent to which our approach reduces human effort in terms of necessary user feedback steps as follows. For each dataset, we obtained candidate correspondences using COMA. We generated a complete interaction graph and required the $1:1$ and cycle constraints

[2] `http://ontobuilder.bitbucket.org/`

[3] BP, PO, UAF, WebForm are available at `http://lsirwww.epfl.ch/schema_matching` and Thalia can be found at: `http://www.cise.ufl.edu/research/dbintegrate/thalia/`

[4] `http://www.dlvsystem.com`, release 2010-10-14

to hold. Then, we simulated user feedback using the exact matches for the dataset. The reconciliation process starts with the matching results, as determined by COMA.

We explored how the quality of the match result in terms of precision improved when eliciting user feedback according to different strategies. For the WebForm and UAF datasets, Figure 3.1 depicts the improvements in precision (Y-axis) with increased feedback percentage (X-axis, out of the total number of correspondences) using four strategies, namely

(1) *Rand_NoReason:* feedback in random order, consequences of feedback are defined as the user input assertions (our baseline that corresponds to the case where humans work without help from our techniques);

(2) *Rand_Reason:* reconciliation using random selection of correspondences, but applying reasoning to conclude consequences;

(3) *MinViol_NoReason:* reconciliation selection of correspondences based on ordering, consequences of feedback are defined as the user input assertions; and finally

(4) *MinViol_Reason:* reconciliation with the combination of ordering and reasoning for concluding consequences.

The results depicted in Figure 3.1 show the average over 50 experiment runs. The dotted line in the last segment of each line represents the situation where no correspondence in the active set violated any constraints, *i.e.*, the reconciliation goal Δ_{NoViol} has been reached. In those cases, we used random selection for the remaining correspondences until we reached a precision of 100%. The other datasets (BP and PO) demonstrate similar results and are omitted here.

The results show a significant reduction of user effort for all strategies with respect to the baseline. Our results further reveal that most improvements are achieved by applying reasoning to conclude on the consequences of user input. Applying ordering for selecting correspondences provides additional benefits. The combined strategy (*MinViol_Reason*) showed the highest potential to reduce human effort, requiring only 40% or less of the user interaction steps of the baseline.

3.4 Reconciliation through crowdsourcing

The involvement of human experts makes the reconciliation phase one of the most costly tasks of database integration projects. One possible alternative to involving experts is to orient a crowd of workers to realize this task for a much lower prise. Of course, there are also disadvantages of this solution. For example, companies are reluctant to give away information about the schemas of their databases. In the case of reconciliation however they only need to give away pairs of attributes, that are candidate matchings and not their entire schema[5].

There is however another problem that we need to address in this context, namely the reliability of workers. The result of a task assigned to the crowd is much less reliable than the answers form the expert collaborators. To overcome this problem one can assign the same task to multiple workers and then aggregate the obtained results. We developed specific aggregation methods with the aim to minimize the effect of possible incorrect answers from crowd workers.

[5] Of course providing more context to the crowd workers could also help their work. We should note that the context information is often taken into account by the matchers.

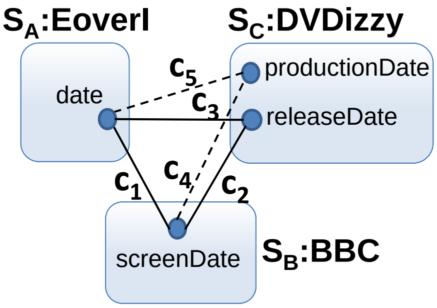


Figure 3.2: A matching network of real-world schemas

There is a large literature on answer aggregation in the context of crowdsourcing [142]. Our work relies on the Expectation Maximization (EM) technique.

We also relaxed the notion of constraints: we essentially use the "soft" versions of constraints of our original model. In [99] we considered "hard" constraints, that enabled us to apply logic-based reasoning. In real situations however it is not easy to formulate all constrains in this form. Sometimes even experts agree to chose correspondences that would be violations to the hard constraints. For this work we relied on the same set of constraints, but we have defined the constraints in a different way to enable possible exceptions.

3.4.1 Integrity constraints

We can express natural expectations that one has w.r.t. the entire network in the form of consistency constraints as follows. Given a network of schemas $N = \langle \mathcal{S}, G_{\mathcal{S}}, C \rangle$, let us denote $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a finite set of constraints that are used to represent the expected consistency conditions on N. We say that a set of correspondences $C' \in C$ violating a constraint $\gamma \in \Gamma$ is a constraint violation. In practice, we are not interested in all possible violations, but the minimal ones: We say that a violation is minimal w.r.t. γ, if none of its proper subsets is violating γ. In the following we give example of the probabilistic versions of constraints of our original work. These constraints have a parameter Δ that can be learned or updated based on a sufficiently large set of answers from the crowd.

Generalized 1-1 constraint. Each attribute of one schema should be matched to at most one attribute of any other schema. For example in Figure 3.2, the set $\{c_3, c_5\}$ violates the 1-1 constraint. However there are some exceptions where this constraint does not hold, such as the attribute *name* of a schema might be a concatenation of the attributes *firstname* and *lastname* of another schema. To capture this observation, we provide a relaxed version of the constraint using probability theory:

$$Pr(\gamma_{1\text{-}1} | X_{c_0}, X_{c_1}, \ldots, X_{c_k}) = \begin{cases} 1 & \text{If } m \leq 1 \\ \Delta \in [0,1] & \text{If } m > 1 \end{cases} \tag{4.7}$$

where $\{c_0, c_1, \ldots, c_k\}$ is a set of correspondences that share a common source attribute and m is the number of X_{c_i} assigned as *true*. When $\Delta = 0$, there is no constraint exception (the constraint is hard). The constraint can be softened by adjusting the Δ value.

Cycle constraint. If multiple schemas are matched in a cycle, the matched attributes should form a closed cycle. For example in Figure 3.2, the set $\{c_1, c_2, c_5\}$ violates the *cycle constraint*. Formally, following the notion of cyclic mappings in [24], we formulate the conditional probability of a cycle as follows:

$$Pr(\gamma_{\circlearrowleft} | X_{c_0}, X_{c_1}, \ldots, X_{c_k}) = \begin{cases} 1 & \text{If } m = k + 1 \\ 0 & \text{If } m = k \\ \Delta \in [0, 1] & \text{If } m < k \end{cases} \tag{4.8}$$

where $c_0, c_1, \ldots, c_k$ forms a sequence of correspondences that starts and ends at the same attribute; and m is the number of X_{c_i} assigned as *true* and Δ is the probability of compensating errors along the cycle (*i.e.*, two or more incorrect assignment resulting in a correct reformation).

3.4.2 Probability calculations

In this Section we discuss the the techniques we used to compute $Pr(X_c)$, the probability that a given correspondence c is true.

The EM algorithm takes as input an answer matrix $[M_{ij}]_{n \times m}$ (n correspondences and m workers) and returns a tuple $\langle P, V \rangle$. V is a vector in which each element v_j is the (estimated) quality of the worker w_j. P is a vector in which each element p_i is the (estimated) probability of correctness for each correspondence c_i. The algorithm alternates between two steps: Expectation step (E-step) and Maximization step (M-step) until it reaches a convergence state where the estimated values of v_j and p_i are stable. In the k-th E-step, it takes the calculated worker quality V^{k-1} estimated in the previous step to calculate the probability of correctness for the correspondences P^k in this step according to the following equation:

$$p_i^k = \sum_{t=1}^{m} v_t^{k-1} \times f(M_{it}) \times \mathbb{1}_{M_{it}=true} \tag{4.9}$$

where f is a function that estimates the correctness of the answers given by the workers and $\mathbb{1}_{cond} = 1$ if *cond* is true and 0 otherwise. In practice, we can estimate the value of f by the probability of correctness for the answer M_{ij} calculated in the previous step. After this step, for each correspondence, the correct answer can be estimated by selecting the one with the highest probability. For correspondences that have been validated from workers, we take the provided answers as the correct values. We denote the estimated correct values at step k as $G^k = \{g_1, g_2, ..., g_n\}$ where g_i is the correct answer for correspondence c_i.

Since the estimated correct values change after each E-step, we need to update the estimated quality of the workers to reflect these changes. In the k-th M-step, we re-estimate the quality of the workers by computing the loss value L_i^k for each worker. This loss value measures how deviating the answers provided by a worker to the estimated correct values:

$$L_j^k = \sum_{i=1}^{n} v_j \times h(M_{ij}, g_i) \tag{4.10}$$

where h is a function that measure the distance between two values. Based on the loss value of each worker, we can re-estimate its quality based on the intuition that the higher the loss value, the lower the quality of the worker.

In the end, the probabilities of possible aggregations of each correspondence c_i are:

$$\begin{cases} Pr(X_{c_i} = true) = p_i \\ Pr(X_{c_i} = false) = 1 - p_i \end{cases} \tag{4.11}$$

3.4.3 Answer aggregation

We compute the aggregation decision $g_\pi(c)$ for each correspondence $c \in C$ that is a pair $g_\pi(c) = \langle a_c, e_c \rangle$, where a_c is the aggregated value (*true* or $false$) and e_c is the error rate. The aggregation decision is obtained as follows:

$$g_\pi(c) = \begin{cases} \langle true, 1 - Pr(X_c = true)\rangle, \text{ if } Pr(X_c = true) \geq 0.5 \\ \langle false, 1 - Pr(X_c = false)\rangle, \text{ otherwise} \end{cases} \tag{4.12}$$

The aggregated value in the aggregation decision thus corresponds to the value that has a higher probability (and lower error rate). The error rate is the probability of making wrong decision.

We would like to reduce this error rate, for each correspondence. We could achieve a lower error rate if we ask more questions, however asking more questions induces higher costs as well. Instead, we will try to lower the error rate given a limited budget of money with the help of the integrity constraints that we explain in the next section. For several crowdsourcing tasks, one could achieve a lower error rate through asking more questions [72, 102]. This is, in fact, a trade-off between the costs and the accuracy [131].

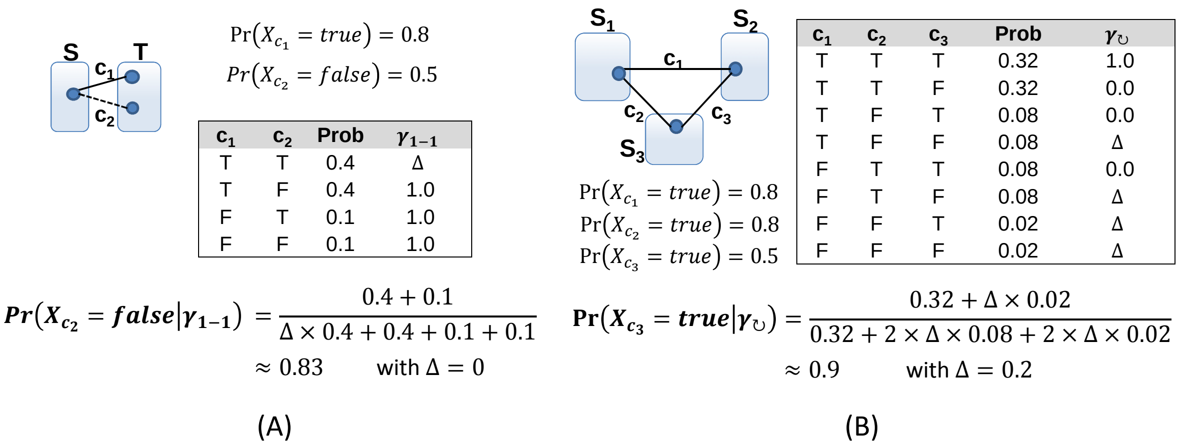
3.4.4 Aggregating with Constraints

Given the aggregation $g_\pi(c)$ of a correspondence c, we compute the justified aggregation $g_\pi^\gamma(c)$ when taking into account the integrity constraint γ. The aggregation $g_\pi^\gamma(c)$ is obtained similarly to equation 4.12, but we use here the conditional probability $Pr(X_c|\gamma)$ instead of $Pr(X_c)$. Formally,

$$g_\pi^\gamma(c) = \begin{cases} \langle true, 1 - Pr(X_c = true|\gamma)\rangle, \text{ If } Pr(X_c = true|\gamma) \geq 0.5 \\ \langle false, 1 - Pr(X_c = false|\gamma)\rangle, \text{ Otherwise} \end{cases} \tag{4.13}$$

In the following, we describe how to obtain the conditional probabilities $Pr(X_c|\gamma)$ in the case of 1-1 constraint and the cycle constraint.

Aggregating with 1-1 Constraint Our approach is based on the intuition illustrated in Figure 3.3(A), depicting two correspondences c_1 and c_2 with the same source attribute. After receiving the answer set from workers and estimating the probabilities, we obtained the probability $Pr(X_{c_1} = true) = 0.8$ and $Pr(X_{c_2} = false) = 0.5$. When considering c_2 independently, it is hard to conclude c_2 being approved or disapproved. However, when taking into account c_1 and 1-1 constraint, c_2 tends to be disapproved since c_1 and c_2 cannot be *true* at the same time. Indeed, following probability theory, the conditional probability $Pr(X_{c_2} = false|\gamma_{1-1}) \approx 0.83 > Pr(X_{c_2} = false)$.



$\Pr(X_{c_1} = true) = 0.8$

$Pr(X_{c_2} = false) = 0.5$

c_1	c_2	Prob	γ_{1-1}
T	T	0.4	Δ
T	F	0.4	1.0
F	T	0.1	1.0
F	F	0.1	1.0

$$\boldsymbol{Pr(X_{c_2} = false|\gamma_{1-1})} = \frac{0.4 + 0.1}{\Delta \times 0.4 + 0.4 + 0.1 + 0.1}$$
$$\approx 0.83 \qquad \text{with } \Delta = 0$$

$\Pr(X_{c_1} = true) = 0.8$

$\Pr(X_{c_2} = true) = 0.8$

$\Pr(X_{c_3} = true) = 0.5$

c_1	c_2	c_3	Prob	$\gamma_{\circlearrowleft}$
T	T	T	0.32	1.0
T	T	F	0.32	0.0
T	F	T	0.08	0.0
T	F	F	0.08	Δ
F	T	T	0.08	0.0
F	T	F	0.08	Δ
F	F	T	0.02	Δ
F	F	F	0.02	Δ

$$\mathbf{Pr}(X_{c_3} = \boldsymbol{true}|\gamma_{\circlearrowleft}) = \frac{0.32 + \Delta \times 0.02}{0.32 + 2 \times \Delta \times 0.08 + 2 \times \Delta \times 0.02}$$
$$\approx 0.9 \qquad \text{with } \Delta = 0.2$$

Figure 3.3: Computing conditional probability with (A) 1-1 constraint and (B) cycle constraint

Computing conditional probability. Given the same set of correspondences $\{c_0, c_1, \ldots, c_k\}$ above, let us denote p_i as $Pr(X_{c_i} = true)$ for short. Without loss of generality, we consider c_0 to be the favorite correspondence whose probability p_0 is obtained from the worker answers. Using the Bayesian theorem and equation 4.7, the conditional probability of correspondence c_0 with 1-1 constraint γ_{1-1} is computed as:

$$Pr(X_{c_0} = true|\gamma_{1-1}) = \frac{Pr(\gamma_{1-1}|X_{c_0} = true) \cdot Pr(X_{c_0} = true)}{Pr(\gamma_{1-1})} = \frac{(x + \Delta(1-x)) \times p_0}{y + \Delta(1-y)} \tag{4.14}$$

$$\text{where} \quad x = \prod_{i=1}^{k} (1 - p_i)$$
$$y = \prod_{i=0}^{k} (1 - p_i) + \sum_{i=0}^{k} [p_i \prod_{j=0, j\neq i}^{k} (1 - p_j)]$$

x can be interpreted as the probability of the case where all other correspondences except c being disapproved. y can be interpreted as the probability of the case where all correspondences being disapproved or only one of them being disapproved.

Aggregating with Cycle Constraint To motivate our definitions we present a small matching network. Figure 3.3(B) depicts an example of cycle constraint for three correspondences c_1, c_2, c_3. After receiving the set of answers from workers and computing the probabilities, we obtaine the probability $Pr(X_{c_1} = true) = Pr(X_{c_2} = true) = 0.8$ and $Pr(X_{c_3} = true) = 0.5$. When considering c_3 independently, it is hard to conclude c_3 being *true* or *false*. However, when taking into account c_1, c_2 under the cycle constraint, c_3 tends to be *true* since the cycle created by c_1, c_2, c_3 shows an interoperability. Therefore, the conditional probability $Pr(X_{c_3} = true|\gamma_{1-1}) \approx 0.9 > Pr(X_{c_3} = true)$.

Computing conditional probability. Given a closed cycle along $c_0, c_1, \ldots, c_k$, let denote the constraint on this circle as $\gamma_{\circlearrowleft}$ and p_i as $Pr(X_{c_i} = true)$ for short. Without loss of generality, we consider c_0 to be the favorite correspondence whose probability p_0 is obtained by the answers of workers in the crowdsourcing process. Following the Bayesian theorem and equation 4.8, the conditional probability of correspondence c_0 with circle constraint is computed as:

$$Pr(X_{c_0} = true|\gamma_{\cup}) = \frac{Pr(\gamma_{\cup}|X_{c_0} = true) \times Pr(X_{c_0} = true)}{Pr(\gamma_{\cup})} = \frac{(\prod_{i=1}^{k}(p_i) + \Delta(1-x)) \times p_o}{\prod_{i=0}^{k}(p_i) + \Delta(1-y)} \tag{4.15}$$

$$\begin{aligned} where \quad x &= \prod_{i=1}^{k}(p_i) + \sum_{i=1}^{k}[(1-p_i)\prod_{j=1,j\neq i}^{k} p_j] \\ y &= \prod_{i=0}^{k}(p_i) + \sum_{i=0}^{k}[(1-p_i)\prod_{j=0,j\neq i}^{k} p_j] \end{aligned}$$

x can be interpreted as the probability of the case where only one correspondence among $c_1, \ldots, c_k$ except c_0 is disapproved. y can be interpreted as the probability of the case where only one correspondence among $c_0, c_1, \ldots, c_k$ is disapproved.

Aggregating with multiple constraints In general settings, we could have a finite set of constraints $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$. Let denote the aggregation with a constraint $\gamma_i \in \Gamma$ is $g_{\pi}^{\gamma_i}(c) = \langle a_c^i, e_c^i \rangle$, whereas the aggregation without any constraint is simply written as $g_{\pi}(c) = \langle a_c, e_c \rangle$. Since the constraints are different, not only could the aggregated value a_c^i be different ($a_c^i \neq a_c^j$) but also the error rate e_c^i could be different ($e_c^i \neq e_c^j$). In order to reach a single decision, the challenge then becomes how to define the multiple-constraint aggregation $g_{\pi}^{\Gamma}(c)$ as a combination of single-constraint aggregations $g_{\pi}^{\gamma_i}(c)$.

Since the role of constraints is to support reducing the error rate and the aggregation $g_{\pi}(c)$ is the base decision, we compute the multiple-constraint aggregation as $g_{\pi}^{\Gamma}(c) = \langle a_c, e_c^{\Gamma} \rangle$, where $e^{\Gamma} = min(\{e_c^i | a_c^i = a_c\} \cup e_c)$. We take the minimum of error rates in order to emphasize the importance of integrity constraints.

3.5 Pay-as-you-go reconciliation

The probabilistic integrity constraints that we discussed in previous section (Section 3.4) enable to formulate integrity constraints with exceptions. They can also be exploited to reduce the necessary human efforts in the reconciliation process. In particular, as we can estimate the probability whether a correspondence is correct in the presence of the probabilistic integrity constraints and based on the obtained user input. For calculating these probabilities we describe the use the Expectation Maximization technique in the previous section (Section 3.4). In our work [67] we go even further to be able to compute these probabilities in a pay-as-you-go fashion, that is use the user input as soon as it arrives. For this we model the matching network as a factor graph. In this way we can compute the probabilities that describe the correctness of correspondences as a marginal probabilities of the corresponding random variables. As the schema matching network could lead to highly cyclic factor graphs, we did not use the Sum-Product algorithm, that could converge slowly in such a case, but we turned to Gibbs sampling techniques [49]. We used the Elementary system, an efficient, state-of-the-are implementation for factor graph computations that was realized by Zhang and Ré based on their results in [139]. We propose a Tabu-search based heuristic technique in [67] to obtain a matching instance (a set of correspondences). We experimentally analysed the efficiency of our methods, that we describe in detail in [67].

3.6 Collaborative reconciliation

We developed a variant of our reconciliation model in [62], where a group of experts can collaboratively reconcile the candidate correspondences. The members of the group can work on reconciling the correspondences at different parts of the matching network. When they merge their results, they might have conflicting views on some correspondences. With the help of argumentation techniques we provide the experts with a tool that they can use in their discussion. In particular the presence of global consistency constraints enables to reason about the consequences for choosing a particular attribute correspondence and the potential conflicts that is can induce, that might not be immediately clear for the individual experts.

3.7 Limitations and perspectives

We modelled the reconciliation phase fo schema matching networks. Our model enabled to quantify the involved human efforts and also to reduce it. One might question our assumptions that might not hold in certain integration setting. For example, while the pairwise matching setting was essential to our industrial partners, in other situations on realises the matchings not "purely" in peer-to-peer way, but for some parts of the network on can define a global database schema. The model of the human interaction is also somewhat limited, for example the experts do not process one correspondence at a time. We assume that the experts validate or invalidate certain attribute correspondences, but this is not exactly the way how humans work. The experts themselves cannot avoid thinking about the consequences of their decisions so naturally they deal with several correspondences together. Moreover they are not involved in the attribute matching, but also in the construction of schema mappings, that rely on the attribute correspondences.

We tried to optimize the involved efforts based on specific criteria, but we have not give attention to keep the work interesting or challenging for humans. Our work was initiated through an industrial need in the context of schema matching, however we believe that the approach can have applications in completely different domains. The methods could be applied whenever we need a consistent set of data, where consistency rules can be expressed in form of (probabilistic) constraints and we need human efforts as our automated tools do not produce the desired level of consistency.

Chapter 4

Worker and task matching in crowdsourcing

Context

I worked on the task assignment problem for knowledge intensive crowdsourcing with Panagiotis Mavridis, who completed his PhD thesis under my supervison. I co-supervised (50%) his thesis with David Gross-Amblard, who was the thesis director. He defended his thesis entitled "Using Hierarchical Skills for Optimal Task Selection in Crowdsourcing " in 2017 at the University of Rennes 1.

Contributions

The results presented in this section were first published at the WWW'2016 conference [89]. This section is based on an extended version of this work that is under review. For this extended version, I have further generalized the definitions and reimplemented some of the experiments. Our paper [89] received the "best student paper award" of the conference WWW'2016. The presentation of this section is based on an extended version of our conference publication, [90] that is under review.

4.1 Knowledge-intensive crowdsourcing

In the previous chapter (Chapter 4) we discussed the use of crowdsourcing in the context of schema matching for validating candidate attribute correspondences. As the answers from the crowd workers potentially erroneous, we applied specific aggregation techniques to obtain a more reliable set of correspondences. In this chapter we discuss a different way of improving the expected quality form the crowd: we try to affect the workers to task such that the workers have all (or most) of the required skills to complete the requested tasks. This is particularly important for knowledge-intensive crowdsourcing [13, 112], where the tasks require some specific skills. For example, workers do not only asked decide whether they see an elephant or a lion on an image, but the labelling tasks requires specific knowledge: one can present images of insects to the workers and ask them to annotate them according to a precise entomological taxonomy of species. The SPIPOLL[1] platform has exactly this task. Other knowledge-intensive platforms include Zooniverse[2] or BumbleBeeWatch[3]. While one can obtain useful results through these platforms for a number of tasks, controlling the quality of the results is a challenging issue, due to the unreliability, volatility or lack of skills of participants.

Generic crowdsourcing platforms already provide basic skill labelling (such as qualifications in Amazon Mechanical Turk[4]: these are short descriptions of qualifications for certain skills a participant might have or the requester might require). Similarly, academic research [13, 112, 140, 122] is also considering skill models to improve result quality. These existing approaches rely on flat, unstructured skill models such as tags or keywords. To obtain a good (expected) quality results from the crowd, one should try to match the crowd workers to the proposed task, such that the skills of the workers correspond to the requested skills for tasks. The task affectation problem can thus be considered as a matching problem, where one would like to find matches between user skill profiles to task descriptions. This section gives an overview of our work for this problem. Our approach relies on a simple form of reasoning about tasks: here we try to match a worker to a task that he is likely able to execute. In the simplest case, he should possess the required skill or in a more realistic case, possess a skill that is similar or more general than the requested skill.

The rest of this section is organized as follows. We give an overview of our hierarchical skill model that enables a basic form of reasoning about skills in Section 4.2. We discuss the formalization of the task assignment problem for knowledge-intensive crowdsourcing in Section 4.3. We discuss algorithms that exploit the skill hierarchies and some aspects of the evaluation in Section 4.4.

4.2 Reasoning about human skills

Applications often require at least some basic forms of reasoning about skills (such as, for example, knowing that the skill *English writing* is “more specific” than the skill *English reading*, in the sense that anyone who can write English can also read). Even such simple reasoning operations are not easy to realize with the above mentioned flat skill models. Many platforms could benefit from such a structured skill approach. On the one hand, it would allow a precise and better targeting of tasks. On the other hand, skill reasoning capacities, especially skill substitutions, would enable the participation of the full available workforce of the platform, even if skills do not correspond exactly to requirements. It is noteworthy that rich skill taxonomies are available and used in other contexts, such as ESCO[5], which is used to help European citizens in their job search and represents 5,000 skills in a structured way.

While current crowdsourcing platforms focus mainly on tasks that require little or no specific skills such as image labelling or text transcription, future platforms might offer a market-

[1] Photographic monitoring of pollinator insects, http://www.spipoll.org/

[2] https://www.zooniverse.org

[3] http://www.bumblebeewatch.org

[4] http://docs.aws.amazon.com/AWSMechTurk/latest/AWSMechanicalTurkRequester/Concepts_QualificationsArticle.html

[5] ESCO: European Skills, Competences Qualifications and Occupations https://ec.europa.eu/esco/home.

place for knowledge-intensive tasks that require more specific skills. Modelling human skills and managing workers with some specific skills is crucial for knowledge-intensive crowdsourcing. If we had information on the required skills for tasks as well as available skills of workers, we could clearly use this information to improve the quality of the task affectation.

If we consider specific skills required to complete a task we often intuitively use skill inference: we consider natural to take more specific or more general skills than the required ones. For example, if someone can translate from Spanish to English, it is reasonable to assume that he speaks English (and also that he understands Spanish). Thus the skill for being able to translate from Spanish to English is more specific than speaking English: anyone who can translate to English is able to speak English. Completing skill models with hierarchies enables some basic forms of reasoning. This was also our strategy, we demonstrated that in this way one can improve the expected quality of the task affectations.

The most simple way to represent skill hierarchies is to use subsumption hierarchies, that we adopted in the earlier version of this work [89]. An example for such a taxonomy is depicted in Figure 4.1. As the skill s' = *Java 1.8 thread* is more specific than the skill s = *core Java* (that we denote as $s \leq s'$), the node s' is a node of the subtree rooted at s.

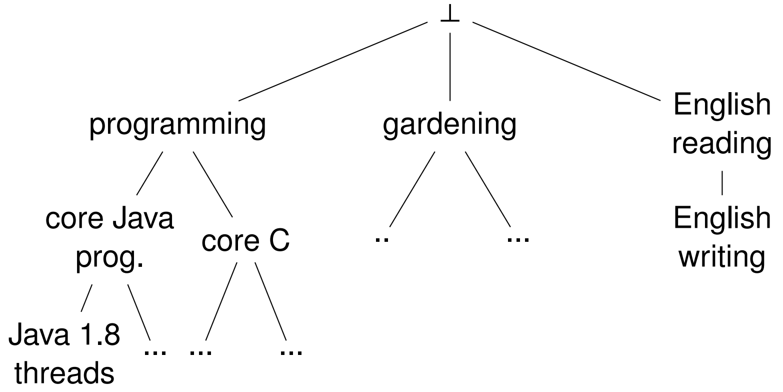


Figure 4.1: a skill taxonomy

Taxonomies, however, have certain limitations: any given skill can only have one parent in a taxonomy, while in real settings this is not always the case. For example, as in our above example, the skill *Translating from English to Spanish* would naturally have two parents: *understanding English* and *Speaking English*, see Figure 4.2. With the above notation, *Speaking Spanish* ≤ *Translating from English to Spanish*.

While in our work [89] we considered skill taxonomies, one can extend this model and consider hierarchical skill models, where a skill has multiple parents. We consider hierarchical structures, where the nodes correspond to skills. It is useful to add some natural restrictions: We consider skill models that have a unique *root* node and any two skills in the hierarchy should have a unique least common ancestor in the hierarchy:

Definition 4.1. (Skill hierarchy) Let S be a set of skill labels $S = \{s_1, s_2, \ldots, s_n\}$. A skill hierarchy is a tuple $\langle S, \leq, r \rangle$ where $\leq$ is a partial order on S that represents the *more specific*

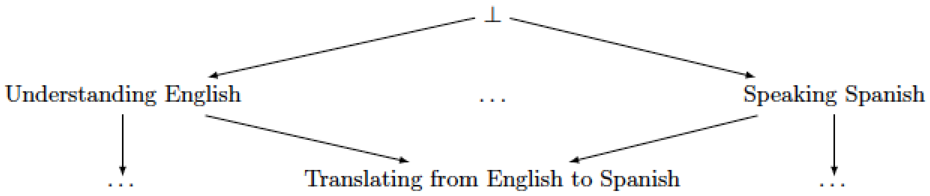


Figure 4.2: Skills with multiple parent nodes

skill relation, $r \in S$ is a root element, and for each pair of skills s_1 and s_2 there exists a unique *least common ancestor* skill $lca(s_1, s_2) \in S$ in the hierarchy.

Such a structure is known as a join semi-lattice [27]. An example for such a skill hierarchy is presented in Figure 4.2. Figure 4.3 depicts a structure that is not a skill hierarchy, because the nodes r_1 and r_2 do not have a common ancestor. Similar hierarchical structures were also used in different contexts, in particular in mathematical psychology and e-Learning systems to represent the knowledge of human learners. The most important hierarchical structures include learning spaces [40], knowledge spaces [34]. While these objects use similar underlying structures, they represent human knowledge at a different level of granularity. In their context, a node could represent the ability of a person to answer a multiple-choice question in a test. Also there models are used for other purposes and do not analyse skill distances as we do in the following section. The child relation represents the partial order on the skill labels.

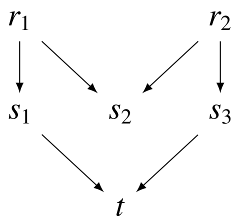


Figure 4.3: Structure that is not a skill hierarchy (based on our Definition 4.1)

Distance between skills

There are different ways one could define distances between skills in a hierarchy. Intuitively, some skills are very similar to each other (such as *Java 1.8 thread* and *Java 1.8 lambda expressions*), while others might seem to be very distant (such as *Java 1.8 thread* and *gardening*). We discuss here several possible distance metrics. There metrics were inspired by some known metrics in the context of concept hierarchies. As in the case of concept hierarchies, there is no single metrics that captures all the subtilities of skill similarity. We discuss the advantages or the potential inconveniences of these metrics in our research report [90].

The distance metrics we discuss make use the concept of information content. Information content describes the amount of information we acquire in case of observing a particular event. Information content[6] is defined as follows:

$$IC(s) = -log_2(P(s)) \tag{2.1}$$

where $P(s)$ is the probability of observing s. That is, if we select a task, how probable is that the skill s is required for this task. Crowdsourcing platforms that dispose a large number of tasks annotated with required skills could use this metrics to characterize the information content for a given skill. In particular, if the number of tasks with the skill label s is N_s, while N is the total number of tasks, then one could the estimate $IC(s)$ as $IC(s) = -log(\frac{N_s}{N})$. This estimation can better be based on a history of proposed task at the platform and should not depend on the currently available tasks. If tasks can have multiple required skills, then N should also consider multiplicities[7].

As in the context of concept hierarchies, besides information content related definitions, researchers also defined distances based on the distances of the corresponding nodes in the concept hierarchy. As we rely on a skill hierarchy that is not necessarily a tree, we need to define the depth of a node, since there could be several paths from the root to a given node. We define *depth*(s), the *depth* of a given skill s in the hierarchy, as the longest path from the root node r to s.

As our goal is to use the metrics to relate required skills for a task and available skills of workers, we use normalized versions of distance metrics knowns for concept hierarchies. In the normalization process, we need to make special attention to avoid division by 0. Below we give the definitions of the metrics, while Table 4.1 gives an overview of distance measures

Definition 4.2. Lin distance.

$$d_L(s_1, s_2) = \begin{cases} 0 & \text{if } s_1 = s_2 = r \\ 1 - \frac{2IC(lca(s_1,s_2))}{IC(s_1)+IC(s_2)} & \text{otherwise.} \end{cases} \tag{2.2}$$

Definition 4.3. (Wu-Palmer distance)

$$d_W(s_1, s_2) = \begin{cases} 0 & \text{if } s_1 = s_2 = r \\ 1 - \frac{2depth(lca(s_1,s_2))}{depth(s_1)+depth(s_2)} & \text{otherwise.} \end{cases} \tag{2.3}$$

Table 4.1 summarizes the distance metrics we use.

4.3 Task assignment problem

We can now formulate the particular task assignment problem in the context of crowdsourcing, that is a specific form of the matching problem. For this, we need to specify the distance or

[6] Information content is also referred in the literature as self-information or surprise. Information entropy is a closely-related concept: it is the expected value of the information content.

[7] Very rare skills could have high information content, that could distort the skill distance metrics we define later in this section. To avoid this problem, if the probability estimate for a skill s is smaller than 1/128 ($P(s) < 1/128$) then we do not use the above estimation, but we set $P(s) = 1/128$ that corresponds to $IC(s) = 7$, since $128 = 2^7$.

Normalized Resnik distance ([111])	$d_R(s_1, s_2) = 1 - \frac{IC(lca(s_1,s_2))}{IC_{max}}$
Lin distance ([86])	$d_L(s_1, s_2) = 1 - \frac{2IC(lca(s_1,s_2))}{IC(s_1)+IC(s_2)}$
Wu-Palmer distance ([130])	$d_W(s_1, s_2) = 1 - \frac{2depth(lca(s_1,s_2))}{depth(s_1)+depth(s_2)}$
Mavridis *et al.* distance ([89]) distance	$d_M(s_1, s_2) = 1 - \frac{depth(lca(s_1,s_2))}{depth_{max}}$

Table 4.1: Distance measures

similarity between a worker with a specific skill profile and a task with a list of required skills. In our work [89] we considered a single skill model, but we extended this to multiple skill models in [90].

We would like to characterize how well a particular worker is suited to execute a given task. We assume that tasks requester specifies what are the necessary skills to complete the task. We will use the following notation: for a task t the set $skills(t)$ denotes the necessary skills for the task t, i.e. $skills(t) = \{s_{t_1}, \ldots, s_{t_N}\}$. We also assume that each worker w has a skill profile, that contains a list of skills where he has expertise. We denote by $skills(w)$ the skill profile of worker w, that is $skills(w) = \{s_{w_1}, \ldots, s_{w_M}\}$. We assume that workers and task proposers use the same vocabulary of the available skill hierarchy. We would like to relate the required skills for a task and the available skills of a worker. We discuss the different possible versions to define the distances.

Tasks with only one required skill

Let us consider first a simple case where there is only one skill s_t required for a task t that is $|skill(t)| = 1$.

Let the skill s_w be a skill in the profile of a worker w that is $s_w \in skills(w)$. If, for example, the skill s_w is a child node of the skill s_t in the skill hierarchy, then s_w is more specific than s_t (that is $s_t \leq s_w$). In this case one could argue that the worker experienced with a more specialized task that is required so one could let him work on the task. If his skill is not a child node in the skill hierarchy, but not far from the node (in terms of the distances we discussed in Section 4.2) then he is likely able to execute the task successfully. We can use this observation to define the distance between a task and a worker. A worker w might have several skills in his profile: we define the distance between a task t and a worker w as the minimum distance of worker skills to the task. We will rely on the distances between skills that we discussed in Section 4.2.

Definition 4.4. Task-worker distance (tasks with one required skill)

$$D(t, w) = \begin{cases} 0 & \text{if } \exists s_w \in skill(w) \text{ such that } skill(t) = s_t \leq s_w, \\ \min\limits_{s \in skill(w)} d(s_t, s) & \text{otherwise.} \end{cases} \tag{3.4}$$

The distance d in the Definition 4.4 is one of the distances that we discussed in the previous section Section.

Multiple required skills

In a more general case, task requesters might want to specify multiple necessary skills for a given task t. In particular, knowledge-intensive macrotasks might require multiple skills . One could try to decompose such macrotasks into multiple microtasks, however such decomposition is not always possible or easy [54], [116].

We denote the set of required skills for a given task t by $skills(t)$ In particular, knowledge-intensive tasks might require multiple skills. For example, is someone needs a simulation of the quantum states of the hydrogen atom, he might specify the need for a skill of understanding the physics of the hydrogen atom, programming skills, and English writing skills to write a documentation of the project. Here we could envisage different scenarios, depending on the particular preferences of the requester.

- He might want that the worker has all of the required skills,
- He might be satisfied with someone who has one of the skills (e.g. programming) and in case the worker has strong development skills, he might care less about the other,
- He might prefer a worker who has multiple skills whose average distance to the task is small,
- He might specify a minimum skill requirement for each of the skills. That is he would like that the task is assigned to someone who fits best, but who has some constraints about the distance to each of the required skills.

Depending on the application setting and the requesters preferences one could prefer different definitions. In the following we give a definition where the task-worker distances are aggregated using the *max* function. This correspond to a preference where each of the skills is required. The distance between a worker w and a task t is determined by the largest distance between the skills in the profile and the required skills.

Definition 4.5. Task-worker distance (tasks with multiple required skills)

$$D(t,w) = \begin{cases} 0 & \text{if } \forall s_t \in skill(t) \exists s_{t_w} \in skill(w), \text{ such that } s_t \leq s_{t_w} \\ \max\limits_{s_t \in skill(t)} \min\limits_{s_w \in skill(w)} d(s_t, s_w) & \text{otherwise.} \end{cases} \tag{3.5}$$

The task assignment problem in crowdsourcing

Given a set of tasks and participants, a task assignment $\mathcal{A}$ is a mapping from T to P that maps a task $t \in T$ to $\mathcal{A}(t) = p \in P$. A task assignment is partial (a task may not be assigned) and injective (a participant can only perform one task during this assignment). As a participant can only participate in one task at a time, the maximum number of tasks that can be assigned is $\min(|T|, |P|)$. Indeed, if there are less tasks than participants, some participants may not be assigned. We focus here on *covering task assignments*, where the available workforce is maximally assigned: the number of assigned tasks is $\min(|T|, |P|)$.

Finally, the quality of an assignment $\mathcal{A}$ is measured by the *cumulative distance* $\mathcal{D}(\mathcal{A})$, which is the sum of distances between each pair of assigned tasks and participants, i.e.:

$$\mathcal{D}(\mathcal{A}) = \sum_{(t,p) \text{ s.t. } \mathcal{A}(t)=p} D(t, p). \tag{3.6}$$

The *normalized cumulative distance* is $\mathcal{D}(\mathcal{A})$ divided by the total number of assigned participants. With this definition, the closer the participants are to the required skill of their task, the smaller is the distance and the better is the assignment.

We make a certain number of assumptions about the task assignment process that we summarize here. We assume that we assign a single worker to a task. We assume that a worker either has or does not have a certain skill, we do not address expertise levels. We also make the assumption that a skill hierarchy is available and it is used to annotate the tasks and also used to construct the skill profiles. Finally, we assume that the skill profiles are good indicators of quality.

We defined the task assignment problem as follows.

Definition 4.6. (Optimal Covering Task Assignment Problem)
`INPUT` : a taxonomy S, a set of tasks T and participants P, *skill* functions.
`OUTPUT`: a covering task assignment $\mathcal{A}$ such that $\mathcal{D}(\mathcal{A})$ is minimized.

4.4 Algorithms and evaluation methods

We presented a model to formulate the task assignment problem in crowdsourcing in the framework of minimal bipartite graph matching problem. Our model relies on a hierarchical skill model that is used at the crowdsourcing platform to annotate tasks and to construct skill profiles for workers. This formulation enables to use the known algorithms for this problem, in particular the Hungarian method [80]. This method can compute an optimal solution, that is a minimum-weight perfect match for a bipartite graph in polynomial time.

Nevertheless, we also proposed heuristic algorithms for the (Optimal Covering Task Assignment Problem, as one might face limitations with the optimal algorithm, because the involved distance matrix could be impractically large. We have proposed different heuristics in [90] and in [89], which are specific variants of the greedy algorithm.

The evaluation showed important improvements in the assignment quality, and constructing the matching was reasonably fast, such that computing the matching is feasible for crowdsourcing platforms. We present here some representative graphs, a more complete experimental evaluation is discussed in our papers [90] and in [89]. Our baselines were the random assignments, where tasks are affected without considering information on skills and the ExactThenRandom technique, that models a simple keyword based matching. If the keyword of the required skill appears in the skill profile of a worker, he is a candidate for this task otherwise the affectation is random.

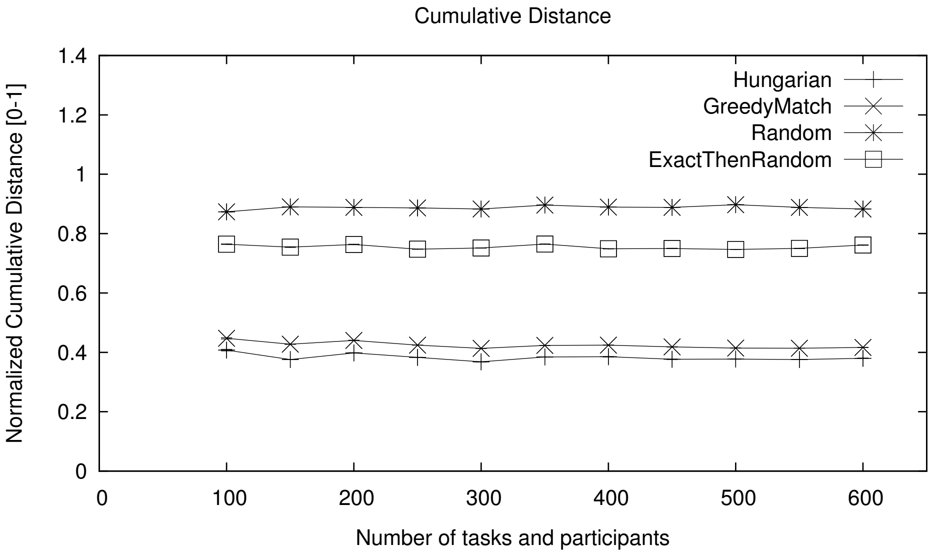


Figure 4.4: Normalized cumulative distance of assignment with respect to the number of participants. Our baselines are the methods *Random* (that corresponds to a strategy for assigning tasks to workers without considering the skill requirements) and *ExactThenRandom* that is a way to simulate the string matching based methods (if a skill is in the profile, we use it for assignment, otherwise we rely on a random assignment)). Our methods that use the skill hierarchy are the *Hungarian* (that gives the optimal matching, based on the Hungarian method) and the *GreedyMatch* that is a greedy heuristics, using our hierarchical skill model, described in [90].

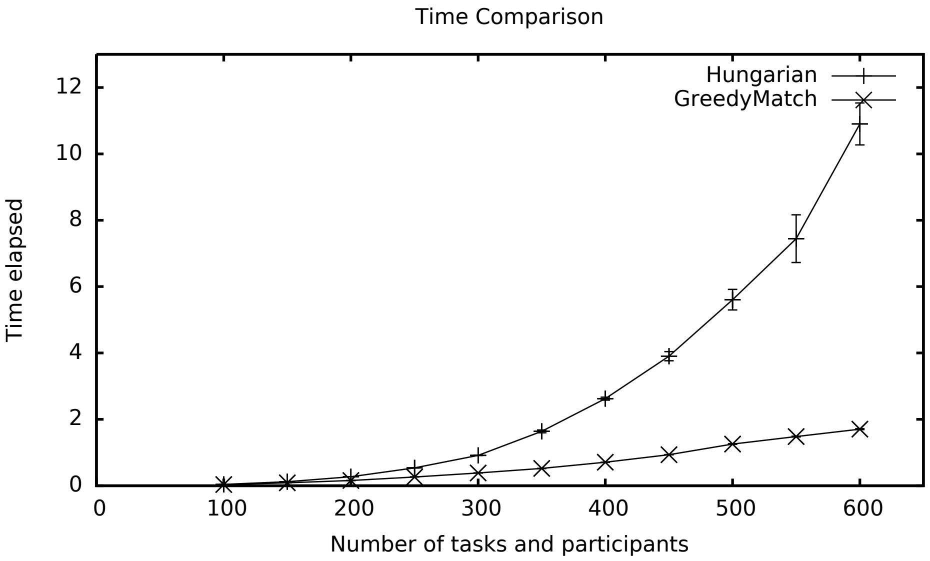


Figure 4.5: Running times of the algorithm with respect to the number of participants.

4.5 Limitations and perspectives

Our model has certainly some limitations.

Multiple skills: While it is natural to assume that a task requires multiple skills, some of these might be more important than others, while in other cases a diversity of skills is preferable to an expertise in specific areas. Our task/worker similarity estimation could model this. We did not explore further this area also because we do not have specific data to validate the specific possible functions.

Cold start and skill updates: We assume the existence of skill profiles. If we do not have them, one could envisage to create and update user profiles as workers complete the tasks. Our model would require further adaptations to deal with this.

Level of expertise of workers Our model considers that a worker has or has not a specific skill, but does not deal with the level of expertise. Dealing with this issue also would need further adaptations to the model.

Skill hierarchies: There is a number of taxonomies available, however they might not have the right granularity w.r.t. a particular task set or available skill profiles. One could construct suitable skill hierarchy, with the help of data mining methods, that is potentially not an easy task, but we did not explore these methods.

Task assignment vs. task selection: In some crowdsourcing platforms, the tasks are not affected but the workers can select themselves based on their preferences.

Worker availability: Workers might not be available to complete tasks only for a certain period of time. One could apply the task affectation methods at a given moment for the available workers and tasks and then repeat the affectation later if needed. This strategy poses further questions, such as how to avoid the frequent solicitation of certain highly-skilled workers.

Chapter 5

Conclusion and perspectives

This habilitation thesis gives an overview of our research efforts in the past years. We developed new ways to address the data matching problems, where we need to establish correspondences between terms that appear in data, originating from independent sources. The correspondences should reflect our human understanding of the meaning of the involved terms. In particular, we proposed novel solutions to the entity matching problem, for Web documents. We modelled the reconciliation phase for schema matching networks and we proposed ways to improve this phase. Our work on knowledge-intensive crowdsourcing platforms aims to improve the quality of obtained data through better task affectation. In the following we give an overview of our ongoing research and we discuss some of our plans for future directions.

5.1 Ongoing work

Context

Currently, I (co-)supervise 4 PhD students: Ian Jeantet, from December 2017 (with David Gross-Amblard, financed through the projet ANR EPIQUE), Rituraj Singh, from January 2018 (with Loïc Hélouët, financed through the projet ANR HEADWORK), Francois Mentec, from October 2019 (with David Gross-Amblard financed through CIFRE with the company ALTEN) and Maria Massri, from October 2019 (with David Gross-Amblard financed through CIFRE with the company Orange). Moreover, I am a mentor for Mickael Foursov, Maître de conférences at University of Rennes 1, who restarted his research activities after a long pause.

Understanding evolving graphs

Graphs can be used to model a number of phenomena, from the Internet to systems biology or to social sciences. Analysing the properties of these networks already gave a lot of insights about a large number of phenomena [84]. The underlying networks in these models can evolve and change dynamically [60], [36]. Understanding the nature of these evolutions and making sense of changes in a large network is a challenging task, given the large size and complex structure of these networks. Our ongoing efforts try to develop methods and tools to support the analysis of the dynamics of such networks.

In our ongoing research project (ANR EPIQUE[1]) we develop new methods and tools that can help social scientists in understanding the evolution of scientific fields, with the help of text analysis techniques that we apply on publication databases. These large collections of scientific publications enable us to extract the scientific terms and analyse their co-utilisation. We can do this analysis for each year and then analyse the evolution of the co-occurrence networks. Indeed, this is the work we are following with Ian Jeantet.

We work on identifying and extracting the hierarchical structure of scientific terms. Analysing the evolution of groups of words (organized in hierarchies) could lead to more robust estimations whether two scientific fields come closer over the time and also a hierarchical organization of the terms can simplify the interpretation of the results. We have developed a hierarchical clustering technique that can deal with overlapping clusters. We are also analysing how to correlate and match the obtained hierarchical structures between consecutive years (year n and $n+1$). We can view this problem also as a specific data matching problem where we need to match hierarchical structures. However, this problem is different from the data matching problems that we discussed in this habilitation thesis, since here we can match the terms in the two structures easily (as they are identical, except a small fraction of the vocabulary), but they are organized in slightly different ways. We explore the use of graph convolutional networks [29], [141] in this context.

In the context of the ANR EPIQUE project, I also coordinate the research efforts of Mickael Foursov. With him we try to understand the evolution of term co-occurrence graphs through the use of methods of spectral graph theory [21]. In particular, we try to analyse the changes in the spectrum of the Laplacian of the term co-occurrence graph [96].

There are a number of other areas that can be modelled through dynamic networks. For example, telecommunication companies would like to understand the connectivity graphs of IoT devices. We have launched a collaboration with OrangeLabs where we try to understand efficient ways of storing, querying and analysing dynamic networks of connected IoT devices. In the course of this collaboration I supervise the thesis of Maria Massri. With her we work on generating temporal graphs to construct suitable benchmarks for temporal graph databases. Such graph generation methods were recently proposed by Bagan et al. [6], for graph databases, without the temporal aspects. Besides model based generation of workloads, we plan to use machine learning methods, to generate large graphs that have the same characteristics as the smaller samples of available graphs that was constructed from the analysis of a modes set of IoT devices.

Crowdsourcing and future of work

Crowdsourcing platforms enable to affect simple tasks to humans. In this way, one can realize specific tasks that require human intelligence that cannot be completed through computers alone. These platforms were also used to construct large labelled datasets (such as the ImageNet [31] dataset) that could be used for training supervised classification methods.

We estimate that besides simple image labelling tasks, there will be different types of crowdsourcing tasks that orient towards other aspects of human intelligence. Moreover, one could

[1] `https://iscpif.fr/epique/`

imagine not only to affect simple tasks to workers but also to request a small workflow from the crowd, who could even communicate or collaborate to realize the tasks. In our ongoing work, we are working on this question. A particularly important aspect here is that the workflow involves data: the results obtained from some crowd workers could be the input to another task, realized by other workers. For example, a crowd worker could transcribe the text that one can find in some images and another worker translate it to another language. I work on this problems with Rituraj Singh. We have developed a model for such workflows and we have studied the properties of these models. We need to make sure that not only individual tasks but the overall workflow delivers results of sufficiently high quality. We realize this research in the context of our other ongoing project ANR HEADWORK[2].

Recommendation services for human resources

Matching worker profiles and job offers is not only relevant for crowdsourcing task assignment. One faces such problems on the daily basis at a human resources (HR) department of a consulting company. In this setting, one would like to obtain a matching between a job profile and CV of a candidate. Today, this matching is done by humans (HR experts) on a daily basis, based on a large set of tacit knowledge. We estimate that one could change the way the HR experts work, if we make the tacit knowledge more explicit and build tools that can give recommendations to HR experts. We started a collaboration with the company ALTEN[3] to develop methods that can derive recommendations to HR experts. I co-supervise the thesis of Francois Mentec in the context of this collaboration. This thesis will build on his master thesis that he realized at the same company that I co-supervised as well.

5.2 Perspectives for future research

The fields of data management and artificial intelligence have a number of further research opportunities.

Semantic heterogeneity problems arise in various new application contexts. For example, enterprises try to integrate larger and larger amounts of data to extract and mine business relevant information. Data lakes[4] (such as the Azure Data Lake Store [108]) offer methods for easy data integration. In such a setting, one needs to address the semantic heterogeneity issues, at extreme scales. This setting requires robust methods, that can tolerate the incompleteness and contradictory facts in the data. Data lakes are of course only one example in this context, semantic heterogeneity problems arise in various other “BigData” applications as well.

Besides the numerous and important practical problems, we lack a more general theory for understanding the semantics of data. While formal representations of semantics (for example, ontologies) play an important role in a lot of specific domains, it is unlikely that we can have such formal models for all possible cases where we need to deal with data integration. The quest

[2] `http://headwork.gforge.inria.fr/`

[3] `https://www.alten.fr/`

[4] `https://www.pwc.com/us/en/technology-forecast/2014/cloud-computing/assets/pdf/pwc-technology-forecast-data-lakes.pdf`

for understanding the semantics of data has a long history. Woods [129] in 1975 tries to define data semantics as "the meaning and the use of data" in his seminal work on semantic networks, but we need more precise notion of these concepts. A. Sheth [119] summarizes the different views perspectives on data semantics. There are diverging views how should we approach data semantics: as a set of relationship between objects [127] or by describing the similarities between objects [8]. The paper [12] also gives a survey of some of the approaches. In general, understanding data semantics is a challenging problem as it is closely linked to natural language semantics and other questions in artificial intelligence, including common sense reasoning.

We would like point out here that information theory of Shanon, that revolutionized various fields, focuses on the transmission of messages and avoids making reference to their meaning [118]: *"Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem."* The engineering problem Shanon refers to in the above citation is the reconstruction of messages between a sender and a receiver. We now face another engineering problem that focuses on establishing common understanding of the meaning that we attach to symbols that are present in different datasets or exchanged between autonomous agents.

When we try to understand the semantics of pieces of information, we should also study its utility: having a shared understanding of the meaning of exchanged symbols (for example, words in a natural language) enables an efficient communication. Moreover, we should not consider the semantic of terms static, but rather dynamic. While one could have the impression that the semantic of terms in our language is static, linguists agree that the meaning of the words change [124]. This phenomenon is known as semantic change or semantic shift. The syntax of our languages and the meaning of the used terms is a result of an evolutionary development [68], [85], even if there are different models of this evolution. The Web and the various available datasets, offer a laboratory for analysing and understanding this process, for example Steels [121] has studied how a group of users of a collaborative tagging system developed a common understanding of terms. Aberer *et al.* have analysed the emergence of semantic understanding in large peer-to-peer systems. We believe that one can follow up these works and develop a more complete understanding of some specific aspects of the evolution or emergence of shared knowledge, in large-scale systems. We can also view our ongoing work on dynamic graphs from this perspective: we try to understand the emergence of specific scientific domains. Of course, for understanding the emergence of semantics, we should analyse a longer history of language development. We think that analysis of dynamic graphs will lead to a better understanding in a number of other fields, including the study of collective intelligence [98] or the emergence of hierarchies [138]. We could combine the analysis of graph dynamics with other techniques including graph signal processing [120], [115]. We think that an evolutionary perspective has a high potential to make advances in a wide range of areas.

Human experts do not only understand the semantics of terms, but in their work they also reason using their knowledge. In our work we used various computational models of reasoning. Some models of the human mind also rely on probabilistic reasoning [30]. How such probabilistic reasoning models could emerge from more basic components such as prediction -that are considered as a basic task of the mind [22]- is not yet clear. However, reasoning methods inspired by human reasoning [81] were successfully used for concept learning. A better understanding of how children learn [82] could also lead to new insights. Even if we cannot build copy the reasoning methods of human mind, understanding the basic principles of human reasoning could lead to better computational reasoning techniques. Combining symbolic reasoning with machine learning approaches [88] is also a very promising direction of research.

Deep learning techniques require a large amounts of data for training [50]. In most cases, the trained models cannot be transferred from one domain to another. Self-supervised methods [26] provide an alternative, but they still require a lot of data that is not available in all domains. This potentially large demand for labelled or unlabelled data could generate new requirements for crowdsourcing platforms. For example, tasks that are more complex or challenging could attract more attention. Mixing machine learning and human judgment does not necessarily lead to better results [125]. We should develop a better understanding, how to decompose tasks between humans and algorithms. These questions are challenging and require interdisciplinary efforts. One should however move away from the model where human-in-the-loop systems only rely on humans as a source of labels and we should develop new models where people can contribute with their full creativity, and where can create new jobs [75]. One needs to address another societal challenge in this context: to assure the fairness based on our societal norms. The data that we use to train machine learning models contains errors and biases [15] an these biases could even be amplified through the use of algorithms. This creates new challenges and an ethical dimension for algorithm design [76].

Bibliography

[1] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth. Start making sense: The Chatty Web approach for global semantic agreements. *Journal of Web Semantics*, 1(1):89–114, 2003.

[2] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[3] Enrique Amigó, Javier Artiles, Julio Gonzalo, Damiano Spina, Bing Liu, and Adolfo Corujo. Weps3 evaluation campaign: Overview of the on-line reputation management task. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[4] Grigoris Antoniou, Paul Groth, Frank van van Harmelen, and Rinke Hoekstra. *A Semantic Web Primer*. The MIT Press, 2012.

[5] Dan Ariely. *Predictably Irrational, Revised and Expanded Edition: The Hidden Forces That Shape Our Decisions*. Harper Perennial, 2010.

[6] Guillaume Bagan, Angela Bonifati, Radu Ciucanu, George H. L. Fletcher, Aurélien Lemay, and Nicky Advokaat. Generating flexible workloads for graph databases. *Proc. VLDB Endow.*, 9(13):1457–1460, September 2016.

[7] Henning Beck. *Scatterbrain: How the Mind's Mistakes Make Humans Creative, Innovative, and Successful*. Greystone Books, 2019.

[8] David Beech. Data semantics on the information superhighway. In *Database Applications Semantics, Proceedings of the Sixth IFIP TC-2 Working Conference on Data Semantics (DS-6), Stone Mountain, Atlanta, Georgia, USA, May 30 - June 2, 1995*, pages 12–33, 1995.

[9] Ron Bekkerman and Andrew McCallum. Disambiguating Web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470, 2005.

[10] Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors. *Schema Matching and Mapping*. Data-Centric Systems and Applications. Springer, 2011.

[11] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18(1):255–276, January 2009.

[12] Alexander Borgida and John Mylopoulos. Data semantics revisited. In Christoph Bussler, Val Tannen, and Irini Fundulaki, editors, *Semantic Web and Databases*, pages 9–26, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[13] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. 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, 2013.

[14] Erik Brynjolfsson. The productivity paradox of information technology. *Commun. ACM*, 36(12):66–77, December 1993.

[15] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[16] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 77–90, 1977.

[17] Surajit Chaudhuri, Venkatesh Ganti, and Rajeev Motwani. Robust Identification of Fuzzy Duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pages 865–876, 2005.

[18] Zhaoqi Chen, Dmitri V. Kalashnikov, and Sharad Mehrotra. Adaptive graphical approach to entity resolution. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 204–213, 2007.

[19] François Chollet. On the measure of intelligence. `https://arxiv.org/abs/1911.01547`, 2019.

[20] Peter Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Publishing Company, Incorporated, 2012.

[21] F R K Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[22] Andy Clark. *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. Oxford University Press, 2016.

[23] E.F. Codd. Relational database: A practical foundation for productivity. In John Mylopolous and Michael Brodie, editors, *Readings in Artificial Intelligence and Databases*, pages 60 – 68. Morgan Kaufmann, San Francisco (CA), 1989.

[24] P. Cudré-Mauroux, Karl Aberer, and A. Feher. Probabilistic message passing in peer data management systems. In *ICDE*, page 41, 2006.

[25] Philippe Cudré-Mauroux. *Emergent Semantics*, pages 982–985. Springer US, Boston, MA, 2009.

[26] Yann Le Cun. *Quand la machine apprend (in French)*. Odile Jacob, 2019.

[27] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition, 2002.

[28] Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[29] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[30] Stanislas Dehaene. *Les talents du cerveau, le défi des machines (in French)*. Odile Jacob, 2018.

[31] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[33] Hong Hai Do and Erhard Rahm. COMA - A System for Flexible Combination of Schema Matching Approaches. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB'02)*, pages 610–621, 2002.

[34] Jean-Paul Doignon and Jean-Claude Falmagne. *Knowledge Spaces*. Springer, 1999.

[35] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 85–96, 2005.

[36] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, 2003.

[37] Jacob Eisenstein. *Introduction to Natural Language Processing*. MIT press, 2019.

[38] Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. Answer set programming: A primer. In *Reasoning Web*, pages 40–110, 2009.

[39] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.

[40] Jean-Claude Falmagne and Jean-Paul Doignon. *Learning Spaces*. Springer, 2010.

[41] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1999.

[42] Iván Fellegi and Alan Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, December 1969.

[43] A. Gal, T. Sagi, M. Weidlich, E. Levy, V. Shafran, Z. Miklós, and N.Q.V. Hung. Making sense of top-k matchings: A unified match graph for schema matching. In *Proceedings of SIGMOD Workshop on Information Integration on the Web (IIWeb'12)*, 2012.

[44] Avigdor Gal. *Uncertain Schema Matching*. Morgan & Calypool Publishers, 2011.

[45] Avigdor Gal, Michael Katz, Tomer Sagi, Matthias Weidlich, Karl Aberer, Zoltan Miklos, Nguyen Quoc Viet Hung, Eliezer Levy, and Victor Shafran. Completeness and Ambiguity of Schema Cover. In *21st International Conference on Cooperative Information Systems (CoopIS 2013)*, 2013.

[46] Howard Gardner. *Frames of mind: The theory of multiple intelligences*. Basic Books, 1983.

[47] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070–1080. MIT Press, 1988.

[48] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *Journal of New Generation Computing*, 9(3/4):365–386, 1991.

[49] W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1996.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[51] Georg Gottlob, Gianluici Greco, Zoltán Miklós, Francesco Scarcello, and Thomas Schwentick. *Graph Theory, Computational Intelligence and Thought. Essays Dedicated to Martin Charles Golumbic on the Occasion of His 60th Birthday*, volume 5420 of *LNCS*, chapter Tree Projections: Game Characterization and Computational Aspects, pages 87–99. Springer, 2009.

[52] Georg Gottlob, Zoltan Miklos, and Thomas Schwentick. Generalized Hypertree Decompositions: NP-hardness and Tractable Variants. *Journal of the ACM*, 56(6):1–32, September 2009.

[53] David Alan Grier. *When Computers Were Human*. Princeton University Press, Princeton, NJ, USA, 2007.

[54] Daniel Haas, Jason Ansel, Lydia Gu, and Adam Marcus. Argonaut: Macrotask crowdsourcing for complex data processing. *Proc. VLDB Endow.*, 8(12):1642–1653, August 2015.

[55] Alon Halevy. Why your data won't mix. *Queue*, 3(8):50–58, October 2005.

[56] Yuval Noah Harari. *Homo Deus: a brief history of tomorrow*. Harper Collins Publishers, 2016.

[57] Mauricio A. Hernández and Salvatore J. Stolfo. The merge/purge problem for large databases. *ACM SIGMOD Record*, 24(2):127–138, May 1995.

[58] Pascal Hitzler, Markus Krtzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 1st edition, 2009.

[59] Erik Hollnagel, David D. Woods, and Nancy Leveson. *Resilience Engineering - Concepts and Precepts*. Ashgate Pub Co, 2006.

[60] Petter Holme and Jari Saramäki, editors. *Temporal Networks*. Springer, 2013.

[61] Boyi Hou, Qun Chen, Jiquan Shen, Xin Liu, Ping Zhong, Yanyan Wang, Zhaoqiang Chen, and Zhanhuai Li. Gradual machine learning for entity resolution. In *The World Wide Web Conference*, WWW '19, pages 3526–3530, New York, NY, USA, 2019. ACM.

[62] Nguyen Quoc Viet Hung, Xuan Hoai Luong, Zoltan Miklos, Tho Thanh Quan, and Karl Aberer. Collaborative Schema Matching Reconciliation. In *21st International Conference on Cooperative Information Systems (CoopIS 2013)*, 2013.

[63] Nguyen Quoc Viet Hung, Xuan Hoai Luong, Zoltan Miklos, Tho Quan Thanh, and Karl Aberer. An MAS Negotiation Support Tool for Schema Matching (Demonstration). In *Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'2013)*, 2013.

[64] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Zoltan Miklos, and Karl Aberer. On Leveraging Crowdsourcing Techniques for Schema Matching Networks. In *DASFAA 2013*, 2013.

[65] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Zoltán Miklós, and Karl Aberer. Reconciling schema matching networks through crowdsourcing. *EAI Endorsed Trans. Collaborative Computing*, 1(2):e2, 2014.

[66] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Zoltan Miklos, Karl Aberer, Avigdor Gal, and Matthias Weidlich. Pay-as-you-go Reconciliation in Schema Matching Networks. In *30th International Conference on Data Engineering (ICDE 2014)*, 2014.

[67] Nguyen Quoc Viet Hung, Matthias Weidlich, Nguyen Thanh Tam, Zoltán Miklós, Karl Aberer, Avigdor Gal, and Bela Stantic. Handling probabilistic integrity constraints in pay-as-you-go reconciliation of data models. *Information Systems*, 83:166 – 180, 2019.

[68] James R. Hurford. *The Origins of Meaning*. Studies in the Evolution of Language. Oxford University Press, 2007.

[69] Yannis E. Ioannidis. Query optimization. *ACM Comput. Surv.*, 28(1):121–123, March 1996.

[70] Ekaterini Ioannou, Saket Sathe, Nicolas Bonvin, Anshul Jain, Srikanth Bondalapati, Gleb Skobeltsyn, Claudia Niederée, and Zoltán Miklós. Entity Search with NECESSITY (demo paper). In *12th International Workshop on the Web and Databases (WebDB 2009)*, 2009.

[71] Ekaterini Ioannou and Yannis Velegrakis. Embench^{++}: Data for a thorough benchmarking of matching-related methods. *Semantic Web*, 10(2):435–450, 2019.

[72] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 64–67, New York, NY, USA, 2010. ACM.

[73] Panagiotis G. Ipeirotis, Vassilios S. Verykios, and Ahmed K. Elmagarmid. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, January 2007.

[74] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, 2008.

[75] Jerry Kaplan. *Humans Need Not Apply: A Guide to Wealth and Work in the Age of Artificial Intelligence*. Yale University Press, New Haven, CT, USA, 2015.

[76] Michael Kearns and Aaron Roth. *The Ethical Algorithm. The Science of Socially Aware Algorithm Design*. Oxford University Press, 2019.

[77] Kevin Kelly. *Inevitable: understanding the 12 technological forces that will shape our future*. Penguin Random House LLC, 2016.

[78] Ph. G. Kolaitis and M. Y. Vardi. Conjunctive query containment and constraint satisfaction. *J. Comput. System Sci.*, 61:302–332, 2000.

[79] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theor.*, 47(2):498–519, September 2006.

[80] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[81] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338, Dec 2015.

[82] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *CoRR*, abs/1604.00289, 2016.

[83] George Lakoff. *Woman, Fire, and Dangerous Things. What Categories Reveal about the Mind.* The University of Chicago Press, 1987.

[84] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex Networks: Principles, Methods and Applications*. Cambridge University Press, 2017.

[85] Philip Lieberman. *Toward an evolutionary biology of language*. Belknap Press of Harvard University Press, 2006.

[86] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

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

[88] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[89] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 843–853, 2016.

[90] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. (under review), 2019.

[91] Pamela McCorduck. *Machines who think : a personal inquiry into the history and prospects of artificial intelligence*. A K Peters/CRC Press, 2004.

[92] David Menestrina, Omar Benjelloun, and Hector Garcia-Molina. Generic Entity Resolution with Data Confidences. In *In First International VLDB Workshop on Clean Databases*, 2006.

[93] Hugo Mercier and Dan Sperber. *The Enigma of Reason*. Harvard University Press, 2019.

[94] Zoltán Miklós. *Understanding tractable decompositions for constraint satisfaction*. PhD thesis, University of Oxford, 2008.

[95] Zoltán Miklós, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, Antonio Mana, Claudia Niederée, Themis Palpanas, and Heiko Stoermer. From Web Data to Entities and Back. In *The 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10)*, volume 6051 of *LNCS*, pages 302–316. Springer, 2010.

[96] Zoltan Miklos, Mickaël Foursov, Franklin Lia, Ian Jeantet, and David Gross-Amblard. Understanding the evolution of science: analyzing evolving term co-occurrence graphs with spectral techniques. Third international workshop on advances on managing and mining evolving graphs (LEG@ECMLPKDD), September 2019.

[97] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[98] Geoff Mulgan. *Big Mind: How Collective Intelligence Can Change Our World*. Princeton University Press, Princeton, NJ, USA, 2017.

[99] Quoc Viet Hung Nguyen, Tri Kurniawan Wijaya, Zoltan Miklos, Karl Aberer, Eliezer Levy, Victor Shafran, Avigdor Gal, and Matthias Weidlich. Minimizing Human Effort in Reconciling Match Networks. In *32nd International Conference on Conceptual Modeling (ER 2013)*, 2013.

[100] C. Ogden and I. Richards. *The Meaning of Meaning - A Study of the Influence of Language upon Thought and of the Science of Symbolism.* Routledge, 1923.

[101] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. Comparative analysis of approximate blocking techniques for entity resolution. *Proc. VLDB Endow.*, 9(9):684–695, May 2016.

[102] Aditya G. Parameswaran, Hector Garcia-Molina, Hyunjung Park, Neoklis Polyzotis, Aditya Ramesh, and Jennifer Widom. Crowdscreen: algorithms for filtering data with humans. In *SIGMOD*, pages 361–372, 2012.

[103] Kyung S Park. *Human reliability : analysis, prediction, and prevention of human errors*. Elsevier Science Ltd, 1986.

[104] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[105] E. Peukert, J. Eberius, and E. Rahm. AMC - A framework for modelling and comparing matching systems as matching processes. In *Proceedings of the 27th International Conference on Data Engineering (ICDE'11)*, pages 1304–1307, 2011.

[106] Kun Qian, Lucian Popa, and Prithviraj Sen. Active learning for large-scale entity resolution. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1379–1388, New York, NY, USA, 2017. ACM.

[107] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[108] Raghu Ramakrishnan, Baskar Sridharan, John R. Douceur, Pavan Kasturi, Balaji Krishnamachari-Sampath, Karthick Krishnamoorthy, Peng Li, Mitica Manu, Spiro Michaylov, Rogério Ramos, Neil Sharman, Zee Xu, Youssef Barakat, Chris Douglas, Richard Draves, Shrikant S. Naidu, Shankar Shastry, Atul Sikaria, Simon Sun, and Ramarathnam Venkatesan. Azure data lake store: A hyperscale distributed file service for big data analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 51–63, New York, NY, USA, 2017. ACM.

[109] J. Reason. *Human Error*. Cambridge University Press, 1990.

[110] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[111] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research (JAIR)*, 11:95–130, 1999.

[112] Senjuti Basu Roy, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB Journal*, 24(4):467–491, 2015.

[113] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.

[114] Tomer Sagi and Avigdor Gal. In schema matching, even experts are human: Towards expert sourcing in schema matching. In *Workshops Proceedings of the 30th International Conference on Data Engineering Workshops, ICDE 2014, Chicago, IL, USA, March 31 - April 4, 2014*, pages 45–49, 2014.

[115] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, April 2013.

[116] Heinz Schmitz and Ioanna Lykourentzou. Online sequencing of non-decomposable macrotasks in expert crowdsourcing. *Trans. Soc. Comput.*, 1(1):1:1–1:33, January 2018.

[117] Giovanni Seni and John Elder. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan and Claypool Publishers, 2010.

[118] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[119] Amit P. Sheth. Panel: Data semantics: What, where and how? In *Proceedings of the Sixth IFIP TC-2 Working Conference on Data Semantics: Database Applications Semantics*, DS-6, pages 601–610, London, UK, UK, 1996. Chapman & Hall, Ltd.

[120] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013.

[121] Luc Steels. Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21(3):32–38, May 2006.

[122] Stefano Tranquillini, Florian Daniel, Pavel Kucherbaev, and Fabio Casati. Modeling, enacting, and integrating custom crowdsourcing processes. *ACM Trans. Web*, 9(2):7:1–7:43, May 2015.

[123] A. M. Turing. I.—Computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950.

[124] Stephen Ullmann. *Grundzüge der Semantik (in German).* Walter de Gruyter, 1967.

[125] Michelle Vaccaro and Jim Waldo. The effects of mixing machine learning and human judgment. *Commun. ACM*, 62(11):104–110, October 2019.

[126] Damir Vandic, Flavius Frasincar, Uzay Kaymak, and Mark Riezebos. Scalable entity resolution for web product descriptions. *Information Fusion*, 53:103 – 111, 2020.

[127] Gio Wiederhold. Value-added mediation in large-scale information systems. In *Database Applications Semantics, Proceedings of the Sixth IFIP TC-2 Working Conference on Data Semantics (DS-6), Stone Mountain, Atlanta, Georgia, USA, May 30 - June 2, 1995*, pages 34–56, 1995.

[128] Ludwig Wittgenstein. *Philosophical investigations.* Macmillan Publishing Company, 1953.

[129] William Woods. What's in a link: Foundations for semantic networks. *Representation and Understanding*, page 76, 11 1975.

[130] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual meeting of the Associations for Computational Linguistics*, pages 133–138, 1994.

[131] Tingxin Yan and Vikas Kumar. CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. *MobiSys*, pages 77–90, 2010.

[132] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. It was easy, when apples and blackberries were only fruits. In *Third WePS Evaluation Workshop: Searching Information about Entities in the Web, CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[133] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. Towards better entity resolution techniques for Web document collections. In *1st International Workshop on Data Engineering meets the Semantic Web (DESWeb'2010) (co-located with ICDE'2010)*, 2010.

[134] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. What have fruits to do with technology? The case of Orange, Blackberry and Apple. In *International Conference on Web Intelligence, Mining and Semantics (WIMS'2011)*. ACM press, 2011.

[135] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. Quality-aware similarity assessment for entity matching in Web data. *Information Systems*, 37:336–351, 2012.

[136] Surender Reddy Yerva, Zoltán Miklós, Flavia Grosan, Tandrau Alexandru, and Karl Aberer. TweetSpector: Entity-based retrieval of Tweets. In *SIGIR'2012*, 2012. (demo paper).

[137] Surrender Reddy Yerva, Zoltán Miklós, and Karl Aberer. Entity-based Classification of Twitter Messages. *International Journal of Computer Science & Applications*, 9(1):88–115, 2012.

[138] Anna Zafeiris and Tamas Vicsek. *Why We Live in Hierarchies? A Quantitative Treatise*. Springer, 2018.

[139] Ce Zhang and Christopher Ré. Towards high-throughput gibbs sampling at scale: A study across storage managers. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 397–408, New York, NY, USA, 2013. ACM.

[140] Jing Zhang, Jie Tang, and Juan-Zi Li. Expert finding in a social network. In Kotagiri Ramamohanarao, P. Radha Krishna, Mukesh K. Mohania, and Ekawit Nantajeewarawat, editors, *DASFAA*, volume 4443 of *Lecture Notes in Computer Science*, pages 1066–1069. Springer, 2007.

[141] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):11, Nov 2019.

[142] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proc. VLDB Endow.*, 10(5):541–552, January 2017.