POLITECNICO DI MILANO
Department of Electronic, Informatics and Bioengineering
Master of Science in Computer Science and Engineering

Source Authority by Data Ownership: an approach to reliable data fusion

Supervisor: Prof. Letizia Tanca
Co-supervisor: Fabio Azzalini

Author: Leonardo Rossi
Student id: 893992

Academic Year 2018-2019

Dedicato ai miei genitori

Ringraziamenti

In primo luogo vorrei ringraziare la mia famiglia che mi è stata sempre accanto fino a questo momento sostenendomi e consigliandomi nel momento del bisogno. Se sono arrivato fino a qui lo devo soprattutto a loro e allo spirito di dedizione che mi hanno trasmesso in tutti questi anni.

Un altro ringraziamento particolare va ai miei colleghi universitari perché grazie a loro sono riuscito ad affrontare questa sfida con impegno, ma soprattutto con gioia e divertendomi.

Vorrei inoltre ringraziare tutti gli amici di una vita che mi hanno supportato per tutti questi anni e con i quali ho condiviso il mio percorso di crescita.

Infine vorrei ringraziare l'intero corpo docenti in quanto durante il mio corso di studi, grazie al loro lavoro, mi hanno fatto appassionare alle materie studiate stimolando la mia curiosità.

In particolar modo ringrazio la professoressa Letizia Tanca che è sempre stata gentile e disponibile e mi ha dato l'opportunità di svolgere questa interessante tesi all'interno del suo laboratorio dove ho trovato un ambiente sempre positivo e stimolante.

Un ringraziamento particolare va anche a Fabio Azzalini il quale è sempre stato disponibile nel guidarmi attraverso i suoi preziosi consigli.

Abstract

The era of Big Data, the World Wide Web and the new pervasive and interactive technologies have lead to an information ecosystem where a huge number of players provide their own information even about the same real world entities. When we want to integrate the data coming from such heterogeneous sources, besides the information volume another challenge consists in the impossibility, in most cases, to monitor the information procurement process of each source in order to exclude from the integration data that has been copied from another source which is already available.
The Data Fusion phase of Data Integration has the main purpose of identify, within all this noisy data where sources provide conflicting or partial information, the true values that correspond to each data item.
In the described context the Majority Voting approach used to select the most proposed values is ineffective, because of the frequent phenomenon of copying between sources, encouraged by the availability of the Web. Many state-of-the-art Data Fusion systems rely on the strategy of evaluating sources' trustworthiness in terms of the veracity of the values they propose and vice-versa using an iterative approach to reach a convergence of these measures. These systems are demonstrated to be very effective but lack scalability, so that efficiency in the face of large quantities of data is low.
In this thesis we propose a new approach that aims at providing a more versatile algorithm which does not rely on an iterative process and thus is more agile and scalable. This algorithm can address the truth discovery task in both single-truth and multi-truth contexts by relying on the concept of "data ownership". Assuming that the real world owner of a given entity always provides the most reliable and copy-free information about it, we decided to focus on identifying the "owner source" (if any) and rely only on the information provided by it, differentiating our solution from the iterative approaches that apply a weighted voting on values based on their sources'

authorities.
The experiments performed on both single-truth and multi-truth real world datasets highlight the fact that our solution is the most efficient in returning the proposed truth, showing similar, and in some cases higher, performances.

Keywords : Data Integration, Data Fusion, truth discovery, multi-truth, single-truth, Data ownership

Estratto

L'era dei Big Data, del World Wide Web e delle nuove tecnologie sempre più pervasive ed interattive ha portato alla creazione di un ecosistema informativo nel quale sono presenti un numero elevatissimo di sorgenti che forniscono informazioni, spesso anche per le stesse entità. Quando vogliamo integrare informazioni provenienti da sorgenti eterogenee, oltre ai problemi legati alla grande quantità di dati disponibile, si aggiunge l'impossibilità, nella maggior parte dei casi, di controllare il processo d'acquisizione delle informazioni da parte delle singole sorgenti, in modo da evitare di considerare nell'integrazione dati già disponibili che sono stati copiati da altre sorgenti.
La fase della Data Fusion all'interno della Data Integration ha come scopo principale quello di identificare, tra tutte le sorgenti che forniscono informazioni contrastanti o parziali, i veri valori da attribuire a ogni oggetto.
In questo contesto, selezionare come vero il valore più frequente è spesso inefficace, a causa dei frequenti fenomeni di copia tra le sorgenti. Molti sistemi di Data Fusion utilizzano una strategia basata sulla valutazione dell'affidabilità delle sorgenti, utilizzando un approccio iterativo per raggiungere una convergenza di queste misure. Questi sistemi si sono dimostrati molto efficaci, ma spesso non presentano la scalabilità per lavorare con grandi quantità di dati.
In questa tesi proponiamo un nuovo approccio volto a fornire un algoritmo più versatile, non basato su un processo iterativo e quindi più veloce e scalabile. Il metodo presentato può essere utilizzato sia in contesti single-truth che in contesti multi-truth. Assumendo che il proprietario reale di una determinata entità è anche colui il quale fornisce l'informazione più affidabile e priva di copia, abbiamo deciso di concentrarci sull'identificazione di tale "sorgente proprietaria" (se presente) e di affidarci solo all'informazione da lei presentata, differenziando la nostra soluzione da quelle che sfruttano un approccio iterativo e che infine utilizzano una votazione pesata per determinare il valore vero da assegnare ad un determinato oggetto.

Gli esperimenti effettuati, sia su dataset single-truth che multi-truth, evidenziano la maggiore efficienza della nostra soluzione, mostrando allo stesso tempo una simile, se non migliore, precisione.

Contents

List of figures

List of tables

Chapter 1

Introduction

In this chapter in Section 1.1 we will first introduce the main problems that our solution has to face in the field of Big Data Integration, then in Section 1.2 we will provide a first description of our solution's approach and finally in Section 1.3 we will conclude with a description of the overall structure of the thesis and how its contents are distributed between the chapters.

1.1 Motivations

During the last decade we are experiencing a shift of values in the society towards data. The importance of data, and more in general of information, which is continuously ascending, is highlighted by the fact that companies and governments are taking more and more measures in order to protect and exploit them in a regulated manner. In fact with the introduction of technologies such as the Web and Internet of Things data has become part of our daily life and the information produced everyday is both growing enormously in volume and has also a major value since it is becoming closer to the users. The whole process of digitalization has lead to the existence of a huge amount of data sources that provide and share information.
In this noisy "chaos" of information, Data Integration is the set of techniques that aim at providing a uniform access to all the information proposed by a set of heterogeneous sources, in order to present a unified view of the requested information. The heterogeneity of the sources is a consequence of the different designing process and the different scopes for which each of them has been created and it can be reflected at multiple levels (i.e. data schema

level, format level, semantic level) bringing to inconsistencies and conflicts. In this scenario, even for the same domain, independently of how niche it is, it is often possible to find multiple sources which provide conflicting or partial information about the same real world object. Data Fusion is the step of the Data Integration process which is mainly focused on the discovery of the truthful values, from all the conflicting values provided by the different sources, to assign to each data object in order to provide the most complete and reliable information possible.

1.2 Introduction of the solution

In this thesis we will present our solution for both multi and single truth Data Fusion which leverages on the ownership of a source towards the data items for which it provides information. Where as single-truth we describe a context in which for each attribute corresponds a single true value while as multi-truth we describe a context in which for the same attribute might exist more than one true values. As will be explained in detail later we noted that sources which results as being the "owner" of the entities for which they provide information are usually the most reliable and copy free. So our solution aims at identifying those sources by means of an "ownership score" which estimates, given a source and an attribute value, how much the source is behaving like an owner towards the data item(s) associated with that value. The solution proposed differs from other state-of-the-art Data Fusion techniques since it focuses on the concept of "ownership" of the information and also because it doesn't leverage on an iterative approach, thus being more scalable that the others.
We implemented our solution in two different phases, expanding it by considering also the probabilities of copying between couples of sources.

1.3 Thesis outline

The thesis is composed by the following chapters:

- In Chapter 2 we provide a description of the useful preliminaries concepts used in the description of the solution such as Data Integration, Big Data, Copy Detection and Bayesian Inference.

- In Chapter 3 we provide a detailed description of the most significant Data Fusion algorithms in the literature focusing on the different insights that brought to their implementations.
- In Chapter 4 we give an overview of our solution and describe the design process behind its implementation.
- In Chapter 5 we formalize the concepts introduced in the previous chapter giving a detailed definition of the implementation of the algorithm providing also its relative pseudocode.
- In Chapter 6 we present the experimental evaluation of the solution proposed preformed by using both single and multi truth real world datasets.
- In Chapter 7 we draw the conclusion of the thesis and provide suggestions for possible future works in order to improve the algorithm.

Chapter 2

Preliminary studies

"Science is built up of facts, as a house is with stones.
But a collection of facts is no more a science than a heap of stones is a house. "

Henri Poincaré

In this section we will describe formally the main topics that are fundamental in order to fully understand the problem and the work proposed in this document. In particular we will first illustrate the concept of traditional Data Integration and its main phases in Section 2.1, then we will outline what is the meaning of Big Data and the new challenges that arise with it in Section 2.2, after that in Section 2.3 we will provide a description of the copy detection problem and the insights that lead to the design of most copy detection algorithms (presented in Section 3.6) and lastly, in Section 2.4, we will provide a description of the Bayesian inference process used in the algorithms presented in Section 3.5.

2.1 Data Integration

Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data [12]. Data Integration can be tackled using 2 main approaches which are physical and virtual data integration depending on the use that will be done of the integrated data.

2.1.1 Physical data integration

Physical data integration is mostly used for data warehouses, and is based on the Extract Transform Load paradigm (ETL) [25] [3] for which the data stored in the original (or external) sources is periodically extracted from there, transformed resolving the conflicts guaranteeing data quality and consistency standards in order to align it with the target schema (i.e. normalization, elimination of duplicates, check on integrity constraints) and loaded into a new database called data warehouse in a presentation-ready format so that application developers can build application over the data and end users can make decisions. Data warehouses are read-only, big capacity databases designed in order to perform analysis on historical data of enterprises thanks to their *cube data model* (Figure 2.1) which organizes data into multiple dimensions, including the time dimension. In this type of physical data inte-

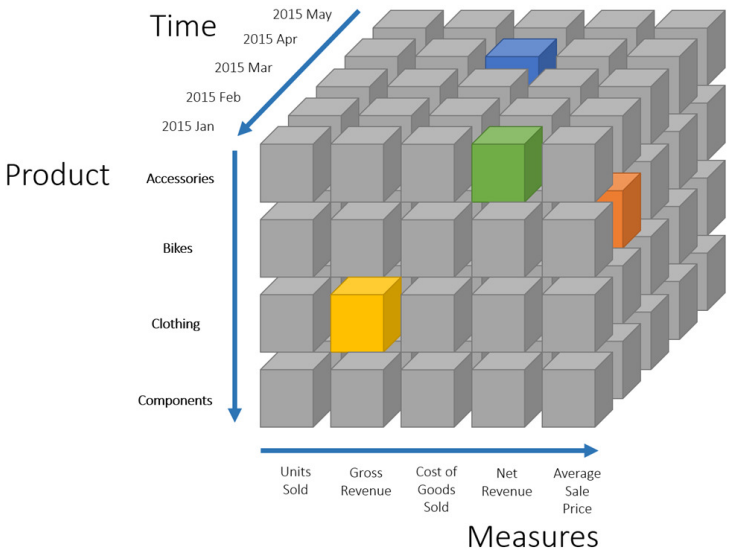


Figure 2.1: Example of data cube model inside a data warehouse

gration the user, ignoring the underlying structure of the external databases, sends queries to the data warehouse and receives data updated to the last time the ETL process was performed. Since the ETL process is only performed periodically the data returned won't probably be up to date so this approach is not used in case there's a need of retrieval of real-time data as for example for the ones contained in operational databases which quickly goes out-of-date.

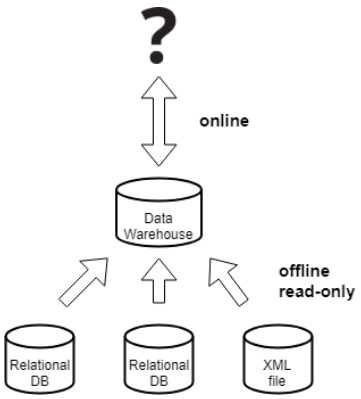


Figure 2.2: Physical database integration schema

2.1.2 Virtual data integration

Virtual data integration systems are characterized by an architecture based on a global schema and a set of sources. In this case, differently to physical data integration, the real data is kept stored in the original data sources, while the global schema provides a reconciled, integrated, and virtual view of the underlying sources [12]. The query coming from the users are expressed as directed to the global schema and then is reformulated for each data source involved in the result in order to retrieve their portion of data required. Since the data is queried and retrieved directly from the data sources in this case the query answers will always be up-to-date, making this approach the best in the case of operational databases for an enterprise.

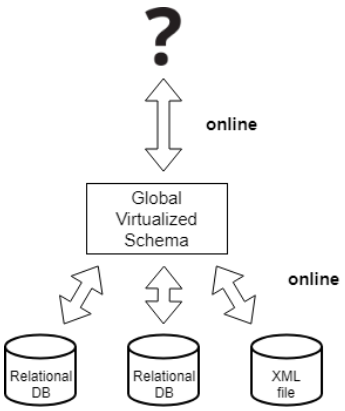


Figure 2.3: Virtual database integration schema

The main problem in data integration arises from the heterogeneity of the sources involved which, even when they provide information about the same domain, introduces three different categories of challenges [8]:

- **Semantic ambiguity** when it is needed to *align* the source tables in order to provide a unique global schema the same conceptual information may be modeled quite differently in different sources and different conceptual information may be modeled similarly in different sources.
- **Instance representation ambiguity** when it is needed to *link* the same data instance from multiple sources, one needs to take into account that instances may be represented differently, reflecting the autonomous nature of the sources.
- **Data inconsistency** when it is needed to *fuse* the data from multiple sources, one needs to resolve the instance-level ambiguities and inconsistencies between the sources.

2.1.3 Conflicts

The heterogeneity of data sources introduced before is reflected and categorized in the following types of conflicts that create discrepancies in the representation of the same real world concepts:

- **Name conflicts** in the case of synonyms consist on the assigning of different names for semantically same schema elements (attributes, entities and relations) as for example *store* and *sales point*. In the case of homonyms consists in the assigning of the same name for semantically different schema elements as for example an attribute named *price* can be interpreted as the price before or after tax.
- **Type conflicts** in the case of attributes where information can be expressed in different types for example in the case of the attribute *gender* the value can be expressed using Male/Female, M/F or similar type of values.
- **Data semantics conflicts** in the case of attributes where the information can be expressed using different measure systems (i.e. meters or yards, kilos or pounds), using different currencies (i.e. euros, yuan, US dollars) and using different granularities (i.e. grams or kilos)

- **Structural conflicts** in the case of real world concepts that can be described using different elements depending on the purpose for which the data source is designed. For example the publisher can be expressed as the attribute of an entity book, if its information is not relevant to the purpose of the data source, or stored an an entity related to the book's one otherwise.

- **Cardinality conflicts** in the case of concepts that can have different type of relation based on the design context of the data source. For example we can have a source that implements a relationship between two entities using a one-to-many relation and another sources that for the same entities uses a middle "relation entity" with a one-to-one relation.

- **Key conflicts** in the case of entities that, in different sources, can be identified locally using a different set of attributes.

Data integration addresses these challenges respectively by the sequential execution of the 3 major steps which are schema alignment, record linkage and data fusion as shown in Figure 2.4.

Figure 2.4: 3 steps of data integration

2.1.4 Schema Alignment

The schema alignment phase is critical in the data integration process since it outputs the conceptual design of the global schema, that will be later used to express the users' queries, and how its attributes are composed starting from the external sources' schemata.
This phase is composed of three main components that here described:

- production of a **global schema** (or mediated schema) that gives a unified view, in the broadest way possible, of all the important aspect of the domain considered

- production of an **attribute matching** system in order to match the attributes expressed in the global conceptual schema with the attributes presented in the external sources' logical schemata
- production of a **schema mapping** that describes the reconciliation of the global schema with the single logical schemata of each external source in terms of semantic relationships between the contents of the sources and the one to propose in the global data. In case one of global schema's attributes that are not reconcilable to any attribute of a source's schema, NULL values are inserted

This phase helps overcome the semantic ambiguity problem outlined before.

2.1.5 Record Linkage

The record linkage phase has the scope of finding across the records provided by each external source different partitioning in order to make each partition correspond to a real world entity.
This phase helps overcome the instance representation ambiguity outlined before.

2.1.6 Data Fusion

The Data Fusion phase is responsible of discovering the true values to assign to each data item presented for the domain of interest of the data integration scope. This challenge arises by the fact that sources, especially in the context of the big data that will be later described in Section 2.2, do not tend to provide the same information for a single data item and moreover tend to copy, completely or partially, the information provided from other sources. This discrepancy in the information provided can be introduced mainly by the data semantic conflicts explained before or by the different type of information's fruition objectives for which the data sources are originally created. There are many types of different Data Fusion techniques that apply different criteria in the process of finding the truth but the main conceptual structure shared by most of them is composed by three steps which are shown in Figure 2.5 are the following:

- **Trustworthiness evaluation** is the quantification of how much a source is reliable, with respect to the partiality or completeness of its

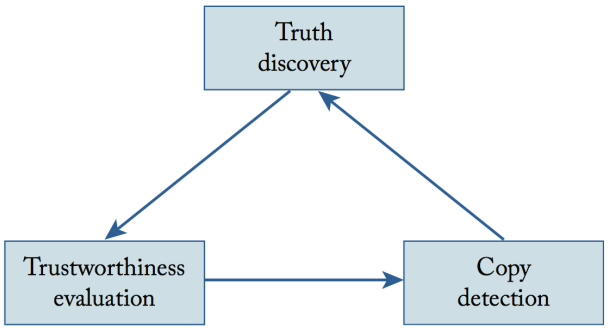


Figure 2.5: 3 steps of Data Fusion

provided values, in order to assign a weighted veracity to it its values

- **Copy detection** is the process of discovering the copying relationships that may take place between different sources, this topic is better described in Section 2.3

- **Truth discovery** is the valuation of the veracity of the value provided in order to select the most trusted ones and assign it to the respective data item

Furthermore the Data Fusion problem can be divided in two domains of interests basing on the type of information to be integrated and definition of the true value to assign to each data item. In particular those domains are the **single-truth** and the **multi-truth** Data Fusion problem.

In a **single truth** context it is assumed that for each data item taken into consideration there's only one true value and if the value proposal of a source is not equal to that one it is considered as a value opposing to the truth. For example this might be the case of a flight's departure date where any information provided that is not the correct date is considered as completely erroneous.

Contrarily in a **multi-truth** context there are multiple true values that can be assigned to a data item and the truth level of the value proposed by a source is proportional to the amount of correct information that the source provides with respect to the overall true value. So in this case there's no binary evaluation (right or wrong) of the truthfulness of a value proposes as

in the case of single-truth context.
For example consider the more complex case of a book's author list where there are multiple contributors:

Source	***value proposed***
s_1	*George Orwell*
s_2	*James Joyce, Franz Kafka*
s_3	*Alessandro Manzoni*
True value	*George Orwell, James Joyce, Franz Kafka*

Table 2.1: Table representing a multi-truth case scenario

In the case of sources s_1 and s_2 it would be wrong to consider the sources as opposing to the truth value since they provide some truth information and the union of their value composes the true value. Similarly it should be also considered the fact that source s_2 provides 2/3 of the truth while s_1 provides only 1/3. In a single-truth scenario the three sources would be considered all as wrong so the value of source s_3, which doesn't contain any truthful information, would have the same consideration of the ones of s_1 and s_2 that are partially correct.

2.2 Big Data

Nowadays the arising of new technologies, even more interactive and penetrating (i.e. Internet Of Things), the digitalization process that is involving both little and large entities mixed with the information availability that the web is able to provide, are resulting in an information explosion. Because of that, data and information more in general are becoming a crucial asset for organizations thanks to the new data analysis and information retrieval technique that makes them meaningful and useful for a large spectrum of scope. In order to effectively explain the explosion mentioned above we now present the main challenges of Big Data and in particular in Big Data integration which can be represented by the 4 V's.

2.2.1 Volume

Regarding the volume of data we start by giving some measures to give an idea of the amount of data created nowadays. In 2011 a report from International Data Corporation (IDC) estimated that the overall created and copied data volume in the world was 1.8 ZB (almost 1021 B), which increased by nearly nine times within the next five years [10]. To get more practical consider that Google processes data of hundreds of Petabyte (PB), Facebook generates log data of over 10 PB per month, Baidu processes data of tens of PB, and Taobao, a subsidiary of Alibaba, generates data of tens of Terabyte (TB) for online trading per day. Furthermore consider that on average, 72 hours of videos are uploaded to YouTube in every minute [22]. Therefore, we are confronted with the main challenge of collecting and integrating massive data from widely distributed data sources [4]. All these examples are caused not only by the fact that nowadays sources may store a previously improbable amount of data, but also by the fact that the number of data sources has grown to be in the millions and, even for a single very specific domain, the number of sources that provide information has grown to be in the tens to hundreds of thousands. In addition to that is important to consider the fact that the web has made the information exchange much easier so that sources have an higher probability of copying introducing redundancies without considering the veracity of the information provided [8].

2.2.2 Velocity

As a direct consequence of the rate at which data is being created and continuously made available, many of the data sources are very dynamic and their number is exploding. The dynamism introduced by technology and connectivity bring to a much faster paced process of data creation since data might collected in almost every task accomplished. Data might be also extremely volatile as for example in the case of IoT's sensor data where the stream of information coming from the sensors is not possible to be stored continuously and a live analysis of the content has to be made in order to evaluate the information retrieved. From the data integration point of view there are techniques such as SOLARIS [20] that propose an online Data Fusion technique in order to perform truth discovery that starts returning answers from the probed sources and then refreshes the answers as it probes other

uncovered sources.

2.2.3 Veracity

The multiplicity of the number of sources reflects also on the different quality of information provided. In fact sources tend to vary under the aspects of many dimensions like the coverage, the accuracy and the timeliness of data provided. Depending on their specific scope of creation in fact data source may decide to give information about the same domain with different level of specification focusing more on one aspect than others. For example a data source that stores the information about the purchases of a bookstore may not give too much importance on the aspect of the books itself, since they may be already identified by its ISBN, and can provide approximate values for the titles of the books or the lists of authors while other sources such as the book publishers, which scope is to create a detailed catalog of the books published, probably will focus more on those aspect providing a more complete and updated information. This example will be later deeply explored in the analysis of our solution.

2.2.4 Variety

Data sources (even in the same domain) are extremely heterogeneous both at the schema level, regarding how they structure their data for example having structural conflicts, and at the instance level, regarding how they describe the same real world entity for example having type or semantic conflicts, showing a considerable variety even for substantially similar entities. This variety problem is also reflected on the variety of the nature and the data model applied by different sources. In fact nowadays meaningful information can be represented using text, videos, social networks (considering relationships between individuals) and images. Based on the level of the organizational structure used to store information it is possible to categorize data, based on how it is expressed and stored in order to be retrievable, in three main categories:

Structured data

Structured data is organized into a formatted repository where the information is organized into multiple attributes in order to make it addressable for

an effective analysis. This in the case of the well known relational databases (i.e. PostgreSQL) which store the information into tables where for each instance contained in a row the data is categorized into a set of attributes represented by its columns. From the schema adopted to represent the domain of interest it is already possible to infer information about the interpretation given to the different real world entities.

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Figure 2.6: Example of structured data about students and researchers

Semi-structured data

Semi-structured data is not organized in a relational database but still has some organizational properties that allow to perform easier analysis. This is done in order to provide a more dynamic data structures such as trees and graphs. Main examples of this kind of organizations are adopted in the so called non relational databases (or noSQL) such as XML data, the Neo4J and the MongoDB data model schemata. These noSQL database perform the persistence of data without using a rigid relational schema as the one proposed in structured data, in fact they are also called "schemaless" since the structure used for organizing the information is not fixed allowing them to be more scalable but on the other hand being slower in performing the joining of data not previously put in relation at the time of persistence. There are many different types of non relational databases based on their type of data model but the main categories are document based, graph based, key-value based, object based and tuple based. For example in the case of XML databases and MongoDB (which adopt a document based data model) data is stored in tags which hierarchy describes a tree structure where the leafs are different types of elements identified by tags as seen in Figure 2.7. In the case of Neo4J (which adopts a graph based data model) data is stored using

nodes and links between them as shown in Figure 2.8.

```
<University>
 <Student ID="1">
  <Name>John</Name>
  <Age>18</Age>
  <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
  <Name>David</Name>
  <Age>31</Age>
  <Degree>Ph.D. </Degree>
</Student>
....
</University>
```

Figure 2.7: Example of semi-structured data (document based) about students and researchers

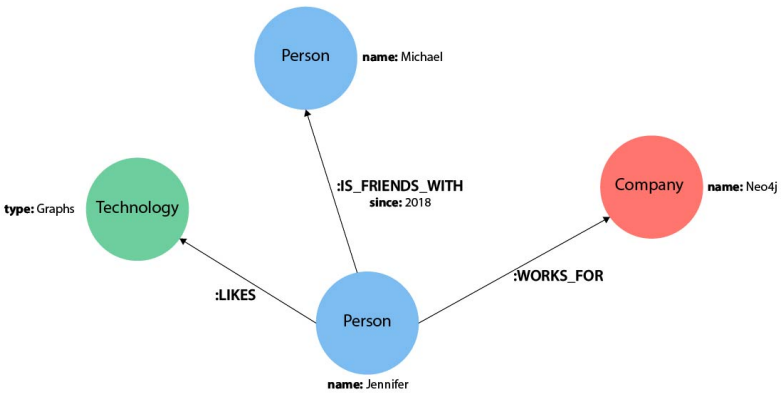


Figure 2.8: Example of semi-structured data (graph based)

Unstructured data

Unstructured data is not organized in a predefined manner and is not applicable to any predefined data model. Main examples of this kind of data are raw text data, images, videos, PDF and media logs. Differently from the other two types of data it is very scalable and flexible, given its simplicity and lack of organization respectively, but is not addressable for any kind of straightforward analysis, it requires ad hoc techniques.

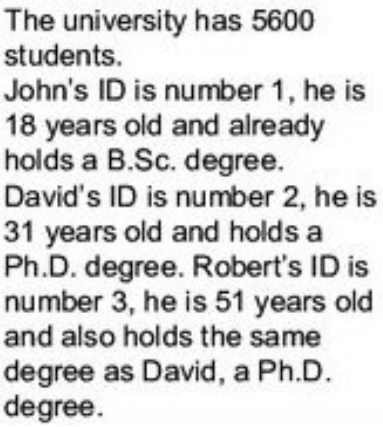


Figure 2.9: Example of unstructured data about students and researchers

2.3 Copy Detection

The copy detection phase consist in detecting copying relationships between data sources, so that it will be possible to compute a discounted vote count to be assigned to a copied value in voting. Copying exists between two data sources if they both present some common data which can be derived directly from each other or transitively from a common data source. Based on the above definition is possible to divide all the data sources into two main categories:

- *independent sources* are sources that provide their proposed values independently without relying on any other sources (copying) and, for this reason, should be considered more trustworthy that the others. Notice that their values are not to be considered always true since they can have an incorrect knowledge of the real world or many other types of problems.

- *copiers* are sources that copy part or the entirety of their provided information from one or more sources. They can also provide some modifications (revisiting or adding additional values) to the values copied in which case those values are considered as an independent contribution of the copier. Moreover in [26] there's a further categorization of the concept of copiers into *blind copiers*, which assume the independence between the copying probability and the veracity of sources, and *smart copiers* that on the contrary are more likely to copy true (and com-

plete in case of multi-truth) values instead of false (or incomplete in case of multi-truth) ones. These sources are considered in general less trustworthy than the independent ones because usually the process of copying might introduce some type of error in the data. An example of this phenomenon are the automatic system that perform copying such as crawlers that retrieve the information from websites and might misinterpret the information presented.

Usually when confronting data sources it is not possible to have information about how they procured their data so we have to rely only on their data snapshots in order to infer the relationship between different sources. Furthermore it also has to be considered the fact that when a dependency between a couple of sources is discovered is not always trivial to identify which of the two is the copier [6].
Copy detection complexity varies on the type of data analyzed and results to be easier for detect copying in an unstructured data context rather than in a structured data one [8]. In fact for example in raw text document when there's a reuse of sufficiently large text fragments is possible to infer a copying relationship while in the case of structure data we have to take into consideration multiple factors. For example the sharing of common data doesn't imply copying since it might be the case of independent sources that provide true values independently and, vice versa, the sharing of only a small portion of common data doesn't imply no copying since this may be the case of a partial copier, which was defined above. For these reasons in the case of structured data the typical approach consists into focusing on the sharing of uncommon data between source since it is much less probable that multiple independent sources provided the same false value independently because of the high number of false values.

2.4 Bayesian inference

Bayesian inference is used in order to derive a *posterior probability* as a consequence of two antecedents that are a *prior probability* and a *likelihood function*. We start by defining the *conditional probability* as the probability of occurrence of an event which is in relationship with one or more other events, for given two events A and B the conditional probability can be calculated as the *likelihood* of B given the occurrence of the event A as

shown in Equation 2.1:

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \tag{2.1}$$

Where as *posterior probability* is intended the conditional probability that is assigned after the observation of an event (evidence), as *prior probability* is intended, in a specular way, the unconditional probability that is assigned before any relevant event is considered. Bayesian inference is based on the computation of the posterior probability defined in the Bayes' theorem (Equation 2.2) which inverted the conditional probability above in order to focus on the B event:

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} \tag{2.2}$$

Where A is considered the hypothesis which probability is influenced by the evidence, $P(A)$ is the prior probability of the hypothesis, $P(A|B)$ is the likelihood expressed before, B is the evidence that was not considered in the computing of the prior probability and $P(B)$ is the prior probability of the observation of the evidence B. For example consider drawing a card from a playing deck of cards composed by a total of 52 cards where 26 of them are red and the remaining 26 are black. Suppose that we are looking for the probability of having drawn a King knowing that the card is black. We can express with A the event that the card picked is a King and with B the event that the card picked is black. Consequently we can compute the elements of Bayes' theorem:
$P(B|A) = P(black|King) = \frac{1}{2}$ (since the deck has 2 red Kings in the deck)
$P(A) = P(King) = \frac{4}{52} = \frac{1}{13}$ (since the deck has 4 Kings of both colours)
$P(B) = P(black) = \frac{1}{2}$ (since half of the cards are black)
Finally using Bayes' theorem we can infer that:

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} = \frac{\frac{1}{2}\ \frac{1}{13}}{\frac{1}{2}} = \frac{1}{13} \tag{2.3}$$

So the posterior probability of the picked card being a King conditioned by the observation that it was black is equal to $\frac{1}{13}$.

Chapter 3

State of the art

"All truths are easy to understand once they are discovered;
the point is to discover them'

Galileo Galilei

In this chapter we will provide a description of the Data Fusion context in section 3.1 and in the following sections we will describes the most important state-of-the-art solution developed in order to be applicable in the Data Fusion scenario. In particular we will describe in section 3.2 the basic voting approach, in section 3.3 the web-link based approaches, in section 3.4 the information retrieval based approaches, in section 3.4 the Bayesian inference based approaches and in section 3.5 the algorithms that included copy detection.

3.1 Data Fusion

Nowadays with the huge increment of the number of sources given by the introduction of the big data analysis there's an explosion of information which reflects on the heterogeneity of the information provided. In fact data is produced constantly from an increasing number of sources and this results in a need to perform a truth discovery analysis in order to trace the information that represents the truth inside the whole noise of data that surrounds it. An example presented in [17] analyses the fact that the results returned by Google for the query "the height of Mount Everest" include multiple values as 29.035, 29.002 and 29.029 feet. As in this example little variations, in most

uses cases, might not be considered as a big trouble. But it has been shown [15] that even for critical information that should be highly reliable like the flight and stock domains, there's more than one value provided for 70% of the data items, 50% of which differs because of various kinds of ambiguity furthermore only 70% of the correct values are provided by the majority of the sources.
In the described scenario process of truth discovery happens to become crucial in order to face the noises difficulties introduce by the *variety* feature of the big data explained before. In particular the application of truth discovery methods can benefit multiple fields of interest like healthcare [23], crowdsourcing [1][14] , information extraction [13][28] and much others.
We will now described a selection of the main Data Fusion truth discovery methods proposed in the literature describing their main characteristics and analyzing the best scenarios for which they were designed. We start by doing a macro classification of the algorithm based on the type of the approach used in order to address the Data Fusion problem:

- *Iterative methods* : which uses a voting approach based on source trustworthiness and value veracity that depend on each other and are computer iteratively until convergence.

- *Optimization Based methods* : which aim at minimizing the distance function that measures the difference between the information provided by each source and the identified truth.

- *Probabilistic graphical model based methods* : which reformulate the problem of truth discovery by representing it using a PGM in order to describe probability distributions over complex domains.

In the following sections we will only describe the algorithm using iterative methods since they're the ones we used to confront our solution with in Chapter 6.

3.2 Basic voting

Majority Voting: This algorithm applies a simple vote count as if each sources votes for the value they propose for a data item and, at the end of the computation, considers as true the value which has the majority of the votes.

This approach is based on the fact that sources are equally reliable and for this reason just considers the redundancy of a value. As explained before this approach doesn't hold when large veracity exists [8] but, nevertheless usually provides very good solid performances.

3.3 Web-link based

The following algorithm are characterized by the fact that they use techniques for studying the link structure of hypermedia environments (i.e. the World Wide Web) in order to perform the truth-discovery task. They identify sources as webpages and analyze the link structure in order to evaluate the level of trustworthiness of a page to then be able to apply a weighted voting strategy to compute the probability that a value is true.

HUB [11]: This algorithm is inspired by the problem of measuring web pages' authorities based on the analysis of web links. Considering pages as data sources it assigns each page a hub score and an authority score and aims in finding the hub pages which are pages that have links to multiple relevant authoritative pages. This describes a mutually reinforcing relationship: a good hub is a page that points to many good authorities while a good authority is a page that is pointed to by many good hubs (Figure 3.1 presents a visual representation of the concept of hub and authority). So basically the concept to evaluate the trustworthiness of a source is based on the vote count obtained by each of its values and both scores are computed until convergence. Here a normalization of the vote counts an trustworthiness is required in order to prevent them from unbounded growing.

Average Log [24]: This algorithm applies the same reasoning of the **HUB** algorithm but it introduces the averaging of the trustworthiness score and multiplies it by the logarithm of the number of values provided in order to avoid to assign the same score for sources which have the same accuracy on much different number of claims. As in the case of the **HUB** algorithm also here is required a normalization for the vote count and the trustworthiness score.

Investment [24]: In this algorithm each source "invests" its trustworthiness uniformly among its claims. The confidence of each claim grows accord-

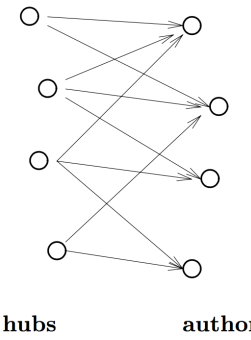


Figure 3.1: Example of hubs and authorities

ingly to a non-linear function defined on the sum of invested trustworthiness of its providers. Then the sources "collects credit back" by being assigned as trustworthiness value the sum of the confidence in their claims, weighted by the proportion of trust previously contributed to each (relative to the other sources). Also in this case, as in the previous algorithms, a normalization is required.

PooledInvestment [24]: This algorithm exploits the same investment-return logic of **Investment** but the confidence of each claim is now linearly scaled in order to maintain the vote count on a data item equal to the accumulated investment. Thanks to the introduction of the linear scaling the normalization is no longer required in this case.

3.4 Information Retrieval based

The following algorithms are based on the concept of corroborating the different claims (views) by the sources by confronting them using similarity measures also used in information retrieval. The main limitation of these approach is that they assume that there's one and only single value (fact) that's true for for each data item and they try to discover it using a complementary vote.

Cosine [9]: This algorithm iteratively evaluates the trustworthiness of sources based on the comparison, using the cosine similarity measure [21], between their proposed views and the estimated golden truth for each fact. This comparison might have three outcomes that are +1, 0, -1 in case of true,

undetermined and false fact respectively. The idea is then to compute, for each view, the similarity between their statements, viewed as a set of 0, +1 and -1 statements on each fact, and the predicted real world values. Then the new views for the sources of the estimation are computed using a linear combination of the old view and the cosine similarity. Finally the estimated true values are computed using an average of the estimated trustworthiness of the views giving more weight to those which tend to remain constant during the iterations.

2-Estimates [9]: 2-Estimates uses two estimators for the truth of facts and the error of views. In particular it computes the trustworthiness of sources differently from **HUB** because of two aspects. The first one is that it computes the score by averaging the votes received by all its values and the second one is that when a source provides a value is considered as if it opposes to every other possible value.

3-Estimates [9]: This algorithm is an extension of the **2-Estimates** where is also taken into account the likelihood of each value. In other words for each fact it also estimates how hard is to be exact about it (i.e. the propensity of sources to be wrong on this fact).

3.5 Bayesian based

These algorithms leverage on the Bayesian inference method described in Section 2.4 in order to find the values that most probably will be true.

Truth finder [27]: This algorithm applies the Bayesian inference process in order to iteratively compute the trustworthiness of a source s (or website w as in Figure 3.2) as a combination of the expected confidence of facts it provides. The confidence of a fact is determined by the trustworthiness of the source providing it and by the other facts that are provided about the same object. In particular, in the case shown in Figure 3.3, the confidence for the fact $f1$ is computed by the addition of the confidence of the other facts for the same objects (in this case $f2$) multiplied by the degree of similarity between the two values (which in the case of conflicting values will be less than zero). Notice that in this way the facts proposed by many sources are more likely to be correct.

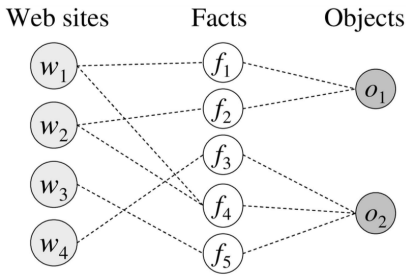


Figure 3.2: Example websites, facts, and object structure

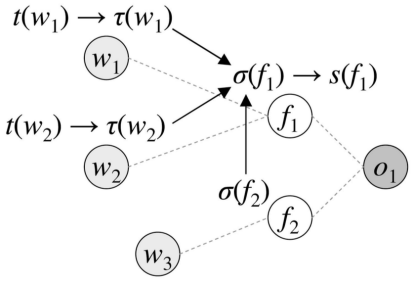


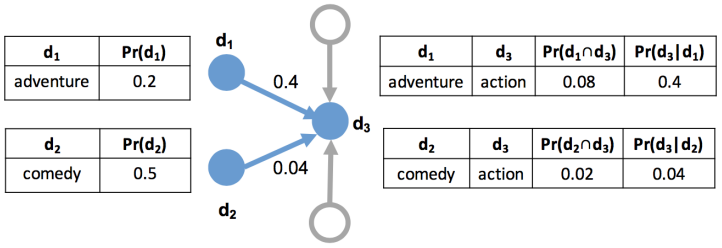
Figure 3.3: Example of conflicting facts for the same object o_1

Accu [6]: This algorithm differs from **Truth Finder** in that it take into consideration that different values provided on the same data item are disjoint and their probabilities should sum up to 1. So in other words, as the IR based algorithm described in Section 3.4, when a source s provides a value $v \neq v'$ on item d, s is considered to vote against v'. To make the Bayesian analysis possible it assumes that there are N false values in the domain of d and they are uniformly distributed.

PopAccu [7]: This algorithm improves the **Accu** algorithm by removing the assumptions of the uniformly distributed wrong values by replacing it with an empirical distribution calculated on the observed data.

AccuSim [6]: This algorithm improves the **PopAccu** algorithm by considering the similarities between proposed values for a same data item as in the case of **Truth finder**.

AccuFormat [15]: This algorithm improves the **AccuSim** algorithm by considering also the formatting of values. In this case a source that provides



d_1	$Pr(d_1)$
adventure	0.2

d_2	$Pr(d_2)$
comedy	0.5

d_1	d_3	$Pr(d_1 \cap d_3)$	$Pr(d_3 \mid d_1)$
adventure	action	0.08	0.4

d_2	d_3	$Pr(d_2 \cap d_3)$	$Pr(d_3 \mid d_2)$
comedy	action	0.02	0.04

Figure 3.4: Example of implication between domains

a value v is considered also a partial provider for values that subsume v (i.e. a source that rounds numerical value to the million that provides a value of '5M' will be also considered a partial provider for the value '4.689.039').

DART [18]: This algorithm is designed to perform truth-discovery in a multi-truth context by considering also for single sources their level of expertise regarding each domain present in the input dataset (i.e. for a dataset about books or about movies the domains might be the genres to which they belong). The algorithm follows these major steps starting from a dataset and a predefined set of domains:

- computes a score $e_{d_i}(s)$ that reflects the expertise of each source s in each domain d considering its coverage (intended as the percentage of data quantity provided by s with respect to the total data quantity provided for d) and the influence between d and other different domains regarding the data item for which s proposes information (intended as the conditional probability of another domain d' given d). In Figure 3.4 there's an example of the probabilistic graph that represents the implication between different domains for a source s that in a book dataset provides information about books related to multiple genres.

- computes, for each source s and each value v provided by s for an object o, its confidence score $c_s(v)$ using a multi-truth approach that, differently than the single-truth ones seen before which assume that when a source supports one or more values it opposes to other potential values, computes the amount of information provided using Formula 3.1

(we used the same notation that it is explained in Section 4.5)

$$c_s^o(v) = \begin{cases} \left(1 - \frac{|V(o) \setminus V_s(o)|}{|V(o)|^2}\right) \cdot \frac{1}{|V_s(o)|} & \text{if } v \in V_s(o) \\ \frac{1}{|V(o)|^2} & \text{if } v \notin V_s(o) \end{cases} \quad \text{with } V(o) \bigcup_{s \in S} V_s(o) \tag{3.1}$$

- using a Bayesian inference approach (Equation 3.2) iteratively computes the veracity of each values $\sigma(v)$ (the probability that the value v is true) starting from the values of sources trustworthiness in recall τ^{rec} and specificity τ^{sp} (since we're in a multi-truth context) and vice-versa (Equations 3.3 and 3.4 respectively) until convergence.

$$\begin{aligned} P(v|\psi(o)) &= \frac{P\left(\psi\left(o\right)|v\right) P\left(v\right)}{P\left(\psi\left(o\right)\right)} \\ &= \frac{P\left(\psi\left(o\right)|v\right)\ \sigma_o\left(v\right)}{P\left(\psi\left(o\right)|v\right)\ \sigma_o\left(v\right) \quad + \quad P\left(\psi\left(o\right)|\bar{v}\right)\left(1 - \sigma_0\left(v\right)\right)} \\ &= \frac{1}{1 + \frac{1-\sigma_o(v)}{\sigma_o(v)} \cdot \frac{P(\psi(o)|\bar{v})}{P(\psi(o)|v)}} \end{aligned} \tag{3.2}$$

$$\tau_d^{rec}(s) = \frac{\sum_{o \in O^d(s)} \sum_{v \in V_s(o)} \sigma_o(v)}{\sum_{o \in O^d(s)} |V_s(o)|} \tag{3.3}$$

$$\tau_d^{sp}(s) = \frac{\sum_{o \in O^d(s)} \sum_{v' \in \overline{V_s(o)}} (1 - \sigma(v'))}{\sum_{o \in O^d(s)} \left|\overline{V_s(o)}\right|} \tag{3.4}$$

where $\overline{V_s(o)}$ denotes the set of values claimed for o by other sources except s. And where the probabilities of the observation $\psi(o)$ conditioned to the fact that v is true (v) or is false ($\bar{v}$) are computed using Equations 3.5 and 3.6

$$P(\psi(o)|v) = \prod_{s \in S_o^d(v)} \tau_d^{rec}(s)^{e_d(s)c_s(v)} \prod_{s \in S_o^d(\bar{v})} \left(1 - \tau_d^{sp}(s)\right)^{e_d(s)c_s(v)} \tag{3.5}$$

$$P(\psi(o)|\bar{v}) = \prod_{s \in S_o^d(\bar{v})} \tau_d^{sp}(s)^{e_d(s)c_s(v)} \prod_{s \in S_o^d(v)} \left(1 - \tau_d^{rec}(s)\right)^{e_d(s)c_s(v)} \tag{3.6}$$

Please note that in Section 5.1.2 there will be a deeper description of some of the concepts introduced briefly here since they're also used in the ADAM algorithm for which we will present our modified version that was used in order to integrate it to our solution.

3.6 Copying affected

The following algorithms, when computing sources' trustworthiness, also take into consideration the probabilities of copying between the sources in order to penalize copiers and promote values coming from independent sources which, more probably, originally provide the values that therefore are more likely to be correct.

AccuCopy [6]: This algorithm extends **AccuFormat** algorithm by introducing the concept of weightning the vote count of each source by the probability that the source provides the value independently. In particular the algorithm starts by setting the same accuracy of each source and the same veracity to each value and then proceeds iteratively with three different phases until the values are stable:

- computes the probabilities of copying based on the veracity of values computed in the last iteration
- sorts the sources based on the probability of being copiers
- updates the veracity of values
- updates the accuracy of the sources

MBM [26]: This algorithm, differently from **AccuCopy** considers the problem of defining copying relationships between sources in a multi-truth context. It starts by reformulating the problem of multi-truth finding by using a *group mapping* model that consists in a series of many-to-many mappings computed by clusterizing sources and values into *source groups* and *value groups* respectively in order to reduce computation. Each source-group represents the maximum number of sources that claim the same group of values while each value-source represents the maximum number of values that are claimed by the same group of sources. An example of this model and how

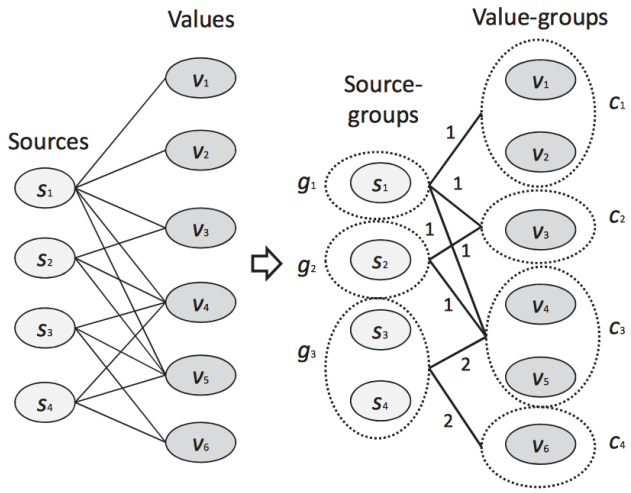


Figure 3.5: Example of transition from single mapping to group mapping

it is computed starting from a more straightforward *single-mapping* model is shown in Figure 3.5. It then applies the classical Bayesian inference iterative process, to the *source groups* and *value groups* granularity, to compute the veracity of each value group c (in this case defined as $a(c)$) based on the observation $\mathcal{X}$. This process is enriched with respect to the DART's one by considering three factors when updating sources trutworthiness:

- *Degree of claim* $w(g, c)$, that is represented on the arcs of the group mapping graph shown in Figure 3.5, and is computed as the cardinality of the source group g that claims a value group c.

- *Confidence score* $\mu(g, c)$ that quantifies the strength with which a the source group s supports or opposes an assertion (value group) c and is computed as shown in Equation 3.7) where $\mathcal{C}$ represents the set of all the value groups and $\mathcal{C}(g)$ is the set of all the value groups proposed by source group s

$$\mu(g,c) = \begin{cases} \frac{1}{|\mathcal{C}(g)|} * \left(1 - \frac{1}{|\mathcal{C}|}\right) & \text{if } c \in \mathcal{C}(g) \\ \frac{1}{|\mathcal{C} \setminus \mathcal{C}(g)|} * \frac{1}{|\mathcal{C}|} & \text{if } c \in \mathcal{C} \setminus \mathcal{C}(g) \end{cases} \tag{3.7}$$

  So a source group by supporting a value group also opposes to others unclaimed value groups at the same time. Note that, as the confidence score definition in **DART**, this score also involves unclaimed values in the computation but, differently from **DART**, in this case if a source

provides a smaller sized value sets it will be assigned a lower confidence score to its values proposed.

- *Independence score* $\mathcal{I}(g, c)$ represent a quantification of sources' dependency impact on the multi-truth discovery process. It is computed in Equation 3.8 as an aggregation of the Independence score computed at the granularity level of the single sources (Equation 3.9)

$$\mathcal{I}(g, c) = \frac{\sum_{s \in g} \mathcal{I}(g, s)}{|g|} \tag{3.8}$$

$$\mathcal{I}(s, c) = \prod_{s_i \in \mathcal{S} \ \wedge \ s_i \text{ provides } c \ \wedge \ s_i \neq s} \frac{1 + P(s \perp s_i | \psi_c)}{2} \tag{3.9}$$

where ψ_c represents the observation that two sources provide the same value-group c. From this point the algorithm proceeds with the computation of the probability $P(s \perp s_i | \psi_c)$ of independence between the two sources s and s_i conditioned to the observation ψ_c. This computation is omitted in this section since it follows a Bayesian inference method which is very similar to the one that will be described in detail in Section 5.1.2 in the implementation of our adapted ADAM algorithm.

ADAM [2]: This algorithm merges MBM's [26] copy detection process, extending it to compute copying probabilities between sources at the domain granularity, with DART's [18] domain aware truth discovery process, which is shown to be better than MBM's [19], extending it with the introduction of the concept of authority score $A_d(s)$ of a source s in a domain d. This score represents how much s is copied in d, with respect to how much all sources are copied in d and is integrated inside DART's Bayesian inference process by substituting Equations 3.5 and 3.6 with Equations 3.10 and 3.11 respectively.

$$P(\psi(o)|v) = \prod_{s \in S_o^d(v)} \tau_d^{rec}(s)^{e_d(s) c_s(v) + A_d(s)} \prod_{s \in S_o^d(\bar{v})} \left(1 - \tau_d^{sp}(s)\right)^{e_d(s) c_s(v) + A_d(s)} \tag{3.10}$$

$$P(\psi(o)|\bar{v}) = \prod_{s \in S_o^d(\bar{v})} \tau_d^{sp}(s)^{e_d(s) c_s(v) + A_d(s)} \prod_{s \in S_o^d(v)} \left(1 - \tau_d^{rec}(s)\right)^{e_d(s) c_s(v) + A_d(s)} \tag{3.11}$$

Chapter 4

Project outline

"The search for truth is more precious than its possession"

Albert Einstein

In this chapter we will describe the functionalities of our algorithm, what are its main characteristics and how it is designed in order to solve the problem of multi-truth Data Fusion in an unsupervised manner.

4.1 Observations

As introduced before the Data Fusion problem arises essentially from 2 main jointed components: high number of sources and conflicting values for the same objects. Starting from this scenario the main idea that we took into consideration during the design and implementation of the algorithm is that in a Data Fusion scenario where usually sources provide conflicting values for a particular object, if there's a source that is the owner of the object in the real world, more probably it will be the one that provides the correct value or, at least, the one that hasn't been contaminated during the copying process that may occur between sources and so that is the most probable to be true [6].
For example considering the case of a dataset concerning flights departure and arrival information, usually the company that provides the flight, as 'Alitalia' or 'Delta', has the most updated information, in terms of delays or cancellations, with respect to other aggregator sources that provide data coming from multiple sources, as 'Skyscanner' or 'eDreams'. Furthermore

most of the proposed copy-detection techniques in the literature (introduced in Chapter 2) [26] conduct the truth finding and copy detection process iteratively until there's a convergence on the veracity of values or copying probabilities. This kind of approach and techniques could be problematic from the scalability point of view, especially in cases of datasets that contains a large number of sources as in the case of web scraped data [16].

4.2 Strategy

For this reason we decided to focus on the identification of what we defined as "owner source", if present in the dataset, or the source that has the most similar characteristics otherwise. In order to identify a source as owner of a particular object we start by recognising in the dataset an attribute that helps categorizing objects based on their provenience and we defined it as "ownership attribute", for example in a dataset that contains information about various books this attribute would be the publisher's name. After that we started identifying, for each value of the "ownership attribute", their respective sources and how those sources covered all the information provided regarding that value across the whole dataset. In particular we defined a score to be assigned to each source that represents the level of "ownership" that the specific source has towards every value of the "ownership attribute". In order to calculate such score we noticed that real world owners usually provide values exclusively for their owned objects and in this way they differentiate themselves from other sources that tend to aggregate information coming from different owners.
Our approach, to find the most reliable values to assign to each data item, is based on the identification of the subset of sources that have the highest probability of being the owners of the single data object and then select for that object the combination of the values provided by those sources since, as explained before, the information provided by them would be at the top of the copying relationships that subsists between sources. The key of our algorithm, that differentiate it from the other presented in Chapter 3, is the computational velocity with which it is able to provide those sources and values before mentioned by approximations and estimation that were not used in the previous iterative Bayesian based algorithms.

4.3 Reference

We decided to leverage on the ADAM algorithm since it is the most recent state-of-the-art algorithm in the Data Fusion scenario and it implements and joints crucial features of the milestones algorithms in the same field such as DART and MBM that will be further discussed in Chapter 5.

4.3.1 Multi-truth

ADAM , as our algorithm does, faces the challenge of the multi-truth possibility where an attribute might have one or more acceptable true value that may vary in the completeness of the response. On the other hand traditional single-truth finding methods believe that when a source support a single or multiple values it opposes to the other potential answers, so it only give credits to sources that provide for the attribute the exact value otherwise, in case of partially wrong or incomplete answers, considers the proposed values as incorrect.

4.3.2 Bayesian approach

ADAM follows the MBM and DART algorithms' Bayesian inference approach in order to evaluate the sources trustworthiness and, in the copy detection process, to compute the copying probabilities that allowed us to maintain an unsupervised approach in the resolution of the Data Fusion problem.

4.3.3 Copy Detection

The best feature for which we decided to use ADAM is the analysis of the relationships that intercourse between sources at value level by confronting the values proposed by couples of sources. In particular it takes into consideration the same values proposed by both sources for common objects and the ones that are different from each other. This whole process generates as output the probabilities, for each couple of sources, that one source is copying from the other (in both directions of copying). In Section 5.1.2 we will describe in detail how these probabilities are computed, how we modified this process and how they're integrated in our truth discovery process.

4.4 Criticism and adaptation

Our work relies and exploits the ADAM algorithm introduced in Chapter 3 especially for the copy detection phase. In particular, in order to further reduce the computation time of the algorithm, we implemented a new version of the algorithm that we called OACD (Owner Aware Copy Detection) that is different from the one presented in three different ways based on three insights:

- Firstly our implementation of ADAM doesn't rely on domain detection since it is probable that datasets in general don't have an attribute that contains information about the domain to which a particular object belongs. For example the two datasets used in the test phase in Chapter 6 aren't suitable for a well-defined domain division. In fact one is focused on US domestic flights while the other is focused about books that belong to the Computer Science genre. Otherwise we decided to focus on a more general and common aspect that is the owner of the information provided since it is very common and most of the time easily inferable from other attributes (as in the case of one of the two test datasets).

- Secondly our implementation of ADAM lacks of multiple iterations because of two main reasons. The first one, that was anticipated before, is that for our purpose we only exploited ADAM to gain copying probabilities between different couples of sources without relying on its computation of the sources' trustworthiness. The second one consists in the fact that we noticed, during the Bayesian iterative process, that the copying probabilities, from one iteration to the next one, were subject to a small change in terms of value compared to the ones computed in the first iteration, so in order to be able to propose a quick solution, we considered the first iteration's probabilities reliable enough to be included in our computation.

- Lastly our implementation of ADAM during the copy detection process relies on the ownership score of each source instead of using the percentage of objects for which each source provides a value. This aspect will be deeply explained in Section 5.1.2.

4.5 Notation

In Table 4.1 is described the notation that will be used in the following chapters in order to formally describe our algorithm.

Notation	Description
$\mathcal{S}$	Set of all sources
$\mathcal{O}$	Set of all data object provided by at least one source
$\mathcal{O}(s)$	Set of all objects provided by source s
$\mathcal{O}^p(s)$	Set of all data object with owner p provided by source s
O	Set of all data owners that provide at least one object
$\texttt{Score}^s(p)$	Ownership score of source s regarding owner p
Θ_{ij}	Set of common objects provided by sources i and j
$V_s(o)$	Set of all values proposed by source s for object o
$c_{ij}(o) = c$	Values provided by both sources i and j for object o
$\psi(o)$	Observation of values provided for object o
$\psi^o_{ij} = \psi_c$	Observation of c values
$s_i \rightarrow s_j$	Source s_i copies from source s_j
$s_i \perp s_j$	Source s_i and source s_j are independent
$\mathcal{S}_{copy}(s)$	Sources in $\mathcal{S}$ for which the $P(s_i \rightarrow s_j)$ is greater than 0.5
τ^{sp}	Specificity of source s
τ^{rec}	Recall of source s
$e(s)$	Expertise of source s
$\sigma_o(v)$	Veracity of value v for obect o

Table 4.1: Table representing the notation used and its meaning

Chapter 5

Design and implementation

"Truth is ever to be found in simplicity, and not in the multiplicity and confusion of things."

Isaac Newton

In this chapter we'll present, discuss and formalize completely the algorithm in detail providing also it is pseudocode. In particular we'll describe, also through a practical example, the evolution of our algorithm and the insights that lead us defining the different components that came together to form the final result that we used in the testing phase described in Chapter 6.

5.1 Concept

The algorithm is designed in order to identify for each object in the dataset it is owner source and use its values proposals in order to return the values that have the best fit for the object. It works by the classification of the sources based on the value of the ownership score which calculation evolved in two different phases. We will describe these two phases referring to the example regarding a dataset that contains information about 4 different books provided by 5 different sources, one of which (S3) simulates a real world owner, show at Table 5.1. In Table 5.2 we present also the golden truth values for the two books taken into consideration.

Bookstore dataset				
ISBN	**Source**	**Book title**	**Book author**	**Publisher**
1558609350	S1	Understanding Your Users	Kathy Baxter, Catherine Courage	Elsevier
1558609350	S2	Understanding Your Users	Baxter, Kathy	Elsevier
1555582990	S1	Leveraging WMI Scripting	Alain Lissoir	Elsevier
0201633582	S2	MFC Programming	Alan Feuer	Pearson
0201633582	S3	MFC Programming	Alan R. Feurer	Pearson
0130284467	S3	Data Structures and Algorithm Analysis	Shaffer, Clifford A.	Pearson
0201633582	S4	Understanding Your Users	Catherine Courage	Elsevier
1555582990	S4	Leveraging WMI Scripting	Alain Lissoir	Elsevier
1558609350	S5	Understanding Your Users	Kathy Baxter, Catherine Courager	Elsevier
1555582990	S5	Leveraging WMI Scripting	Alain Lissoir	Elsevier
0201633582	S5	MFC Programming	Alan R. Feurer	Pearson

Table 5.1: Table representing an example of a multi-truth dataset where only source S3 is a real world owner

5.1.1 Phase one

In the first phase of the algorithm we first identify the 3 main elements needed to classify the sources :

- an attribute that identifies uniquely every source that from now on we'll refer to as '*source id*'
- an attribute that identifies uniquely every real world owner for every object that from now on we'll refer to as '*owner id*'
- one or more attributes which truth values are to be discovered that

Bookstore dataset			
ISBN	**Book title**	**Book author**	**Publisher**
1558609350	Understanding Your Users	Kathy Baxter, Catherine Courage	Elsevier
1555582990	Leveraging WMI Scripting	Alain Lissoir	Elsevier
0201633582	MFC Programming	Alan R. Feuer	Pearson
0130284467	Data Structures and Algorithm Analysis	Clifford A. Shaffer	Pearson

Table 5.2: Table representing the truth values of the books presented in Table 5.1

from now on we'll refer to as '*value attribute*'

In the example these attributes were assigned as shown in Table 5.3:

source id	Source
owner id	Publisher
value attribute	Book author

Table 5.3: Table representing the choice of the key attributes for the algorithm

Then, after performing a data cleaning in order to be able to compare the different values and id attributes, we started to compute a first version of the ownership score that took into account how much a source behaves similarly as a real world owner. In order to accomplish this we firstly identified and collected all the different values that were included in the dataset for the *owner id* and *source id* attributes as shown in Table 5.4.

source id	S1, S2, S3, S4, S5
owner id	Elsevier, Pearson

Table 5.4: Table representing the sets of values for the owner id and source id attributes for the example

Then, for each combination, we computed the coverage of each source with reference to every *owner id* while iterating over each row of the dataset. In

the next we create a matrix containing the results of the score computation with reference to each source and each owner attribute, which formula is described in the next section.

First Ownership score

The first ownership score calculated for a source s with reference to an owner p only takes into account the structural properties of the dataset and it is computed by calculating the coverage, in terms of how much information (tuples) about p's objects inside the dataset was provided by source s, and weight it by the of uncoverage (1 - coverage) by s of all the information regarding other owners' objects except p's.

$$\texttt{Score}^s(p) = 100 * \frac{|\mathcal{O}^p(s)|}{\sum_{s' \in \mathcal{S}} |\mathcal{O}^p(s')|} * \left(1 - \frac{\sum_{p'|p' \neq p} |\mathcal{O}^{p'}(s)|}{\sum_{s'|s' \neq s} \sum_{p'|p' \neq p} |\mathcal{O}^{p'}(s')|} \right) \tag{5.1}$$

In this way the score will enhance the specificity and the completeness of the coverage of s for the objects that fall into the ones owned by the owner o. To give a better readability of the results and to give a better balance for the second phase of the ownership score calculation that will be described below, we then decided add to the formula the multiplication by 100 term at the beginning. In particular this formula gives a value in the range [0,100] which, in the ideal case of a source that's a real world owner and that's the only provider of the information, will be maximized and, in case of aggregator sources, will be lowered by the fact that the source propose values about object relative to other owners or by the fact that the source doesn't have information about one or more objects in its supposed domain of ownership. So the score can be lowered locally by the single sources by providing values for data objects that belong to different owners (losing specificity) or can be lowered globally by the presence of other sources providing information for the same objects. Referring to the example dataset (Table 5.1), in Table 5.5 we present the matrix containing the results for the computation of the first ownership score.

From these results is it possible to notice that:

- sources S1 and S4 have an equal score that's the highest regarding the Elsevier's books since they both provide full coverage of that domain

Score	**Elsevier**	**Pearson**
S1	$100\times\frac{2}{7}\times(1-\frac{0}{4}) = \mathbf{28.6}$	$100 \times \frac{0}{4} \times (1 - \frac{2}{7}) = \mathbf{0}$
S2	$100\times\frac{1}{7}\times(1-\frac{1}{4}) = \mathbf{10.7}$	$100\times\frac{1}{4}\times(1-\frac{1}{7}) = \mathbf{21.4}$
S3	$100 \times \frac{0}{7} \times (1 - \frac{2}{4}) = \mathbf{0}$	$100\times\frac{2}{4}\times(1-\frac{0}{7}) = \mathbf{50}$
S4	$100\times\frac{2}{7}\times(1-\frac{0}{4}) = \mathbf{28.6}$	$100 \times \frac{0}{4} \times (1 - \frac{2}{7}) = \mathbf{0}$
S5	$100\times\frac{2}{7}\times(1-\frac{1}{4}) = \mathbf{21.4}$	$100\times\frac{1}{4}\times(1-\frac{2}{7}) = \mathbf{17.9}$

Table 5.5: Table representing the first ownership score calculation for the example

and don't propose any information about other domains, their score is lowered by the popularity of the tuples regarding those objects inside the dataset. In fact they would have accomplished the maximum ownership score (100) if wasn't that other sources were also proposing values for the 2 Elsevier's books bringing the denominator of the first term of the formula equals to 7.

- source S3, since it represents a real world owner that proposes all the authors for its books, has the highest value for Pearson's book for the same reasons as S1 and S4 with Elsevier. This score is much higher

than theirs because of the different popularity of the objects provided. In fact in case of Pearson's books we have only 4 total value proposal in the dataset against the 7 of Elsevier's book mentioned before.

- sources S2 and S5 have lower scores than the other sources because of a local issue that is regarding the objects for which they propose information or, in other words, because of their lack of specificity in either one of the *owner id*'s values. In fact we can notice that the unbalance between the objects provided by S5 (2 Elsevier and 1 Pearson) opposed to S2's behaviour (1 Elsevier and 1 Pearson), is reflected by the fact that its score is higher than S2's one for Elsevier and lower for Pearson. Furthermore the different popularity of those owners' objects in the dataset makes the difference in scores higher in the case of Elsevier (17.9) tha in the case of Pearson (3.5) because in this way sources that provide information about uncommon objects are rewarded.

True value selection

In this version of the algorithm we adopted a first, more straightforward approach in order to select the value to be proposed as true for each object. In fact we decided to assign the ownership of each *owner id* value to the source that resulted having the highest ownership score regarding its objects and to give as a result for the *value attribute* the one proposed by the owner source found (if it has a solution for that particular object). This strategy revealed to be intuitive and yet very effective for a single-truth scenario but didn't exploited the multi-truth features of some datasets. In fact, as can be seen from the example, while for books with a single author such as Elsevier's *Leveraging WMI Scripting* and Pearson's books the algorithm gives the complete answer in terms of authors, otherwise in the multi-truth case as for the Elsevier's book *Understanding Your Users* the solution proposed may be incomplete because the source selected as owner proposed only one of the two authors expected. In fact the owner selected (S1 or S4) provide one single author instead of the two requested.

During a first phase of testing this approach revealed to be promising in terms of results and computationally fast in giving an approximated solution to our problem. So we decided to refine it by considering not only the relationship in terms of coverage between sources and owner but also con-

sidering the relationship that occurs between different sources. Furthermore we designed a more refined strategy for the composition of the result for the *value attribute* that fits better into the multi-truth context.

5.1.2 Phase two

In the second phase of the algorithm we decided to incorporate those copying relationship mentioned above by evaluating the fact that sources, especially in a web based context, tend to exchange, modify and copy information between each other. So we used a modified version of the ADAM algorithm, that we defined "OACD" as Ownership Aware Copy Detection, in order to compute the copying probabilities between each pair of sources taking into consideration the values and the coverage they presented.

OACD

As introduced before in Chapter 4 the ADAM algorithm has been modified in order to provide, as soon as possible during the computation, reliable copying probabilities without leveraging on the separation in domains for the data objects. In this section there's a deeper description of how our version of ADAM, given in input the datasets, the *source id*, *value id* and values attributes, it computes the before mentioned probabilities by leveraging the copy detection approach also used in MBM and focusing on the ownership score of the first phase described before.

Consideration of common values : We start by defining as ψ_{ij}^{o} the observation of the common values $c_{ij}(o)$ provided by two different sources s_i and s_j regarding an object $o \in \Theta_{ij}$ that they both provide defined in Equations 5.2, 5.3.

$$\Theta_{ij} = \mathcal{O}(s_i) \cap \mathcal{O}(s_j) \qquad \text{with } s_i \neq s_j \tag{5.2}$$

$$c_{ij}(o) = V_{s_i}(o) \cap V_{s_j}(o) \qquad \forall o \in \Theta_{ij} \tag{5.3}$$

From now on for each triple $\langle s_i,\ s_j,\ o \rangle$ with $s_i, s_j \in \mathcal{S},\ o \in \Theta_{ij}$ we will omit some parameters for ease of notation by using $c_{ij}(o) = c$ and $\psi_{ij}^{o} = \psi_c$.
Using the notation introduced in Section 4.5, we will now describe how we

will use the Bayesian inference approach, explained in Section 2.4, in order to compute the posterior probabilities, given the observation of the common values ψ_c, that a couple of sources s_i and s_j are involved in a copying relationship (so either $P(s_i \to s_j|\psi_c)$ or $P(s_j \to s_i|\psi_c)$) or are independent sources ($P(s_i \perp s_j|\psi_c)$). As explained before the first step consists in obtaining the elements needed to compute the Equation 2.2 which are the likelihood of the observation ψ_c and the prior probabilities of the 3 types of relationships described above ($s_i \to s_j$, $s_j \to s_i$ and $s_i \perp s_j$).
Notice that we are considering only directional copying relationships between sources since the main assumption that we take into consideration in this copy detection process is that we assume there's no mutual copying between sources. So for every couple of sources each source can be either a copier source or an independent source as shown in Equation 5.4.

$$P(s_i \to s_j|\psi_c) + P(s_j \to s_i|\psi_c) + P(s_i \perp s_j|\psi_c) = 1 \tag{5.4}$$

Defining $\sigma(v)$ the veracity of value v, in Equation 5.5 we then define the probability of the values observed ψ_c of being correct as the probability that all the values in c are correct.

$$\sigma(c) = \prod_{v \in c} \sigma(v) \tag{5.5}$$

Now it is possible to define the formulas that express the likelihood of ψ_c in the different cases of source dependence and value veracity using Equations 5.6, 5.7 and 5.8.

$$\begin{cases} P(\psi_c|s_i \to s_j,\ c \text{ true}) = P(\psi_c|s_j \to s_i,\ c \text{ true}) = 1 \\ P(\psi_c|s_i \to s_j,\ c \text{ false}) = P(\psi_c|s_J \to s_i,\ c \text{ false}) = 1 \end{cases} \tag{5.6}$$

$$P(\psi_c|s_1 \perp s_2,\ c \text{ true}) = \tau^{rec}(s_1)\ \tau^{rec}(s_2)\ [1 - \tau^{sp}(s_1)]\ [1 - \tau^{sp}(s_2)] \tag{5.7}$$

$$P(\psi_c|s_1 \perp s_2,\ c \text{ false}) = \tau^{sp}(s_1)\ \tau^{sp}(s_2)\ [1 - \tau^{rec}(s_1)]\ [1 - \tau^{rec}(s_2)] \tag{5.8}$$

Note that, as the definition of copier suggests, in Equation 5.7 we have also formalized the fact that if 2 different sources are copiers they will provide the same common values c independently from their veracity as was already introduced in [26].

Once the likelihood of ψ_c is obtained, in order to apply the Bayesian equation we now only need to express the *prior probabilities* related to the copying relationships between the two sources which are $P(s_i \rightarrow s_j)$, $P(s_j \rightarrow s_i)$ and $P(s_j \perp s_i)$ that for ease of notation will be defined as described in Equation 5.9.

$$\begin{cases} P(s_i \rightarrow s_j) & = \rho_{ij} \\ P(s_j \rightarrow s_i) & = \rho_{ji} \\ P(s_j \perp s_i) & = 1 - \rho_{ij} - \rho_{ji} \end{cases} \tag{5.9}$$

These ρ_{ji} values are initialized based on the number of object for which each source has a claim as explained in Equation 5.10

$$\rho_{ij} = [1 - e(s_i)]\, e(s_j) \qquad \forall s_i,\ s_j \in \mathcal{S} \ \wedge \ s_i \neq s_j \tag{5.10}$$

where $e(s)$ is the *expertise score* of the source s. Originally in ADAM this score represents the portion of objects for which s is proposing some value and is computed as shown in Equation 5.11 (using the α parameter to modulate its weight)

$$e(s) = \sqrt{1 - \big(\alpha \cdot P(s) - 1\big)^2} \tag{5.11}$$

$$P(s) = \frac{|\mathcal{O}(s)|}{\left|\bigcup_{s' \in S} \mathcal{O}(s')\right|} \tag{5.12}$$

Note the fact that the ADAM expertise score is designed in order to give a lower copying probability to sources which provide values for a large amount of data objects. Since this approach is conflicting with our definition of owner source and the idea that it doesn't provide values for a copious amount of objects but only for those which are related to that owner, we decided to redefined the expertise score of a source s in terms of the maximum ownership score obtained by s as shown in Equation 5.13. We adopted this measure since it defines how much s is an owner for any of the *owner id* attribute's values (note that a real world owner is characterized by one high ownership score towards the owned items and many low ownership scores towards other *owner id* attribute's values).

$$e(s) = \frac{\max \texttt{Score}^s(p)}{100} \tag{5.13}$$

We multiplied the maximum ownership score by $\frac{1}{100}$ in order to have an ownership score value in the range of [0,1] and maintain in such way the properties

of the probabilities. The multiplication by 100 introduced in Equation 5.1 will be shown to be useful when we will combine the copying probabilities in the ownership score that will be show in Equation 5.18. In this way we will penalize, by attributing higher copying probabilities, sources that are not probable owners (ownership score < 0.5) proportionally to their volume of information provided, without penalizing owner sources which provide an high volume of values with an high specificity. Once we computed the prior probabilities we have all the elements needed in order to apply the Bayesian inference approach to calculate the probabilities, for a couple of sources, of the 3 possible different copying relations that can occur between them $Y = \{s_i \to s_j;\ s_j \to s_i;\ s_i \perp s_j\}$ that is shown in Equation 5.14.

$$\begin{aligned} P\left(y|\psi_c\right) &= \frac{P\left(\psi_c|y\right)\ P\left(y\right)}{\sum_{y' \in Y} P\left(\psi_c|y'\right)\ P\left(y'\right)} \\ &= \frac{P\left(y\right)\left[P\left(\psi_c|y,\ c \text{ true}\right)\ \sigma\left(c\right) + P\left(\psi_c|y,\ c \text{ false}\right)\ \left(1-\sigma\left(c\right)\right)\right]}{\sum_{y' \in Y} P\left(y'\right)\ \left[P\left(\psi_c|y',\ c \text{ true}\right)\ \sigma\left(c\right) + P\left(\psi_c|y',\ c \text{ false}\right)\ \left(1-\sigma\left(c\right)\right)\right]} \end{aligned} \tag{5.14}$$

Then, by replacing Equations 5.5, 5.6, 5.7, 5.8 and 5.9 into Equation 5.14 we obtain the following result for the posterior probabilities of copying

$$P\left(s_i \to s_j|\psi_c\right) = \frac{\rho_{ij}}{\rho_{ij} + \rho_{ji} + \left(1 - \rho_{ij} - \rho_{ji}\right)\ P_u} \tag{5.15}$$

where

$$\begin{aligned} P_u = \ & \sigma(c)\ \left[\tau^{rec}(s_i) \cdot \tau^{rec}(s_j) \cdot (1 - \tau^{sp}(s_i)) \cdot (1 - \tau^{sp}(s_j))\right] + \\ & + \big(1 - \sigma(v)\big)\ \left[\tau^{sp}(s_i) \cdot \tau^{sp}(s_j) \cdot (1 - \tau^{rec}(s_i)) \cdot (1 - \tau^{rec}(s_j))\right] \end{aligned} \tag{5.16}$$

Consideration of uncommon values : Using Equation 5.15 we only expressed the probability that a source s_i copies from another source s_j their common values proposed c for an object o. Now we need to take into consideration also the eventual values that they don't share in order to have a complete overview of the behaviour of the two sources. In order to do that

we scale the probability calculated before by means of the Jaccard similarity of the sets of value that the two sources s_i and s_j propose for object o. The computation of such similarity score is described in Equation 5.17.

$$J_{ij}(o) = J_{ji}(o) = \frac{\left|V_{s_i}(o) \cap V_{s_j}(o)\right|}{\left|V_{s_i}(o) \cup V_{s_j}(o)\right|} \tag{5.17}$$

As explained in Chapter 4 we considered only the first iteration of the ADAM algorithm since we observed that the copying probabilities are not affected by a significant variation in terms of values. In order to do so we performed a parameter tuning to discover the best way to initialized the parameters requested which are:

- The default τ^{rec}
- The default τ^{sp}

Final Ownership score

The final ownership score calculated for a source s with reference to an owner o is meant to adjust and weight the first phase's score in order to take into account the copying relationship between sources. In fact a typical characteristics of real world owner sources is that they tend to not copy from other sources since they're the ones that first publish the information. On the other hand they tend to be the target sources to be copied by the aggregators. In order to consider a source s as copier of another source s' we decided that its probability of copying should have been higher than 0.5, for this reason during the computation of the score we considered only the probabilities that satisfied this condition. As illustrated in Table 4.1 and accordingly to the definition of the concept of 'copier' provided above, we define the set of sources copiers of a source s as $\mathcal{S}_{copy}(s)$. During the definition of this score we evaluated multiple ways to incorporate in a effective way the probabilities in order to gain the best results and we found the best fit in the Equation 5.18.

$$\texttt{Score}^s(p) = \left(100 * \frac{|\mathcal{O}^p(s)|}{\sum_{s' \in \mathcal{S}} |\mathcal{O}^p(s')|} * \left(1 - \frac{\sum_{p'|p' \neq p} |\mathcal{O}^{p'}(s)|}{\sum_{s'|s' \neq s} \sum_{p'|p' \neq p} |\mathcal{O}^{p'}(s')|} \right) \right)^{exponent} \tag{5.18}$$

where

$$exponent = \left(1 - \frac{\sum_{s' | s' \in \mathcal{S}_{copy}(s)} P\left(s' \rightarrow s\right)}{|\mathcal{S}_{copy}(s)|} \right) \tag{5.19}$$

In this way in our classification system we enhance sources that tend to be the least copiers so that their information can be considered "original", the most updated and, for those reasons, the most reliable.

Notice that the OACD's process of computing the copying probabilities gives results which accuracy is proportional to the dimension on the dataset analyzed, since, using larger datasets, it is possible to have a more complete observation of the behaviour of the sources involved. For this reason in Table 5.5 we present the estimated copying probabilities that, as the ones computed by OACD, are based on the number of common object covered and the number of common values proposed between each pair of sources, supposing to extend those behaviours to a larger dataset.

p-copy	**S1**	**S2**	**S3**	**S4**	**S5**	***Mean***
S1	0	0.1	0	0.3	0.5	0
S2	0.3	0	**0.6**	0	0.5	0.6
S3	0	0.2	0	0	0.4	0
S4	0.3	0	0	0	0.4	0
S5	**0.9**	**0.7**	**0.6**	0.3	0	0.73

Table 5.6: Table representing the probabilities of copying between each pair of sources for the example

Referring to the example dataset (Table 5.1), Table 5.6 presents the matrix containing the results for the computation of the final ownership score.

From these results is it possible to notice the fact that the source S5, which copied source S1 for the values proposed for objects with *owner id* equals to Elsevier, and the source S2 which, as S5, copies S3's values about one of Pearson's books are strongly penalized. In fact in Table 5.6 is possible to see that their scores dropped and, regarding Elsevier, S5 now has a lower score than S2 which provides one book less.

Score	**Elsevier**	**Pearson**
S1	$28.6^{(1-0)} = \mathbf{28.6}$	$0^{(1-0)} = \mathbf{0}$
S2	$10.7^{(1-0.6)} = \mathbf{2.58}$	$21.4^{(1-0.6)} = \mathbf{3.41}$
S3	$0^{(1-0)} = \mathbf{0}$	$50^{(1-0)} = \mathbf{50}$
S4	$28.6^{(1-0)} = \mathbf{1}$	$0^{(1-0)} = \mathbf{0}$
S5	$21.4^{(1-0.73)} = \mathbf{2.29}$	$17.9^{(1-0.73)} = \mathbf{2.17}$

Table 5.7: Table representing the final ownership score calculation for the example

True value selection

As introduced before in order to present as much information as possible to allow our solution to work efficiently also in multi-truth cases we introduced a new strategy to select the results once the ownership scores are computed. First of all we redefined the concept of owner source extending it to more than just the one source with the maximum score value. In fact we decided to consider as *owners* the sources with the top 3 scores and combined their results in order to match the true one. Furthermore in order to accomplish that we used the *FuzzyWuzzy* library and two different string comparison

methods, that will all be described later in Chapter 6, to express all the values using the same format and so to be able to compare it. Finally, having the top 3 sources with their proposed values, we propose as solution the union of all the proposed values avoiding eventual repetitions. This technique has revealed to be very effective since in multi-truth web based scenarios sources tend to miss values instead of proposing a wrong one and, those which do error, won't be taken into consideration since probably will not have an ownership score high enough. As a result it is possible to notice that for the example the final value found for the Elsevier's book *Understanding Your Users* is not anymore the single author *Kathy Baxter* proposed by S1, but now it is the combination of S1's and S5's *Kathy Baxter* with S2's *Catherine Courage* that matches the true value.

5.2 Algorithm's pseudocode

Algorithm 1: Ownership score algorithm

input: $O,\ \mathcal{O},\ \mathcal{S},\ V_s(o)\ \forall s \in \mathcal{S},\ o \in \mathcal{O}$

foreach $p \in O$ **do**
  foreach $s \in \mathcal{S}$ **do**
    compute $|\mathcal{O}^p(s)|$;
  end
end
foreach $p \in O$ **do**
  foreach $s \in \mathcal{S}$ **do**
    p by $s \longleftarrow |\mathcal{O}^p(s)|$;
    foreach $p' \in O$ **do**
      if $p' \neq p$ **then**
        p' by s += $|\mathcal{O}^{p'}(s)|$;
      else
      end
    end
  end
  compute OACD to obtain $P\left(s_i \rightarrow s_j\right)\ \forall\ s_1,\ s_2 \in \mathcal{S}$;
  foreach $s \in \mathcal{S}$ **do**
    compute $\mathcal{S}_{copy}(s)$;
  end
  $\texttt{Score}^s(p) \leftarrow$ result of Equation 5.11 ;
end
$S^p \leftarrow \{s_1,\ s_2,\ s_3 \mid \nexists\ s' \in \mathcal{S},\ \texttt{Score}^s(p)\ >\ \texttt{Score}^{s_1}(p)\ \&$
$\texttt{Score}^s(p)\ >\ \texttt{Score}^{s_2}(p)\ \&$
$\texttt{Score}^s(p)\ >\ \texttt{Score}^{s_3}(p)\}$;
return $\{V_{s_1}(o) \bigcup V_{s_2}(o) \bigcup V_{s_3}(o) |\ s_1,\ s_2,\ s_3 \in \mathcal{S}^p,\ o \in \mathcal{O}^p\}\ \forall\ o \in \mathcal{O}$

Chapter 6

Experimental results

"Experience never errs; it is only your judgments that err by promising themselves effects such as are not caused by your experiments."

Leonardo da Vinci

In this chapter we will describe in which context we tested our solution and we will provide comparable measures in order to confront its performances with the ones of ADAM, DART and Majority Vote. In section 6.1 we will describe the two datasets used for the experimentation phase, in section 6.2 we will explain which operations were performed in order to clean them, in section 6.3 we will describe the metrics of comparison used to confront our solution's performances with the other ones already proposed and we present our tests' results in term of such metrics.

6.1 Dataset description

In this section we will describe the two datasets used for the testing phase of our algorithm in order to assess its features. We used two datasets taken from [5] which contains datasets created for making experiments using Data Fusion techniques. From Luna Dong's pool of dataset we selected the one regarding the information about books, which inspired the example used in Chapter 5, and the one regarding the information about flights. We chose to use these two datasets because they enhance different aspects of the Data Fusion problem (i.e. multi-truth vs single-truth) and also because they're commonly proposed in experiments for most of the Data Fusion techniques introduced at Chapter 3.

6.1.1 Books' dataset

This dataset contains information about books related to the Computer Science field that have been collected from the online bookstore aggregator AbeBooks.com in 2007. It comprehends 1263 books and 894 data sources which are all bookstores: aggregators of books from different publishers [27][6] for a total of 33965 tuples. For each book it provides the information presented in table 6.1

Bookstore dataset			
ISBN	Source	Book title	Author list

Table 6.1: Table representing the attributes of the books' dataset

As a comparison for our results we took as a reference the gold standard proposed in [5] which contains the precise author lists manually obtained from book covers of 100 randomly selected books.
It's worth notice that this dataset lacks of sources that are real world owners (in this case publishers) and, according to the example used in Chapter 5, it requires a multi-truth approach since the number of authors per book can be greater than one.
In particular in Table 6.2 we show how the cardinalities of the authors parameter is distributed across the books claims and golden truth.

Number of authors	**Books in dataset**	**Books in golden truth**
1	23531	43
2	6724	40
3	2589	11
4	672	3
5	236	1
6	123	1
7	51	1
>7	39	0
Total	**33965**	**100**

Table 6.2: Table representing the cardinalities of the Author list attribute's values

6.1.2 Flights' dataset

This dataset contains information about over 1200 flights from 38 sources over 1-month period (December 2011) [15] for a total of 70688 tuples. Note that all the flights contained in the dataset are provided by 3 providers which are *American Airlines*, *Continetal Airlines* and *United Airlines*. For each flight it provides the information presented in table 6.3, note that not every tuple has all the information described below

Flights dataset			
Source	Flight number	Scheduled departure	Actual departure
Departure gate	Scheduled arrival	Actual arrival	Arrival gate

Table 6.3: Table representing the attributes of the flights' dataset

As a comparison for our results we took as a reference the gold standard proposed in [5] which contains the departure/arrival information on 100 randomly selected flights provided by the corresponding airline website for a total of 2964 tuples (because the same flight might be repeated in different days).
Differently from the books' dataset, this one contains the 3 sources that are real world owners (in this case providers) of the flights. Furthermore, in order to have a more complete observation of the behaviour of our algorithm, we decided to approach this dataset using a single-truth methodology. In fact during the tesitng phase we considered and compared the value proposed only for the *Scheduled departure* attribute that, since is composed of a date and a time, doesn't leave room for incomplete answers and allow us to take a binary decision in whether or not the value proposed by a source was correct. Between all the attributes in the dataset we decided to take the *Scheduled departure* into consideration since it is one of the few which is proposed in every tuple of the dataset unlike other attributes (i.e. *Departure gate*).
In Table 6.4 we present the cardinalities of the 3 providers regarding all the tuples in the dataset and the ones contained in the golden truth.

Flight provider	Flights in dataset	Flights in golden truth
American Airlines	38062	1491
Continetal Airlines	11690	527
United Airlines	20936	946
Total	**70688**	**2964**

Table 6.4: Table representing the cardinalities of the Author list attribute's values

6.2 Dataset cleaning

6.2.1 Books' dataset

The Books' dataset had been cleaned and extended in order to perform a significant test phase with the following 3 operations: Author list cleaning, Publisher retrieval and Publisher scraping.

Author list cleaning

As explained before the values proposed in the books dataset came from different data sources which clearly didn't use the same formats to represent their information. In particular this was a problem for the *Author list* because it wasn't possible to compare values for the same object since some sources tend to add noisy words between authors (i.e. PhD, CISM) and used different formats in the presentation of the same names.
For example the authors from the golden truth for the book *Wavelets for Computer Graphics*

- *stollnitz, eric j.; derose, tony d.; salesin, david h.;*

inside the dataset were proposed using the following formats :

- *Stollnitz, Eric J.; Derose, Tony D.; Salesin, David H.*
- *Eric Stollnitz*
- *Stollnitz, Eric*
- *Stollnitz, Eric J./ Derose, Tony D./ Salesin, David H.*

In order to clean those values and to be able to compare them we firstly split each raw value, using as separator strings like “and”, “&”, “;”, “with”, “/” and we transformed every string in lowercase. Then we exploited Python’s *FuzzyWuzzy* library’s *token sort ratio* scorer which tokenizes a string by spaces, it sorts the tokens alphabetically and assigns a score related to the similarity of these sorted strings. In the last phase of the algorithm, when we had to compare the resulting strings in order to select the final results, we used two different methods of comparison and evaluated the string as equals if at least one of the following gave a sufficient score of similarity:

- by using a similar method as *token sort ratio*
- by using the Levenshtein distance ratio of similarity

Publishers retrieval

Other than the authors comparison for the sake of our algorithm we also had to find the publishers for the dataset’s and the golden truth’s books since they were not provided. So in order to create a *Publisher* column, as the one proposed in Chapter 5’s example, we leveraged another Python’s library called *isbnlib* which, given a book’s ISBN, could find its information consulting two websites : openlibrary.org and books.google.com.

Publishers scraping

Lastly, since there was no real world owner source inside the data, we decided to select the 3 most popular publishers inside the dataset (Pearson, Elsevier and McGraw-Hill) and extend it by inserting them as legitimate sources. To do so we decided to perform a scraping operation of those publisher’s websites Computer Science related sections in order to retrieve the information needed for our attributes. Since we noticed that all the websites’ contents related to books were dynamically inserted inside the HTML of the webpage via javascript, it was impossible to perform a static analysis of the raw HTML using the *BeautifulSoup* library. For this reason we decided to use Python’s library *Selenium* which allows to simulate a user web browser’s session and to automatically interact with the web page’s interactive elements (i.e. links, buttons) giving in this way the possibility to navigate into the websites to reach the individuals book pages loaded including the dynamic content. Once a book page with all its information was reached *Selenium* also allowed to

scrape it by retrieving the content of the HTML tags of interest. Lastly, after the acquisition of all the Computer Science books from those 3 publishers, we applied a join operation and inserted in the dataset, specifiyng the publisher as sources, only the tuples regarding books already proposed by at least another sources. This join was performed in order not to unbalance the score computation in favour of the real publisher sources just inserted.

6.2.2 Flights' dataset

Regarding the Flight's dataset we have different sources that propose different hours of departure/arrival for various flights. We noticed that there was no attribute among the ones described in Section 6.1.2 which value was unique for each physical flight in order to identify and consider it as a data object. In fact the *flight number* provided identifies the route travelled by a flight but this code would be the same if the flight is repeated on another day. Therefore we decided to identify each single flight object by creating a *flight id* attribute for each tuple by concatenating its *Flight number* and *Scheduled departure* date values. The main problem in this case was that each source provided the *Scheduled departure* date in very different formats that sometimes even lacked of the information about the year. In particular we noticed that in total there were twelve different formats for the dates and some source presented information in two or more of them inside the whole dataset. Because of these two reasons we decided to build our own parser, instead of using a library, in order to convert dates in a common format in order to create the identifier described above.
Furthermore, since in the verification process of our algorithm we decided to confront as value proposed by sources the whole *Scheduled departure* date and time attribute in order to create a single-truth context, we implemented also another parser to convert the different time formats into a unique one and to create an attribute *Parsed Scheduled departure* that contained values for each tuple that were able to be confronted.

6.3 Results

In this section we describe the metrics used for the comparison of our algorithm with other state-of-the-art solutions covering most of the different Data Fusion techniques listed in Chapter 3.

6.3.1 Metrics

In order to compare the performances of the algorithms we decided to give a quantitative description of their results and computations using the computation time, the precision score (which represents the percentage of values provided that were correct), the recall score (which represents the percentage of the correct values that were provided) and the F1 score (which is the harmonic mean of the scores former described).

6.3.2 Outcome

Our experiments were performed using a MacBook Pro equipped with a dual core 2.9 Ghz Intel Core i5 processor with 8GB RAM and a 256 GB SSD.
We performed two different kind of analysis on both single and multi truth datasets where the first one aimed at the comparison the two different phases of our solution and the second one compared our second phase's solution with others state-of-the-art solutions such as ADAM, DART and Majority Voting.

Multi-truth : Regarding the multi-truth book dataset we started by calculating the difference in terms of performances of our two phases of the algorithm in order to measure the impact of the inclusion of the copying probabilities into our computations. We implemented the two different phases as described in Chapter 5 and we observed their performance measures evolving while we used an increasing number (k) of sources to be considered "owners" of the information of each owner attribute value. From the results presented in Figures 6.1, 6.2 and 6.3, emerges the fact that, by including the OACD algorithm, the performances of our solution tend to slightly increase uniformly from every point of view. The fact that the change in the performances' measures values is not so evident can be caused by the lack of significant copious copy relationships inside the books' datasets taken into consideration. As a matter of fact is useful to remember that in the Phase 2 we will only modify the score related to sources that has been demonstrated to be copiers (copying probability > 0.5), so evidently our dataset lacks of a big number of potential owners sources which are copiers. Another important aspect to highlight from this analysis is the fact that the precision of both approaches starts decreasing when considering more than three owners in order to compute the final results, while the recall is still increasing at a lower rate causing

a decrease of the F1 measures. This highlights the fact that combining the value propositions coming from more than three sources the results tend to be more slightly more complete in terms of number of true authors returned but the other sources included in the solution, which have a less ownership score, introduce a much higher number of wrong value propositions lowering the overall performance.

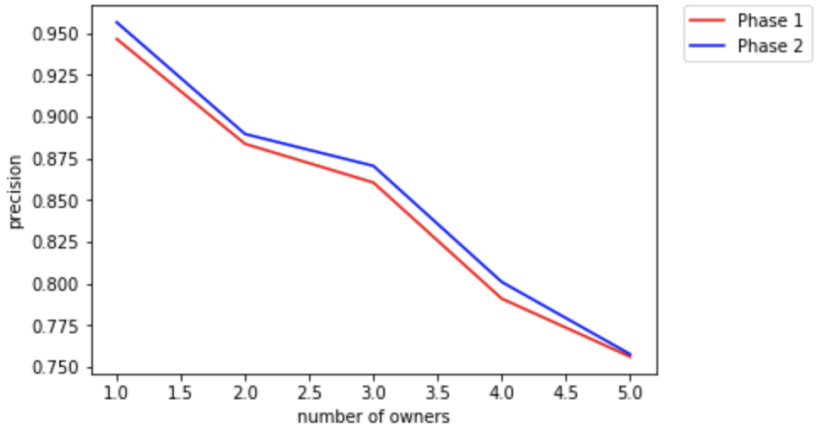


Figure 6.1: Precision comparison between the two phases with k owners

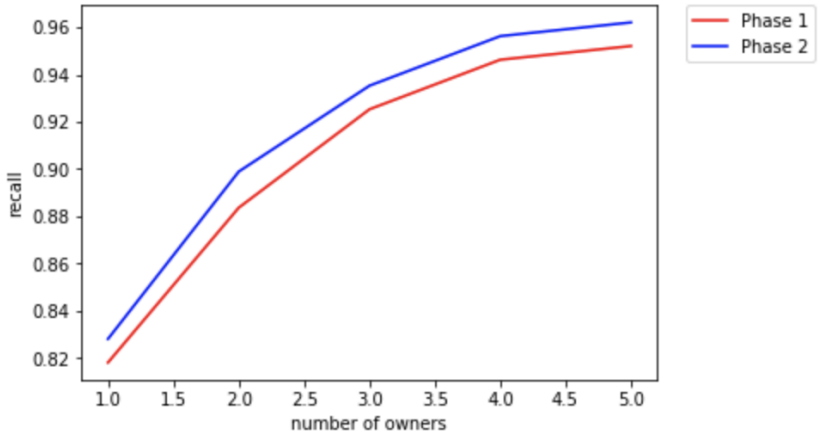


Figure 6.2: Recall comparison between the two phases with k owners

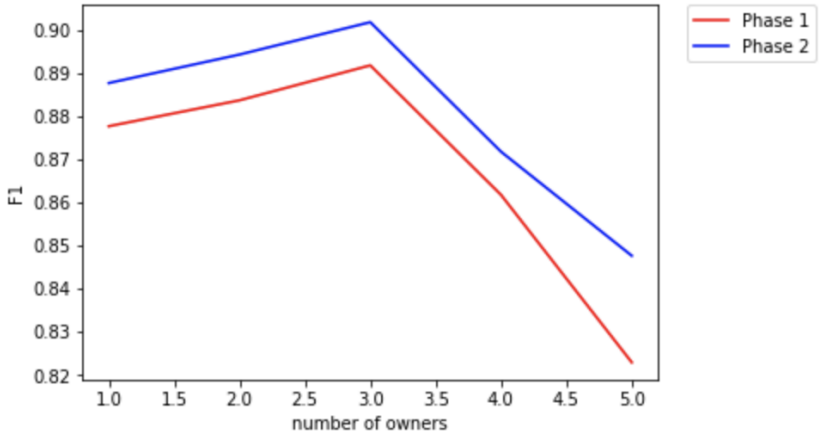


Figure 6.3: F1 comparison between the two phases with k owners

In Figure 6.4 we present the results of the comparison of our solution with the other ones listed before. From the chart it is possible to notice that the lack of presence of well defined domain structure in the dataset strongly penalizes ADAM and DART's approaches which performances overall are similar. The big difference in terms of precision and recall for these algorithms is motivated by the fact that ADAM didn't provide any value for a portion of the books in the dataset differently than DART, so it has a higher precision and a lower recall (ADAM by construction aims at having an high precision [2]). In terms of computation time the fastest algorithm was DART which was faster than our solution by almost 10 seconds since it doesn't take into consideration the computation of the copying probabilities. Finally it is possible to conclude that our approach provides better overall performances than the others in this case of a low copying, multi-truth, domain free datasets.

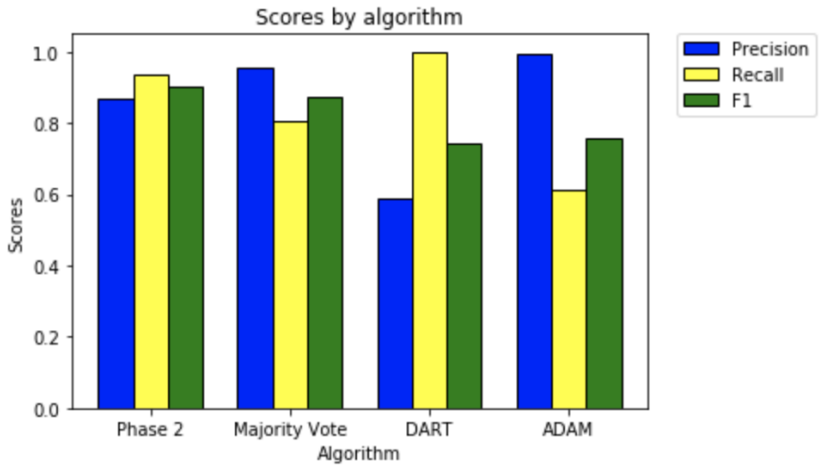


Figure 6.4: Performance comparison between the algorithms on books' dataset

Single-truth : Regarding the single-truth flights dataset our solution, in both phases, provides all and only the true values for each object regardless of the numbers of owners considered. In fact it is important to highlight the fact that in this dataset the real world owners were included as sources and they provided a full coverage of their provided flights. This aspects also emerges by the fact that we found a small number of high ownership score sources and a big number of low ownership score sources. Furthermore the golden truth related to the dataset was composed only by values coming from those real world owner sources. Our solution was able to appropriately identify such real world owner sources giving credits to their values that resulted to be the truth. For this reasons we noted that in this case our precision, recall and F1 scores were equals to 1. Despite this situation our algorithm in the second phase still successfully penalized some aggregator sources that were identified as copiers by lowering their score by a maximum of almost 40% while avoiding lowering real world owners' scores as shown by the scores reported in table 6.5 (we refer to the ownership score towards one owner since it is proportional for the others).

Source	Phase 1 score	Phase 2 score	% change
den	0.478	0.293	38.7
mco	0.486	0.311	36.0
United Airlines	4.518	4.518	0.0

Table 6.5: Table representing the change of ownership value between the two phases

So also in this case our solution provided the best performances overall providing a solution that was the best fit for the objects in the dataset. From the computational time point of view the most rapid approach the fastest approaches were the ones that didn't rely on the computation of copying probabilities such as the Majority Vote and DART followed by our solution.

Chapter 7

Conclusions and future works

> *"Whenever a theory appears to you as the only possible one, take this as a sign that you have neither understood the theory nor the problem which it was intended to solve."*
>
> Karl Popper

In this thesis we presented our work about the studying and developing of a new, non-iterative, multi-truth, ownership-based truth discovery approach.
In particular in Chapter 2 we described the concept of Data Integration by defining the challenges that it has to face, its two main different approaches (physical and virtual) and all the different steps that are required to overcome the heterogeneity of sources. Then we introduced the concept of Big Data, listing its main characteristics and how they affect Data Integration, we presented the problem of copy detection in Data Fusion and lastly we formally defined the Bayesian inference process.
In Chapter 3 we gave an overview of the different types of state-of-the-art strategies used in the Data Fusion in order to perform truth discovery using a categorization based on their characteristics and the different insights that led to their implementations.
In Chapter 4, starting from some observations about the real world datasets, we provided a description of the underlying strategy of our solution including a brief description of its contributions and the adaptation put in place in order to achieve the best results.
In Chapter 5 we formally defined the two phases of our solution showing through an example the different aspects taken into consideration during the computation and their motivations.

In Chapter 6 we provided an experimental analysis of the features of our algorithm comparing both its two phases and its final version with DART, ADAM and Majority Vote on two real world datasets.

7.1 Discussion

From the experiments performed in Chapter 6 it is possible to affirm that our solution's ownership based approach is valid and the focusing on this aspect of information is worth further exploration. This aspect in enhanced when using datasets where the real world owners are present with a good coverage since, as shown for the flights' dataset, we are able to identify them and provide a truthful solution regarding their data objects.

7.2 Future works

In this section we introduce some of the possible future works that would probably increase the algorithm's performances and enlarge its scope making it more complete:

- **Testing :** the tests that we performed on the two dataset revealed that Phase 2's performances are quite promising but didn't show a very big difference from Phase 1 because of the lack of copious significant copying relationship in both datasets. For this reason we propose to perform another testing phase using datasets that are composed by sources which apply a more intense copying in order to see if the performances are heavily affected by bigger changes of the source ownership scores.

- **Result aggregation :** in the case of the books' dataset, when considering multiple (k) sources as owners, we proposed the union of their value proposals in order to provide the results. Next versions of the algorithms might explore new result aggregation techniques for the case of multi-truth discovery that may higher its precision performances and that may result more adaptable to domains of interest other than the authors of a book.

- **Data cleaning :** before applying our algorithm we had to perform a data cleaning for both datasets in order to standardize their val-

ues. Further extension of our algorithm might result more robust to uncleaned data in order to avoid such practices.

- **Copy detection :** our copy detection method (OACD) was based on the previous state-of-the-art algorithms and adapted to suit our solution. Furthermore from the computational time point of view it was the bottleneck taking up to more than 50% of the time. Next version of algorithm might explore new ways to apply copy detection that are closer to our solution and that take less time in computing the probabilities.

Bibliography

[1] Bahadir Ismail Aydin, Yavuz Selim Yilmaz, Yaliang Li, Qi Li, Jing Gao, and Murat Demirbas. Crowdsourcing for multiple-choice question answering. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 2946–2953. AAAI Press, 2014.

[2] Fabio Azzalini, Davide Piantella, and Letizia Tanca. Data fusion with source authority and multiple truth. In Massimo Mecella, Giuseppe Amato, and Claudio Gennaro, editors, *Proceedings of the 27th Italian Symposium on Advanced Database Systems, Castiglione della Pescaia (Grosseto), Italy, June 16-19, 2019*, volume 2400 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[3] Srividya K. Bansal. Towards a semantic extract-transform-load (ETL) framework for big data integration. In *2014 IEEE International Congress on Big Data, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 522–529. IEEE Computer Society, 2014.

[4] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *MONET*, 19(2):171–209, 2014.

[5] Luna Dong. Data sets for data fusion experiments, URL http://lunadong.com/fusionDataSets.htm, Accessed 01/04/2020.

[6] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Integrating conflicting data: The role of source dependence. *Proc. VLDB Endow.*, 2(1):550–561, August 2009.

[7] Xin Luna Dong, Barna Saha, and Divesh Srivastava. Less is more: Selecting sources wisely for integration. *Proc. VLDB Endow.*, 6(2):37–48, December 2012.

[8] Xin Luna Dong and Divesh Srivastava. *Big Data Integration*, volume 7. Morgan & Claypool publishers, 2015.

[9] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. Corroborating information from disagreeing views. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 131–140, New York, NY, USA, 2010. ACM.

[10] John Gantz and David Reinsel. Extracting value from chaos. *IDC iview*, 1142(2011):1–12, 2011.

[11] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98, pages 668–677, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[12] Maurizio Lenzerini. Data integration: A theoretical perspective. In Lucian Popa, Serge Abiteboul, and Phokion G. Kolaitis, editors, *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, pages 233–246. ACM, 2002.

[13] Furong Li, Mong-Li Lee, and Wynne Hsu. Entity profiling with varying source reliabilities. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 1146–1155. ACM, 2014.

[14] Hongwei Li, Bo Zhao, and Ariel Fuxman. The wisdom of minority: discovering and targeting the right group of workers for crowdsourcing. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 165–176. ACM, 2014.

[15] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava. Truth finding on the deep web: Is the problem solved? *CoRR*, abs/1503.00303, 2015.

[16] Xian Li, Xin Luna Dong, Kenneth B. Lyons, Weiyi Meng, and Divesh Srivastava. Scaling up copy detection. In Johannes Gehrke, Wolfgang Lehner, Kyuseok Shim, Sang Kyun Cha, and Guy M. Lohman, editors, *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, pages 89–100. IEEE Computer Society, 2015.

[17] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *SIGKDD Explorations*, 17(2):1–16, 2015.

[18] Xueling Lin and Lei Chen. Domain-aware multi-truth discovery from conflicting sources. *PVLDB*, 11(5):635–647, 2018.

[19] Xueling Lin and Lei Chen. Domain-aware multi-truth discovery from conflicting sources. *Proc. VLDB Endow.*, 11(5):635–647, January 2018.

[20] Xuan Liu, Xin Luna Dong, Beng Chin Ooi, and Divesh Srivastava. Online data fusion. *PVLDB*, 4(11):932–943, 2011.

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[22] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[23] Subhabrata Mukherjee, Gerhard Weikum, and Cristian Danescu-Niculescu-Mizil. People on drugs: Credibility of user statements in health communities. *CoRR*, abs/1705.02522, 2017.

[24] Jeff Pasternack and Dan Roth. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 877–885, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[25] Kimball Ralph and Caserta Joe. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data*. John Wiley & Sons, Inc. USA, 2004.

[26] Xianzhi Wang, Quan Z. Sheng, Xiu Susie Fang, Lina Yao, Xiaofei Xu, and Xue Li. An integrated bayesian approach for effective multi-truth discovery. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 493–502, New York, NY, USA, 2015. ACM.

[27] Xiaoxin Yin, Jiawei Han, and Philip S. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE Trans. Knowl. Data Eng.*, 20(6):796–808.

[28] Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, Shi Zhi, Jiawei Han, Clare R. Voss, and Malik Magdon-Ismail. The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In Jan Hajic and Junichi Tsujii, editors, *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1567–1578. ACL, 2014.

Appendix A

Implementation appendix

In this appendix we present some significant code snippets used for the computation of the ownership score in order to give a full description of the computation process. The programming language we have chosen to use is Python 3.7.

A.1 First phase ownership score

A.1.1 Sources Map

Here we first introduce our function used in both scores computation in order to count the tuples provided by each source.

```
1  def compute_sources_map(data, source_attr, own_attr):
2      """
3      Computes a map containing for each source's and owner's
4          value the number of tuples provided
5          by the source fot the owner
6      :param data: the dataset
7      :param source_attr: the attribute of sources_id
8      :param owner_attr: the attribute of the owner_id
9      """
10     sources = data[source_attr].unique()
11     sources_map = {s:{} for s in sources}
12     owner_map = {}
13     for index, row in data.iterrows():
```

```
        owner_count_map = sources_map[row[source_attr]]
        row_owner = row[own_attr]
        if row_owner in owner_map:
            owner_count_map[row_owner] += 1
        else:
            owner_count_map[row_owner] = 1
    return sources_map
```

A.1.2 First score

Here we present the implementation of the Equation 5.1

```
def ownership_score(data, source_attr, owner_attr):
    """
    Computes a map containing for each source's and owner's
        value the first phase ownershp score
    :param data: the dataset
    :param source_attr: the attribute of sources_id
    :param owner_attr: the attribute of the owner_id
    """
    sources_map = compute_sources_map(data, source_attr,
                                      owner_attr)
    source_ownership_map = {s:{} for s in sources_map}
    sources_map = compute_sources_map(data, source_attr,
                                      owner_attr)
    owners = data[owner_attr].unique()
    #map containing number of tuples for each owner's value
    owner_tuples_map = {c:{} for c in owners}
    for c in owners:
        owner_tuples_map[c] = 0
        for s in sources_map:
            owner_tuples_map[c] += sources_map[s][c]
    #first phase ownership score computation
    for c in owner_tuples_map:
        for s in sources_map:
```

```
            #number of tuples of other owners
            total_others = 0
            #number of tuples of other owners by s
            other_by_s = 0
            for c1 in owner_tuples_map:
                if c1 != c:
                    other_by_s += sources_map[s][c1]
                    total_others += owner_tuples_map[c1]

            ownership_score = 100*( (sources_map[s][c])
                                    /(owner_tuples_map[c]) )
                                  *(1-(other_by_s/total_others))
            source_ownership_map[s][c] = ownership_score
    return source_ownership_map
```

A.2 Second phase ownership score

Here we present the implementation of the Equation 5.18. We introduced the copying probabilities that come from our OACD algorithm inside a map that's given as input to the function.

```
from sklearn import preprocessing

def ownership_score_2(data, source_attr, owner_attr, id_attr,
                      p_copy_map, source_ownership_map):
    """
    Computes a map containing for each source's and owner's
        value the first phase ownershp score
    :param data: the dataset
    :param source_attr: the attribute of sources_id
    :param owner_attr: the attribute of owner_id
    :param id_attr: the attribute of object_id
    :param p_copy_map: map containing copying
                       probabilities for each
```

```
                        pair of sources
    :param source_ownership_map: map of first
                                 ownership scores
    """
    sources_map = compute_sources_map(data, source_attr,
                                      owner_attr)
    source_ownership_map_new = {s:{} for s in sources_map}
    #create a vector with the mean of the probabilities
    #that each source is copying another one
    p_copy_means = []
    for s in sources_map:
        n_sources = 0
        total_p_copy_source = 0
        for s1 in p_copy_map[s]:
            #only consider copying relationship where
            #p_copy >= 0.5
            if p_copy_map[s][s1] >= 0.5:
                n_sources += 1
                total_p_copy_source += p_copy_map[s][s1]
        if n_sources != 0:
            mean_p_copy = total_p_copy_source / n_sources
        else:
            mean_p_copy = 0
        p_copy_means.append(mean_p_copy)
    #normalization of the means in a [0,1] range
    #(because of the new definition of source expertise)
    p_copy_means = preprocessing.minmax_scale(p_copy_means,
                                              f_range=(0,1),
                                              axis=0,
                                              copy=True)
    #phase 2 ownership score computation
    index = -1
    for s in sources_map:
        index += 1
        mean_p_copy = p_copy_means[index]
        for c in source_ownership_map[s]:
```

```
            #phase 1 ownership score
            own_score = source_ownership_map[s][c]
            #we flatten to 0 very low ownership scores
            if own_score >= 1:
                new_score = own_score ** (1 - mean_p_copy)
            else:
                new_score = 0
            source_ownership_map_new[s][c] = new_score
    return source_ownership_map_new
```