Morgan & Claypool Publishers

Big Data Integration

Xin Luna Dong
Divesh Srivastava

Synthesis Lectures on Data Management

Z. Meral Özsoyoğlu, *Series Editor*

Big Data Integration

Synthesis Lectures on Data Management

Editor
Z. Meral Özsoyoğlu, *Case Western Reserve University*

Founding Editor
M. Tamer Özsu, *University of Waterloo*

Synthesis Lectures on Data Management is edited by Meral Özsoyoğlu of Case Western Reserve University. The series publishes 80- to 150-page publications on topics pertaining to data management. Topics include query languages, database system architectures, transaction management, data warehousing, XML and databases, data stream systems, wide-scale data distribution, multimedia data management, data mining, and related subjects.

Big Data Integration
Xin Luna Dong, Divesh Srivastava
March 2015

Instant Recovery with Write-Ahead Logging: Page Repair, System Restart, and Media Restore
Goetz Graefe, Wey Guy, Caetano Sauer
December 2014

Similarity Joins in Relational Database Systems
Nikolaus Augsten, Michael H. Böhlen
November 2013

Information and Influence Propagation in Social Networks
Wei Chen, Laks V. S. Lakshmanan, Carlos Castillo
October 2013

Data Cleaning: A Practical Perspective
Venkatesh Ganti, Anish Das Sarma
September 2013

Data Processing on FPGAs
Jens Teubner, Louis Woods
June 2013

Perspectives on Business Intelligence
Raymond T. Ng, Patricia C. Arocena, Denilson Barbosa, Giuseppe Carenini, Luiz Gomes, Jr., Stephan Jou, Rock Anthony Leung, Evangelos Milios, Renée J. Miller, John Mylopoulos, Rachel A. Pottinger, Frank Tompa, Eric Yu
April 2013

Semantics Empowered Web 3.0: Managing Enterprise, Social, Sensor, and Cloud-Based Data and Services for Advanced Applications
Amit Sheth, Krishnaprasad Thirunarayan
December 2012

Data Management in the Cloud: Challenges and Opportunities
Divyakant Agrawal, Sudipto Das, Amr El Abbadi
December 2012

Query Processing over Uncertain Databases
Lei Chen, Xiang Lian
December 2012

Foundations of Data Quality Management
Wenfei Fan, Floris Geerts
July 2012

Incomplete Data and Data Dependencies in Relational Databases
Sergio Greco, Cristian Molinaro, Francesca Spezzano
July 2012

Business Processes: A Database Perspective
Daniel Deutch, Tova Milo
July 2012

Data Protection from Insider Threats
Elisa Bertino
June 2012

Deep Web Query Interface Understanding and Integration
Eduard C. Dragut, Weiyi Meng, Clement T. Yu
June 2012

P2P Techniques for Decentralized Applications
Esther Pacitti, Reza Akbarinia, Manal El-Dick
April 2012

Query Answer Authentication
HweeHwa Pang, Kian-Lee Tan
February 2012

Declarative Networking
Boon Thau Loo, Wenchao Zhou
January 2012

Full-Text (Substring) Indexes in External Memory
Marina Barsky, Ulrike Stege, Alex Thomo
December 2011

Spatial Data Management
Nikos Mamoulis
November 2011

Database Repairing and Consistent Query Answering
Leopoldo Bertossi
August 2011

Managing Event Information: Modeling, Retrieval, and Applications
Amarnath Gupta, Ramesh Jain
July 2011

Fundamentals of Physical Design and Query Compilation
David Toman, Grant Weddell
July 2011

Methods for Mining and Summarizing Text Conversations
Giuseppe Carenini, Gabriel Murray, Raymond Ng
June 2011

Probabilistic Databases
Dan Suciu, Dan Olteanu, Christopher Ré, Christoph Koch
May 2011

Peer-to-Peer Data Management
Karl Aberer
May 2011

Probabilistic Ranking Techniques in Relational Databases
Ihab F. Ilyas, Mohamed A. Soliman
March 2011

Uncertain Schema Matching
Avigdor Gal
March 2011

Fundamentals of Object Databases: Object-Oriented and Object-Relational Design
Suzanne W. Dietrich, Susan D. Urban
2010

Advanced Metasearch Engine Technology
Weiyi Meng, Clement T. Yu
2010

Web Page Recommendation Models: Theory and Algorithms
Şule Gündüz-Ögüdücü
2010

Multidimensional Databases and Data Warehousing
Christian S. Jensen, Torben Bach Pedersen, Christian Thomsen
2010

Database Replication
Bettina Kemme, Ricardo Jimenez-Peris, Marta Patino-Martinez
2010

Relational and XML Data Exchange
Marcelo Arenas, Pablo Barcelo, Leonid Libkin, Filip Murlak
2010

User-Centered Data Management
Tiziana Catarci, Alan Dix, Stephen Kimani, Giuseppe Santucci
2010

Data Stream Management
Lukasz Golab, M. Tamer Özsu
2010

Access Control in Data Management Systems
Elena Ferrari
2010

An Introduction to Duplicate Detection
Felix Naumann, Melanie Herschel
2010

Privacy-Preserving Data Publishing: An Overview
Raymond Chi-Wing Wong, Ada Wai-Chee Fu
2010

Keyword Search in Databases
Jeffrey Xu Yu, Lu Qin, Lijun Chang
2009

Copyright © 2015 by Morgan & Claypool Publishers

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews—without the prior permission of the publisher.

Big Data Integration

Xin Luna Dong, Divesh Srivastava

www.morganclaypool.com

ISBN: 978-1-62705-223-8 paperback
ISBN: 978-1-62705-224-5 ebook

DOI: 10.2200/S00578ED1V01Y201404DTM040

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON DATA MANAGEMENT
Series ISSN: 2153-5418 print 2153-5426 ebook

Lecture #40
Series Editor: M. Tamer Özsu, *University of Waterloo*

First Edition
10 9 8 7 6 5 4 3 2 1

Big Data Integration

Xin Luna Dong
Google Inc.

Divesh Srivastava
AT&T Labs-Research

SYNTHESIS LECTURES ON DATA MANAGEMENT #40

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

The big data era is upon us: data are being generated, analyzed, and used at an unprecedented scale, and data-driven decision making is sweeping through all aspects of society. Since the value of data explodes when it can be linked and fused with other data, addressing the *big data integration* (BDI) challenge is critical to realizing the promise of big data.

BDI differs from traditional data integration along the dimensions of *volume*, *velocity*, *variety,* and *veracity*. First, not only can data sources contain a huge volume of data, but also the number of data sources is now in the millions. Second, because of the rate at which newly collected data are made available, many of the data sources are very dynamic, and the number of data sources is also rapidly exploding. Third, data sources are extremely heterogeneous in their structure and content, exhibiting considerable variety even for substantially similar entities. Fourth, the data sources are of widely differing qualities, with significant differences in the coverage, accuracy and timeliness of data provided.

This book explores the progress that has been made by the data integration community on the topics of schema alignment, record linkage and data fusion in addressing these novel challenges faced by big data integration. Each of these topics is covered in a systematic way: first starting with a quick tour of the topic in the context of traditional data integration, followed by a detailed, example-driven exposition of recent innovative techniques that have been proposed to address the BDI challenges of volume, velocity, variety, and veracity. Finally, it presents emerging topics and opportunities that are specific to BDI, identifying promising directions for the data integration community.

KEYWORDS

big data integration, data fusion, record linkage, schema alignment, variety, velocity, veracity, volume

To Jianzhong Dong, Xiaoqin Gong, Jun Zhang, Franklin Zhang,
and Sonya Zhang

To Swayam Prakash Srivastava, Maya Srivastava,
and Jaya Mathangi Satagopan

Contents

List of Figures . . . xv
List of Tables . . . xvii
Preface . . . xix
  Acknowledgments . . . xix

1. Motivation: Challenges and Opportunities for BDI . . . 1
1.1 Traditional Data Integration . . . 2
  1.1.1 The Flights Example: Data Sources . . . 2
  1.1.2 The Flights Example: Data Integration . . . 6
  1.1.3 Data Integration: Architecture & Three Major Steps . . . 9
1.2 BDI: Challenges . . . 11
  1.2.1 The “V” Dimensions . . . 11
  1.2.2 Case Study: Quantity of Deep Web Data . . . 13
  1.2.3 Case Study: Extracted Domain-Specific Data . . . 15
  1.2.4 Case Study: Quality of Deep Web Data . . . 20
  1.2.5 Case Study: Surface Web Structured Data . . . 23
  1.2.6 Case Study: Extracted Knowledge Triples . . . 26
1.3 BDI: Opportunities . . . 27
  1.3.1 Data Redundancy . . . 27
  1.3.2 Long Data . . . 28
  1.3.3 Big Data Platforms . . . 29
1.4 Outline of Book . . . 29

2. Schema Alignment . . . 31
2.1 Traditional Schema Alignment: A Quick Tour . . . 32
  2.1.1 Mediated Schema . . . 32
  2.1.2 Attribute Matching . . . 32
  2.1.3 Schema Mapping . . . 33
  2.1.4 Query Answering . . . 34
2.2 Addressing the Variety and Velocity Challenges . . . 35
  2.2.1 Probabilistic Schema Alignment . . . 36
  2.2.2 Pay-As-You-Go User Feedback . . . 47

- 2.3 Addressing the Variety and Volume Challenges . . . 49
  - 2.3.1 Integrating Deep Web Data . . . 49
  - 2.3.2 Integrating Web Tables . . . 54

3. Record Linkage . . . 63

- 3.1 Traditional Record Linkage: A Quick Tour . . . 64
  - 3.1.1 Pairwise Matching . . . 65
  - 3.1.2 Clustering . . . 67
  - 3.1.3 Blocking . . . 68
- 3.2 Addressing the Volume Challenge . . . 71
  - 3.2.1 Using MapReduce to Parallelize Blocking . . . 71
  - 3.2.2 Meta-blocking: Pruning Pairwise Matchings . . . 77
- 3.3 Addressing the Velocity Challenge . . . 82
  - 3.3.1 Incremental Record Linkage . . . 82
- 3.4 Addressing the Variety Challenge . . . 88
  - 3.4.1 Linking Text Snippets to Structured Data . . . 89
- 3.5 Addressing the Veracity Challenge . . . 94
  - 3.5.1 Temporal Record Linkage . . . 94
  - 3.5.2 Record Linkage with Uniqueness Constraints . . . 100

4. BDI: Data Fusion . . . 107

- 4.1 Traditional Data Fusion: A Quick Tour . . . 108
- 4.2 Addressing the Veracity Challenge . . . 109
  - 4.2.1 Accuracy of a Source . . . 111
  - 4.2.2 Probability of a Value Being True . . . 111
  - 4.2.3 Copying Between Sources . . . 114
  - 4.2.4 The End-to-End Solution . . . 120
  - 4.2.5 Extensions and Alternatives . . . 123
- 4.3 Addressing the Volume Challenge . . . 126
  - 4.3.1 A MapReduce-Based Framework for Offline Fusion . . . 126
  - 4.3.2 Online Data Fusion . . . 127
- 4.4 Addressing the Velocity Challenge . . . 133
- 4.5 Addressing the Variety Challenge . . . 136

5. BDI: Emerging Topics . . . 139

- 5.1 Role of Crowdsourcing . . . 139
  - 5.1.1 Leveraging Transitive Relations . . . 140
  - 5.1.2 Crowdsourcing the End-to-End Workflow . . . 144

        - 5.1.3 Future Work 146
    - 5.2 Source Selection 146
        - 5.2.1 Static Sources 148
        - 5.2.2 Dynamic Sources 150
        - 5.2.3 Future Work 153
    - 5.3 Source Profiling 153
        - 5.3.1 The Bellman System 155
        - 5.3.2 Summarizing Sources 157
        - 5.3.3 Future Work 160

6. Conclusions 163

Bibliography 165

Authors' Biographies 175

Index 177

List of Figures

1.1 Traditional data integration: architecture. 9
1.2 K-coverage (the fraction of entities in the database that are present in at least *k* different sources) for phone numbers in the restaurant domain [Dalvi et al. 2012]. 18
1.3 Connectivity (between entities and sources) for the nine domains studied by Dalvi et al. [2012]. 19
1.4 Consistency of data items in the Stock and Flight domains [Li et al. 2012]. 22
1.5 High-quality table on the web. 23
1.6 Contributions and overlaps between different types of web contents [Dong et al. 2014b]. 27

2.1 Traditional schema alignment: three steps. 32
2.2 Attribute matching from Airline1.Flight to Mediate.Flight. 33
2.3 Query answering in a traditional data-integration system. 34
2.4 Example web form for searching flights at Orbitz.com (accessed on April 1, 2014). . . 50
2.5 Example web table (Airlines) with some major airlines of the world (accessed on April 1, 2014). 54
2.6 Two web tables (CapitalCity) describing major cities in Asia and in Africa from nationsonline.org (accessed on April 1, 2014). 58
2.7 Graphical model for annotating a 3x3 web table [Limaye et al. 2010]. 61

3.1 Traditional record linkage: three steps. 65
3.2 Pairwise matching graph. 67
3.3 Use of a single blocking function. 69
3.4 Use of multiple blocking functions. 70
3.5 Using MapReduce: a basic approach. 72
3.6 Using MapReduce: BLOCKSPLIT. 74
3.7 Using schema agnostic blocking on multiple values. 79
3.8 Using meta-blocking with schema agnostic blocking. 81
3.9 Record linkage results on $\overline{\text{Flights}}_0$. 84
3.10 Record linkage results on $\overline{\text{Flights}}_0 + \Delta\overline{\text{Flights}}_1$. 85

3.11 Record linkage results on $\overline{\mathsf{Flights}}_0 + \Delta\overline{\mathsf{Flights}}_1 + \Delta\overline{\mathsf{Flights}}_2$. 85

3.12 Tagging of text snippet. 91

3.13 Plausible parses of text snippet. 92

3.14 Ground truth due to entity evolution. 95

3.15 Linkage with high value consistency. 96

3.16 Linkage with only name similarity. 97

3.17 K-partite graph encoding. 103

3.18 Linkage with hard constraints. 104

3.19 Linkage with soft constraints. 104

4.1 Architecture of data fusion [Dong et al. 2009a]. 110

4.2 Probabilities of copying computed by AccuCopy on the motivating example [Dong et al. 2009a]. An arrow from source S to S' indicates that S copies from S'. Copyings are shown only when the sum of the probabilities in both directions is over 0.1. 121

4.3 MapReduce-based implementation for truth discovery and trustworthiness evaluation [Dong et al. 2014b]. 126

4.4 Nine sources that provide the estimated arrival time for *Flight 49*. For each source, the answer it provides is shown in parenthesis and its accuracy is shown in a circle. An arrow from S to S' means that S copies some data from S'. 128

4.5 Architecture of online data fusion [Liu et al. 2011]. 129

4.6 Input for data fusion is two-dimensional, whereas input for extended data fusion is three-dimensional [Dong et al. 2014b]. 136

4.7 Fixing #provenances, (data item, value) pairs from more extractors are more likely to be true [Dong et al. 2014b]. 138

5.1 Example to illustrate labeling by crowd for transitive relations [Wang et al. 2013]. . . . 141

5.2 Fusion result recall for the Stock domain [Li et al. 2012]. 147

5.3 Freshness versus update frequency for business listing sources [Rekatsinas et al. 2014]. . 151

5.4 Evolution of coverage of the integration result for two subsets of the business listing sources [Rekatsinas et al. 2014]. 152

5.5 TPCE schema graph [Yang et al. 2009]. 158

List of Tables

1.1 Sample data for Airline1.Schedule . . . 3
1.2 Sample data for Airline1.Flight . . . 3
1.3 Sample data for Airline2.Flight . . . 4
1.4 Sample data for Airport3.Departures . . . 4
1.5 Sample data for Airport3.Arrivals . . . 5
1.6 Sample data for Airfare4.Flight . . . 6
1.7 Sample data for Airfare4.Fares . . . 6
1.8 Sample data for Airinfo5.AirportCodes, Airinfo5.AirlineCodes . . . 6
1.9 Abbreviated attribute names . . . 7
1.10 Domain category distribution of web databases [He et al. 2007] . . . 16
1.11 Row statistics on high-quality relational tables on the web [Cafarella et al. 2008b] . . . 25

2.1 Selected text-derived features used in search rankers. The most important features are in italic [Cafarella et al. 2008a] . . . 56

3.1 Sample Flights records . . . 65
3.2 Virtual global enumeration in PAIRRANGE . . . 76
3.3 Sample Flights records with schematic heterogeneity . . . 78
3.4 Flights records and updates . . . 83
3.5 Sample Flights records from Table 3.1 . . . 89
3.6 Traveller flight profiles . . . 95
3.7 Airline business listings . . . 101

4.1 Five data sources provide information on the scheduled departure time of five flights. False values are in italics. Only S1 provides all true values. . . . 109
4.2 Accuracy of data sources computed by ACCUCOPY on the motivating example . . . 122
4.3 Vote count computed for the scheduled departure time for *Flight 4* and *Flight 5* in the motivating example . . . 122
4.4 Output at each time point in Example 4.8. The time is made up for illustration purposes 128

4.5 Three data sources updating information on the scheduled departure time of five flights. False values are in italic. 133
4.6 CEF-measures for the data sources in Table 4.5 135

Preface

Big data integration is the confluence of two significant bodies of work: one quite old—data integration—and the other relatively new—big data.

As long as there have been data sets that people have sought to link and fuse to enhance value, data integration has been around. Even before computer scientists started investigating this area, statisticians had already made much progress, given their pressing need to correlate and analyze census data sets collected over time. Data integration is challenging for many reasons, not the least being our ability to represent and misrepresent information about real-world entities in very diverse ways. To effectively address these challenges, considerable progress has been made over the last few decades by the data integration community on the foundational topics of schema alignment, record linkage, and data fusion, especially for well-structured data.

Recent years have seen a dramatic growth in our ability to capture each event and every interaction in the world as digital data. Concomitant with this ability has been our desire to analyze and extract value from this data, ushering in the era of big data. This era has seen an enormous increase in the amount and heterogeneity of data, as well as in the number of data sources, many of which are very dynamic, while being of widely differing qualities. Since the value of data explodes when it can be linked and fused with other data, data integration is critical to realizing the promise of big data of enabling valuable, data-driven decisions to alter all aspects of society.

Data integration for big data is what has come to be known as big data integration. This book explores the progress that has been made by the data integration community in addressing the novel challenges faced by big data integration. It is intended as a starting point for researchers, practitioners and students who would like to learn more about big data integration. We have attempted to cover a diversity of topics and research efforts in this area, fully well realizing that it is impossible to be comprehensive in such a dynamic area. We hope that many of our readers will be inspired by this book to make their own contributions to this important area, to help further the promise of big data.

ACKNOWLEDGMENTS

Several people provided valuable support during the preparation of this book. We warmly thank Tamer Özsu for inviting us to write this book, Diane Cerra for managing the entire publication process, and Paul Anagnostopoulos for producing the book. Without their gentle reminders, periodic nudging, and prompt copyediting, this book may have taken much longer to complete.

Much of this book's material evolved from the tutorials and talks that we presented at ICDE 2013, VLDB 2013, COMAD 2013, University of Zurich (Switzerland), the Ph.D. School of ADC 2014 and BDA 2014. We thank our many colleagues for their constructive feedback during and subsequent to these presentations.

We would also like to acknowledge our many collaborators who have influenced our thoughts and our understanding of this research area over the years.

Finally, we would like to thank our family members, whose constant encouragement and loving support made it all worthwhile.

Xin Luna Dong and Divesh Srivastava
December 2014

CHAPTER 1

Motivation: Challenges and Opportunities for BDI

The big data era is the inevitable consequence of *datafication*: our ability to transform each event and every interaction in the world into digital data, and our concomitant desire to analyze and extract value from this data. Big data comes with a lot of promise, enabling us to make valuable, data-driven decisions to alter all aspects of society.

Big data is being generated and used today in a variety of domains, including data-driven science, telecommunications, social media, large-scale e-commerce, medical records and e-health, and so on. Since the value of data explodes when it can be linked and fused with other data, addressing the *big data integration* (BDI) challenge is critical to realizing the promise of big data in these and other domains.

As one prominent example, recent efforts in mining the web and extracting entities, relationships, and ontologies to build general purpose knowledge bases such as Freebase [Bollacker et al. 2008], the Google knowledge graph [Dong et al. 2014a], ProBase [Wu et al. 2012], and Yago [Weikum and Theobald 2010] show promise of using integrated big data to improve applications such as web search and web-scale data analysis.

As a second important example, the flood of geo-referenced data available in recent years, such as geo-tagged web objects (e.g., photos, videos, tweets), online check-ins (e.g., Foursquare), WiFi logs, GPS traces of vehicles (e.g., taxi cabs), and roadside sensor networks has given momentum for using such integrated big data to characterize large-scale human mobility [Becker et al. 2013], and influence areas like public health, traffic engineering, and urban planning.

In this chapter, we first describe the problem of data integration and the components of traditional data integration in Section 1.1. We then discuss the specific challenges that arise in BDI in Section 1.2, where we first identify the dimensions along which BDI differs from traditional data integration, then present a number of recent case studies that empirically study the nature of data sources in BDI. BDI also offers opportunities that do not exist in traditional data integration, and we highlight some of these opportunities in Section 1.3. Finally, we present an outline of the rest of the book in Section 1.4.

1.1 TRADITIONAL DATA INTEGRATION

Data integration has the goal of providing *unified access to data* residing in multiple, autonomous data sources. While this goal is easy to state, achieving this goal has proven notoriously hard, even for a small number of sources that provide structured data—the scenario of traditional data integration [Doan et al. 2012].

To understand some of the challenging issues in data integration, consider an illustrative example from the Flights domain, for the common tasks of tracking flight departures and arrivals, examining flight schedules, and booking flights.

1.1.1 THE FLIGHTS EXAMPLE: DATA SOURCES

We have a few different kinds of sources, including two airline sources Airline1 and Airline2 (e.g., United Airlines, American Airlines, Delta, etc.), each providing flight data about a different airline, an airport source Airport3, providing information about flights departing from and arriving at a particular airport (e.g., EWR, SFO), a comparison shopping travel source Airfare4 (e.g., Kayak, Orbitz, etc.), providing fares in different fare classes to compare alternate flights, and an informational source Airinfo5 (e.g., a Wikipedia table), providing data about airports and airlines.

Sample data for the various source tables is shown in Tables 1.1–1.8, using short attribute names for brevity. The mapping between the short and full attribute names is provided in Table 1.9 for ease of understanding. Records in different tables that are highlighted using the same color are related to each other, and the various tables should be understood as follows.

Source Airline1

Source Airline1 provides the tables Airline1.Schedule(Flight Id, Flight Number, Start Date, End Date, Departure Time, Departure Airport, Arrival Time, Arrival Airport) and Airline1.Flight(Flight Id, Departure Date, Departure Time, Departure Gate, Arrival Date, Arrival Time, Arrival Gate, Plane Id). The underlined attributes form a key for the corresponding table, and Flight Id is used as a join key between these two tables.

Table Airline1.Schedule shows flight schedules in Table 1.1. For example, record r_{11} in table Airline1.Schedule states that Airline1's flight *49* is scheduled to fly regularly from *EWR* to *SFO*, departing at *18:05*, and arriving at *21:10*, between *2013-10-01* and *2014-03-31*. Record r_{12} in the same table shows that the same flight *49* has different scheduled departure and arrival times between *2014-04-01* and *2014-09-30*. Records r_{13} and r_{14} in the same table show the schedules for two different segments of the same flight *55*, the first from *ORD* to *BOS*, and the second from *BOS* to *EWR*, between *2013-10-01* and *2014-09-30*.

Table Airline1.Flight shows the actual departure and arrival information in Table 1.2, for the flights whose schedules are shown in Airline1.Schedule. For example, record r_{21} in table Airline1.Flight

TABLE 1.1: Sample data for Airline1.Schedule

	FI	FN	SD	ED	DT	DA	AT	AA
r_{11}	123	49	2013-10-01	2014-03-31	18:05	EWR	21:10	SFO
r_{12}	234	49	2014-04-01	2014-09-30	18:20	EWR	21:25	SFO
r_{13}	345	55	2013-10-01	2014-09-30	18:30	ORD	21:30	BOS
r_{14}	346	55	2013-10-01	2014-09-30	22:30	BOS	23:30	EWR

TABLE 1.2: Sample data for Airline1.Flight

	FI	DD	DT	DG	AD	AT	AG	PI
r_{21}	123	2013-12-21	18:45	C98	2013-12-21	21:30	81	4013
r_{22}	123	2013-12-28	21:30	C101	2013-12-29	00:30	81	3008
r_{23}	345	2013-12-29	18:30	B6	2013-12-29	21:45	C18	4013
r_{24}	346	2013-12-29	22:35	C18	2013-12-29	23:35	C101	4013

records information about a specific flight, corresponding to the regularly scheduled flight r_{11} (the Flight Id *123* specifies the join key), using a plane with id *4013*, actually departing on *2013-12-21* at *18:45* (40 minutes later than the scheduled departure time of *18:05*) from gate *C98*, and actually arriving on *2013-12-21* at *21:30* (20 minutes later than the scheduled arrival time of *21:10*) at gate *81*. Both r_{11} and r_{21} use yellow highlighting to visually depict their relationship. Record r_{22} in the same table records information about a flight on a different date, also corresponding to the regularly scheduled flight r_{11}, with a considerably longer delay in departure and arrival times. Records r_{23} and r_{24} record information about flights on *2013-12-29*, corresponding to regularly scheduled flights r_{13} and r_{14}, respectively.

Source Airline2

Source Airline2 provides similar data to source Airline1, but using the table Airline2.Flight(Flight Number, Departure Airport, Scheduled Departure Date, Scheduled Departure Time, Actual Departure Time, Arrival Airport, Scheduled Arrival Date, Scheduled Arrival Time, Actual Arrival Time).

Each record in table Airline2.Flight, shown in Table 1.3, contains both the schedule and the actual flight details. For example, record r_{31} records information about Airline2's flight *53*, departing from *SFO*, scheduled to depart on *2013-12-21* at *15:30*, with a 30 minute delay in the actual departure time, arriving at *EWR*, scheduled to arrive on *2013-12-21* at *23:35*, with a 40 minute

TABLE 1.3: Sample data for Airline2.Flight

	FN	DA	SDD	SDT	ADT	AA	SAD	SAT	AAT
r_{31}	53	SFO	2013-12-21	15:30	16:00	EWR	2013-12-21	23:35	00:15 (+1d)
r_{32}	53	SFO	2013-12-22	15:30	16:15	EWR	2013-12-22	23:35	00:30
r_{33}	53	SFO	2014-06-28	16:00	16:05	EWR	2014-06-29	00:05	23:57 (-1d)
r_{34}	53	SFO	2014-07-06	16:00	16:00	EWR	2014-07-07	00:05	00:09
r_{35}	49	SFO	2013-12-21	12:00	12:35	EWR	2013-12-21	20:05	20:45
r_{36}	77	LAX	2013-12-22	09:15	09:15	SFO	2013-12-22	11:00	10:59

TABLE 1.4: Sample data for Airport3.Departures

	AL	FN	S	A	GT	TT	T	G	R
r_{41}	A1	49	2013-12-21	2013-12-21	18:45	18:53	C	98	2
r_{42}	A1	49	2013-12-28	2013-12-28	21:29	21:38	C	101	2

delay in the actual arrival time; its arrival on *2013-12-22* (the day after its scheduled arrival) is indicated by the *(+1d)* associated with the actual arrival time. Note that this table contains a record r_{35} for Airline2's flight *49*, which is different from Airline1's flight *49*, illustrating that different airlines can use the same flight number for their respective flights.

Unlike source Airline1, source Airline2 does not publish the departure gate, arrival gate, and the plane identifier used for the specific flight, illustrating the diversity between the schemas used by these sources.

Source Airport3

Source Airport3 provides tables Airport3.Departures(Air Line, Flight Number, Scheduled, Actual, Gate Time, Takeoff Time, Terminal, Gate, Runway) and Airport3.Arrivals(Air Line, Flight Number, Scheduled, Actual, Gate Time, Landing Time, Terminal, Gate, Runway).

Table Airport3.Departures, shown in Table 1.4, publishes information only about flight departures from *EWR*. For example, record r_{41} in table Airport3.Departures states that Airline1's flight *49*, scheduled to depart on *2013-12-21*, departed on *2013-12-21* from terminal *C* and gate *98* at *18:45* and took off at *18:53* from runway *2*. There is no information in this table about the arrival airport, arrival date, and arrival time of this flight. Note that r_{41} corresponds to records r_{11} and r_{21}, depicted by the consistent use of the yellow highlight.

TABLE 1.5: Sample data for Airport3.Arrivals

	AL	FN	S	A	GT	LT	T	G	R
r_{51}	A2	53	2013-12-21	2013-12-22	00:21	00:15	B	53	2
r_{52}	A2	53	2013-12-22	2013-12-23	00:40	00:30	B	53	2
r_{53}	A1	55	2013-12-29	2013-12-29	23:35	23:31	C	101	1
r_{54}	A2	49	2013-12-21	2013-12-21	20:50	20:45	B	55	2

Table Airport3.Arrivals, shown in Table 1.5, publishes information only about flight arrivals into *EWR*. For example, record r_{51} in table Airport3.Arrivals states that Airline2's flight *53*, scheduled to arrive on *2013-12-21*, arrived on *2013-12-22*, landing on runway *2* at *00:15*, reaching gate *53* of terminal *B* at *00:21*. There is no information in this table about the departure airport, departure date, and departure time of this flight. Note that r_{51} corresponds to record r_{31}, both of which are highlighted in lavender.

Unlike sources Airline1 and Airline2, source Airport3 distinguishes between the time at which the flight left/reached the gate and the time at which the flight took off from/landed at the airport runway.

Source Airfare4

Travel source Airfare4 publishes comparison shopping data for multiple airlines, including schedules in Airfare4.Flight(Flight Id, Flight Number, Departure Airport, Departure Date, Departure Time, Arrival Airport, Arrival Time) and fares in Airfare4.Fares(Flight Id, Fare Class, Fare). Flight Id is used as a join key between these two tables.

For example, record r_{61} in Airfare4.Flight, shown in Table 1.6, states that Airline1's flight *A1-49* was scheduled to depart from Newark Liberty airport on *2013-12-21* at *18:05*, and arrive at the San Francisco airport on the same date at *21:10*. Note that r_{61} corresponds to records r_{11}, r_{21}, and r_{41}, indicated by the yellow highlight shared by all records.

The records in table Airfare4.Fares, shown in Table 1.7, gives the fares for various fare classes of this flight. For example, record r_{71} shows that fare class *A* of this flight has a fare of \$5799.00; the flight identifier *456* is the join key.

Source Airinfo5

Informational source Airinfo5 publishes data about airports and airline in Airinfo5.AirportCodes(Airport Code, Airport Name) and Airinfo5.AirlineCodes(Air Line Code, Air Line Name), respectively.

TABLE 1.6: Sample data for Airfare4.Flight

	FI	FN	DA	DD	DT	AA	AT
r_{61}	456	A1-49	Newark Liberty	2013-12-21	18:05	San Francisco	21:10
r_{62}	457	A1-49	Newark Liberty	2014-04-05	18:05	San Francisco	21:10
r_{63}	458	A1-49	Newark Liberty	2014-04-12	18:05	San Francisco	21:10
r_{64}	460	A2-53	San Francisco	2013-12-22	15:30	Newark Liberty	23:35
r_{65}	461	A2-53	San Francisco	2014-06-28	15:30	Newark Liberty	23:35
r_{66}	462	A2-53	San Francisco	2014-07-06	16:00	Newark Liberty	00:05 (+1d)

TABLE 1.7: Sample data for Airfare4.Fares

	FI	FC	F
r_{71}	456	A	$5799.00
r_{72}	456	K	$999.00
r_{73}	456	Y	$599.00

TABLE 1.8: Sample data for Airinfo5.AirportCodes, Airinfo5.AirlineCodes

	Airinfo5.AirportCodes			Airinfo5.AirlineCodes	
	AC	AN		ALC	ALN
r_{81}	EWR	Newark Liberty, NJ, US	r_{91}	A1	Airline1
r_{82}	SFO	San Francisco, CA, US	r_{92}	A2	Airline2

For example, record r_{81} in Airinfo5.AirportCodes, shown in Table 1.8, states that the name of the airport with code *EWR* is *Newark Liberty, NJ, US*. Similary, record r_{91} in Airinfo5.AirlineCodes, also shown in Table 1.8, states that the name of the airline with code *A1* is *Airline1*.

1.1.2 THE FLIGHTS EXAMPLE: DATA INTEGRATION

While each of the five sources is useful in isolation, the value of this data is considerably enhanced when the different sources are integrated.

TABLE 1.9: Abbreviated attribute names

Short Name	Full Name	Short Name	Full Name
A	Actual	AA	Arrival Airport
AAT	Actual Arrival Time	AC	Airport Code
AD	Arrival Date	ADT	Actual Departure Time
AG	Arrival Gate	AL	Air Line
ALC	Air Line Code	ALN	Air Line Name
AN	Airport Name	AT	Arrival Time
DA	Departure Airport	DD	Departure Date
DG	Departure Gate	DT	Departure Time
ED	End Date	F	Fare
FC	Fare Class	FI	Flight Id
FN	Flight Number	G	Gate
GT	Gate Time	LT	Landing Time
PI	Plane Id	R	Runway
S	Scheduled	SAD	Scheduled Arrival Date
SAT	Scheduled Arrival Time	SD	Start Date
SDD	Scheduled Departure Date	SDT	Scheduled Departure Time
T	Terminal	TT	Takeoff Time

Integrating Sources

First, each airline source (e.g., Airline1, Airline2) benefits by linking with the airport source Airport3 since the airport source provides much more detailed information about the actual flight departures and arrivals, such as gate time, takeoff and landing times, and runways used; this can help the airlines better understand the reasons for flight delays. Second, airport source Airport3 benefits by linking with the airline sources (e.g., Airline1, Airline2) since the airline sources provide more detailed information about the flight schedules and overall flight plans (especially for multi-hop flights such as Airline1's flight *55*); this can help the airport better understand flight patterns. Third, the comparison shopping travel source Airfare4 benefits by linking with the airline and airport sources to provide additional information such as historical on-time departure/arrival statistics; this can be very useful to customers as they make flight bookings. This linkage makes critical use of the informational source Airinfo5, as we shall see later. Finally, customers benefit when the various sources are integrated since they do not need to go to multiple sources to obtain all the information they need.

For example, the query "*for each airline flight number, compute the average delays between scheduled and actual departure times, and between actual gate departure and takeoff times, over the past one month*" can be easily answered over the integrated database, but not using any single source.

However, integrating multiple, autonomous data sources can be quite difficult, often requiring considerable manual effort to understand the semantics of the data in each source to resolve ambiguities. Consider, again, our illustrative Flights example.

Semantic Ambiguity

In order to *align* the various source tables correctly, one needs to understand that (i) the same conceptual information may be modeled quite differently in different sources, and (ii) different conceptual information may be modeled similarly in different sources.

For example, source Airline1 models schedules in table Airline1.Schedule within date ranges (specified by Start Date and End Date), using attributes Departure Time and Arrival Time for time information. However, source Airline2 models schedules along with actual flight information in the table Airline2.Flight, using different records for different actual flights, and differently named attributes Scheduled Departure Date, Scheduled Departure Time, Scheduled Arrival Date, and Scheduled Arrival Time.

As another example, source Airport3 models both actual gate departure/arrival times (Gate Time in Airport3.Departures and Airport3.Arrivals) and actual takeoff/landing times (Takeoff Time in Airport3.Departures, Landing Time in Airport3.Arrivals). However, each of Airline1 and Airline2 models only one kind of departure and arrival times; in particular, a careful examination of the data shows that source Airline1 models gate times (Departure Time and Arrival Time in Airline1.Schedule and Airline1.Flight) and Airline2 models takeoff and landing times (Scheduled Departure Time, Actual Departure Time, Scheduled Arrival Time, Actual Arrival Time in Airline2.Flight).

To illustrate that different conceptual information may be modeled similarly, note that Departure Date is used by source Airline1 to model actual departure date (in Airline1.Flight), but is used to model scheduled departure date by source Airfare4 (in Airfare4.Flight).

Instance Representation Ambiguity

In order 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.

For example, flight numbers are represented in sources Airline1 and Airline2 using digits (e.g., *49* in r_{11}, *53* in r_{31}), while they are represented in source Airfare4 using alphanumerics (e.g., *A1-49* in r_{61}). Similarly, the departure and arrival airports are represented in sources Airline1 and Airline2 using 3-letter codes (e.g., *EWR*, *SFO*, *LAX*), but as a descriptive string in Airfare4.Flight (e.g., *Newark Liberty*, *San Francisco*). Since flights are uniquely identified by the combination of attributes (Airline, Flight Number, Departure Airport, Departure Date), one would not be able to link the data in Airfare4.Flight with the corresponding data in Airline1, Airline2, and Airport3 without additional tables mapping airline codes to airline descriptive names, and airport codes to airport descriptive names, such as Airinfo5.AirlineCodes and Airinfo5.AirportCodes in Table 1.8. Even with such tables, one might need approximate string matching techniques [Hadjieleftheriou and Srivastava 2011] to match *Newark Liberty* in Airfare4.Flight with *Newark Liberty, NJ, US* in Airinfo5.AirportCodes.

Data Inconsistency

In order to *fuse* the data from multiple sources, one needs to resolve the instance-level ambiguities and inconsistencies between the sources.

For example, there is an inconsistency between records r_{32} in Airline2.Flight and r_{52} in Airport3.Arrivals (both of which are highlighted in blue to indicate that they refer to the same flight). Record r_{32} states that the Scheduled Arrival Date and Actual Arrival Time of Airline2's flight *53* are *2013-12-22* and *00:30*, respectively, implying that the actual arrival date is the same as the scheduled arrival date (unlike record r_{31}, where the Actual Arrival Time included *(+1d)* to indicate that the actual arrival date was the day after the scheduled arrival date). However, r_{52} states this flight arrived on *2013-12-23* at *00:30*. This inconsistency would need to be resolved in the integrated data.

As another example, record r_{62} in Airfare4.Flight states that Airline1's flight *49* on *2014-04-05* is scheduled to depart and arrive at *18:05* and *21:10*, respectively. While the departure date is consistent with record r_{12} in Airline1.Schedule (both r_{12} and r_{62} are highlighted in green to indicate their relationship), the scheduled departure and arrival times are not, possibly because r_{62} incorrectly used the (out-of-date) times from r_{11} in Airline1.Schedule. Similary, record r_{65} in Airfare4.Flight states that Airline2's flight *53* on *2014-06-28* is scheduled to depart and arrive at *15:30* and *23:35*, respectively. While the departure date is consistent with record r_{33} in Airline2.Flight (both r_{33} and r_{65} are highlighted in greenish yellow to indicate their relationship), the scheduled departure and arrival times are not, possibly because r_{65} incorrectly used the out-of-date times from r_{32} in Airline2.Flight. Again, these inconsistencies need to be resolved in the integrated data.

1.1.3 DATA INTEGRATION: ARCHITECTURE & THREE MAJOR STEPS

Traditional data integration addresses these challenges of semantic ambiguity, instance representation ambiguity, and data inconsistency by using a pipelined architecture, which consists of three major steps, depicted in Figure 1.1.

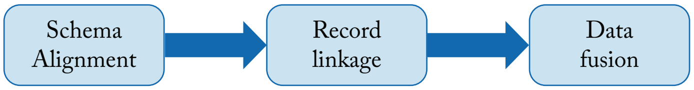


FIGURE 1.1: Traditional data integration: architecture.

The first major step in traditional data integration is that of *schema alignment*, which addresses the challenge of semantic ambiguity and aims to understand which attributes have the same meaning and which ones do not. More formally, we have the following definition.

Definition 1.1 (Schema Alignment) Consider a set of source schemas in the same domain, where different schemas may describe the domain in different ways. *Schema alignment* generates three outcomes.

1. A *mediated schema* that provides a unified view of the disparate sources and captures the salient aspects of the domain being considered.
2. An *attribute matching* that matches attributes in each source schema to the corresponding attributes in the mediated schema.
3. A *schema mapping* between each source schema and the mediated schema to specify the semantic relationships between the contents of the source and that of the mediated data.

The result schema mappings are used to reformulate a user query into a set of queries on the underlying data sources for query answering.

This step is non-trivial for many reasons. Different sources can describe the same domain using very different schemas, as illustrated in our Flights example. They may use different attribute names even when they have the same meaning (e.g., Arrival Date in Airline1.Flight, Actual Arrival Date in Airline2.Flight, and Actual in Airport3.Arrivals). Also, sources may apply different meanings for attributes with the same name (e.g., Actual in Airport3.Departures refers to the actual departure date, while Actual in Airport3.Arrivals refers to the actual arrival date).

The second major step in traditional data integration is that of *record linkage*, which addresses the challenge of instance representation ambiguity, and aims to understand which records represent the same entity and which ones do not. More formally, we have the following definition.

Definition 1.2 (Record Linkage) Consider a set of data sources, each providing a set of records over a set of attributes. *Record linkage* computes a partitioning of the set of records, such that each partition identifies the records that refer to a distinct entity.

Even when schema alignment has been performed, this step is still challenging for many reasons. Different sources can describe the same entity in different ways. For example, records r_{11} in Airline1.Schedule and r_{21} in Airline1.Flight should be linked to record r_{41} in Airport3.Departures; however, r_{11} and r_{21} do not explicitly mention the name of the airline, while r_{41} does not explicitly mention the departure airport, both of which are needed to uniquely identify a flight. Further, different sources may use different ways of representing the same information (e.g., the alternate ways of representing airports as discussed earlier). Finally, comparing every pair of records to determine whether or not they refer to the same entity can be infeasible in the presence of billions of records.

The third major step in traditional data integration is that of *data fusion*, which addresses the challenge of data quality, and aims to understand which value to use in the integrated data when the sources provide conflicting values. More formally, we have the following definition.

Definition 1.3 (Data Fusion) Consider a set of data items, and a set of data sources each of which provides values for a subset of the data items. *Data fusion* decides the true value(s) for each data item.

Such conflicts can arise for a variety of reasons including mis-typing, incorrect calculations (e.g., the conflict in actual arrival dates between records r_{32} and r_{52}), out-of-date information (e.g., the conflict in scheduled departure and arrival times between records r_{12} and r_{62}), and so on.

We will describe approaches used for each of these steps in subsequent chapters, and move on to highlighting the challenges and opportunities that arise when moving from traditional data integration to big data integration.

1.2 BDI: CHALLENGES

To appreciate the challenges that arise in big data integration, we present five recent case studies that empirically examined various characteristics of data sources on the web that would be integrated in BDI efforts, and the dimensions along which these characteristics are naturally classified.

> When you can measure what you are speaking about, and express it in numbers, you know something about it. —Lord Kelvin

1.2.1 THE “V” DIMENSIONS

Big data integration differs from traditional data integration along many dimensions, paralleling the dimensions along which big data is characterized as differing from traditional databases.

Volume

In the big data era, not only can data sources contain a huge volume of data, but also the number of data sources has grown to be in the millions; even for a single domain, the number of sources has grown to be in the tens to hundreds of thousands.

There are many scenarios where a single data source can contain a huge volume of data, ranging from social media and telecommunications networks to finance.

To illustrate a scenario with a large number of sources in a single domain, consider again our Flights example. Suppose we would like to extend it to all airlines and all airports in the world to support flexible, international travel itineraries. With hundreds of airlines worldwide, and over

40,000 airports around the world,[1] the number of data sources that would need to be integrated would easily be in the tens of thousands.

More generally, the case studies we present in Sections 1.2.2, 1.2.3, and 1.2.5 quantify the number of web sources with structured data, and demonstrate that these numbers are much higher than the number of data sources that have been considered in traditional data integration.

Velocity

As a direct consequence of the rate at which data are being collected and continuously made available, many of the data sources are quite dynamic, and the number of data sources is also rapidly exploding.

To illustrate the scenario with dynamic data sources, in our (extended) Flights example, there are tens of thousands of data sources that provide information changing over time. Some of this information changes at the granularity of minutes and hours, such as the estimated departure and arrival times of flights, and the current locations of flights. Other information changes more slowly at the granularity of months, weeks, and days, such as the changes in scheduled departure and arrival times of flights. Providing an integrated view of such dynamically changing data across all these sources is beyond that ability of traditional methods for data integration.

To illustrate the growth rate in the number of data sources, the case study we present in Section 1.2.2 illustrates the explosion in the number of deep web sources within a few years. Undoubtedly, these numbers are likely to be even higher today.

Variety

Data sources from different domains are naturally diverse since they refer to different types of entities and relationships, which often need to be integrated to support complex applications. Further, data sources even in the same domain are quite heterogeneous both at the schema level regarding how they structure their data and at the instance level regarding how they describe the same real-world entity, exhibiting considerable variety even for substantially similar entities. Finally, the domains, source schemas, and entity representations evolve over time, adding to the diversity and heterogeneity that need to be handled in big data integration.

Consider again our Flights example. Suppose we would like to extend it to other forms of transportation (e.g., flights, ships, trains, buses, taxis) to support complex, international travel itineraries. The variety of data sources (e.g., transportation companies, airports, bus terminals) that would need to be integrated would be much higher. In addition to the number of airlines and airports worldwide, there are close to a thousand active seaports and inland ports in the world;[2] there are over a thousand operating bus companies in the world;[3] and about as many operating train companies in the world.[4]

---

[1] https://www.cia.gov/library/publications/the-world-factbook/fields/2053.html (accessed on October 1, 2014).

The case studies we present in Sections 1.2.2, 1.2.4, and 1.2.5 quantify the considerable variety that exist in practice in web sources.

Veracity

Data sources are of widely differing qualities, with significant differences in the coverage, accuracy, and timeliness of data provided.

Our Flights example illustrates specific quality issues that can arise in practice. These quality issues only get exacerbated with an increasing number and diversity of data sources, due to copying between the sources and different types of correlations between the sources in practice.

The case studies we present in Sections 1.2.3, 1.2.4, and 1.2.6 illustrate the significant coverage and quality issues that exist in data sources on the web, even for the same domain. This provides some context for the observation that "one in three business leaders do not trust the information they use to make decisions."[5]

1.2.2 CASE STUDY: QUANTITY OF DEEP WEB DATA

The deep web consists of a large number of data sources where data are stored in databases and obtained (or surfaced) by querying web forms. He et al. [2007] and Madhavan et al. [2007] experimentally study the *volume*, *velocity*, and domain-level *variety* of data sources available on the deep web.

Main Questions

These two studies focus on two main questions related to the "V" dimensions presented in Section 1.2.1.

- What is the scale of the deep web?

  For example, how many query interfaces to databases exist on the web? How many web databases are accessible through such query interfaces? How many web sources provide query interfaces to databases? How have these deep web numbers changed over time?

---

[2] http://www.ask.com/answers/99725161/how-many-sea-ports-in-world (accessed on October 1, 2014).

[3] http://en.wikipedia.org/wiki/List_of_bus_operating_companies (accessed on October 1, 2014).

[4] http://en.wikipedia.org/wiki/List_of_railway_companies (accessed on October 1, 2014).

[5] http://www-01.ibm.com/software/data/bigdata/ (accessed on October 1, 2014).

- What is the distribution of domains in web databases?

  For example, is the deep web driven and dominated by e-commerce, such as product search? Or is there considerable domain-level variety among web databases? How does this domain-level variety compare to that on the surface web?

Study Methodology

In the absence of a comprehensive index to deep web sources, both studies use sampling to quantify answers to these questions.

He et al. [2007] take an IP sampling approach to collect server samples, by randomly sampling 1 million IP addresses in 2004, using the Wget HTTP client to download HTML pages, then *manually* identifying and analyzing web databases in this sample to extrapolate their estimates of the deep web to the estimated 2.2 billion valid IP addresses. This study distinguishes between deep web sources, web databases (a deep web source can contain multiple web databases), and query interfaces (a web database could be accessed by multiple query interfaces), and uses the following methodology.

1. The web sources are crawled to a depth of three hops from the root page. All the HTML query interfaces on the retrieved pages are identified.

   Query interfaces (within a source) that refer to the same database are identified by manually choosing a few random objects that can be accessed through one interface and checking to see if each of them can be accessed through the other interfaces.
2. The domain distribution of the identified web databases is determined by manually categorizing the identified web databases, using the top-level categories of the http://yahoo.com directory (accessed on October 1, 2014) as the taxonomy.

Madhavan et al. [2007] instead use a random sample of 25 million web pages from the Google index from 2006, then identify deep web query interfaces on these pages in a *rule-driven manner*, and finally extrapolate their estimates to the 1 billion+ pages in the Google index. Using the terminology of He et al., this study mainly examines the number of query interfaces on the deep web, not the number of distinct deep web databases. For this task, they use the following methodology.

1. Since many HTML forms are present on multiple web pages, they compute a signature for each form by combining the host present in the action of the form with the names of the visible inputs in the form. This is used as a lower bound for the number of distinct HTML forms.

2. From this number, they prune away non-query forms (such as password entry) and site search boxes, and only count the number of forms that have at least one text input field, and between two and ten total inputs.

Main Results

We categorize the main results of these studies according to the investigated "V" dimensions.

Volume, Velocity. The 2004 study by He et al. [2007] estimates a total of 307,000 deep web sources, 450,000 web databases, and 1,258,000 distinct query interfaces to deep web content. This is based on extrapolation from a total of 126 deep web sources, containing 190 web databases and 406 query interfaces identified in their random IP sample. This number of identified sources, databases, and query interfaces enables much of their analysis to be accomplished by manually inspecting the identified query interfaces.

The subsequent 2006 study by Madhavan et al. [2007] estimates a total of more than 10 million distinct query interfaces to deep web content. This is based on extrapolating from a total of 647,000 distinct query interfaces in their random sample of web pages. Working with this much larger number of query interfaces requires the use of automated approaches to differentiate query interfaces to the deep web from non-query forms. This increase in the number of query interfaces identified by Madhavan et al. over the number identified by He et al. is partly a reflection of the *velocity* at which the number of deep web sources increased between the different time periods studied.

Variety. The study by He et al. [2007] shows that deep web databases have considerable domain-level *variety*, where 51% of the 190 identified web databases in their sample are in non e-commerce domain categories, such as health, society & culture, education, arts & humanities, science, and so on. Only 49% of the 190 identified web databases are in e-commerce domain categories. Table 1.10 shows the distribution of domain categories identified by He et al., illustrating the domain-level variety of the data in BDI. This domain-level variety of web databases is in sharp contrast to the surface web, where an earlier study identified that e-commerce web sites dominate with an 83% share.

The study by Madhavan et al. [2007] also confirms that the semantic content of deep web sources varies widely, and is distributed under most directory categories.

1.2.3 CASE STUDY: EXTRACTED DOMAIN-SPECIFIC DATA

The documents that constitute the surface web contain a significant amount of structured data, which can be obtained using web-scale information extraction techniques. Dalvi et al. [2012] experimentally study the *volume* and coverage properties of such structured data (i.e., entities and their attributes) in several domains (e.g., restaurants, hotels).

TABLE 1.10: Domain category distribution of web databases [He et al. 2007]

Domain Category	E-commerce	Percentage
Business & Economy	Yes	24%
Computers & Internet	Yes	16%
News & Media	Yes	6%
Entertainment	Yes	1%
Recreation & Sports	Yes	2%
Health	No	4%
Government	No	2%
Regional	No	4%
Society & Culture	No	9%
Education	No	16%
Arts & Humanities	No	4%
Science	No	2%
Reference	No	8%
Others	No	2%

Main Questions

Their study focuses on two main questions related to the "V" dimensions presented in Section 1.2.1.

- How many sources are needed to build a complete database for a given domain, even restricted to well-specified attributes?

  For example, is it the case that well-established head aggregators (such as http://yelp.com for restaurants) contain most of the information, or does one need to go to the long tail of web sources to build a reasonably complete database (e.g., with 95% coverage)? Is there a substantial need to construct a comprehensive database, for example, as measured by the demand for tail entities?

- How easy is it to discover the data sources and entities in a given domain?

  For example, can one start with a few data sources or seed entities and iteratively discover most (e.g., 99%) of the data? How critical are the head aggregators to this process of discovery of data sources?

Study Methodology

One way to answer the questions is to actually perform web-scale information extraction in a variety of domains, and compute the desired quantities of interest; this is an extremely challenging task, for which good solutions are currently being investigated. Instead, the approach that Dalvi et al. [2012] take is to study domains with the following three properties.

1. One has access to a comprehensive structured database of entities in that domain.
2. The entities can be uniquely identified by the value of some key attributes available on the web pages.
3. One has access to (nearly) all the web pages containing the key attributes of the entities.

Dalvi et al. identify nine such domains: books, restaurants, automotive, banks, libraries, schools, hotels & lodging, retail & shopping, and home & garden. Books are identified using the value of ISBN, while entities in the other domains are identified using phone numbers and/or home page URLs. For each domain, they look for the identifying attributes of the entities on each web page in the Yahoo! web cache, group web pages by hosts into sources, and aggregate the entities found on all the web pages of each data source.

They model the problem of ease of discovery of data sources and entities using a bi-partite graph of entities and sources, with an edge (E, S) indicating that an entity E is found in source S. Graph properties like connectivity of the bi-partite graph can help understand the robustness of iterative information extraction algorithms with respect to the choice of the seed entities or data sources for bootstrapping. Similarly, the diameter can indicate how many iterations are needed for convergence. In this way, they don't need to do actual information extraction, and only study the distribution of information about entities already in their database. While this methodology has its limitations, it provides a good first study on this topic.

Main Results

We categorize the main results of this study according to the investigated "V" dimensions.

Volume. First, they find that all the domains they study have thousands to tens of thousands of web sources (see Figure 1.2 for phone numbers in the restaurant domain). These numbers are much higher than the number of data sources that are considered in traditional data integration.

Second, they show that tail sources contain a significant amount of information, even for domains like restaurants with well-established aggregator sources. For example, http://yelp.com is shown to contain fewer than 70% of the restaurant phone numbers and fewer than

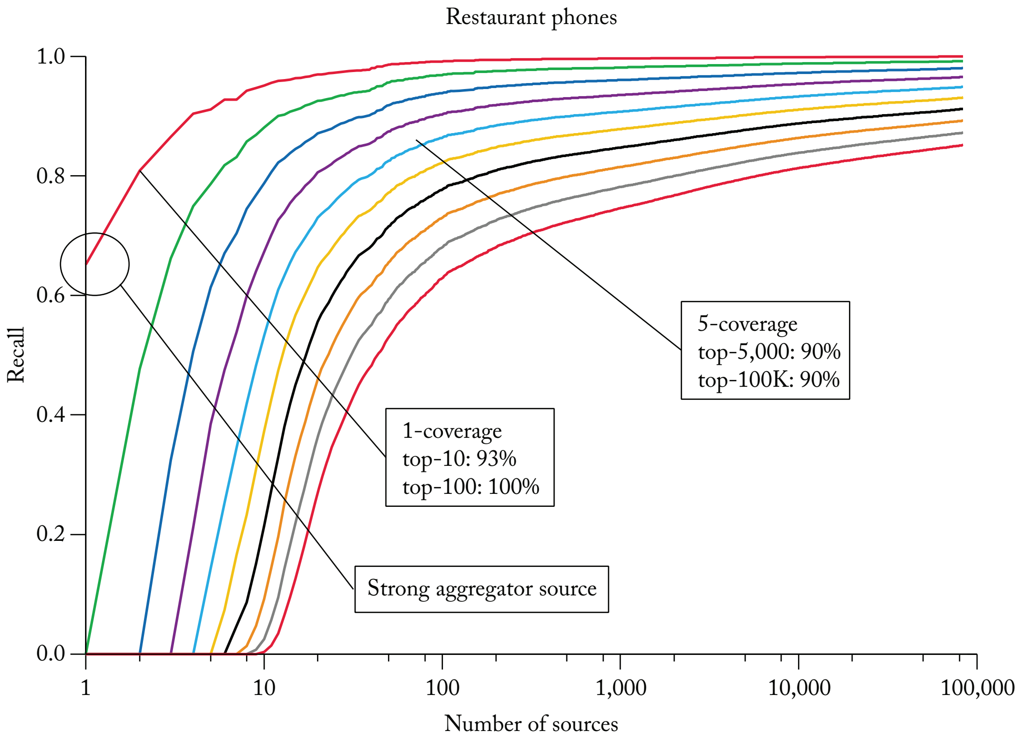


FIGURE 1.2: K-coverage (the fraction of entities in the database that are present in at least *k* different sources) for phone numbers in the restaurant domain [Dalvi et al. 2012].

40% of the home pages of restaurants. With the top 10 sources (ordered by decreasing number of entities found on the sources), one can extract around 93% of all restaurant phone numbers, and with the top 100 sources one can extract close to 100% of all restaurant phone numbers, as seen in Figure 1.2. However, for a less available attribute such as home page URL, the situation is quite different: one needs at least 10,000 sources to cover 95% of all restaurant home page URLs.

Third, they investigate the redundancy of available information using *k*-coverage (the fraction of entities in the database that are present in at least *k* different sources) to enable a higher confidence in the extracted information. For example, they show that one needs 5000 sources to get 5-coverage of 90% of the restaurant phone numbers (while 10 sources is sufficient to get 1-coverage of 93% of these phone numbers), as seen in Figure 1.2.

Fourth, they demonstrate (using user-generated restaurant reviews) that there is significant value in extracting information from the sources in the long tail. In particular, while both

Graph		Avg. no. sites per entity	Diameter	No. conn. comp.	Percent entities in largest comp.
Domain	Attr				
Books	ISBN	8	8	439	99.96
Automotive	phone	13	6	9	99.99
Banks	phone	22	6	15	99.99
Home	phone	13	8	4507	99.76
Hotels	phone	56	6	11	99.99
Libraries	phone	47	6	3	99.99
Restaurants	phone	32	6	52	99.99
Retail	phone	19	7	628	99.93
Schools	phone	37	6	48	99.97
Automotive	homepage	115	6	10	98.52
Banks	homepage	68	8	30	99.57
Home	homepage	20	8	5496	97.87
Hotels	homepage	56	8	24	99.90
Libraries	homepage	251	6	4	99.86
Restaurants	homepage	46	6	146	99.82
Retail	homepage	45	7	1260	99.20
Schools	homepage	74	6	122	99.57

High data redundancy

Low diameter

Highly connected data

FIGURE 1.3: Connectivity (between entities and sources) for the nine domains studied by Dalvi et al. [2012].

the demand for and the availability of review information reduces towards the tail, information availability reduces at a faster rate, suggesting that tail extraction can be valuable in spite of the lower demand.

Fifth, as seen in Figure 1.3, they observe that there is a significant amount of data redundancy (tens to hundreds of sources per entity on average), and the data within a domain is well connected. This redundancy and well connectedness is critical for discovery of sources and entities in BDI. In particular, for almost all the (domain, attribute) pairs, over 99% of the entities are present in the largest connected component of the bi-partite graph, establishing that even a randomly chosen small seed set of entities is sufficient to reach most of the entities in the domain. Further, a small diameter (around 6–8) implies that iterative approaches would converge fairly rapidly. Finally, they show that the graphs remain well connected (with over 90% entities) even after the top 10 aggregator sources are removed, demonstrating that the connectivity does not depend only on the head aggregator sources.

1.2.4 CASE STUDY: QUALITY OF DEEP WEB DATA

While the studies by He et al. [2007] and Madhavan et al. [2007] shed light on the *volume*, *velocity*, and domain-level *variety* of deep web data, they do not investigate the quality of data present in these sources. To overcome this limitation, Li et al. [2012] experimentally study the *veracity* of deep web data.

Main Questions

This study focuses on two main questions related to the "V" dimensions presented in Section 1.2.1.

- What is the quality of deep web data?

  For example, are there a lot of redundant data among deep web sources? Are the data consistent across sources in a domain? Is the quality of data better in some domains than others?

- What is the quality of the deep web sources?

  For example, are the sources highly accurate? Are correct data provided by the majority of the sources? Is there an authoritative source that can be trusted while all the other sources are ignored, in case of inconsistency across sources? Do sources share data with or copy from other sources?

Study Methodology

One way to answer these questions is to actually perform big data integration across all the deep web sources in each of multiple domains; this is an extremely challenging task that has not yet been solved. Instead, the approach that Li et al. [2012] take is to study a few domains with the following properties.

1. The deep web sources in these domains are frequently used, and believed to be clean since incorrect values can have an adverse effect on people's lives.
2. The entities in these domains are consistently and uniquely identified across sources by the value of some key attributes, making it easy to link information across deep web sources.
3. Focusing on a moderate number of popularly used sources is sufficient to understand the quality of data experienced by users in these domains.

The study by Li et al. [2012] identifies two such domains: Stock and Flight. Stocks are consistently and uniquely identified by stock symbols (e.g., *T* for *AT&T Inc.*, and *GOOG* for *Google, Inc.*) across sources, and flight numbers (e.g., *UA 48*) and departure/arrival airport codes (e.g., *EWR* and *SFO*) are typically used to identify flights on a given day across sources. They identify a moderately large number of popular deep web sources in each of the domains by: (i) using domain-specific search terms on popular search engines and manually identifying deep web sources from the top 200 returned results; (ii) focusing on those sources that use the GET method (i.e., the form data are encoded in the URL itself), and don't use Javascript. This results in 55 sources (including popular financial aggregators such as Yahoo! Finance, Google Finance, and MSN Money, official stock exchange sources such as NASDAQ, and financial news sources such as Bloomberg and MarketWatch) in the Stock domain and 38 sources (including 3 airline sources, 8 airport hub sources, and 27 third-party sources such as Orbitz, Travelocity, etc.) in the Flight domain.

In the Stock domain, they pick 1000 stock symbols from the Dow Jones, NASDAQ, and Russell 3000, and query each stock symbol on each of the 55 sources every week day in July 2011. The queries are issued one hour after the stock market closes each day. Extracted attributes are manually matched across sources to identify globally distinct attributes; of these, 16 popular attributes whose values should be fairly stable after the stock market closes (such as daily closing price) are analyzed in detail. A gold standard is generated for 200 stock symbols by taking the majority voting results from 5 popular financial sources.

In the Flight domain, they focus on 1200 flights departing from or arriving at the hub airports of the three airlines, United, Continental, and American, and query for each flight at least one hour after the scheduled arrival time every day in December 2011. Extracted attributes are manually matched across sources to identify globally distinct attributes; of these, six popular attributes are analyzed in detail. A gold standard is generated for 100 flights by taking the data provided by the corresponding airline source.

Main Results

We categorize the main results of this study according to the "V" dimensions presented in Section 1.2.1, as in the previous study.

Although the primary focus of this study is *veracity*, the results of this study also cast some light on the schema-level *variety* of deep web sources.

Variety. Li et al. [2012] identify considerable schema-level *variety* among the deep web sources examined. For example, the 55 sources in the Stock domain provide different numbers of attributes, ranging from 3–71, for a total of 333 attributes. After manually matching these attributes across sources, they identify a total of 153 globally distinct attributes, many of which are computed using other attributes (e.g., 52 week high and low prices). The distribution of the number of providers for these attributes is highly skewed, with only 13.7% of the attributes (a total of 21) provided by at least one third of the sources, and over 86% attributes provided by fewer than 25% of the sources. The Flight domain does not exhibit as much schema-level variety, with the 38 sources providing 43 attributes, which are manually matched to obtain 15 globally distinct attributes.

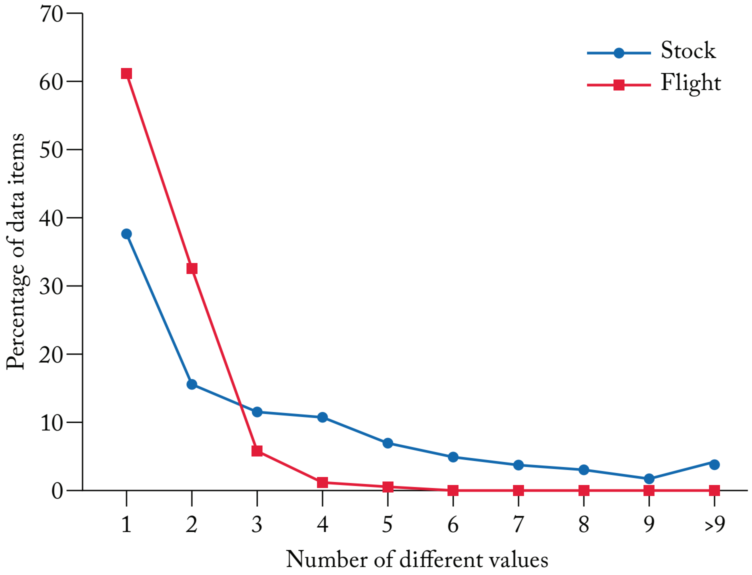


FIGURE 1.4: Consistency of data items in the Stock and Flight domains [Li et al. 2012].

Veracity. The quality of data is not as high as expected, given that the data in the domains studied are expected to be quite clean. In particular, the data in these domains exhibit significant levels of inconsistency. In the Stock domain, for example, the number of different values (even after allowing for some value tolerance) for a data item ranges from 1–13, with an average of 3.7; further, inconsistent values are provided by different sources for over 60% of the data items. Value inconsistency is much lower in the Flight domain, where the number of different values (after allowing for value tolerance) for a data item ranges from 1–5, with an average of 1.45; further, inconsistent values are provided by different sources for fewer than 40% of the data items. There are different reasons for the observed inconsistencies, including semantic ambiguity, out-of-date data, and errors. Figure 1.4 illustrates the distribution of number of values for data items for both domains. Li et al. show that these inconsistencies cannot be effectively addressed by using *naive voting*, which often has an even lower accuracy than the highest accuracy from a single source.

Similarly, they observe that the accuracy of deep web sources can vary a lot. In the Stock domain, the average source accuracy is just 0.86, and only 35% of the sources have an accuracy above 0.9. While most of the authoritative sources have an accuracy above 0.9, their coverages are all below 0.9, implying that one cannot rely on a single authoritative source and ignore all other sources. In the Flight domain, the average source accuracy is even lower, just 0.8, and

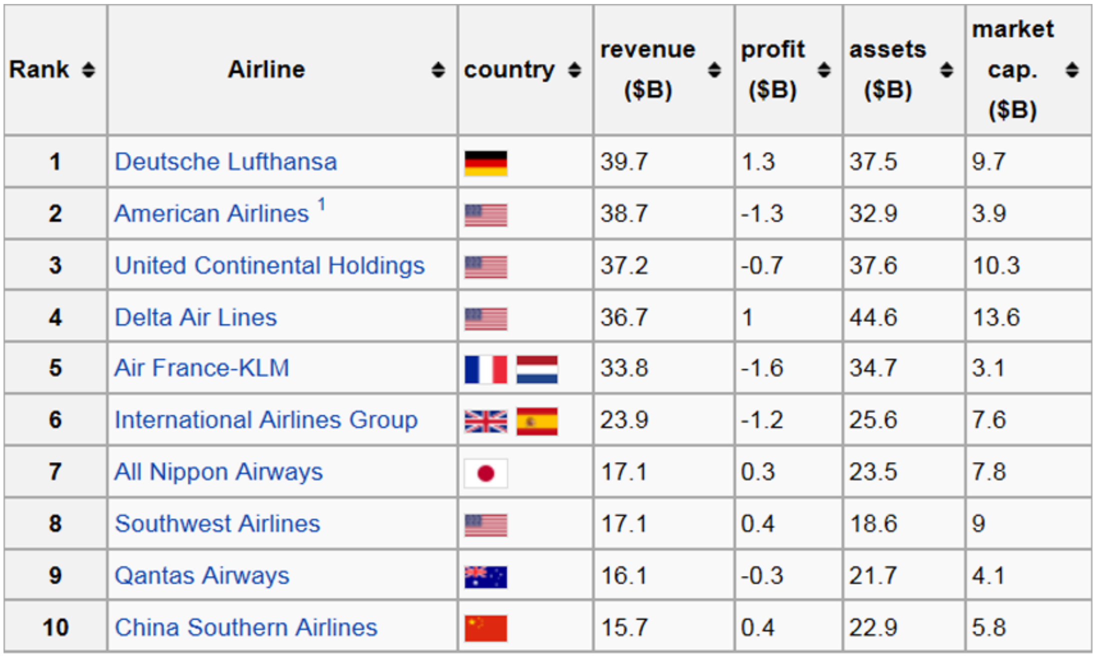
29% of sources have an accuracy below 0.7. Authoritative sources in this domain again have accuracies above 0.9, but their coverages are all below 0.9.

Finally, Li et al. [2012] observe copying between deep web sources in each domain. In some cases, the copying is claimed explicitly, while in other cases it is detected by observing embedded interfaces or query redirection. Interestingly, the accuracy of the original sources that are copied is not always high, ranging from 0.75–0.92 for Stock, and from 0.53–0.93 for Flight.

1.2.5 CASE STUDY: SURFACE WEB STRUCTURED DATA

The static HTML pages on the surface web obviously contain a vast volume of unstructured data, but also include a huge volume of structured data in the form of HTML tables, such as the table in Figure 1.5. Cafarella et al. [2008b] and Lautert et al. [2013] experimentally study the *volume* and structural *variety* of such tables on the web.

This work is motivated by the fact that the surface web is typically modeled as a hyperlinked collection of unstructured documents, which tends to ignore the relational data contained in web documents. For example, most wikipedia pages contain high-quality relational data that provide



Rank ⬍	Airline ⬍	country ⬍	revenue ($B) ⬍	profit ($B) ⬍	assets ($B) ⬍	market cap. ($B) ⬍
1	Deutsche Lufthansa	🇩🇪	39.7	1.3	37.5	9.7
2	American Airlines [1]	🇺🇸	38.7	-1.3	32.9	3.9
3	United Continental Holdings	🇺🇸	37.2	-0.7	37.6	10.3
4	Delta Air Lines	🇺🇸	36.7	1	44.6	13.6
5	Air France-KLM	🇫🇷 🇳🇱	33.8	-1.6	34.7	3.1
6	International Airlines Group	🇬🇧 🇪🇸	23.9	-1.2	25.6	7.6
7	All Nippon Airways	🇯🇵	17.1	0.3	23.5	7.8
8	Southwest Airlines	🇺🇸	17.1	0.4	18.6	9
9	Qantas Airways	🇦🇺	16.1	-0.3	21.7	4.1
10	China Southern Airlines	🇨🇳	15.7	0.4	22.9	5.8

FIGURE 1.5: High-quality table on the web.

valuable information on just about every topic. By explicitly recognizing relational tables on the surface web, which are accessible to crawlers, web search engines can return such tables as well in response to user keyword queries.

Main Questions

These studies focus on two main questions related to the "V" dimensions presented in Section 1.2.1.

- How many high-quality relational tables are present on the surface web? How does one distinguish them from other uses of HTML tables (for example, form layout)?
- How heterogeneous are these tables?

  For example, what is the distribution of table sizes, in terms of number of rows and columns? How many of these tables have a richer structure (for example, nested tables, cross-tabs) than conventional relational tables?

Study Methodology

Cafarella et al. [2008b] start from a multi-billion page english language portion of the Google crawl, and use an HTML parser to obtain all occurrences of the HTML `table` tag. Only a small fraction of the identified tables are high-quality relational tables, and they use the following methodology to distinguish them from non-relational uses of the HTML tag.

1. They use parsers to eliminate obviously non-relational tables, including extremely small tables (fewer than two rows or two columns), those that are embedded in HTML forms (which are used for visual layout of user input fields), and calendars.
2. They use a sample of the remaining tables, and human labeling to estimate the total fraction of high-quality relational tables.
3. They train a classifier to distinguish between relational tables and other uses of the HTML `table` tag, such as page layout and property sheets, using a variety of table-level features. They subsequently collect distributional statistics using the output of the classifer.

Lautert et al. [2013] observe that even high-quality tables on the web are structurally heterogeneous, with horizontal, vertical, and matrix structures, some having cells that span multiple rows or columns, some with multiple values in individual cells, and so on. They use the following methodology to quantify the structural heterogeneity of tables on the web.

1. They extract all HTML tables from a collection of crawled sources starting from wikipedia, e-commerce, news, and university sources, visiting a total of 174,927 HTML pages, and extracting 342,795 unique HTML tables.

TABLE 1.11: Row statistics on high-quality relational tables on the web [Cafarella et al. 2008b]

Number of Rows	Percent of Tables
2-9	64.07
10–19	15.83
20–29	7.61
30+	12.49

2. They develop a supervised *neural network* classifer to classify tables into different categories, using a list of 25 layout, HTML, and lexical features. The training set uses 4,000 web tables.

Main Results

We categorize the main results of these studies according to the investigated "V" dimensions.

Volume. First, Cafarella et al. [2008b] extract approximately 14.1 billion raw HTML tables from the crawl. Of these, 89.4% (or 12.5 billion) are eliminated as obviously non-relational (almost all of which are extremely small tables) using their parsers. Of the remaining tables, human judgement is used on a sample to determine about 10.4% (or 1.1% of raw HTML tables) as high-quality relational tables. This results in an estimate of 154 million high-quality relational tables on the web.

Second, Cafarella et al. [2008b] train a classifier using features such as numbers of rows and columns, number of rows with mostly nulls, number of columns with non-string data, average and standard deviation of string lengths in cells, and so on, to identify high-quality relational tables with a high recall of 0.81, even though the precision is lower at 0.41. Using the results of the classifier, they identify distributional statistics on numbers of rows and columns of high-quality relational tables. More than 93% of these tables have between two and nine columns; there are very few high-quality tables with a very large number of attributes. In contrast, there is a greater diversity in the number of rows among high-quality tables, as shown in Table 1.11.

Variety. Lautert et al. [2013] determine that there is considerable structural variety even among the high-quality tables on the web. Only 17.8% of the high-quality tables on the web are akin to traditional RDBMS tables (each cell contains a single value, and does not span more than one row or column). The two biggest reasons for tables on the web differing from RDBMS tables are: (i) 74.9% of the tables have cells with multiple values (of the same type or of different types) and (ii) 12.9% of the tables have cells that span multiple rows or columns.

1.2.6 CASE STUDY: EXTRACTED KNOWLEDGE TRIPLES

Our final case study is about domain-independent structured data represented as ⟨subject, predicate, object⟩ knowledge triples, obtained using web-scale information extraction techniques. In our Flight example, the triples ⟨*Airline1_49*, *departs_from*, *EWR*⟩ and ⟨*Airline1_49*, *arrives_at*, *SFO*⟩ represent that the Departure Airport and Arrival Airport of Airline1's flight *49* are *EWR* and *SFO*, respectively. Dong et al. [2014b] experimentally study the *volume* and *veracity* of such knowledge triples obtained by crawling a large set of web pages and extracting triples from them.

This work is motivated by the task of automatically constructing large-scale knowledge bases by using multiple extractors to extract (possibly conflicting) values from each data source for each data item, then resolving various ambiguities present in the extracted triples to construct a high quality knowledge base.

Main Questions

This study focuses on two main questions related to the "V" dimensions presented in Section 1.2.1.

- What are the number and distributional properties of knowledge triples that can be extracted from web pages?

  For example, how many triples can be extracted from DOM trees found in web pages vs. using natural language processing techniques on unstructured text?
- What is the quality of the extracted triples, and the accuracy of the extractors that are used for this purpose?

Study Methodology

Dong et al. [2014b] crawl over 1 billion web pages, to extract knowledge triples from four types of web content, using the following methodology.

1. They extract knowledge triples from: (i) text documents, by examining phrases and sentences; (ii) DOM trees, which can be found on surface web pages (e.g., web lists), as well as in deep web sources; (iii) web tables, which contain high quality relational information, where rows represent subjects, columns represent predicates, and the corresponding cells contain the objects of the triples; and (iv) web annotations, manually created by webmasters using standard web ontologies such as http://schema.org.
2. They limit attention to extracting triples whose subjects and predicates exist in the manually curated *Freebase* knowledge base [Bollacker et al. 2008].
3. The quality of the extracted knowledge is also evaluated against the Freebase knowledge base as a gold standard. Specifically, if an extracted triple $\langle s, p, o\rangle$ occurs in Freebase, it is considered to be true; if $\langle s, p, o\rangle$ does not occur in Freebase, but $\langle s, p, o'\rangle$ does, then the extracted triple $\langle s, p, o\rangle$ is considered to be false; otherwise it is not included in the gold standard.

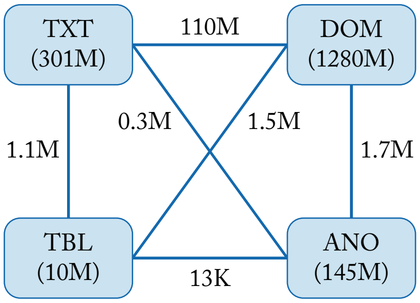


FIGURE 1.6: Contributions and overlaps between different types of web contents [Dong et al. 2014b].

Main Results

We categorize the main results of this study according to the investigated "V" dimensions.

Volume. First, Dong et al. [2014b] extract 1.6 billion distinct triples, with about 80% of the triples from DOM trees, followed by about 19% from text documents, with little overlap between the triples extracted from the different types of web content, as shown in Figure 1.6.

Second, these extracted triples are associated with 43 million subjects and 4.5 thousand predicates (with 337 million (subject, predicate) pairs) from Freebase. Most distributions (such as #triples per subject) are highly skewed, with a long tail; for example, there are over 1 million triples for each of the top 5 entities, whereas for 56% entities they extract no more than 10 triples each.

Veracity. Among the 1.6 billion triples, 40% (or 650 million) have gold standard labels, of which 200 million are considered as true. Thus, the overall accuracy of extracted triples is only about 30%. Most of the errors are due to errors in the extractions, but a small percentage are due to wrong information provided by the sources.

The study also shows a high variance in the accuracy of the extractors.

1.3 BDI: OPPORTUNITIES

BDI does not only come with difficult challenges, characterized along the "V" dimensions, as we discussed in Section 1.2. There are also interesting opportunities enabled by BDI and the infrastructures used for managing and analyzing big data, to effectively address these challenges. We focus on three such opportunities.

1.3.1 DATA REDUNDANCY

The data obtained from different sources often overlap, resulting in a *high data redundancy* across the large number of sources that need to be integrated.

This is evident in our motivating Flights example, where information such as Departure Airport, Scheduled Departure Time, Arrival Airport, and Scheduled Arrival Time about Airline1's flight *49* can be obtained from each of the sources Airline1, Airport3 and Airfare4.

The case studies presented in Sections 1.2.3 and 1.2.4 illustrate the redundancy that exists in many domains. Specifically, the study by Dalvi et al. [2012] mentions that the average number of sources per entity is quite large in all the domains studied, with 56 sources per hotel phone number and 251 sources per library home page as particularly notable, as shown in Figure 1.3. Further, this high average value is not just due to extreme skew; for example, over 80% of restaurant phone numbers are present in at least 10 distinct sources, as shown by the 10-coverage plot in Figure 1.2. Similarly, the study by Li et al. [2012] identifies 16 popular attributes in the Stock domain and 6 popular attributes in the Flight domain that are provided by at least one third of the sources analyzed in each of these domains.

One key advantage of this data redundancy is to effectively address the *veracity* challenge in BDI, as we discuss in detail in Chapter 4. Intuitively, if there are only a few sources that provide overlapping information, and the sources provide conflicting values for a particular data item, it is difficult to identify the true value with high confidence. But with a large number of sources, as is the case in BDI, one can use sophisticated data fusion techniques to discover the truth.

A second advantage of data redundancy is to begin to address the *variety* challenge in BDI, and identify attribute matchings between the source schemas, which are critical for schema alignment. Intuitively, if there is significant data redundancy in a domain, and the bi-partite graph of entities and sources is well connected (as in the domains studied by Dalvi et al. [2012]), one can start with a small seed set of known entities, and use search engine technology to discover most of the entities in that domain. When these entities have different schemas associated with them in the different sources, one can naturally identify attribute matchings between the schemas used by the different sources.

A third advantage of data redundancy is the ability to discover relevant sources for BDI in a domain, when sources are not all known *a priori*. The key intuition again is to take advantage of a well-connected bi-partite graph of entities and sources, start with a small seed set of known entities, and use search engine technology to iteratively discover new sources and new entities, in an alternating manner.

1.3.2 LONG DATA

A significant source of big data in practice is *long data*, that is, data collected about evolving entities over time.

In our motivating Flights example, the schedules of airlines evolve over time, as illustrated in the table Airline1.Schedule. In practice, airline and airport sources typically provide estimated flight departure and arrival times, which can vary considerably over short time periods; airplane maintenance and repair logs provide insight about airplane quality over time, and so on.

While the case studies that we presented earlier in this chapter do not specifically deal with long data, some of the techniques that we will describe in subsequent chapters, especially for record linkage (Chapter 3) and data fusion (Chapter 4), take considerable advantage of the presence of long data.

Intuitively, entities in the real world evolve, which result in their attribute values changing over time. The information provided by data sources that contain such entities is not always fresh, and out-of-date values are common, as illustrated in the table Airfare4.Flight. Record linkage and data fusion in such scenarios are challenging, but can take advantage of the fact that evolution of entities is typically a gradual and relatively smooth process: (i) even when some attributes of a flight (e.g., Scheduled Departure Time) evolve, other attributes (e.g., Departure Airport) do not necessarily change; and (ii) even when entities evolve over short time periods, changes in attribute values are usually not erratic (e.g., the changes to estimated arrival time of a flight as reported by the airline).

1.3.3 BIG DATA PLATFORMS

The management and analysis of big data has benefited considerably from significant advances in recent years from scalable big data platforms on clusters of commodity hardware (e.g., Hadoop), and distributed programming models (e.g., MapReduce).

Big data integration can be extremely resource intensive, with each of the tasks of schema alignment, record linkage, and data fusion requiring significant computational resources. While much work remains to be done to take full advantage of the big data platforms available, recent work in this area has brought hope that these tasks can in fact be effectively parallelized. We present a few such techniques, especially for record linkage and data fusion, in subsequent chapters.

1.4 OUTLINE OF BOOK

The rest of the book is structured as follows. In the next three chapters, we focus on each of the main tasks of data integration. Chapter 2 focuses on *schema alignment*, Chapter 3 focuses on *record linkage*, and Chapter 4 focuses on *data fusion*. Each of these chapters is organized similarly: we start with a quick tour of the task in the context of traditional data integration, before describing how the various BDI challenges of volume, velocity, variety, and veracity have been addressed in the recent literature. In Chapter 5, we outline emerging topics that are specific to BDI and identify promising directions of future work in this area. Finally, Chapter 6 summarizes and concludes the book.

Authors' Biographies

XIN LUNA DONG

Xin Luna Dong is a senior research scientist at Google Inc. Prior to joining Google, she worked for AT&T Labs-Research. She received her Ph.D. from University of Washington, received a Master's Degree from Peking University in China, and a Bachelor's Degree from Nankai University in China. Her research interests include databases, information retrieval, and machine learning, with an emphasis on data integration, data cleaning, knowledge bases, and personal information management. She has published more than 50 papers in top conferences and journals in the field of data integration, and got the Best Demo award (one of top-3) in Sigmod 2005. She is the PC co-chair for WAIM 2015 and has served as an area chair for Sigmod 2015, ICDE 2013, and CIKM 2011.

DIVESH SRIVASTAVA

Divesh Srivastava is the head of Database Research at AT&T Labs-Research. He is a fellow of the Association for Computing Machinery (ACM), on the board of trustees of the VLDB Endowment, the managing editor of the *Proceedings of the VLDB Endowment* (*PVLDB*), and an associate editor of the *ACM Transactions on Database Systems*. He received his Ph.D. from the University of Wisconsin, Madison, and his Bachelor of Technology from the Indian Institute of Technology, Bombay, India. His research interests and publications span a variety of topics in data management. He has published over 250 papers in top conferences and journals. He has served as PC Chair or Co-chair of many international conferences including ICDE 2015 (Industrial) and VLDB 2007.