# Understanding Quantities in Web Tables and Text

**Yusra Ibrahim**

Dissertation
zur Erlangung des Grades
*Doktor der Ingenieurwissenschaften (Dr.-Ing.)*
Fakultät für Mathematik und Informatik
der Universität des Saarlandes

Saarbrücken
2019

| | |
|---|---|
| Dean | Prof. Dr. Sebastian Hack |
| Colloqium | 08.10.2019<br>Saarbrücken |

### Examination Board

| | |
|---|---|
| Supervisor and Reviewer | Prof. Dr.-Ing. Gerhard Weikum |
| Reviewer | Prof. Dr. Mirek Riedewald |
| Reviewer | Prof. Dr. Klaus Berberich |
| Chairman | Prof. Dr. Dr. h.c. mult. Kurt Mehlhorn |
| Research Assistant | Dr. Paramita Mirza |

# Abstract

There is a wealth of schema-free tables on the web, arranging valuable information about quantities on sales and costs, the environmental footprint of cars, health data, and more. The text accompanying these tables explains and qualifies the numerical quantities given in the tables. Despite this ubiquity of tabular data, there is little research that harnesses this wealth of data by semantically understanding the information that is conveyed rather ambiguously in these tables. This information can be disambiguated only by the help of the accompanying text. Understanding quantities in tables and text would provide opportunities for answering queries about numerical quantities like “Internet companies with annual income above 5 Mio. USD”, “electric cars with energy consumption below 100 MPGe”, or “clinical trials with a daily anti-coagulant dosage above 30 mg”.

In the process of understanding quantity mentions in tables and text, we are faced with the following challenges; First, there is no comprehensive knowledge base for anchoring quantity mentions. Second, tables are created ad-hoc without a standard schema and with ambiguous header names; also table cells usually contain abbreviations. Third, quantities can be written in multiple forms and units of measures–for example “48 km/h” is equivalent to “30 mph”. Fourth, the text usually refers to the quantities in tables using aggregation, approximation, and different scales.

In this thesis, we target these challenges through the following contributions:

- We present the Quantity Knowledge Base (QKB), a knowledge base for representing Quantity mentions. We construct the QKB by importing information from Freebase, Wikipedia, and other online sources. In the QKB, we organize quantities in a simple four-level taxonomy of dimensions, units, measures, and thematic domains. The QKB enables the canonicalization of quantity mentions into a triple of the form <measure, value, unit>.

- We propose Equity: a system for automatically canonicalizing header names and cell values onto concepts, classes, entities, and uniquely represented quantities registered in a knowledge base. We devise a probabilistic graphical model that captures coherence dependencies between cells in tables and candidate items in the space of concepts, entities, and quantities. Then, we cast the inference problem into an efficient algorithm based on random walks over weighted graphs. We give specific consideration to quantities, which we map to a <measure, value, unit> triple over a taxonomy of physical, monetary, temporal, and dimensionless measures. Our experiments with web tables from diverse domains demonstrate the viability of our method and its benefits over baselines.

- We introduce the *quantity alignment problem*: computing bidirectional links between textual mentions of quantities and the corresponding table cells. We propose BriQ: a system for computing such alignments. It supports navigation between explanations in text and details in tables. In addition, it enables advanced content summarization. BriQ copes with the specific challenges of approximate quantities, aggregated quantities, and calculated quantities. In these cases, the align-

ment is more complex than a mere surface form match. We judiciously combine feature-based classification with joint inference using random walks over candidate alignment graphs. Experiments with a large collection of tables from the Common Crawl project demonstrate the viability of our methods.

- We design ExQuisiTe: a web application that identifies mentions of quantities in text and tables, aligns quantity mentions in the text with related quantity mentions in tables, and generates salient suggestions for extractive text summarization systems. ExQuisiTe handles exact single-cell references as well as rounded or truncated numbers and aggregations such as a row or a column totals, and supports user-friendly exploration.

## Kurzfassung

Schemalose Tabellen sind im Internet allgegenwärtig. Dargestellt werden beispielsweise Finanzdaten von Unternehmen, Gesundheitsdaten oder Angaben zur Umweltbelastung durch verschiedene Automodelle. Texte, in denen solche Tabellen eingebettet sind, erklären und beschreiben wichtige Quantitäten in den Tabellen. Obwohl tabellarische Daten weit verbreitet sind, ist weitgehend unerforscht wie diese reichhaltige Datensammlung automatisiert semantisch interpretiert werden kann. Hierzu ist es unerlässlich den Begleittext zu verstehen und die relevanten Informationen zu extrahieren. Dies würde auch die Möglichkeit eröffnen, Fragen zu Quantitäten zu beantworten, zum Beispiel nach ,,Internetfirmen mit einem jährlichen Umsatz von mehr als 5 Millionen US-Dollar" oder ,,elektrischen Autos mit einem Energieverbrauch unter 14 kWh/100km" oder ,,klinischen Studien mit einer Gabe von mehr als 30 mg Gerinnungshemmer täglich".

Um Quantitäten in Tabellen und Text zu verstehen, müssen einige Herausforderungen gemeistert werden. Erstens existiert keine allgemeine Wissensbank über Quantitäten. Zweitens werden Tabellen üblicherweise ohne die Anwendung standardisierter Schemata für den jeweiligen Einzelfall erstellt. Dies betrifft insbesondere die Ausdrucksweise in Kopfzeilen sowie die verwendeten Abkürzungen innerhalb der einzelnen Zellen. Drittens können dieselben Quantitäten auf unterschiedliche Weise, unter Verwendung verschiedener Maßeinheiten, ausgedrückt werden. So sind ,,48 km/h" beispielsweise äquivalent zu ,,30 mph". Und viertens wird bei der Erklärung einer Tabelle im begleitenden Text häufig gerundet oder zusammengefasst, oder auch mit anderen Maßeinheiten gearbeitet.

In dieser Dissertation begegnen wir den beschriebenen Herausforderungen mit folgenden Beiträgen:

- Wir präsentieren die Quantity Knowledge Base (QKB), eine Wissensbank für Quantitäten. Die QKB wird durch den Import von Quellen wie Freebase und Wikipedia konstruiert. In ihr organisieren wir Quantitäten mit Hilfe einer vierstufigen Taxonomie: Dimensionen, Einheiten, Maßangaben und Themenbereiche. QKB erlaubt somit die vereinheitlichte Darstellung von Quantitäten in Form von Tripeln <Maßangabe, Wert, Einheit>.

- Wir stellen Equity vor, ein System zur automatischen Vereinheitlichung der Kopfzeilen und Zellen von Tabellen. Der Tabelleninhalt wird durch Konzepte, Klassen, Entitäten und eindeutige Quantitäten in einer Wissensbank repräsentiert. Hierzu entwickeln wir ein probabilistisches grafisches Modell zur Darstellung der Abhängigkeiten der Tabellenzellen untereinander sowie der Tabellenzellen und der möglichen Objekte im Raum der Konzepte, Entitäten und Quantitäten. Für die Inferenz auf diesem Modell entwickeln wir einen effizienten Algorithmus, der auf Random Walks über gewichteten Graphen basiert. Dabei achten wir besonders auf Quantitäten, die wir als <Maßangabe, Wert, Einheit> Tripel von physikalischen, monetären, zeitlichen und räumlichen Maßen repräsentieren. Unsere Experimente

mit Webtabellen aus verschiedenen Bereichen belegen die Leistungsfähigkeit unserer Methode sowie ihre Vorzüge gegenüber anderen Vergleichsmethoden.

- Wir führen das *Quantitätenzuordnungsproblem* ein, bei dem bidirektionale Links zwischen Quantitäten im Text und Quantitäten in entsprechenden Tabellenzellen ermittelt werden. Zur Berechnung der Links schlagen wir BriQ vor. Dieses System erlaubt es zwischen den Erklärungen im Text und den Details in den Tabellen zu navigieren und ermöglicht potentiell das Generieren von Zusammenfassungen. BriQ kann mit approximativen, aggregierten und umgerechneten Quantitäten umgehen. Die Methode kombiniert maschinelles Lernen für merkmalsbasierte Klassifikation mit unüberwachter algorithmischer Inferenz mittels Random Walks auf geeignet konstruierten Kandidatengraphen. Experimente mit einer großen Sammlung an Tabellen aus dem Common Crawl Projekt demonstrieren die Leistungsfähigkeit unserer Methode.

- Als letzten Beitrag entwickeln wir ExQuisiTe: eine Webapplikation, die Quantitäten in Texten und Tabellen identifiziert, sie im Text mit den dazugehörigen Zellen in Tabellen verbindet und daraus Vorschläge für extrahierende Textzusammenfassung generiert. ExQuisiTe beherrscht zusätzlich zu Quantitäten in einzelnen Zellen auch Aggregationen wie beispielsweise Zeilen- und Spaltensummen oder Differenzen und Verhältnisse zwischen den Werten zweier Tabellenzellen.

## Acknowledgments

It has been a long way, and this achievement was only feasible with the help and support of great people surrounding me.

First, I would like to thank my supervisor Prof. Dr. Gerhard Weikum. His guidance and support were crucial for completing this work. Working under his supervision, taught me how to carry out strong scientific research from the first step of selecting the research problem to the last step of presenting it.

Also, I would like to thank my main collaborator Dr. Mirek Riedewald. He helped me to structure my thought process and improve my scientific writing. I've learned from him many essential skills, that had a great impact on completing this work.

Second, I would like to thank my family, and especially my parents. They are always there for me, providing me with exceptional support and encouragement. I won't be without them, and I'm very grateful to have them.

I would like to thank my friends who always believed in me on times I had doubts about myself.

Finally, I would like to thank all the Max-Planck-Institute family for their continuous support and help. Also, I would like to thank all my colleagues at D5. I've learned a lot from interacting with each one of them.

I'm grateful for so many people in my life who helped me to make this accomplishment.

# Contents

# 1 Introduction

Numbers are an integral part of language, though they are often overlooked by *Natural Language Understanding* (NLU) and *Information Extraction* research. Numerical quantities appear in scientific research results, financial reports, and medical records, among others. They are arranged in tabular formats or infused in natural text. The web contains more than 150 million tables [CHW$^+$08, CHZ$^+$08], and it is estimated that over 40% of columns in web tables contain numerical quantities [SC14]. Tapping into this wealth of resources by understanding quantities in web tables and text surrounding them can lay the first brick in building next-generation Information Retrieval systems that are capable of answering complex queries about quantities. To this end, we pursue answers to the following questions:

- How can we semantically represent quantities?
- How can we disambiguate quantities in web tables and their surrounding text?
- How can we identify relations between mentions of quantities in text and tables?

## 1.1 Motivation

There is a wealth of under-utilized resources on the web including millions of tables [CHW$^+$08, CHZ$^+$08]. These tables hold valuable information about quantities on revenues and costs, the environmental footprint of cars, drug trials and more. Unlike text, tables have a structure and their structure implies certain relations. For example, a row in a table can hold a single entity record, and each column can represent a specific measure of that entity.

**Example 1.1.** *Table with ambiguous column headers, abbreviated names, and unspecified units. Each row specifies within-town and motorway speed-limits for a country.*

| | Within towns | Expressways/motorways |
|---|---|---|
| Germany | 50 (30 residential areas) | No limit (130 advisory) |
| UAE | 50–60 | 100–160 |
| UK | 48 (30 mph) | 113 (70 mph) |
| US | 40–120 (25–75 mph) | 97–129 (60–80 mph) |
| Uruguay | 50 | 90–110 |
| Uzbekistan | 70 | 100 |

Despite the structured nature of web tables they poses a great challenge for interpretation. Web tables are often created in an ad-hoc manner without proper schema design and with highly heterogeneous formats and attribute values. In Example 1.1, the first column has a *missing header* and *abbreviated names*; and the second and third columns hold *ambiguous quantities* without specified units.

This ambiguity limits the reuse of such tables and creates a huge heterogeneity problem when *comparing* or *aggregating multiple tables*. It also implies that table contents can be properly interpreted only in conjunction with the textual explanation surrounding them. Hence, it is important to devise algorithms for: (i) canonicalizing table cells to a *Knowledge Base*; (ii) understanding table contents in the light of their surrounding text and vice versa.

Though some prior work has touched upon entity canonicalization for tables [LSC10, BND15], none handles the canonicalization of quantity mentions. Sarawagi et al. [SC14] addressed quantities, but focused on the specific tasks of searching with numerical values and extracting numerical relations from text [MMM$^+$16].

Long documents with multiple tables are hard to read and navigate. For example, financial documents and environmental reports sometimes span hundreds of pages; with tables spread across the pages or assembled at the end. A user might want to understand quantity mentions in a table by reading the corresponding textual explanation. Also, she might need to drill down into more details about a quantity in the text by looking at the corresponding table(s). Both of these tasks demand the *alignment of quantity mentions* in text and tables.

Prior work has focused on linking mentions of entities in text [SWH15] and tables [LSC10, BND15, RLB15] to a knowledge base. However, the focus has been names rather than quantities, and those works heavily depend on the availability of a knowledge base. None of the prior work focused on aligning mentions of quantities within the same document.

Example 1.2 illustrates the type of possible alignments between mentions of quantities in text and tables. A quantity mention in text can reference two types of quantity mentions in tables:

- *explicit quantities* or *single-cell* quantity mentions such as “`204.3`” and “`121.9`”.
- *implicit quantities* or *composite* quantity mentions computed as an aggregation of one or more table cells such as “20%”, meaning “$percentage(\frac{\texttt{204.3}}{\texttt{587.8}})$”

In addition, a quantity mention in the text can be:

- an *exact* mention of a quantity in the table,
- an *approximation* of a quantity mention in the table.

In Example 1.2, the quantity “`204.3 million tonnes`” is an exact mention and “`122 million tonnes`” is an approximate mention.

**Example 1.2.** *A text snippet accompanied with its related table. The text contains explicit and implicit quantity mentions from the table. The table encloses ambiguous quantities with undefined units and measures. The units are specified in text: "tonnes of CO2 equivalent", "%", and "tonnes", while the measures are specified in the table caption and the last column's header: "Greenhouse Gas Emission by Sector" and "% change".*

> Generating energy is responsible for the biggest single wedge of UK carbon emissions- `204.3 million tonnes` of CO2 equivalent, or `35%` of the total for 2010. Transport is not far behind though- `122 million tonnes` of CO2 equivalent, or `20%` of the total. The biggest part of that? Passenger cars, which generated `68 million tonnes` in 2010.

| | 1990 | 2000 | 2010 | % change, |
|---|---|---|---|---|
| 2000 to 2010 | | | | |
| Energy Supply | 273.4 | 220.1 | `204.3` | -7.1 |
| Business | 113.2 | 111.3 | 89.0 | -20.0 |
| Transport | 121.5 | 126.7 | `121.9` | -3.8 |
| Public sector | 13.1 | 11.6 | 8.5 | -26.7 |
| Residential | 80.8 | 90.1 | 89.9 | -0.3 |
| Agriculture | 63.1 | 58.0 | 50.7 | -12.7 |
| Industrial Process | 54.4 | 24.6 | 10.9 | -55.7 |
| Land use | 3.9 | 0.4 | -3.8 | -1129.5 |
| Waste Management | 45.9 | 29.3 | 16.5 | -43.7 |
| Grand Total | 769.4 | 672.0 | `587.8` | -12.5 |

Greenhouse Gas Emission by Sector

Identifying mentions of implicit and approximate table-quantities in the text is challenging. A simple surface form match will not work in such cases. In addition the search space for composite quantity mentions can become exponential in the table size. All of these challenges highlight the need for efficient algorithms for canonicalizing and aligning quantity mentions in text and tables.

Our work on canonicalizing quantity mentions can facilitate comparing multiple tables and fusing their data towards analytic insights. Also, it can support a new generation of search engines capable of answering complex queries such as "Internet companies with annual income above 5 Mio. USD" and "electric cars with energy consumption below 100 MPGe". Aligning quantity mentions in text and tables can support faster navigation between explanations in text and numerical figures in tables. In addition, it can aid in summarizing long and complex documents.

## 1.2 Challenges

**Absence of Knowledge Base Support for Quantities:** Grounding quantity mentions requires a specialized knowledge base for quantities. Entity knowledge bases such as Yago[1] do not support quantity-specific concepts such as units and measures. Other knowledge bases, such as Wikipedia[2], DBpedia[3], and Freebase[4] have limited support for units and measures. They have good coverage of simple units such as "meters" and "miles", but *low coverage* for complex units such as "miles per gallon (mpg)". More *complex ontologies* such as *QUDT*[5] are hard to use and integrate. Therefore, we construct a specialized knowledge base for quantities with high coverage of units and measures and less complex ontology.

**Heterogeneity of Web Tables** Web tables are typically created in an ad-hoc manner with human users in mind. Thus, they pose a variety of challenges regarding heterogeneity and incompleteness as follows:

- Tables do not follow a predefined schema and contain heterogeneous attributes and values.
- Column headers are sometimes missing or abbreviated such as the header of the first column in Example 1.2.
- Table cells contain abbreviated names, such as "UAE" and "US" in the first column in Example 1.1.
- Tables sometimes have underspecified quantity mentions with missing units or measures.

**Ambiguity of Quantity Mentions:** We can write a single quantity in multiple forms and units of measures. For example, one can express the in-town speed limit in the UK as "48 km/h" or "30 mph" as in Example 1.1. Though both quantities are equivalent, their surface forms are different. A mention like "120m" can represent the height of a building in meters, the duration of a movie in minutes, or even 120 million monetary value with missing currency (unit). Though these quantities are different, their surface forms are the same. The following points characterize the challenges of quantity mention ambiguity:

- Abbreviating units and measures in quantity mentions.
- Varying precision points and scales in quantity mentions' value.
- Expressing quantity mentions in different unit systems.

[1] http://yago-knowledge.org

[2] http://wikipedia.org

[3] https://wiki.dbpedia.org

[4] http://freebase.com

[5] http://www.qudt.org

This renders quantity mentions ambiguous, and it urges the need for grounding quantity mentions in a knowledge base.

**Quantity Alignment:** Aligning quantity mentions in text and tables is more complex than a mere surface form match. The alignment problem poses the following challenges:

- The use of approximations when referencing table-quantities in text, such as"`122 million tonnes`" in Example 1.2.
- Quantity mentions in the text can use a different scale from the table, such as "200,000" vs "0.2 million".
- The use of aggregate quantity mentions in text, such as "`35%`" in Example 1.2.
- The search space of the possible alignments is exponential in the size of the table and as large as the number of possible subsets of cells.

Hence, one needs to employ more sophisticated techniques to align quantity mentions in the text to their corresponding quantities in tables.

## 1.3 Contributions

### 1.3.1 QKB: A Knowledge Base for Quantity Mentions

We construct a *Quantity Knowledge Base (QKB)* to counter the absence of knowledge base support for quantities. QKB provides the semantic space to ground quantity mentions in tables and text. QKB is constructed by importing and combining data from Freebase and other sources. This data is restructured into a light-weight taxonomy that supports the representation of quantity as a triple $\langle measure, value, unit \rangle$ where

- The $m$easure is a name referring to a certain quantifiable aspect of an object or process (e.g., the height of a building, the power of a car's engine).
- The $v$alue is a numerical literal.
- The $u$nit is a defined and widely used magnitude of a quantity, such as meter, kg, Watt, kWh, USD, EURO, etc.

For each quantity, QKB also keeps a set of alias names for the measure, a regular expression for feasible surface forms of value and unit, and conversion rules for units. Our light-weight taxonomy covers physical, monetary and temporal measures and also unit-less numbers like ratios, rates, counts, and scores. The QKB is publicly available and *Equity* – our web tables disambiguation framework that is presented in the following section – relies on it for grounding quantity mentions.

### 1.3.2 Equity: Disambiguation of Entities and Quantities in Web Tables and Surrounding Context

To overcome the heterogeneity of web tables and the ambiguity of quantity mentions, we propose *Equity:* a framework to fully canonicalize mentions in ad-hoc tables and their surrounding contexts to a knowledge base. It exploits the textual context surrounding a table to jointly link names and numerical quantities from both tables and text. It canonicalizes mentions of entities, classes, concepts, and quantities as follows:

- Quantity mentions are linked to a specialized knowledge base, the QKB.
- Entity mentions are linked to entities in Yago.
- Class mentions in table headers are linked to classes of entities in Yago.
- Concepts mentions in table headers are linked to concepts in Wikipedia, because Yago does not support a notion of concepts.

Equity employs a Markov Random Field (MRF) model, distantly supervised by relatedness measures from a KB. Then, it drives a reduced acyclic MRF model, for which it casts the inference into an efficient algorithm based on random walks over normal weighted graphs. Equity results were presented at CIKM 2016 [IRW16].

### 1.3.3 BriQ: Understanding the Relation Between Quantity Mentions in Text and Tables.

To align quantity mentions in text and tables, we introduce *BriQ*: a framework that computes bidirectional linking between textual mentions of quantities and the corresponding table cells. BriQ is designed to cope with the specific challenges of approximate quantities, aggregated quantities, and calculated quantities in the text that are common but cannot be directly matched in table cells.

The BriQ algorithm consists of two main stages: local resolution and global resolution. The *local resolution* assigns a confidence score for each candidate alignment in isolation without considering other neighboring alignments. The *global resolution* then takes as input the candidate alignments from the previous stage and outputs the final alignment of quantities between text and tables. It uses the local resolution's confidence score as prior weights and employs a global inference algorithm based on random walks over graphs to resolve the alignments.

BriQ can handle a broad range of aggregation functions, such as sum, average, difference, percentage, and change ratio. Thus, it can support advanced content summarization and faster navigation between explanations in text and details in tables.

We conducted an extensive user study for annotating web pages to train BriQ. The annotated dataset is publicly available; we presented it along with BriQ at ICDE 2019 [IR$^+$19].

### 1.3.4 ExQuisiTe: A Tool for Explaining Quantities in Text

To prove the viability of our proposed models we introduce `ExQuisiTe`, a tool that applies our proposed algorithms to help users perform the following:

- identifies mentions of quantities in text and tables,
- aligns quantities in the text to their relevant quantities in tables,
- and generates salient suggestions for *Extractive Text Summarization (ETS)* systems.

The ExQuisiTe allows users to perform these task through an easy to use web interface. We presented ExQuisiTe at WWW 2019, and the source code is publicly available at `https://www.mpi-inf.mpg.de/briq/` .

## 1.4 Publications

Preliminary results from this work, for which I'm the main author, have been published in :

- CIKM 2016 [IRW16]
- ICDE 2019 [IR$^+$19]
- WWW 2019 [IW19]

## 1.5 Organization

This thesis is organized as follows: in Chapter 2 we lay the thesis background and discuss the related work. Chapter 3 discusses the QKB and its hierarchy. Then, in Chapter 4 we explain the Equity system for disambiguating entities and quantities in web tables and its surrounding context. In Chapter 5 we explain the BriQ system for the alignment of quantity mentions in tables and text. In Chapter 6 we present the ExQuisiTe application for smart navigation of documents and enabling extractive summarization that considers table contents. Finally, Chapter 7 concludes the thesis and emphasizes the future research directions.

# 2 Background & Related Work

This chapter provides the basics for this thesis. It presents the main concepts and algorithms we build on and discusses the related research areas and the current state-of-the-art techniques. The aim of this chapter is to position this thesis within the relevant research areas.

## 2.1 Knowledge Bases

In our context, a *Knowledge Base (KB)* is a repository of structured data about factual knowledge that computers can consume. A knowledge base can cover a general or a specific domain. General knowledge bases focus on the breadth of knowledge by modeling a broad range of world knowledge. Specialized knowledge bases focus on the depth of knowledge by modeling specific domain knowledge, such as the *biomedical*, the *financial*, or the *enterprise* domains.

General knowledge bases model facts about real-world *entities* and *concepts* along with their *attributes*, *classes*, and *relations*. These terms are defined as follows:

**Definition 2.1 (Entity).** *An entity $E$ is a thing that exists and can be uniquely identified and distinguished.*

Examples of entities are Angela Merkel (Person), Google LLC (Company), The Nile (River), and La Tour Eiffel (Monument).

**Definition 2.2 (Concept).** *A concept $C$ is a term that coins an abstract notion.*

Example concepts are Knowledge Base, Programming Language, and Physical Dimensions: Length, Height, and Width.

**Definition 2.3 (Attribute).** *An attribute $A$ is a property of an entity $E$ or a concept $C$ and it takes a value from a defined range $\Sigma_A$.*

Example attributes of a Person entity are Name, Date of Birth, and Salary. Example attributes of a Dimension concept are Unit of Measure (e.g., kilogram and meter ) and Thematic Domain (e.g., physical and financial).

**Definition 2.4 (Class).** *A class $C$ is a set of entities with common attributes. Classes of entities are organized into a hierarchy exhibiting that each class inherits all of the attributes of its ancestor(s).*

Example classes of entities are Companies, Actors, and Mammals.

**Definition 2.5 (Relation).** *A relation connects two entities and it can have a direction, such as* Parent-of, *or be bidirectional, such as* Spouse.

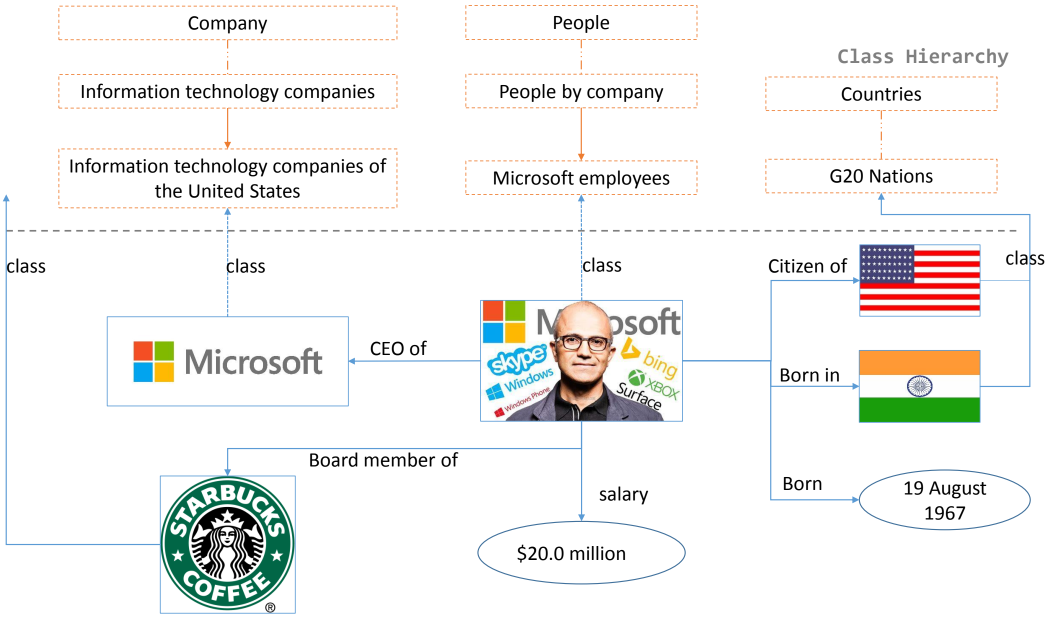


Figure 2.1: A snippet of a Knowledge Base represented as a Knowledge Graph

### 2.1.1 Knowledge Base Representation

Knowledge bases model world knowledge in a format that is consumable by computers. Therefore it can automate several applications, such as semantic annotation [HYB$^+$11], question answering [YBE$^+$12], inference and reasoning about data [NMTG16].

Facts in knowledge bases are stored as triples of *subject, predict, and object (SPO)*. For example, {S: `Sundar Pichai`, P: `CEO of`, O: `Google`} and {S: `Daniel Radcliffe`, P: `Acted in`, O: `Harry Potter and the Philosopher's Stone`}. These facts are usually cast into a *Knowledge Graph (KG)*, in which each entity or concept is represented by a node in the graph; and each relation is represented by a directed edge between entities or concepts.

**Example 2.1.** *Figure 2.1 shows a snippet of a knowledge graph with:*

- Entities: *`Satya Nadella`, `Microsoft`, `Starbucks Coffee`, `India`, and `United States`.*
- Attributes: *`Salary`, `Born`*
- Classes, *such as `Companies`, `Countries`, and `People`.*
- Relations: *`CEO of`, `Born in`, `Board Member of`, and `Citizen of`.*

*Some of the triples in the KG in Figure 2.1 are as follows:*

- *{S: `Satya Nadella`, P: `Born in`, O: `India`},*

- {*S:* `Satya Nadella`, *P:* `CEO of`, *O:* `Microsoft`},

- {*S:* `Satya Nadella`, *P:* `Board Member of`, *O:* `Starbucks Coffee`}.

### 2.1.2 Knowledge Base Construction

A knowledge base can be automatically constructed or manually curated by experts. Automatically constructed knowledge bases employ *Information Extraction(IE)* techniques, which we will introduce in the next section, to extract high confidence factual knowledge. Hence, the quality of the facts in automatically constructed knowledge bases heavily depends on the extraction method. Examples of automatically constructed knowledge bases are: DBpedia [ABK$^+$07], Freebase [BEP$^+$08], and Yago [SKW07]. Manually curated knowledge bases require domain experts to model and represent knowledge. Hence, the manually curated facts in these knowledge bases tend to be accurate and have high confidence. Examples of manually curated knowledge bases are WordNet [Mil98] and Cyc [Len95]. Though manually curated knowledge is more accurate than automatically constructed knowledge, it has less coverage and requires tremendous manual effort.

### 2.1.3 Numerical Values in Knowledge Bases

A KB entity can have numerical attributes, such as age and salary for a person; or height for a building. Nevertheless, KB numerical attributes are limited in coverage and lack the semantic modeling of measures and unit systems. In Example 2.1, the salary attribute has a value of “20.0 million”; however, this value is a string that has no associated semantics.

On the other hand, complex ontologies of quantities and measures, such as $QUDT$[1], are impractical to integrate into real-life applications. Hence, there is a need to construct a simple quantity knowledge base with high coverage that can be easily integrated into real-life applications. Chapter 3 discusses how to overcome the absence of semantic representation of numerical values by developing a simple Quantity Knowledge Base(QKB). The QKB captures the semantics of units of measures and is aware of their various representation in text.

**Example 2.2.** *Figure 2.2 illustrates the semantic representation of the quantity salary with value \$20.0 million using QKB.*

## 2.2 Information Extraction

Constructing a knowledge base is only realized through efficient Information Extraction techniques. The advance in IE has a direct influence on the quality and coverage of knowledge bases. This section presents an overview of the IE research field and its recent progress.

[1] `http://www.qudt.org/`

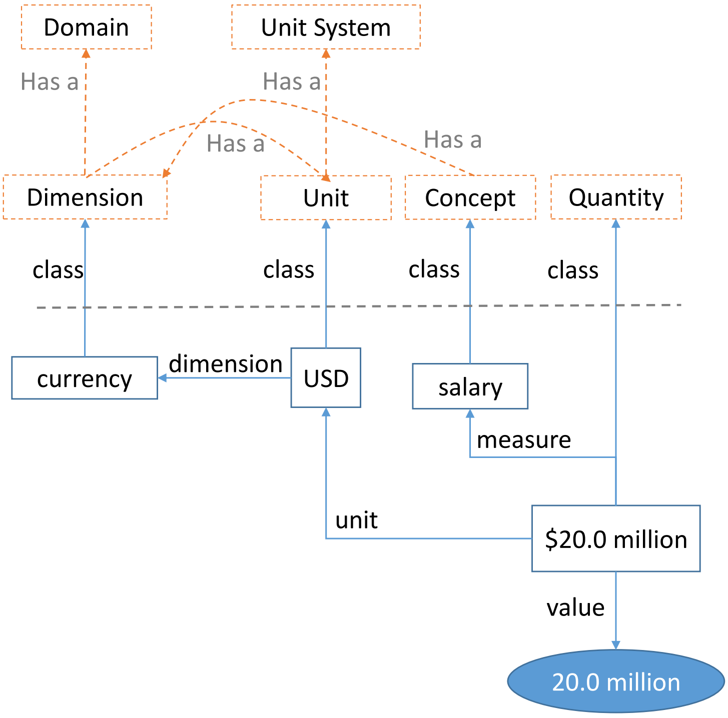


Figure 2.2: A snippet of Quantity Knowledge Base represented as a graph.

Information Extraction aims at extracting structured knowledge from unstructured data. It harnesses non-structured or semi-structured resources, such as *Wikipedia infoboxes or article texts*, to extract high confidence factual knowledge and represent it in a structured form that is machine-readable. It exploits *Natural Language Processing (NLP)* techniques to process unstructured textual resources. Information Extraction can be classified under the following main dimensions (a more fine grained taxonomy is given in [Sar08]): sources of extraction, target of extraction, and the methodology of extraction.

## 2.2.1 Sources of Extraction

The source of extraction differentiates between the type of resources the IE techniques exploit to harvest structured information. In this thesis, we differentiate between two main sources of data:

- *Non-structured sources*, such as natural language text in web pages or enterprise documents. These sources are difficult as they are often ambiguous and lack structure cues.
- *Semi-structured sources*, such as web forms, Wikipedia infoboxes, and web tables. These sources have a structure that implies certain relations between the content.

However, they do not follow a standard schema and contain abbreviations and incomplete snippets.

### 2.2.2 The Target of Extraction

The target of extraction differentiates IE systems based on their end goal. The following are the common extraction tasks as defined by [JM09]:

- *Entity Extraction*, also known as *Named Entity Recognition (NER)*, aims at finding mentions of named entities in text and labeling them with their corresponding classes. The classes of named entities are application specific. Examples of entity classes are persons, organizations, financial assets, genes or protein names.

  The challenges in NER are to find the span of the entity mention and to identify its type. *Conditional Random Field (CRF)* models have proven effectivness in NER; see for example the work of [ML03] and the widely used Stanford NER Tagger [FGM05]. Recently, neural sequence tagging models using Bidirectional Long Short-Term Memory (BiLSTM) prove effectiveness in recognizing entity mentions [LBS+16].

  Once the mentions of entities have been recognized, they can be grouped in sets corresponding to real-world entities. *Named Entity Disambiguation* and *Coreference Resolution*, which we will introduce in Section 2.3, are the tasks responsible for linking mentions of entities to real-world entities. For example, the mentions "Barack Obama" and "Obama" will be linked together to the set corresponding to the entity `Barack Obama`

- *Relation Extraction* aims at finding and classifying semantic relations between entities or concepts. The most common type of relations is binary relations, such as parent-of, citizen, employer, and president. These relations are represented as SPO triples of subject, predicate, and object; for example {S: `Microsoft`, P: `employer of`, O: `Satya Nadella`} and {S: `Satya Nadella`, P: `citizen of`, O: `United States`}. Recent relation extraction systems have also addressed higher-arity relations [MSB+12, MSB13, DCG13, ESW18].

  Numerical relation extraction received attention from the research community as well [MMM+16, SPM17]. This research direction has focused on extracting SPO triples at which the object is a numerical value, such as {`Germany`, `population`, `83 million`} and {`China`, `Inflation Rate`, `2.5`%}.

- *Event Extraction* aims at finding events in which entities of interest participate. Most event mentions correspond to verbs, such as: took a flight, conducted a concert, played in the Olympic Games.

- *Temporal Expression Extraction* aims at finding when an event took place [SG16]. Temporal expressions have different levels of granularity, such as decades, years, weekdays, times, and dates. Temporal expressions can be absolute, such as 1 April 2019 and the 4th quarter of 2018; or relative, such as tomorrow, last week, or two years ago. Moreover, they can represent a duration, such as 1 hour, 2 weeks, or 5 years. Thus, temporal expressions are different from numerical quantities and have different semantic attributes.

HeidelTime [SG15] and SUTime [CM12] are examples of systems for temporal expressions extraction. They recognize and normalize temporal expressions in textual documents; then they annotate documents with TIMEX3 standards [2].

### 2.2.3 The Methodology of Extraction

The methodology of extraction differentiates IE systems based on the algorithmic technique of extraction. The two main dimensions, presented in [Sar08], under which the method of extraction falls are:

- *Hand Crafted vs. Learned Rules:* Domain experts can define a set of extraction heuristics that the IE system follows to extract structured information. In this case, the extraction is limited to the set of handcrafted rules and the IE system requires a significant amount of manual labor to define and maintain these rules [Ril93, ARHB$^+$93].

  On the other hand, rules can be learned automatically from a manually annotated corpus. In this case, domain experts are also needed; to first build the machine learning model for learning the rules, and second to annotate the corpus [CM99, Sod99, Ait02]. In both cases, human experts are needed to craft extraction rules, design and model the learning space, or annotate data.

- *Rule-based vs. Statistical:* Rule-based extraction relies on logical rules to extract information, whereas statistical extraction relies on the assertion signal from repeated predicates. Rule-based systems are easier to interpret and are best suited for closed domains at which human experts are available to define the rules [CMBT02, SDNR07]. On the other hand, statistical methods are good for open-ended domains, such as web extraction [BCS$^+$07, CBK$^+$10, NZRS12].

## 2.3 Related Tasks

In this section we will discuss the tasks most closely related to this thesis and their state-of-the-art techniques. In the following part we will discuss Named Entity Disambiguation, Coreference Resolution, and Table Canonicalization.

### 2.3.1 Named Entity Disambiguation

After recognizing mentions of entities in the text we can map them to real-world entities in a KB. This task is known as *Named Entity Disambiguation (NED)*, and it aims at resolving ambiguous mentions of entities to a standard representation in the KB. NED systems can link textual mentions of entities in the text to Wikipedia pages (in this case,

[2] http://www.timeml.org

it is called *Wikification* which was first coined by [MC07]) or to a knowledge base such as Yago, DBpedia, and Freebase.

**Definition 2.6 (A Surface Form or A Mention).** *A surface form or a mention is a single word or a sequence of words that corresponds to an entity in the knowledge base.*

**Definition 2.7 (Candidate Entity).** *For a given surface form, a candidate entity is a knowledge base entity that has a similar surface form.*

**Definition 2.8 (Anchor Text).** *In hypertext, an anchor text is the visible word or sequence of words representing* a hyperlink.

**Example 2.3.** *A sentence taken from a news article about Michelle Obama's new book "Becoming".*

> `Obama` emphasizes how important role models are, especially for young women of color in a culture that isn't changing fast enough.

Example 2.3 shows an ambiguous surface form "Obama". It can refer to the candidate entity `Barack Obama` or the candidate entity `Michelle Obama`. This ambiguity can be resolved using contextual cues from the text.

### NED Dictionaries

NED systems rely on dictionaries of entities to find candidate entities for the surface forms in the text. These dictionaries map entities to possible surface forms with associated weights which score the confidence of the mapping. The quality and completeness of surface-form/entity pairs in a dictionary directly affect the quality of NED algorithms. Too noisy dictionaries result in spurious matches and low precision; incomplete dictionaries result in more misses and low recall. Thus, it is crucial to construct high-quality high-coverage dictionaries. Wikipedia is the primary source for constructing NED dictionaries because of its high-quality manually curated content.

In most NED dictionaries [CSMA16], entities correspond to Wikipedia pages and their surface forms are collected from the following sources:

**Wikipedia Page Title:** the title of the Wikipedia page corresponding to the entity is a possible surface form, such as "United States".

**Redirect Pages' Title:** in Wikipedia, a redirect page is solely created to point to the original Wikipedia page. The titles of redirect pages are common *noun phrase(NPs)* used to refer to an entity, such as "United States of America"

**Disambiguation Pages:** in Wikipedia, a disambiguation page is created to list all the entities having the same *ambiguous* surface form. For example, the disambiguation page for the surface form "America" contains the following Wikipedia pages: `The Americas`, `America (Julio Iglesias album)`, and `America, Illinois, U.S.`

**Anchor Text:** Wikipedia href anchors that link to other Wikipedia pages are common surface forms of the entity represented by this page, such as the "US" and "U.S.".

Each surface-form/entity pair is weighted by the number of times the surface form is used to refer to the entity.

### NED Input Type

NED algorithms differ according to their types of input. NED algorithms have been designed for long textual documents, short text or social media text, and web tables.

Long documents, such as news articles and Wikipedia pages, received the most attention from the research community [Cuc07, MW08, HYB$^+$11, KSRC09, RRDA11, SWH15]. These documents tends to be long, well-formed, and with sufficient context. [BP06] was the first to introduce a system to disambiguate mentions of entities in a document to Wikipedia entities. [Cuc07] was the first system to perform both entity recognition and disambiguation. It employs a joint disambiguation algorithm that accounts for the category agreement of entities in the same document.

With the rise of social media, it was important to design NED algorithms that handle this type of input. Social media text is more challenging than news articles and Wikipedia pages, as it is short, informal, and has insufficient context. Entity mentions in social media are cryptic and difficult to disambiguate. Hence, there was a need to design NED algorithms for social media and short text, such as [GCK13, IAYW].

The algorithm in [GCK13] jointly solves the recognition and disambiguation tasks on tweets. It starts with extracting all the possible k-grams matching the surface form of at least one entity. These k-grams designate the set of candidate entity mentions. Then, for each candidate mention the method generates a set of candidate entities including the "NIL" entity, where a candidate mention mapped to "NIL" indicates an out of knowledge base entity.

The algorithm in [IAYW] includes cues from user profiles, hash-tags, other similar tweets, and external web pages to enrich the context of the tweet. It starts with normalizing the mentions in tweets to an intermediate normal from, then finding candidate entities for each normalized mention. It uses a notion of temporal importance in addition to other features to find the best matching entities.

For web tables, several algorithms have been proposed to canonicalize table cells to a KB which we will discuss in Section 2.3.3.

### NED Features

The most widely used features in NED are as follows:

**Surface Form Similarity:** is a string-based similarity between the mention in text and the possible surface form of a candidate entity.

**Context Similarity** is the similarity between the context of the mention in a document and the context of the KB entity. In [HYB$^+$11], the context of the entity is the set of *keyphrases* extracted from the corresponding Wikipedia page. The context similarity between the mention and the candidate entity is then computed using an aggregated weighted score.

**Coherence Measures:** are used to estimate the homogeneity of entity mentions in a document. It is usually computed using pair-wise entity relatedness measures between candidate entities. For example, Wikipedia link structure is used in [HYB+11] to estimate the pair-wise relatedness of entities. Recently, *Vector Space Similarity* is used to estimate the relatedness between entities using their entity-embedding vectors as in [ZSG16].

**Importance Measures:** are used to estimate the prior importance of an entity. For example, in [RRDA11] entity importance is estimated using the fraction of Wikipedia pages referring to this entity, and in [IAYW] a temporal entity importance measure is proposed.

### NED Algorithms

Named Entity Disambiguation can be jointly performed with Named Entity Recognition as in [DK14, NTW16], which use a CRF model for the recognition and the disambiguation tasks.

Otherwise, NED exploits existing NER systems' output to identify mentions of entities as in [HYB+11, RRDA11, KSRC09]. These algorithms disambiguate entities using similarity measures between mentions in text and entities in a knowledge base, in addition to the coherence between all entity mentions in a document. They employ joint disambiguation algorithms that collectively resolve all mentions of entities in a document. For example the AIDA system, [HYB+11] uses a combination of the prior probability of an entity mention, the context similarity between mentions and KB entities, and the overall coherence of entity mentions to disambiguate mentions of entities to their canonical entities in the Yago KB. This method casts the disambiguation problem into a weighted graph model and solves it by finding a dense subgraph with the best mapping of mentions to KB entities. [KSRC09] casts the joint disambiguation problem into an *Integer Linear Program (ILP)* objective function and then relaxes it to an LP problem. [RRDA11] uses local and global features to train two *Support Vector Machine (SVM)* models; the first model is a Ranking Linear SVM to estimate the coefficients of the features to rank candidate entities; and the second model is a Linear SVM Classifier to decide if linking the mention to "NIL" improves the objective function or not.

Graph algorithms, such as *Random Walks with Restarts (RWR)*, fit nicely with the nature of the joint disambiguation problem. Hence, several NED algorithms exploit graph algorithms [HYB+11, GB14, PHG15, ZSG16]. These algorithms transform the disambiguation problem into the graph-space, at which the candidate entities are nodes in the graph. Pair-wise entity relatedness edges connect entity nodes in the graph, such that the resulting set of entity-mappings is coherent. The weight of the edges is computed using several similarity measures, and recently vector space similarity between entity-embedding vectors is used to estimate pair-wise entity relatedness as in [ZSG16].

### 2.3.2 Coreference Resolution

When *noun phrases (NPs)* refer to the same entity, they are called coreferent. Coreference Resolution is the task of finding NPs that refer to the same entity and constructing coreference chains.

> **Example 2.4.** *A sentence taken from a news article about Michelle Obama's new book "Becoming".*

> In her new memoir, "Becoming" — a book whose reportedly enormous advance rendered its contents almost as closely guarded as the bullion at Fort Knox — `Michelle Obama` puts to rest any speculation about her political ambitions.

A NP referring back to a preceding NP is called an *anaphora*; the referring NP is called an *anaphor*, and the preceding NP is called the *antecedent.* In Example 2.4 the second "her" is an *anaphor* and its *antecedent* is `Michelle Obama`. A NP referring forward to a succeeding NP is called *cataphor*, the referring NP is called the *cataphor*, and the subsequent NP is called the *postcedent.* In Example 2.4 the first "her" is a *cataphor* and its *postcedent* is `Michelle Obama`. Hence, the first "her",`Michelle Obama`, and the second "her" compose a coreference chain.

Coreference resolution is well studied in computational linguistics and discourse analysis, and it can be categorized according to the learning algorithm into Supervised, Unsupervised or Semi-supervised [Ng10]. There are three main coreference resolution models: mention-pair model, entity-mention model, and ranking model.

#### Mention-pair Model

This model trains a binary classifier to decide whether a pair of Noun Phrases (NP) are coreferent or not. The model was first introduced in [AB95] and [ML95]; both use a decision tree classifier. The model falls short of identifying coreferential chains as the transitivity rule is not enforced. For example, if $np_1$ is coreferent with $np_2$ and $np_2$ is coreferent with $np_3$, then $np_1$ should be coreferent with $np_3$; and hence a coreference chain $(np_1, np_2, np_3)$ exists. Therefore, a partitioning algorithm has to be employed to identify such coreference chains.

#### Entity-Mention Model

This model represents each entity as a cluster and the features of the cluster are that of the entity it represents and all the NPs in the cluster. Hence, the decision made in this model is whether a NP is coreferent with another entity (cluster) or not. In this case, a training example consists of a preceding cluster $c$ and a NP $n$ with a binary label indicating the coreference relation. These models did not show significant improvement over the mention-pair models. For example, in [LIJ$^+$04] it does not achieve any improvement over the mention-pair models. Also, in [YSZT04, YSL$^+$08] the entity-mention models introduced only a slight improvement over the mention-pair models.

### Ranking Model

This model selects the most probable antecedent for a given anaphor. Ranking algorithms were first used in [CBD94] with a binary classifier. Recent work applied more advanced ranking techniques to rank more than two candidates at a time. Also, other ranking techniques were proposed to rank clusters (entities) rather than a single mention as in [RN09].

### Rule-based Coreference Resolution

Rule-based methods are more effective than supervised algorithms. Stanford coreference resolution system presented in [RLR$^+$10] is a rule-based system for clustering entities. The model uses carefully-engineered features in successive stages starting from high to low confidence features. The algorithm consists of seven phases, where each phase resolves coreferent mentions based on a set of rules. Surprisingly, this simple algorithm outperforms the state-of-the-art supervised algorithms, which suggests that coreference resolution can be targeted with simple algorithms and well engineered features and rules [HK09, NC02].

### Neural Coreference Resolution

This model is competitive with the current state-of-the-art coreference resolution algorithms. Clark and Manning [CM15, CM16b, CM16a] successfully applied deep learning in coreference resolution. They split the problem into three main parts: mention-pair classification, mention-pair ranking and cluster ranking (entity-centric). They proposed three different algorithms. Two of them are based on deep learning [CM16a, CM16b] and the third is based on agglomerative clustering [CM15].

## 2.3.3 Web Table Search and Canonicalization

Recently, the research community has paid attention to web tables, and several algorithms have been proposed to: process tables for search purposes and schema completion [CHW$^+$08, CHM11, MAAH09, PS12, SC14], fusing tables for data augmentation [YGCC12, ZC13, GS09, CHK09], and canonicalizing table contents to a KB [LSC10, BND15, RLB15].

### Table Search

Cafarella et al. were the first to bring attention to web tables in the seminal work presented in [CHW$^+$08, CHM11]. They explored the problem of searching a huge corpus of tables encompassing 14.1 billion HTML tables extracted from English documents. They used an extraction filter based on [CHZ$^+$08] with low-recall and high-precision to extract 125 million high-quality relational tables. Then, they computed attribute correlation statistics from table headers and populated *attribute statistics database (ACSDb)* which they used later for schema auto-completion.

Sarawagi et al. explored table search in [PS12, SC14]. In [PS12] a 25 million tables corpus was crawled from the web and for each table, the header and a set of weighted phrases from its web page was extracted. The weights of these phrases are computed using the DOM tree of the web page and the frequency of the `HTML` tags. Then, for a given query the algorithm finds all matching tables, extracts table columns that are relevant to the query, and consolidates these columns into a single answer table. It uses a probabilistic graphical model to jointly match query terms to relevant columns and map these columns to an answer table. The main contribution is the graphical model for mapping query terms to relevant table columns.

The model presented in [SC14] focuses on quantity queries on web tables. To this end, it annotates the table columns with units using *a probabilistic context-free grammar (PCFG).* For a given query, it infers the type of the requested quantity $a_q$ and its associated entity $e_q$ and retrieves a set of candidate tables for the query. Then, the algorithm extracts relevant snippets from the candidate tables based on the quantity type $a_q$ and the entity $e_q$. The answer to the quantity query is a distribution of values or a single ranked point value. In this model, the semantic annotation of table columns is solely driven by the search query, that no “hard” labels are assigned to table columns. This serves the purpose of the search framework, but it does not resolve the table canonicalization problem.

### Table Fusion

Aggregating data across multiple tables is explored in [YGCC12, ZC13]. The target of these research is to build a table augmentation framework that is capable of finding direct and holistic matching tables for a given query table. First, the framework finds direct matching tables to the given query table using traditional schema matching techniques and content overlap. Then, it constructs a graph with nodes representing individual tables and edges connecting directly matching tables with a weight corresponding to the matching score. It employs *Topic Sensitive Page Rank (TSP)* algorithm [Hav02], to find holistic matching tables for the query table. [ZC13] complements the work in [YGCC12] by labeling table columns with unit, scale, and timestamps. This work is driven by table search and data augmentation and is not concerned with table cells' canonicalization. On the other end of the spectrum, [GS09] employs a CRF model to augment a query table with semi-structured lists on the web.

### Table Canonicalization

Canonicalizing table columns and cells to a KB is explored in [LSC10, BND15, RLB15]. [LSC10] maps table rows to entities, columns to attributes, and column-pairs to relations in a KB. It models the problem with a probabilistic graphical model and perform message passing algorithm to make the inference. However, it did not consider quantities and their related concepts. [BND15] also uses a probabilistic graphical model to canonicalize table cells, but it only considers linking entity mentions in table cells with KB entities. On the other hand, [RLB15] canonicalizes table headers and rows to concepts and entities in DBPedia, but it only focuses on a small subset of relational tables.

The algorithm introduced in [RMKS15] assigns semantic labels, i.e. attribute names, to table columns based on both the column headers and contents. It learns a semantic labeling function using a labeled corpus from which it extracts attribute names and their corresponding set of values. It considers attributes with string and numerical values. In the case of string columns, the algorithm finds the target attribute using a bag-of-words model. In the case of numerical columns, it uses statistical hypothesis testing to determine if the set of values in the column and the candidate attribute come from the same distribution. However, it considers the whole column and does not consider canonicalizing individual table cells. Thus, this method will fail in handling columns with heterogeneous content.

## 2.4 Related Algorithms

### 2.4.1 Random Walks on Graphs

*Random Walks(RW)* on graphs can measure the similarity between two vertices based on their connectivity. A random walk operates on a directed weighted graph $G = \{V, E\}$, where $V$ is the set of vertices and $E$ is the set of edges in $G$; each edge $e_{ij} \in E$ connects two vertices $(v_i, v_j)$ with a weight $w_{ij}$. The weights on the graphs are normalized such that for each $v_i \in V$ the sum of the outgoing edge weights is equal to one: $\sum_j w_{ij} = 1$.

A random walk starts at a vertex $v_{i_0} \in G$ and randomly moves to another vertex $v_{i_1}$ in the neighborhood of $v_{i_0}$ with a probability equal to the weight of the edge $w_{i_0,i_1}$. This random process is repeated at each vertex, generating a path of random visits. This path connects the start vertex $v_{i_0}$ with all the vertices on the path and hence it can measure the similarity between $v_i$ and other vertices in $G$.

Random walks can be modeled as a Markov chain [Nor98] with state-space $I = \{V\}$ and transition probabilities:

$$P(S_{t+1} = j \mid S_t = i) = p_{i,j} = w_{i,j}$$

For simplicity we refer to a vertex $v_i$ in the graph by its index $i$.

The transition matrix $P \in \mathbb{R}^{n \times n}$, where $n$ is the number of vertices, is called a stochastic matrix; meaning that $p_{i,j} \geq 0$ and for all $i, j \in I$, we have $\sum_{j \in I} p_{ij} = 1$.

A Markov chain has an initial distribution $\lambda$ over the state-space $I$, where $\lambda$ is a row vector $\lambda = (\lambda_i : i \in I)$ , $0 \leq \lambda_i \leq 1$ and $\sum_i \lambda_i = 1$.

We call $S_t$ a Markov chain with initial probability distribution $\lambda$ and transition matrix $P$ if for all $t \geq 0$ and $i_0, i_1, .., i_t, i_{t+1} \in I$,

$$(i) P(S_0 = i_0) = \lambda_{i_0};$$
$$(ii) P(S_{t+1} = i_{t+1} \mid S_0 = i_0, S_1 = i_1, .., S_t = i_t) = P(S_{t+1} = i_{t+1} \mid S_t = i_t) = p_{i_t, i_{t+1}}$$

Then, the probability of a certain state transition sequence $S_t$ (path) is given by:

$$P(S_0 = i_0, S_1 = i_1, .., S_t = i_t) = \lambda_{i_0} p_{i_0,i_1} p_{i_1,i_2} ... p_{i_{t-1},i_t}$$

**Definition 2.9.** *A Markov chain or a transition matrix $P$ is called irreducible, if for every two distinct states $i, j \in I$: $j$ is reachable from $i$, written as $i \to j$, and $i$ is reachable from $j$, written as $j \to i$.*

**Definition 2.10.** *A state i is aperiodic if $p_{ii}^{(n)} > 0$ for all sufficiently large n.*

**Lemma 2.1.** *Suppose P is irreducible and has an aperiodic state i. Then, for all states $j$ and $k$ $p_{jk}^{(n)} > 0$ for all sufficiently large n. In particular, all states are aperiodic.*

**Theorem 2.1 (Stationary Distribution [Nor98]).** *Let $I$ be finite. Suppose for some $i \in I$ that*

$$p_{ij}^{(n)} \to \pi_j \text{ as } n \to \infty \text{ for all } j \in I.$$

*. Then $\pi = (\pi_j : j \in I)$ is a stationary distribution*

**Theorem 2.2 (Convergence to equilibrium [Nor98]).** *Let $P$ be irreducible and aperiodic, and suppose that $P$ has a stationary distribution $\pi$. Let $\lambda$ be any initial distribution over the state-space $I$. Suppose that $(X_n)_{n\geq 0}$ is Markov($\lambda, P$). Then*

$$\mathbb{P}(X_n = j) \to \pi_j \text{ as } n \to \infty \text{ for all } j.$$

*In particular,*

$$p_{ij}^{(n)} \to \pi_j \text{ as } n \to \infty \text{ for all } j.$$

Such a Markov chain is called ergodic.

This Markov chain will converge to an equilibrium state with a stationary distribution $\pi$, such that $\pi P = \pi$ [Nor98]. The stationary distribution $\pi$ is independent of the initial state $\lambda$.

Random walks have many variants, most notably *Random Walks with Restart (RWR)* [PBMW99] where the random surfer can choose to move to a vertex in the neighborhood of the current vertex with probability $\alpha$ or jump back to the start vertex (teleport) with probability $(1 - \alpha)$. In this case the restarts are limited to specific states and the Markov chain is not necessarily ergodic. The probability of a node's reachability serves as a measure of the similarity between the teleport vertex and the other vertices in the graph.

Random-walk similarity measures are related to similarity measures used for semi-supervised learning [ZLR05] and spectral clustering [MS01], hence it provides an elegant measure of similarity between vertices in a graph. Chapter 4 discusses how we efficiently employ RWR to solve the inference problem in `Equity`.

Random walks have been adopted in numerous domains such as: information retrieval [PBMW99, Hav02], schema matching [MGMR02], image segmentation [Gra06], named entity and word-sense disambiguation [TMN04, GB14, MRN14]. Multiple techniques have been proposed to efficiently measure similarities in large graphs using random walks [TFP06, Cha07]. Page et al. [PBMW99] proposed an efficient algorithm to compute Page Rank on web-scale link graphs comprising millions of pages.

### 2.4.2 Probabilistic Graphical Models and Markov Random Fields

*Probabilistic Graphical Models (PGM)* are a family of stochastic models that account for uncertainty and structure. It aims at efficiently modeling a joint probability distribution $P$ over a set of random variables $\mathbb{X} = \{X_1, X_2, .., X_n\}$. The domain of this joint probability distribution grows exponentially with the size of the random variables. For example, given binary-valued random variables, there will be $2^n$ different possible assignments to $P$. Therefore, finding the joint probability distribution $P$ for these random variables becomes computationally intractable as the number of the random variables grows large or as their domain of values expands. PGM enables the estimation of these intractable joint probability distributions through a simplified representation. PGM relies on the conditional independence properties between the random variables to factorize the joint probability distribution into modular components [KFB09].

**Definition 2.11.** *Let $X, Y$ and $Z$ be random variables. Then, $X$ is conditionally independent of $Y$ given $Z$ in a distribution $P$ if:*

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$$

*for all the values of $x \in dom(X), y \in dom(Y), z \in dom(Z)$.*

There are two common types of PGMs: Bayesian Networks and Markov Networks. Bayesian Networks are directed PGMs, and Markov Networks are undirected PGMs. Bayesian Networks are represented as a *directed acyclic graph (DAG)*. The vertices in the graph correspond to the random variables in the model, and the edges indicate the dependencies between these random variables. The direction of the dependency determines the direction of the edge. The joint probability distribution of the variables in Bayesian Networks can be deduced from the DAG. For more information about Bayesian Networks, refer to [KFB09].

#### Markov Random Fields

Markov Networks, also known as *Markov Random Fields (MRF)*, are useful in modeling domains at which the direction of the dependencies is unknown. MRFs often offer a simpler perspective over directed models in terms of the independence structure and the inference [KFB09]. The joint probability distributions of an MRF is modeled using an undirected graph $G_m$, where vertices are random variables, and edges indicate a direct probabilistic relation between random variables. The joint probability distribution is factorized over the graph using potential functions or factors over maximal cliques in the graph [KFB09],

**Parameterization** MRFs are represented by a joint probability distribution. This distribution is parametrized over the graph structure. The parameters of MRF do not correspond to probabilities or conditional probabilities as in the case of Bayesian Networks. Thus, they are less intuitive and harder to estimate from the data.

Given an MRF with a set of random variables $\mathbb{X}$, we define the joint probability distribution of the MRF using the notion of factors.

**Definition 2.12.** *A factor $\pi(X)$ is a function that maps a configuration of values of the variables $\mathbb{X}$ to a real positive number in $\mathbb{R}^+$.*

The joint probability distribution of an MRF is represented as a product of local models. These local models are defined over a subset of variables in the graphs, such that each local model $i$ corresponds to a factor $\pi_i$. MRFs simplify the parameterization of the joint probability distribution by assigning factors to maximal cliques in the graph.

**Definition 2.13 (clique).** *A clique $C_i$ is a subset of vertices in a graph such that every two vertices are directly connected. A clique is a complete graph, and a maximal clique is a clique that cannot be extended by including any additional vertex.*

**Definition 2.14.** *Given an MRF graph $G_m$, a distribution $P$ factorizes over $G_m$ if it is associated with:*
*(i) a set of maximal cliques $C_1, C_2, .., C_k$ in $G_m$;*
*(ii) a set of factors associated with each clique: $\pi_1(C_1), \pi_2(C_2), ..., \pi_k(C_k)$, such that*

$$P(X_1, X_2, .., X_n) = \frac{1}{Z} * \pi_1(C_1) \times \pi_2(C_2) \times ... \times \pi_k(C_k)$$

*where Z is a normalization constant called the partition function.*

MRFs assign factors to maximal cliques in $G_m$. The set of factors $\pi_1(C_1), \pi_2(C_2), ..., \pi_k(C_k)$ are known as the clique potentials. The distribution $P$ that factorizes over $G_m$ is called *Gibbs Distribution.* The clique potentials can be represented by a logarithmic function,

$$\pi(C) = exp(-\epsilon(C)),$$

where $\epsilon(C) = -ln\pi(C)$ is called the energy function. The joint distribution now corresponds to:

$$P(X_1, .., X_n) \propto exp(-\sum_{i=1}^{k} \epsilon_i(C_i))$$

Any positive distribution whose conditional independence properties can be represented by an undirected graphical model can be represented as a product of clique potentials [Mur12].

**Theorem 2.3 (Hammersley-Clifford [Bre01]).** *A positive distribution $P(X) > 0$ satisfies the conditional independence properties of an undirected graph $G_m$ iff $P$ can be represented as a product of factors where each factor corresponds to a maximal clique in the graph.*

A special class of MRF is pairwise MRF where all the factors are defined over a single variable or a pair of variables. Thus, the parameterization is restricted to the edges of the graph. Pairwise MRF are attractive due to their simplicity and their ability to model a broad range of domains [KFB09]. *Conditional Random Fields (CRF)* [LMP01] is another variant of MRF at which clique potentials are conditioned on input features. CRF models have been proposed for various NLP and IE problems such as noun-phrase chunking [SP03], part of speech tagging [LMP01], and named entity recognition [SC05].

### Inference

The joint probability distribution of a PGM allows us to answer three types of queries, which are listed in [KFB09]:

- **Probability Queries**
  A probability query is the most common query type. It consists of two parts: (i) the `evidence`, a subset $E$ of observed random variables in the model and its observed values $e$; and (ii) the *query variables*, a subset $Y$ of unobserved variables in the model. The probability query is the posterior probability distribution over the values $y$ of $Y$, conditioned on the evidence $E = e$, and defined as follows:

  $$P(Y \mid E = e)$$

  It is also defined as the marginal over $Y$ of the joint probability distribution conditioned on $E = e$.

- **Maximum a Posterior (MAP) Queries**
  A MAP query, also known as the *most probable explanation (MPE)*, finds the most likely assignment to all of the random variables in the model except for the evidence $E$. Let $W = \mathbb{X} \; - \; E$, then the MAP query finds the most probable assignment of $W$, given the observed evidence $E = e$:

  $$MAP(W \mid e) = \arg\max_{W} P(w, e)$$

  , where $\arg\max_x f(x)$ gives the value of x at which $f(x)$ is maximum. The difference between the MAP and the probability query is that in MAP query we find the most likely *joint* assignment to $W$. To find the most likely assignment to a single variable A, we can compute the conditional probability distribution $P(A \mid E = e)$, and then pick the assignment with the highest value. However, the most likely joint assignment of a subset of random variables is different from the assignment at which each variable is selected separately.

- **Marginal MAP Queries**
  In the case of rare events such as rare diseases, the MAP assignment might not be able to find the desirable answer. For example, in a medical diagnostic problem, where the most likely disease has multiple possible symptoms, each of them occurs with low probability. In contrast, there exists a rare disease with a small number of symptoms, each of them occurs with a high probability given the disease. The MAP joint assignment for the disease and the symptoms might be higher for the second disease, but not for the first one. Hence, It is more desirable to find the most likely assignment of the disease variable only, given the symptoms. The marginal MAP answers this type of queries, given a subset $Y$ of random variables (disease), and an evidence $E = e$ (symptoms):

  $$MAP(Y \mid e) = \arg\max_{y} P(y \mid e)$$

. Let $\mathbb{Z} = \mathbb{X} - Y - E$, the marginal MAP computes the following:

$$MAP(Y \mid e) = \arg\max_{y} \sum_{\mathbb{Z}} P(Y, \mathbb{Z} \mid e)$$

. Thus, the marginal MAP contains both the elements of a probability query and a MAP query.

Inferring answers for these queries is NP-hard in the worst case [KFB09]. Exact inference is possible for some cases of graphical models. However, for a large number of graphical models, exact inference is intractable, and only approximate inferencing is possible.

Learning graphical models involve two tasks: (i) parameter estimation and (ii) structure learning. For parameter estimation, *Maximum Likelihood Estimation* can estimate the values of the parameters $\theta_{\mathbb{G}}$ of the joint probability distribution. It maximizes the likelihood function $L(\theta_{\mathbb{G}} \mid \mathbb{D})$, and finds the best parameters that can generate the data given the graphical model. However, maximum likelihood estimation usually overfits the data. *Bayesian Parameter Estimation* is another parameter estimation approach that defines a prior distribution over the parameters to prevent overfitting [KFB09]. Learning the structure of the graphical models is explained in [KFB09].

# 3 Quantity Knowledge Base

This chapter discusses the construction and the organization of the *Quantity Knowledge Base (QKB).*

## 3.1 Introduction

The QKB is the semantic space to which we aim to map mentions of quantities. We constructed the QKB by importing information from `freebase`, `Wikipedia`, and other online sources. The QKB is organized into a simple four-level taxonomy of dimensions, units, measures, and thematic domains. It supports the mapping of quantity mentions into triples of $\langle measure, value, unit\rangle$. QKB keeps a set of alias names for each measure as well as conversion rules for each unit to the *International System of Units (SI).* It covers different quantities including physical, monetary and also dimensionless quantities such as ratios, proportions, counts, and scores.

## 3.2 The QKB Taxonomy

The QKB has a lightweight taxonomy with four types of objects:

**Definition 3.1 (Unit).** *A unit is a quantity of a specific magnitude that serves as a reference for measuring and comparing similar quantities. In the SI, there are seven base units: meter, kilogram, seconds, ampere, kelvin, candela, mole; all the other units are derived from these units.*

**Definition 3.2 (Dimension).** *A dimension is a physical measurement of an object. The primary seven dimensions corresponding to the base units in the SI are length, mass, time, electric current, temperature, luminous intensity, and amount of substance. Dimensions have a one-to-one correspondence with units.*

**Definition 3.3 (Measure).** *A measure is a concept referring to a certain quantifiable aspect of an object or process (e.g., the height of a building, the inflation rate). A measure can have a physical dimension such as fuel consumption and greenhouse gas emission or can have dimensionless quantity such as inflation rate and population.*

**Definition 3.4 (Thematic Domain).** *A domain is a group of dimensions commonly used for measuring objects in a certain field such as Mechanics, Heat, Electricity & Magnetism, Acoustics.*

The QKB represents dimensionless quantities with a unit of 1, such as ratios and counts. In the following sections, we explain the construction of the QKB and describe its taxonomy.

## 3.3 QKB Construction

We automatically import and merge units from different sources. Then, we manually refine the processed information to remove incorrect objects, merge similar objects, or add missing information. Hence, the QKB is a high-quality knowledge base for quantity alignment.

### 3.3.1 Sources of Information

The lack of a quantity knowledge base to serve our need for mapping quantity mentions motivated us to combine information from multiple sources. Each of the sources we consider complements the deficiencies of the other sources. The result is a QKB which represents the semantic space for mapping quantity mentions. We use the following sources of information:

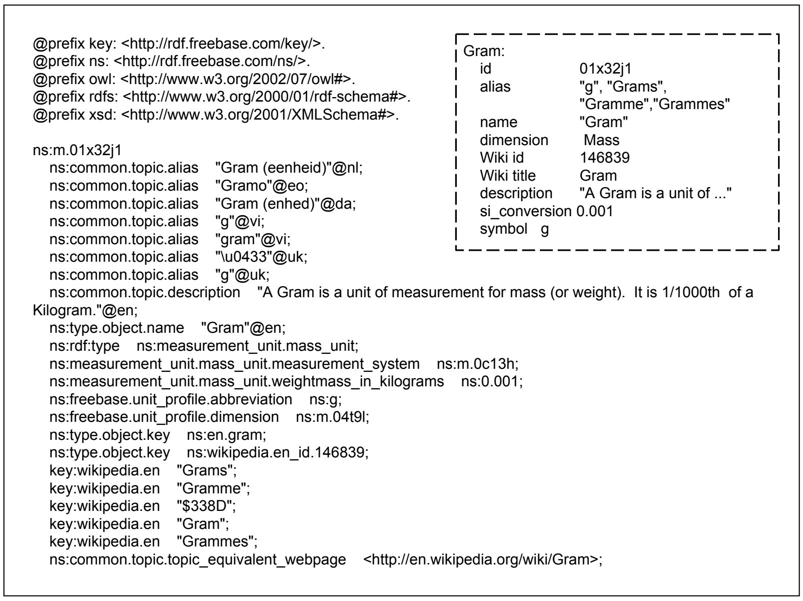
#### Wikipedia

`Wikipedia` is a free collaborative encyclopedia constructed by the community. In `Wikipedia`, the encyclopedic knowledge is arranged into pages, such that each page corresponds to a single entity. `Wikipedia` covers physical entities such as persons, locations, organization as well as concepts such as speed, renewable energy, greenhouse gas. `Wikipedia` is a general textual encyclopedia, and therefore, it is insufficient for representing specialized objects like quantities. At the time of this work, `Wikipedia` covered the SI base units such as meter, kilogram, second. However, it had low coverage for the complex units such as miles per gallon (MPG), cubic meter per kilogram, kilogram per second. Now `Wikipedia` contains lists of SI derived units accompanied by their measure and conversion rules to SI base units. In both cases, the unit information is specified in the text or in tables. Thus, extraction requires extra effort. For each unit in `Wikipedia` we extracted its page title, such as “Kilogram” and “Joule”, as well as its article numerical id.

#### Freebase[1]

`Freebase` is a collaborative knowledge base constructed by the community. It consists of triples of $\langle subject, predict, object\rangle$ (SPO). The objects in `freebase` are linked to their corresponding `Wikipedia` pages through `Wikipedia` page URL and a title. `freebase` has better coverage for units than `Wikipedia`, but it has lower quality. For each unit, we extract the following information from `freebase`: the unit names, abbreviations, `freebase` identifier, dimension, the corresponding `Wikipedia` page title and identifier. Figure 3.1 shows `freebase` triples for the Gram unit, and the information we extracted from these triples. Figure 3.2 gives examples of units extracted from `freebase`.

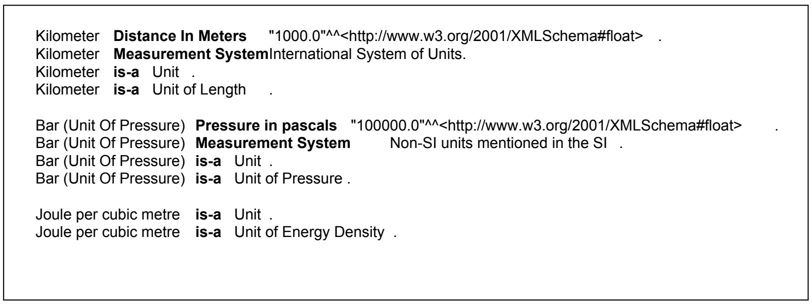
[1] freebase.com



```
@prefix key: <http://rdf.freebase.com/key/>.
@prefix ns: <http://rdf.freebase.com/ns/>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

ns:m.01x32j1
   ns:common.topic.alias    "Gram (eenheid)"@nl;
   ns:common.topic.alias    "Gramo"@eo;
   ns:common.topic.alias    "Gram (enhed)"@da;
   ns:common.topic.alias    "g"@vi;
   ns:common.topic.alias    "gram"@vi;
   ns:common.topic.alias    "\u0433"@uk;
   ns:common.topic.alias    "g"@uk;
   ns:common.topic.description    "A Gram is a unit of measurement for mass (or weight).  It is 1/1000th  of a
Kilogram."@en;
   ns:type.object.name    "Gram"@en;
   ns:rdf:type    ns:measurement_unit.mass_unit;
   ns:measurement_unit.mass_unit.measurement_system    ns:m.0c13h;
   ns:measurement_unit.mass_unit.weightmass_in_kilograms    ns:0.001;
   ns:freebase.unit_profile.abbreviation    ns:g;
   ns:freebase.unit_profile.dimension    ns:m.04t9l;
   ns:type.object.key    ns:en.gram;
   ns:type.object.key    ns:wikipedia.en_id.146839;
   key:wikipedia.en    "Grams";
   key:wikipedia.en    "Gramme";
   key:wikipedia.en    "$338D";
   key:wikipedia.en    "Gram";
   key:wikipedia.en    "Grammes";
   ns:common.topic.topic_equivalent_webpage    <http://en.wikipedia.org/wiki/Gram>;
```

```
Gram:
   id               01x32j1
   alias            "g", "Grams",
                    "Gramme","Grammes"
   name             "Gram"
   dimension         Mass
   Wiki id          146839
   Wiki title       Gram
   description      "A Gram is a unit of ..."
   si_conversion 0.001
   symbol   g
```

Figure 3.1: Example of triples associated with the Gram unit from `freebase`. The information extracted from these triples is shown in the top right corner.

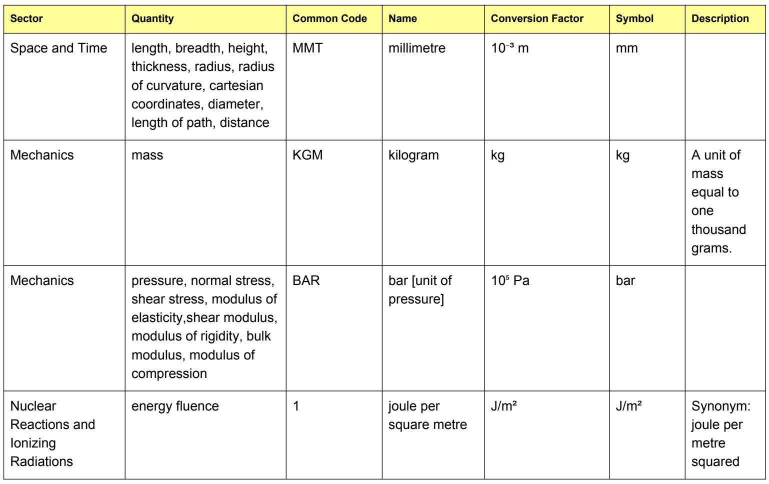


```
Kilometer   Distance In Meters    "1000.0"^^<http://www.w3.org/2001/XMLSchema#float>   .
Kilometer   Measurement SystemInternational System of Units.
Kilometer   is-a   Unit   .
Kilometer   is-a   Unit of Length     .

Bar (Unit Of Pressure)  Pressure in pascals   "100000.0"^^<http://www.w3.org/2001/XMLSchema#float>       .
Bar (Unit Of Pressure)  Measurement System        Non-SI units mentioned in the SI   .
Bar (Unit Of Pressure)  is-a   Unit  .
Bar (Unit Of Pressure)  is-a   Unit of Pressure .

Joule per cubic metre   is-a   Unit  .
Joule per cubic metre   is-a   Unit of Energy Density  .
```

Figure 3.2: Example of units extracted from `freebase`

### United Nations Economic Commission for Europe (UNECE)

We import information from the *Codes for Trade* published by the UNECE. We import the *ISO Country and Currency Codes* for monetary quantities, and the *Codes for Units of Measurement* used in the international trade for other quantities. For each unit, we extract the following information: sector (i.e. mechanics and acoustics), dimension, name, symbol (i.e $kg$ and $m^2$), synonyms, conversion factor to SI units, UNECE code. Figure 3.3 shows an example of unit information organized in an excel sheet from UNECE. We extract the dimension(s) from the “Quantity” column, and the name of the unit from the “Name” column. We extract the synonyms from the description column when it is given.



| Sector | Quantity | Common Code | Name | Conversion Factor | Symbol | Description |
|---|---|---|---|---|---|---|
| Space and Time | length, breadth, height, thickness, radius, radius of curvature, cartesian coordinates, diameter, length of path, distance | MMT | millimetre | $10^{-3}$ m | mm | |
| Mechanics | mass | KGM | kilogram | kg | kg | A unit of mass equal to one thousand grams. |
| Mechanics | pressure, normal stress, shear stress, modulus of elasticity,shear modulus, modulus of rigidity, bulk modulus, modulus of compression | BAR | bar [unit of pressure] | $10^5$ Pa | bar | |
| Nuclear Reactions and Ionizing Radiations | energy fluence | 1 | joule per square metre | J/m$^2$ | J/m$^2$ | Synonym: joule per metre squared |

Figure 3.3: Example unit facts from UNECE: Revision 12 Recommendation 20 for 2016

### 3.3.2 Data Integration

We extract the units from `freebase` and UNECE codes for trades. We combine the units from both sources using exact string matching of the units' name(s). Then, we used the `Wikipedia` identifier in `freebase` to link each unit with the corresponding `Wikipedia` page. We aggregate the information from these sources and create a single record for each unit, dimension, measure, and domain. After that, we extract from `Wikipedia` all the concepts related to each unit as follows:

- *Concepts:* We extract all pages corresponding to non-entities from `Wikipedia`. We identify these concepts as the `Wikipedia` pages that are not included in the `Yago` knowledge base, because `Yago` only contains physical entities. Then, we filter non-concept pages, using the category of the page; First, we identify the category of the page; Second, we extract its parent category from `Yago` simple taxonomy; Third, we filter out pages under certain categories, such as physical entities and artifacts. Finally, we manually inspect the concepts and remove pages that follows a certain pattern, such as page titles containing "book" or "album".

- *Table Headers:* We identify columns with numerical content from `Wikipedia` tables. Then, for each column, we extract its header and hyperlinks to other `Wikipedia` pages. Each `Wikipedia` table has a unique identifier and is attached with a specific `Wikipedia` page. We identify each header and hyperlink by its table, row, and column.

- *Column Content:* We extract hyperlinks to other `Wikipedia` pages form numerical columns' content. We identify each hyperlink with its table, row, and column.

- *Co-occurrence*: We collect co-occurrence statistics of pages from the hyperlinks we obtained from the tables. We collect three types of co-occurrence statistics: (i) same-table, for pages co-occurring on the same table, (ii) same-column, for pages co-occurring on the same column, and (iii) same-row, for pages co-occurring in the same row.

- *Unit-Concepts Co-occurrence*: We identify the subset of co-occurrence statistics corresponding to a unit-concept relation.

As a final step we merge a list of currencies from the UNECE codes for trades to the QKB.

# 4 Equity: Semantically Annotating Web Tables and Surrounding Context

## 4.1 Introduction

### 4.1.1 Motivation

The Web contains a wealth of structured but schema-free data in the form of HTML tables. These are manually created by knowledgeable users who want to share information—about music, food, car companies, renewable energy, traffic statistics etc. The advent of cloud-based editing and publishing tools (e.g., Google Sheets and Fusion Tables, Microsoft Excel Online) makes it even easier for users to post such content on the Internet. Likewise, a huge amount of tabular data exists within enterprise intranets, typically created with spreadsheet software.

There is a great opportunity in comparing and combining multiple tables, towards analytic insight. However, these tables are typically created in an ad-hoc manner, to be shared with human users. And the absence of schemas and, even more, the diversity and potential inconsistency of terminologies among different tables (by different users) makes such data fusion steps impossible—if desired to be automatic—or extremely tedious—if carried out manually. The vocabulary mismatch across tables has several dimensions:

- Names in table headers typically denote *classes* (e.g., car model) or general *concepts* (e.g., CO2 emission), but are chosen ad-hoc on a per-table basis.

- Names in cells of the table body often denote individual *entities* (e.g., Tesla—the car maker, Musk—its CEO, Model S—one of Tesla's models), but the entity names are highly ambiguous.

- Other cells contain *quantities* such as financial measures (e.g., revenue in USD), physical measures (e.g., power in kW or energy consumption per 100 km in kWh), or plain numbers denoting ratios, temporal changes, ratings, etc. The encodings of values and their units can vary heavily across tables (e.g., $1 bn vs. 1000m USD for revenue, MPG vs. l/100 km for fuel consumption).

**Example:** Table 4.1 shows a typical example of a Web table, about environment-friendly cars in the U.S. If we want to compare this data to a table about these (and other) cars in Europe, we face huge heterogeneity issues regarding headers (Manufacturer vs. Company), entities in cells (Toyota Prius Eco vs. Prius Model 2016) and quantities in cells (MPGe vs. kWh/100km). This table is taken from a Wikipedia article; tables

Table 4.1: Example Table: Green Vehicles Comparison

| Vehicle | Manufacturer | Class | GHG emissions (1) | Tailpipe emissions (g/mi of CO2) | EPA Fuel Economy combined (MPG) | Annual Fuel Cost |
|---|---|---|---|---|---|---|
| Toyota Prius phev | Toyota | Hybrid electric | 61 lb CO2 | 133 | 95 MPGe ( [29kWh+0.2 gal] / 100 mi) | $600.00 |
| Toyota Prius Eco All years, gasoline fuel | Toyota | Hybrid electric | 51 lb CO2 | 178 | 50 ( 21.25 km/li ) | $600.00 |
| BMW i3 All years, all fuels | BMW | Electric car | 54 lb CO2 | 0 | 124 MPGe ( 27kWH/100mi ) | <550$ |
| Tesla Model S ( 60/85 kWh battery )2013 Award | Tesla | Electric car | 54 lb CO2 | 0 | 95 MPGe ( 35kWH/100mi ) | $700.00 |
| Chevrolet Volt 2011 Award | GM | Plug-in Hybrid | 61 lb CO2 | 81 | 98 MPGe ( 35kWh/100mi ) | $800.00 |
| Bolloré Bluecar | Cecomp | Electric car | 15.2 kg/100km | 0 | NA | 80 €/mo |

(1) measured per 100 mile.

“from the wild”, appearing in user’s homepages or posted to social media, are an even greater challenge for proper interpretation.

### 4.1.2 Problem Statement

In order to make better sense and enable re-use of ad-hoc tables, we want to *canonicalize* their headers and cells: link classes and concepts to a taxonomic catalog or simply to Wikipedia articles, disambiguate entity names onto uniquely identified entities registered in a knowledge base (KB), and map quantities into a complete and normalized representation with easily interpretable value and unit. This chapter addresses the problem of linking tables and surrounding text to a KB, with emphasis on making sense of entities and quantities.

**Prior work and its limitations:** While entity linking (and so-called Wikification) from text to knowledge bases has received wide attention (see [ERD, SWH15, URNN$^+$15] and references given there), there is fairly little work on semantic annotation and linkage of Web tables. The first work on lifting Web tables to “first-class citizens” for search engines, by Halevy et al. [CHM11], solely aimed at indexing for searchability and did not pursue any form of canonicalization. The seminal work on semantic linking for table cells by Sarawagi et al. [LSC10] devised a probabilistic graphical model to map classes, relations and entities to a knowledge base. The resulting accuracy was in the order of 80%, and the method has high computational complexity. The recent work of Bhagavatula et al. [BND15] improved accuracy above 95%.

None of these prior works considered quantities in tables.

Sarawagi et al. [SC14] addressed quantities, but focused on the specific tasks of searching with numerical values and extracting numerical relations from text [MMM$^+$16]. Fully canonicalizing tables so that they can be compared and joined has been out-of-scope. The work by Chakrabarti et al. [YGCC12] developed table-to-table matching methods, based on entity augmentation, for the purpose of searching Web tables. In their follow-up work [ZC13] the matching problem for a table corpus is extended to consider also numeric attributes. Although this work supports some form of comparing and combining tables, full canonicalization where all cells are mapped to semantic items in a knowledge base has not been considered.

### 4.1.3 Proposed Solution

Our approach is inspired by this prior work, but goes beyond their settings in several ways:

- We completely canonicalize entities and quantities (as well as classes and concepts).
- We exploit the textual context that usually surrounds a table and jointly link names and values from both table and text. For example, Table 4.1 appears in a page with the text shown in Figure 4.1. This allows us to harness semantic redundancy and richer features.
- We devise an efficient algorithm for fast processing of input tables, with the goal of supporting analysts in a responsive manner.

**Our Approach:** Probabilistic graphical models like Markov Random Fields (MRF's, aka. CRF's when inference is focused on conditional probabilities) are a most natural candidate for capturing the interdependencies in the potential linking targets of different entities, quantities, classes and concepts. Therefore, we conceptually start with a judiciously designed MRF model. To avoid the bottleneck of explicitly labeled training data, we employ distant supervision by drawing semantic relatedness weights from a knowledge base (with weights mined from Wikipedia links, unrelated to tables). We merely need a small set of annotated tables for tuning six hyper-parameters. To escape the high complexity of MRF/CRF inference (typically via MCMC sampling), we harness a theorem from [Coh10] and construct a regular weighted graph from the MRF such that, under certain conditions, random walk (RW) algorithms closely approximate marginal probabilities for the MRF. Random walks can be implemented very efficiently. Working out the details of this MRF-to-RW reduction is one of our key contributions. Our end-to-end solution for the table canonicalization problem is implemented in a system called *Equity* (Entity and quantity in tables).[1]

**Contributions:** We present in this chapter the following contributions:

- a comprehensive, distantly supervised MRF model for canonicalizing ad-hoc tables, handling classes, concepts, entities and quantities in both table cells and surrounding text;

[1] More on this project, including code and experimental data, can be found at `www.mpi-inf.mpg.de/equity`.

- an efficient algorithm, based on random walks, for computing high-quality solutions;

- experimental results with a diverse set of Web tables that demonstrate the high accuracy of our method.

> The most efficient cars on the market are all electric cars. In fact, every electric car on the market is more efficient than even the most efficient conventional hybrid car (the Toyota Prius). Some of them are more than twice as efficient. As you scroll through the list below, note that the Prius has a MPG rating of 50 while Model S has a MPGe of 95. If you are not familiar with MPGe, it is a rating created by the EPA to determine the relative efficiency of an electric car compared to a gasoline car. MPGe is generally good for comparing electric cars to conventional gasmobiles and hybrids.

Figure 4.1: Text Snippet from context of Table 4.1

## 4.2 Related Work

### 4.2.1 Entity Linking:

There is ample work on detecting and disambiguating entities that appear in text documents; [SWH15] is a recent survey on this topic. Some of the prominent approaches map to Wikipedia (e.g., [MW08, RRDA11]), thus covering also classes and concepts, whereas others strictly focus on individual entities with DBpedia, Yago or Freebase as their point of reference (e.g., [HYB$^+$11, MJGSB11]). The best-performing methods typically combine a variety of signals and techniques like pair-wise relatedness of entities [Cuc14, PF14], refined context models [LSRP15], graph algorithms [HYB$^+$11], and random walks [GB14]. The Equity system adopts some of these techniques, embedding them into its generalized framework for linking mentions in tables to both entities and quantities.

### Quantity Extraction

Numeric attribute values and numeric expressions in natural-language text have been considered by work on information extraction and knowledge fusion [DGH$^+$14, MMM$^+$16, RVR15, SA15]. This line of research is related to our stage of quantity mention detection, but does not address the issue of canonicalizing quantities.

### Table Search and Matching

Starting with the seminal work of [CHW$^+$08, CHM11, MAAH09], there is growing research on Web tables and spreadsheets, with the goal of searching table contents, matching tables against each other and inferring table header semantics [SC14, VHM$^+$11, YGCC12, ZC13]. Linking table cells to a KB is of no or minor concern in these works. Sarawagi et al. [SC14] and Zhang et al. [ZC13] deal with quantities in tables, using computational expensive techniques like MCMC inference for probabilistic graphical models.

### Table Canonicalization

Closest to our work is the prior research of table canonicalization and quantity Queries [LSC10, SC14, MFJ13, BND15]. Limaye et al. [LSC10], Mulwad et al. [MFJ13] and Bhagavatula et al. [BND15] pursued the same goal as our work, but did not consider quantities at all. Also, these methods use expensive inference algorithms and partly rely on extensive training data; our work avoids both of these potential bottlenecks. Ritze et al. [RLB15] addresses the linking of table headers and rows to concepts and entities in DBpedia, but focuses on small and narrow HTML tables.

Sarawagi et al. [SC14] specifically addressed quantities in tables. It developed a grammar-based technique for column annotation and a supervised classifier for inferring units of columns with numeric values. However, this was driven by the task of searching a heterogeneous table corpus, without resolving the heterogeneity—thus leaving out the task of linkage to a comprehensive KB.

### MRF/CRF and Random Walks

Our approach builds on insights from the work of W. Cohen [Coh10] about the connections between MRF inference and random walk algorithms. Cohen has developed this further into a general framework for reasoning with random walks [WMLC15].

## 4.3 Model and System Overview

This section presents the formal problem definition and introduces important notation.

### 4.3.1 Problem Input

The input to the Equity system is:

- A table $T$ with $m+1$ rows, numbered $0 \dots m$, and $n+1$ columns, numbered $0 \dots n$; where row 0 is the *header row*. We use $m_{ij}$ to refer to the mention in table cell $(i, j)$, i.e., in row $i$ and column $j$. The set of all mentions in table $T$ is denoted as $\mathbf{M_T}$.

- A surrounding context with $\nu$ mentions $m_k$ $(k = 1, \dots, \nu)$. The context is extracted from the web document's title and the table's surrounding text and caption. The set of all mentions in the context is denoted as $\mathbf{M_X}$.

We use $\mathbf{M} = \mathbf{M_T} \cup \mathbf{M_X}$ to refer to the set of all mentions in both table and context. We distinguish between two types of mentions. A *numerical mention* is a number, possibly accompanied by a unit. It represents a quantitative measure such as '27 kWh/mi'. All other mentions are referred to as *string mentions.* They are likely to refer to entities (e.g., GM or General Motors), classes (e.g., car manufacturers), or concepts (e.g., GHG emission).

Equity currently focuses on tables with the following common structural properties: The table header contains string mentions for classes and concepts. If the header is a class, then the non-header cells in the same column contain instances of the class, i.e., entities, as illustrated by the Manufacturer column in Table 4.1. If the header is a concept, then the non-header cells in the column contain quantitative measures, e.g., the Annual Fuel Cost in the example. As a result, Equity distinguishes between the following six *sorts* of mentions based on mention type and location: string/cell, string/header, string/context, numeric/cell, numeric/header, numeric/context.
Note that Equity can easily handle "transposed" tables where the header is not in row 0, but column 0, by working with the transpose of $T$.

### 4.3.2 Knowledge Base

The space of semantic targets to which we aim to map mentions in a table and its context is given by one or more knowledge bases (KB's). For individual *entities* and for *classes* (i.e., semantic types), we use Yago (`yago-knowledge.org`), which is one-to-one interlinked with Wikipedia for entities, and also connects Wikipedia categories with WordNet synsets for its extensive class hierarchy. For general *concepts*—abstractions that are neither classes nor entities (e.g., love, universe, number theory)—Yago is less suitable. In that case targets are Wikipedia articles (which do not have counterparts in Yago).
For quantities—the most challenging kind of targets— we use our QKB introduced in Chapter 3.

### 4.3.3 Algorithm Objective and Output

We are interested in a (potentially partial) mapping $\Psi$ from the set $\mathbf{M}$ of mentions to the set $\mathbf{S}$ of semantic items. Among all possible mappings, we aim to find one where (i) each mention is mapped to the "best" semantic target and (ii) the mapping is "consistent" with constraints implied by the table structure. These intuitive ideas will be formalized in the next sections. Clearly, there can be tension between the two goals that Equity has to address. For instance, given only the string "Tesla", the best match might be the KB entry for the person Nikola Tesla. However, in the context of Table 4.1, the location of this mention in a column of car manufacturers suggests a reference to the car maker.
Similar to the mentions, Equity also distinguishes between different *sorts* of semantic items: entities, classes, concepts and quantities.
Equity should produce the following mappings for sample mentions in Table 4.1:

- Mention $m_{03}$="GHG emission" is mapped to concept `Greenhouse_gas`.
- Mention $m_{11}$="Toyota" is mapped to entity `Toyota` (the company).
- Mention $m_{25}$="50" is mapped to the `physical` measure `EPA_Fuel_Economy` with value `50` and unit `Miles_Per_Gallon`.

If a mention has no proper item in the KB, Equity should map it to Null.

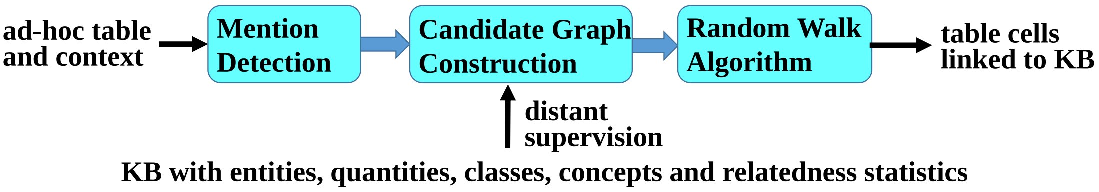


Figure 4.2: System Architecture of Equity

### 4.3.4 System Architecture

Figure 4.2 depicts the major components of the Equity system. We employ standard preprocessing for extracting a table and its context from a web page, and for shallow NLP such as part-of-speech tagging and coarse-grained typing of names—both via the Stanford NLP tools based on trained CRF's [FGM05]. Note that the typing by the Stanford NER tagger merely produces labels like Person, Location, Organization, Date, Money and Misc, for text spans that likely denote entities or values of these kinds.

Equity first *detects mentions* in both table cells and context. This was partly done by the Stanford NER tagger already, but we apply additional regular expressions over token types to detect more mentions. Especially for quantities this is often decisive to ensure high recall.

To identify *semantic item candidates*, we run a light-weight form of Named Entity Disambiguation. This is a specifically configured variant of AIDA [HYB$^{+}$11] using a simple popularity-based prior only and giving it only the mention itself (without any context). As the mention boundaries from the previous stage are not necessarily correct, we re-run AIDA with different choices of mention substrings as input. From the output, Equity keeps the top candidate entities based on AIDA's confidence. For classes and concepts, which are not supported by AIDA, we perform simple string lookups against Yago and Wikipedia to generate candidates. For quantities, we match the input mentions against the alias names and, when applicable, regular expressions, for the measures in our QKB. The point of all this is to generate sufficiently many reasonable candidates. Hence this step does not have to be highly precise.

From the identified semantic items, we construct a *candidate graph*. This graph is constructed so as to approximate a full-fledged MRF with joint inference (see next section). Here we harness the KB for distant supervision, by using its precomputed relatedness scores as input for setting edge weights. These relatedness (aka. coherence) values are precomputed from Wikipedia links. We also apply some heuristic pruning when edge weights are negligibly small.

Finally, we perform random walks over the graph and identify, for each mention, the semantic item that has the highest stationary probability.

## 4.4 Probabilistic Graphical Model

We start with a very natural approach and cast the problem of determining the semantic targets for the given table mentions into a probabilistic model with the following random variables:

- $X_{i,j}$: the observed surface form of a mention in table cell (i,j).
- $X_k$: the observed surface form of the $k$-th mention in the context of the table.
- $Y_{i,j}$: hidden variable for the semantic target corresponding to the mention in cell (i,j).
- $Y_k$: hidden variable for the semantic target corresponding to the $k$-th mention in the table context.

The $X$-variables range over the set of all possible strings, while the $Y$-variables range over the set of possible semantic targets $S$ in the KB.
The desired mapping $\Psi$ from the set of mentions $\mathbf{M}$ to the set of semantic items $\mathbf{S}$ is determined by inferring the $Y$-variables from the given $X$-variables. We propose to use Markov Random Fields (MRF), which have been successfully employed for identifying entities and types in tables [LSC10] and for inference problems in image processing [BKR11] due to their ability to efficiently represent spatial coherence relationships between pixels. Tables are similar to images in the sense that table structure implies implicit coherence relationships. For convenience, we introduce $\mathbf{H} = \langle X_{0,0}, \ldots, X_{m,n}, X_1, \ldots, X_\nu, Y_{0,0}, \ldots, Y_{m,n}, Y_1, \ldots, Y_\nu \rangle$ to refer to the vector of all random variables. Let $H_i$ refer to the $i$-th entry in $\mathbf{H}$, $i \in \{0, 1, \ldots, 2(m+1)(n+1)+2\nu)\}$. In addition to $\mathbf{H}$, the MRF is defined by a set of potential functions $\Phi$, which capture relationships between the random variables. A pairwise relationship between $H_u$ and $H_v$ is modeled by function $\Phi_{u,v}$, which maps each pair of values from the domains of $H_u$ and $H_v$, respectively, to a real number. For the sake of readability, we will also use $\Phi(H_u, H_v)$ to refer to $\Phi_{u,v}$. Similarly, the relationship between three random variables $H_u$, $H_v$, and $H_w$ would be modeled by potential function $\Phi_{u,v,w}$, mapping a 3-tuple of values from the respective domains of the three random variables to a real number, and so on. Let $E_\Phi$ denote the set of sub-vectors of $\mathbf{H}$ over which $\Phi$ is defined. Then the MRF represents joint probability distribution

$$\Pr(\mathbf{H}) = \frac{1}{Z} \prod_{e \in E_\Phi} \Phi_e,$$

where $Z$ serves to scale the values so that they are true probabilities. A crucial design decision for an MRF is to determine $E_\Phi$, i.e., which random variables to connect through a potential function.

### 4.4.1 Potential Functions

We limit all potential functions to be “pairwise”. In addition to tractability of inference, this simplifies specifying the functions themselves. Intuitively, a pairwise potential

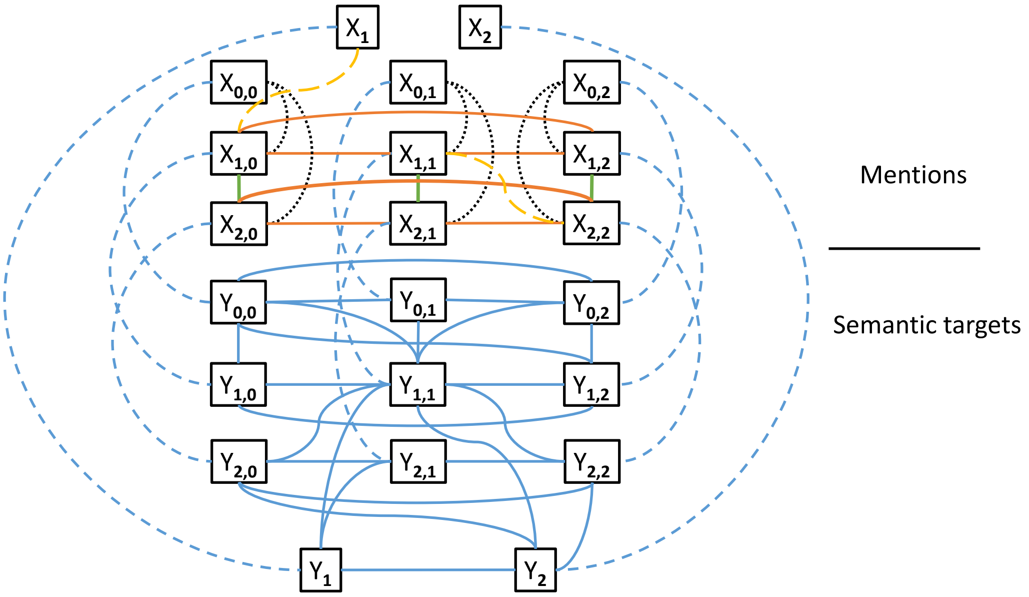


Figure 4.3: MRF model for a hypothetical table with 3 rows and 3 columns, whose context contains 2 mentions. For the $Y$-variables, not all edges of the fully connected subgraph are shown to reduce clutter.

function couples two random variables from **H** based on a relationship induced by table structure and content. The first family of potential functions captures the generic property that the surface form of the mention is closely related to the underlying semantic meaning.

**Mention-target coupling:** This dependency is represented by the blue dashed line in Figure 4.3. The corresponding family of potential functions is defined as $\phi_1(X_{i,j}, Y_{i,j})$, for all $0 \leq i \leq m$ and $0 \leq j \leq n$. Note that the individual functions in this family will differ depending on the sort of the mention and the semantic target (see Section 4.5.3). Similarly, the relationship between surface form and corresponding semantic target for the table context is captured by potential functions $\phi_2(X_k, Y_k)$, for $1 \leq k \leq \nu$. The next families of potential functions capture relationships induced by the table *structure*.

**Header-cell coupling:** This dependency is represented by the vertical black dotted line in Figure 4.3. It reflects that the header determines the information stored in a column. Equity captures this with a family of potential functions between the random variable for a header mention and the cell mentions in the same column: $\phi_3(X_{0,j}, X_{i,j})$, for $i > 0$.

**Same-row coupling:** This dependency is represented by the horizontal orange solid lines in Figure 4.3. It models that the cells in a row contain data for a certain object represented by the row, hence are closely related. Formally this is encoded with potential functions for each pair of random variables for mentions in a row: $\phi_4(X_{i,j}, X_{i,k})$, for $i > 0$ and $j \neq k$.

**Same-column coupling:** This dependency is represented by the vertical green solid lines in Figure 4.3. Since all entries refer to the same "type" of information determined by the header, each cell's mention is closely related to the others in the same column. The corresponding family of potential functions is $\phi_5(X_{i,j}, X_{k,j})$, for $i > 0$, $k > 0$, $i \neq k$. The last potential function families model global coherence properties.
**Same-value coupling:** This dependency is represented by the yellow long-dashed line in Figure 4.3. It captures the notion that given the specialized nature of a table and its context, occurrences of the same surface form are likely to refer to the same semantic target. This is modeled by potential functions that connect the random variables for all pairs of mentions that share the same surface form. One connects context mentions to table mentions: $\phi_6(X_{i,j}, X_k)$, for all $m_{ij} = m_k$. The other connects table mentions with each other: $\phi_7(X_{i,j}, X_{a,b})$, for all $m_{ij} = m_{ab}$. This could easily be relaxed to a coupling based on "similar", instead of identical, surface forms, e.g., to match 'MS Research' with 'Microsoft Research' or numbers such as '1.1 million' and '1,101,925'.
**Candidate-candidate coupling:** This dependency is represented by the blue solid line in Figure 4.3. It is motivated by the fact that all semantic targets for mentions in table and context should refer to a common topic, hence should be coherent. The corresponding families of potential functions are $\phi_8(Y_{i,j}, Y_{a,b})$, $\phi_9(Y_{i,j}, Y_k)$, and $\phi_{10}(Y_c, Y_k)$, for all $(i \neq a) \vee (j \neq b)$, $c \neq k$.

## 4.5 MRF and Random Walks

Cohen [Coh10] proved that marginal probabilities in an acyclic "pairwise"[2] MRF can be computed (almost) exactly through random walks followed by minimal post-processing. The proof includes the construction of an *ordinary-graph analog* of a given MRF, on which the random walks are performed. Even though the MRF for a table will usually contain cycles, Cohen's construction can still be applied to it. Cycles merely imply that equivalence between marginal probabilities in the MRF and the result of the random walk computation in the ordinary-graph analog might not hold any more. However, we argue—and confirm empirically—that the ordinary-graph analog still provides a good starting point for a random-walk based approach.
We now provide a summary of Cohen's approach, emphasizing intuition over detailed formalisms (for details, see [Coh10]). Given an MRF, its ordinary-graph analog is constructed as follows:

- For each random variable $V$ and each possible value $v \in V$, create a node $n_v$.
- Two nodes $n_v$ and $n_w$, $v \in V$, $w \in W$, $V \neq W$, are connected by an undirected edge of weight $\phi_{V,W}(v, w)$, if and only if $V$ and $W$ are connected by an edge in the MRF.
- For each leaf variable $L$, i.e., variable that is connected to only one other variable in the MRF, there is an additional *anchor node* $a_L$. It is connected by an undirected edge of weight 1 to each node $n_l$, $l \in L$.

[2] This is an MRF where all potential functions are defined over pairs of random variables.

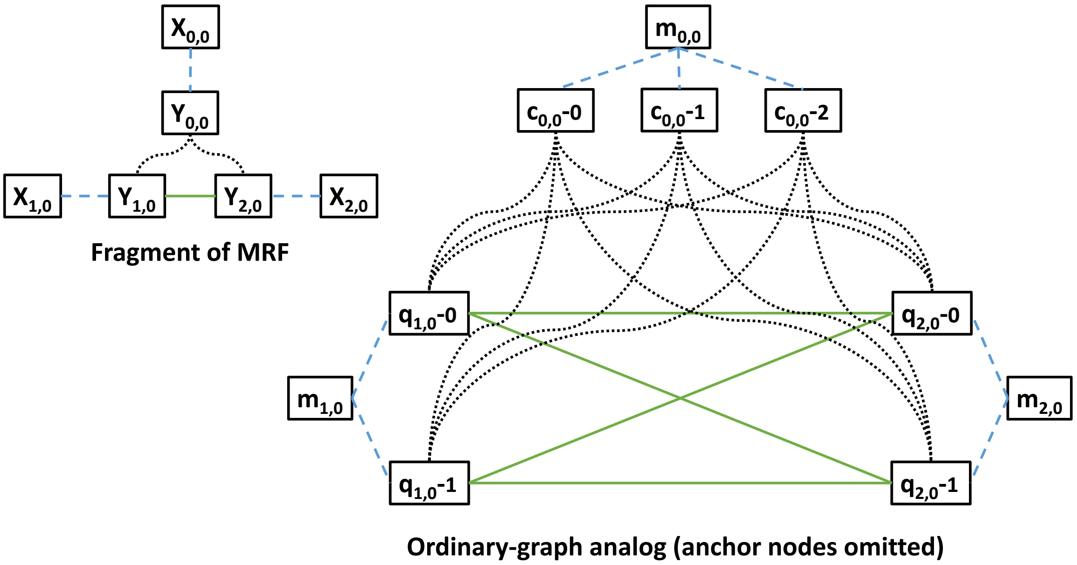


Figure 4.4: Ordinary-graph analog for a fragment of the MRF in Figure 4.3. For each connection of two random variables in the MRF, all values of the first are connected to all values of the second in the ordinary-graph analog. In the example, $Y_{0,0}$ ranges over three values; $Y_{1,0}$ and $Y_{2,0}$ each range over two possible values; and for the $X$-variables the only value is the mention given in the table.

- There are no other nodes or edges.

Figure 4.4 illustrates this construction for a fragment of the MRF in Figure 4.3. To simplify notation, we will simply say "node $v$" to refer to "the node in the ordinary-graph analog that corresponds to value $v$".

In the case of an acyclic MRF, the marginal probability $\Pr(V = v)$ for random variable $V$ can then be computed (almost) exactly using *Personalized PageRank* [Hav03, LM06]. This is a random walk algorithm with random restarts from a single designated start node. Let $\{v_1, v_2, \ldots, v_{|V|}\}$ be the set of possible values for random variable $V$. Cohen's proposed approach is to execute Personalized PageRank $|V|$ times, each time for a different $v_i \in V$ as the start node. Let $\alpha_s$ be the product of the PageRank values of all anchor nodes for the personalized PageRank execution with start node $v_s$. Then $\Pr(V = v_s)$ is obtained as $\frac{\alpha_s}{\sum_{i=1}^{|V|} \alpha_i}$.

## 4.5.1 Reduced Acyclic MRF

We create a reduced version of the MRF by removing edges until the remaining graphical model is acyclic. For clarity, we will refer to the MRF as defined in Section 4.4.1 as *full-MRF*; and to its acyclic version as *reduced-MRF*. More formally, reduced-MRF is the maximum spanning tree of the full-MRF, which Equity computes using Kruskal's algorithm. Ideally we would like to remove edges that have little impact on the marginal probabilities of the $Y$-variables. This impact is determined by the potential functions, which are difficult to learn due to lack of labeled training data. We therefore resort to a heuristic based on *priorities of edge types*.

Edge types are defined as in Section 4.4.1. Since surface form has a strong impact on the choice of semantic target, all *mention-target* edges have highest priority and will never be removed. For the other five edge types, Equity explores all $5! = 120$ possible sort orders of their priorities. Edges of the same type are prioritized based on the sum of the individual weights of the corresponding edges in the ordinary-graph analog (see Section 4.5.3). The winner is selected based on performance on a small validation set of labeled tables.

This approach removes a large fraction of edges. We also explore an alternative that does not remove edges from full-MRF, but is computationally more expensive.

## 4.5.2 Modified Construction for Full-MRF

Depending on table structure and content, full-MRF might not have any leaf variables. This in turn implies that the corresponding ordinary-graph analog might have no anchor nodes, and therefore the computation using PageRank values of anchor nodes would be undefined. (Cohen did not encounter this problem as he only considered acyclic MRF, which are guaranteed to have leaf variables.) Even if there are leaf variables, as in the case of reduced-MRF, the meaning of the product of the PageRank values of the anchor nodes is not clear. Hence we have to re-think (1) the choice of start nodes for personalized PageRank and (2) how to use the PageRank values to select the best semantic target for each $Y$-variable.

Due to their unclear role for MRF with cycles, Equity works with a slightly modified ordinary-graph analog where *all anchor nodes and their adjacent edges are removed.* On the resulting graph, the best semantic target for a random variable $Y_{i,j}$ with candidate set $\{y_1, y_2, \ldots, y_{|Y_{i,j}|}\}$ is determined by executing personalized PageRank with start node $m_{ij}$, i.e., the node for the mention in table cell $(i,j)$. Let $\beta_k$ refer to the PageRank value of node $y_k$. Equity returns the candidate $y_w$ with the largest $\beta$-value and estimates its probability of being the right answer as $\frac{\beta_w}{\sum_{k=1}^{|Y_{i,j}|} \beta_k}$. In general, the semantic candidate for table cell $(i,j)$ is determined by (1) running personalized PageRank with starting node $m_{ij}$ and (2) selecting that node $y_{i,j} \in Y_{i,j}$ with the highest PageRank among all semantic target candidates for cell $(i,j)$.

The approach is motivated by the following intuition. Since mention node $m_{ij}$ is directly connected to all semantic candidates for $Y_{i,j}$, starting there corresponds to a *prior*: greater edge weight results in correspondingly greater PageRank mass. The remainder of the graph then accounts for the effect of the table context. As closely related values of connected random variables will have edges of greater weight, the candidate that is well-connected to, and hence more coherent with, this context receives a greater PageRank value from those other edges.

### 4.5.3 Edge Weights

So far we have only specified the graph structure for personalized PageRank computation. Now we turn our attention to the edge weights. Instead of attempting to first learn the potential functions and then convert them to edge weights, we apply distant supervision using the KB and co-occurrence patterns in Wikipedia to determine those weights directly. Edge weights are defined by edge type. Each is the product of a type-specific weight vector and a feature vector, i.e., for an edge of type $i$ connecting values $u \in U$ and $v \in V$ of random variables $U$ and $V$, it is defined as

$$\mathbf{w}_i^T \mathbf{f}_i(u, v).$$

Due to the small number of labeled training cases, the number of parameters learned from these data has to be small. Hence for most edge types, the vectors are one-dimensional. We constrain all multi-dimensional weight vectors to only contain equal values. As a result, we only have a single *hyper-parameter* for each edge type. The hyper-parameters, each with a value between 0 and 1, are learned from a separately withheld and randomly selected validation set of labeled training tables. Equity performs a grid search to find the parameter combination with the best performance on the validation data. In the following, we introduce the edge weight features.

**Mention-target edges** connect a surface form to a semantic candidate item. For string mentions, we build on previous work and use features based on string similarity [LSC10] and popularity statistics from Wikipedia links [HYB$^+$11]. However, no previous work considered the relationship between surface form and semantic target for quantities. Depending on the sort of mention and semantic candidate, we use the following 1-dimensional feature vectors:

- $m_{ij}$ is a string mention; $Y_{i,j} = c$, where $c$ is a concept or class: Based on the intuition that surface form and semantic target are often textually similar for concepts and classes, we use the Jaro-Winkler distance between $m_{ij}$ and $c$.

- $m_{ij}$ is a string mention; $Y_{i,j} = e$, where $e$ is an entity: We use the *popularity-based prior* that was found most effective for named entity disambiguation by Hoffart et al. [HYB$^+$11]. For string mention $m_{ij}$ and candidate entity $e$, it is defined as the number of Wikipedia links with anchor text $m_{ij}$ that refer to $e$, divided by the total number of Wikipedia links with this anchor text.

- $m_{ij}$ is a numerical mention; $Y_{i,j} = q$, where $q$ is a quantity: We propose a new feature based on links in Wikipedia tables that refer to Wikipedia articles about units of measurement. Let $m'_{ij}$ be the *unit component* of $m_{ij}$, i.e., the leftover after removing the magnitude. Then the feature is defined as the number of links in Wikipedia tables that have anchor text $m'_{ij}$ and refer to a unit that is associated with quantity $q$, divided by the total number of links in Wikipedia tables with this anchor text and referring to any unit of measurement.

**Header-cell edges:** Given header and cell mentions $m_{0j}$ and $m_{ij}$, the 1-dimensional feature vector contains the number of Wikipedia tables where these surface forms co-occur in header and non-header cell, respectively, of a column.
**Same-row edges:** Given same-row mentions $m_{ij}$ and $m_{ik}$, the 1-dimensional feature vector contains the number of Wikipedia tables where these surface forms co-occur in any row.
**Same-column edges:** Given same-column mentions $m_{ij}$ and $m_{kj}$, the 1-dimensional feature vector contains the number of Wikipedia tables where these surface forms co-occur in any column (excluding the header).
**Same-value edges:** We use a 1-dimensional feature vector with value equal to the Jaro-Winkler distance between the two surface forms.
**Candidate-candidate edges:** Equity uses a *relatedness* feature based on Wikipedia link co-occurrences. The relatedness of two semantic items is computed as the number of Wikipedia pages in which they co-occur, normalized so that the maximum value is equal to 1. In case of edges connecting a class and entity semantic targets in the same column, the weight of the edge is updated by the relation between the class and the entity's classes. That is, the edge is weighted using a mixture of the candidates' relatedness and the classes overlap measures.

## 4.6 Implementation

### 4.6.1 Mention Recognition

For detecting mentions in tables and their contexts, we use the state-of-the-art Stanford NER tagger [FGM05]. However, this tool was designed for natural-language sentences as input and shows low recall on tables. Hence we developed an extended mention recognition system as part of the Equity system. Our tool is centered on a rule-based classifier that uses regular expressions to detect occurrences of classes, concepts, entities and quantities in tables and their surrounding text. The major steps are as follows.

**Classify columns:** A column can be classified as numerical, textual or mixed. We run our regular expression classifier on each cell of the column, and then use majority voting.

**Detect concepts and classes:** We annotate the headers of numerical columns as mentions of concepts, and the headers of textual columns as mentions of classes. For mixed columns we base the decision on the majority of their cells.

**Detect quantities:** We use regular expressions to identify mentions of quantities, and to decompose them into value and unit.

**Detect entities:** We use the Stanford NLP parser to detect all possible noun phrases in a textual cell and mark them as entity mentions.

**Enrich mentions:** We further augment mentions in a cell with all sub-strings of the detected noun phrases. We repeatedly call the text-based entity-linking tool AIDA [HYB$^+$11] with each sub-string as the sole input, to determine candidate entities. Then we filter the mention candidates, to select the maximum-length non-overlapping mentions with non-null candidates.

### 4.6.2 Candidate Search

**Quantity Candidates:** We start by finding candidates from our QKB for the unit part of the quantity mention. However, the unit is not always included in the cell. Therefore, we perform an expansion search for quantity candidates. We look for possible units, first in the cell, then in the column header, and eventually in the table context. Moreover, for quantities that do not have units, such as votes or scores, we use the column header to identify the measure.

**Entity Candidates:** We use the AIDA web service[3], to retrieve a set of candidate entities for each mention. The input is a set of possible mentions, and the output is a set of top-k candidate entities based on a simple popularity prior.

**Class Candidates:** We use Locality Sensitive Hashing (LSH) to retrieve candidate classes for mentions; then we filter them based on Jaro-Winkler distance between class name and mention.

**Concept Candidates:** Similar to the previous case, we use LSH followed by a filtering step using Jaro-Winkler distance. Furthermore, we add candidate measures from the QKB as candidates, as some column headers have labels like frequency, width or height. We ensure that the candidate units for the column cells are compatible with the candidate measure for the column header when we perform the final inference over the graph.

### 4.6.3 Random Walk Algorithm

As explained in section 4.5, we construct 2 types of graphs: one for the full-MRF and one for the reduced-MRF. We re-scale all edge weights by multiplying them with the hyper-parameters for the respective edge type. We use the power-iteration technique to

[3] https://gate.d5.mpi-inf.mpg.de/aida/service/disambiguate

compute the stationary vector of random walks with restart on the graph as described in Section 4.5. We check convergence based on the relative ordering of the semantic items, following [Hav03].

## 4.7 Experimental Results

We evaluated the effectiveness of the Equity system on a systematically sampled and fully annotated collection of 69 Web tables (with context). Equity is also compared to previous work on three larger collections with up to 6,085 tables.

### 4.7.1 Setup

**Dataset:** We build a corpus of web tables from two different sources: the Google Tables API[4] and the Wikipedia tables corpus from [BND15][5]. Note that fully annotating a table with the ground-truth for all mentions is a labor-intensive task requiring specialist knowledge. Hence we opted for annotators from our lab and aimed for a sampled and relatively small, but fully annotated collection. We wanted to cover a variety of domains: environment, finance, sport, health and politics. To get this diversity, we used a handful of keywords per domain to search for tables from the two sources and then randomly sampled medium-sized tables from the search results. In total, we obtained 69 tables this way: 63 from Wikipedia articles and 6 from various web sites. Table 4.2 shows statistics about the test dataset.

Table 4.2: Statistics for Test Data Collection

| Average Number per Table | Various Websites | Wikpedia |
|---|---|---|
| # rows | 13.57 | 10.86 |
| # columns | 5.00 | 6.00 |
| # numerical columns | 1.57 | 3.02 |
| # entity mentions | 17.38 | 30.28 |
| # quantity mentions | 23.34 | 29.29 |
| # class & concept mentions | 2.28 | 4.29 |

**Hyper-Parameter Tuning:** We used a withheld set of 7 tables (disjoint from the test data) from a variety of domains (health, finance, etc.), in order to tune the hyper-parameters of Equity: six weights for different kinds of edges (see Section 4.5.3). We performed a grid search over 1000 combinations to obtain the best hyper-parameters for the full-MRF model and, separately, for the reduced-MRF.

[4] https://research.google.com/tables
[5] http://websail-fe.cs.northwestern.edu/TabEL/

Table 4.3: Micro-averaged Precision of Mention Detection

| | Number of Mentions | | | Micro-average Precision % | | |
|---|---|---|---|---|---|---|
| Type | Table | Cxt. | All | Table | Cxt. | All |
| class | 109 | 0 | 109 | 62.4 | - | 62.4 |
| concept | 284 | 0 | 284 | 70.1 | - | 70.1 |
| date | 160 | 165 | 325 | 100.0 | 97.0 | 98.5 |
| entity | 1628 | 0 | 1628 | 49.7 | - | 49.7 |
| location | 221 | 188 | 409 | 98.2 | 94.7 | 96.6 |
| money | 0 | 7 | 7 | - | 100.0 | 100.0 |
| organization | 116 | 225 | 341 | 85.3 | 60.9 | 69.2 |
| percent | 19 | 35 | 54 | 100.0 | 100.0 | 100.0 |
| person | 86 | 55 | 141 | 89.5 | 56.4 | 76.6 |
| quantity | 2011 | 272 | 2283 | 82.7 | 58.1 | 79.8 |
| Total | 4634 | 947 | 5581 | 71.5 | 74.6 | 72.0 |

### 4.7.2 Results

We report and discuss the effectiveness of Equity for mention detection and for mention linking. Our performance measure is precision, micro-averaged over all mentions of all 62 tables of the test collection. The total number of mentions evaluated is 5,581.

#### Mention Detection

Table 4.3 shows the precision of the mention recognition stage, broken down into mentions in tables and mentions in the contexts (Cxt.) and the total over both.
The Stanford NER Tagger alone was able to detect 1,277 mentions (out of which 1,120 are correct mentions) of the following types: date, location, money, organization, percent and person—mainly in the context. The Equity mention detector additionally identified 4,304 mentions (out of which 2,898 are correct mentions) of the following types: class, concept, entity, and quantity. In total, our method discovered 5,581 mentions in tables and their contexts. The micro-averaged precision is about 72%. Table 4.3 breaks this down onto the different kinds of mentions. The weak points are mentions of classes and mentions of entities other than location, organization and person (i.e., the row "entity" in the table)—mostly products or other artifacts (e.g., movies). On the other hand, we achieve almost 80% precision for quantities, which is the main target of this research.

#### Mention Linking

Table 4.4 gives the micro-averaged precision that Equity (in its reduced-MRF configuration) achieves for mapping mentions to semantic items in the KB. We consider only correctly recognized mentions here, as the errors from the previous stage of mention detection would lead to trivial follow-up errors. In total, we evaluated 4,018 mentions at this mention linking stage. Overall, we obtain around 92% precision in linking quan-

Table 4.4: Micro-averaged Precision of Mention Linking, Considering all Mentions (Entities, Concepts, Classes, Quantities) in Table and Context

| | Number of Mentions | | | Micro-average Precision % | | |
|---|---|---|---|---|---|---|
| Type | Table | Cxt. | All | Table | Cxt. | All |
| class | 68 | 0 | 68 | 82.4 | - | 82.4 |
| concept | 199 | 0 | 199 | 84.9 | - | 84.9 |
| date | 160 | 160 | 320 | 100.0 | 100.0 | 100.0 |
| entity | 809 | 0 | 809 | 81.0 | - | 81.0 |
| location | 217 | 178 | 395 | 94.9 | 96.6 | 95.7 |
| money | 0 | 7 | 7 | - | 100.0 | 100.0 |
| organization | 99 | 137 | 236 | 89.9 | 95.6 | 93.2 |
| percent | 19 | 35 | 54 | 100.0 | 100.0 | 100.0 |
| person | 77 | 31 | 108 | 98.7 | 96.8 | 98.1 |
| quantity | 1664 | 158 | 1822 | 97.5 | 60.8 | 94.3 |
| Total | 3312 | 706 | 4018 | 92.2 | 89.4 | 91.7 |

tities, entities, classes and concepts. For entities alone we achieved 88%, and for all kinds of quantities 93%. Table 4.4 shows the break-down for the different kinds of input mentions. Here, the row "quantity" refers to all numeric mentions excluding those of type date and money. The latter two are mostly detected by the Stanford NER tagger, whereas most of the remaining quantity mentions are only detected by our method. The numbers show that the quantities detected by the Equity-specific method exhibit even higher precision for mention linking, around 94%. As dealing with quantities has been the main target of this work, we consider the observed performance as very good. For linking entity mentions, the precision is well above 90% for location, organization and person. Similar to the mention detection stage, the remaining kinds of entities—for example, products such as car models—are a somewhat weaker point. Precision for these is around 81% ("entity" row in Table 4.4).

### Comparison with Other Systems

Although the specific focus of our work is on quantities in tables, we also performed comparisons to prior work on entity linking in tables, using various annotated datasets from these works. We compare two configurations of our Equity system against the systems proposed in [LSC10] and [BND15], restricting all inputs to entity mentions in table cells (i.e., no context, no quantities).
Table 4.5 shows the results, for the following datasets, with results for baselines as reported in the literature:

- web_manual [LSC10], a set of 371 web tables with a total of 9,239 mentions,
- wiki_links [LSC10] with 6,085 Wikipedia tables containing a total of 131,807 mentions, and

Table 4.5: Entity Linking for Different Datasets, Considering only Entity Mentions in Tables, but not in the Context.

| | Micro-average Precision % | | | |
|---|---|---|---|---|
| Data Set | Equity | | Limaye et al. | TabEL |
| | full-MRF | red-MRF | [LSC10] | [BND15] |
| web_manual | 86.11 | 85.11 | 81.37 | 89.41 |
| wiki_links | 96.39 | 96.24 | 84.28 | 97.16 |
| wiki_random | 83.04 | 82.98 | – | 96.17 |
| Equity corpus | 84.11 | 85.36 | – | – |

- wiki_random [BND15] with 3,000 randomly selected Wikipedia tables and about 40,000 mentions.

Equity outperforms [LSC10] on all datasets. In comparison to TabEL [BND15], Equity performs nearly as well on the larger wiki_links collection, which has many tables from prominent Wikipedia articles. On wiki_random, on the other hand, Equity is substantially outperformed by TabEL.

The reason is that this dataset contains many tables from the long tail of Wikipedia with lower curation quality. In particular, these tables contain a substantial fraction of misleading anchor texts. For example, the mention 'Oslo' appears with a link to 'Bislett_Stadium', and 'BMW' is linked to 'BMW_in_Formula_One'. A supervised learning method like TabEL can handle such peculiar instances better. Recall that Equity is designed for coping with quantities and entities together, as opposed to focusing on entities alone.

### 4.7.3 Ablation Study

To study the importance of the different edge types in the graph models, we performed an ablation study where we selectively disabled some of them in both full-MRF and reduced-MRF. Table 4.6 shows the results on mention linking, limited to entities because quantities are only annotated in the Equity corpus.

We observe that the reduced-MRF and the full-MRF have almost the same precision in all configurations. However, the reduced-MRF variant of Equity is much faster (see below). The results on leaving out specific types of edges show that our methods are robust. Missing certain cues affects the output quality only slightly. On the other hand, this also shows that the wiki_links corpus, the by far largest of the datasets, is a fairly easy test case. The other two corpora are rather small; hence there is no final conclusion yet on the importance of edge types.

### 4.7.4 Error Analysis

Many of the linking errors we observed are due to the absence of specific measures or units in our QKB, or caused by very ambiguous column headers. Examples for the latter are

Table 4.6: Ablation Study on Mention Linking, Considering only Entity Mentions in Table and Context

| | Micro-average Precision % | | |
|---|---|---|---|
| | web_manual | wiki_links | Equity corpus |
| full-MRF | 86.11 | 96.39 | 86.69 |
| red-MRF | 85.11 | 96.24 | 87.79 |
| full-MRF w/o cand-cand | 84.81 | 96.17 | 86.63 |
| red-MRF w/o cand-cand | 84.81 | 96.17 | 87.92 |
| full-MRF w/o table-struct. | 84.92 | 96.22 | 86.37 |
| red-MRF w/o table-struct. | 85.09 | 96.25 | 87.86 |
| full-MRF w/o same-value | 86.11 | 96.39 | 86.69 |
| red-MRF w/o same-value | 85.11 | 96.24 | 87.79 |

"$\eta(Observed)$" for measuring the thermal efficiency of a heat engine and "Nat." referring to nationalities with abbreviations of countries such as "GRE" (for Greece, presumably). We also observed cases where the column header gives misleading information such as "Density ($area/km^2$)" while the values in that column indicate population densities in $people/km^2$. Also, Equity sometimes misclassifies a column as numeric; an example is the column "Pollutant" with values like "CO2", "PM10" etc. Conversely, we occasionally miss out on a numeric column; an example is "Govt." with numbers referring to a country's governments at different periods.

### 4.7.5 Run-Time Analysis

We implemented the Equity system in Java using a Postgres database as a KB repository, and measured its run-time on a server with 4×4 Intel Xeon CPU E5-2667 v3 @ 3.20GHz cores, setting the maximum memory allocation pool for Equity to 40GB. The run-times for the reduced-MRF variant of Equity are 15 times faster than those for the full-MRF. Further analyzing the time spent in different components shows that the dominant factors are (i) SQL calls to fetch candidates and associated statistics from the KB and (ii) web service calls to obtain auxiliary information from AIDA. Discounting these components, which could be re-implemented in a much more light-weight manner, the time to process one table is about 2 seconds on average for the reduced-MRF variant of Equity. With some code tuning, this could be further optimized.

## 4.8 Summary

This chapter addressed the task of fully canonicalizing mentions in ad-hoc tables and their surrounding contexts, by linking mentions of entities, classes, concepts and quantities to a knowledge base. To this end, we devised an MRF model, distantly supervised by relatedness measures from a KB, then derived a reduced acyclic MRF, and finally cast the inference over this light-weight model into an efficient algorithm based on random walks over normal weighted graphs. Our experiments with a collection of Web and Wikipedia tables demonstrate that particularly the detection and linking of quantities—our main target—works very well. The reduced-MRF method achieves an overall linking precision of about 92%, and even 93% for quantity mentions. The Equity system introduced in this chapter is a first building block in our longer-term research towards making sense of Web tables and spreadsheets in enterprises.

# 5 BriQ: Understanding the Relation Between Quantity Mentions in Text and Tables.

## 5.1 Introduction

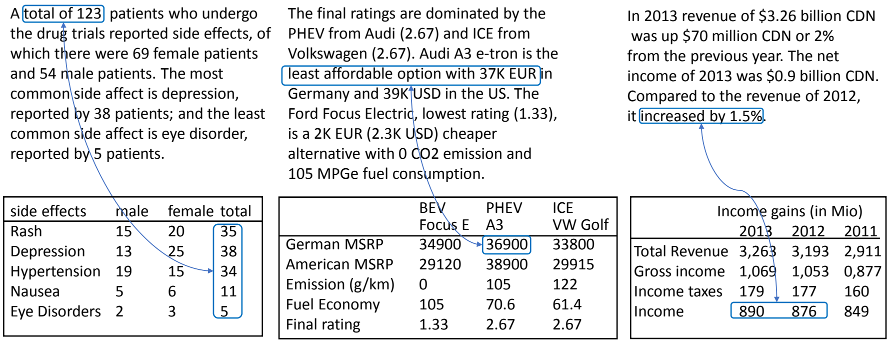
### 5.1.1 Motivation

Tables not just epitomize relational databases, but are also widely used to represent data on the Web (embedded in HTML pages) and in enterprises (in spreadsheets). Unlike in databases, these tables are often created in an ad-hoc manner, without proper schema design and with highly heterogeneous formats of attribute values. Therefore, the interpretation of tables, by human analysts and other users, often hinges on additional text that discusses the table content.

Figure 5.1 shows excerpts of Web pages from the domains of health, environment and finance. The currency of the financial numbers in Figure 5.1c becomes clear only when reading the text. Likewise, it is the text of Figure 5.1b that points the user to the most expensive of the three cars.

To make sense of tables, it is thus crucial that table rows, columns and individual cells are connected with relevant snippets in the surrounding text. For entire rows and for cells with names of products, companies, locations, etc., this is the problem of *entity linking* [SWH15]. Specific methods for tables as input have been developed [BND15, LB17, LSC10]. However, this does not capture the *quantities* in individual cells. Linking quantities has been addressed in [IRW16, SC14], but these works assume that a knowledge base or reference system of canonicalized quantities (with standardized measures, proper units, etc.) is available. In practice, knowledge bases for quantities are merely small and limited to special domains.

In this paper, we aim to link quantities without making such assumptions. We do so by linking table cells with relevant pieces of the text that accompanies a table. This supports users in two ways. First, in going from tables to text, they obtain explanations of the mere numbers in cells and their relevance for the topic at hand. Second, in going from text to tables, the user can drill down on statements in terms of detailed numbers. Figure 5.1 illustrates these benefits by the overlaid bidirectional edges. Quantity alignment links the text to data from the tables, and vice versa. Hence, it can be combined with entity linking techniques to augment knowledge bases. Furthermore, quantity alignment creates an opportunity for advanced automatic text summarization [NM12, GG17], which currently does not include table data. Once our system identifies aligned quanti-



A total of 123 patients who undergo the drug trials reported side effects, of which there were 69 female patients and 54 male patients. The most common side affect is depression, reported by 38 patients; and the least common side affect is eye disorder, reported by 5 patients.

| side effects | male | female | total |
|---|---|---|---|
| Rash | 15 | 20 | 35 |
| Depression | 13 | 25 | 38 |
| Hypertension | 19 | 15 | 34 |
| Nausea | 5 | 6 | 11 |
| Eye Disorders | 2 | 3 | 5 |

a) Example about Health

The final ratings are dominated by the PHEV from Audi (2.67) and ICE from Volkswagen (2.67). Audi A3 e-tron is the least affordable option with 37K EUR in Germany and 39K USD in the US. The Ford Focus Electric, lowest rating (1.33), is a 2K EUR (2.3K USD) cheaper alternative with 0 CO2 emission and 105 MPGe fuel consumption.

| | BEV<br>Focus E | PHEV<br>A3 | ICE<br>VW Golf |
|---|---|---|---|
| German MSRP | 34900 | 36900 | 33800 |
| American MSRP | 29120 | 38900 | 29915 |
| Emission (g/km) | 0 | 105 | 122 |
| Fuel Economy | 105 | 70.6 | 61.4 |
| Final rating | 1.33 | 2.67 | 2.67 |

b) Example about Environment

In 2013 revenue of $3.26 billion CDN was up $70 million CDN or 2% from the previous year. The net income of 2013 was $0.9 billion CDN. Compared to the revenue of 2012, it increased by 1.5%.

| | Income gains (in Mio) | | |
|---|---|---|---|
| | 2013 | 2012 | 2011 |
| Total Revenue | 3,263 | 3,193 | 2,911 |
| Gross income | 1,069 | 1,053 | 0,877 |
| Income taxes | 179 | 177 | 160 |
| Income | 890 | 876 | 849 |

c) Example about Finance

Figure 5.1: Examples of Web Tables with Explanatory Texts

ties, it is possible to determine which table rows, columns, and individual cells are referenced by the text summary—so that they can be added to it. And since our approach distinguishes between simple single-cell references and aggregates, it can provide hints to an automatic text summarizer. For instance, knowing that one sentence references a row sum, while another discusses individual values in the same row, the summarization algorithm could decide to include the former in the summary, but not the latter.

### 5.1.2 Problem Statement

We formalize the problem of bridging quantities in tables and text as a *quantity alignment problem*: For a text document with one or more tables,

- detect quantity mentions in text that refer to table cells
- and map these mentions to their proper cells.

Here, quantity mentions are textual expressions that contain numbers, but also include phrases that refer to aggregation, ranking and change rates. For example, in Figure 5.1a, the phrase "total of 123 patients" refers to an aggregate value, namely, the sum of the values in the sales column. In Figure 5.1b, "the least affordable option" refers to the maximum price in a column, and in Figure 5.1c, "increased by 1.5%" refers to the rate of change.

Although the problem resembles that of entity linking, it is more challenging (and unexplored) for several reasons:

- There is no explicit knowledge base that contains all targets (namely, entities) of the desired mapping. In our setting, the targets of the alignment are the values in table cells (often in incomplete or noisy formats), and the number of possible mention-cell pairs that could be aligned is huge.
- Quantity mentions in text often differ in their formats from their counterparts in table cells. For example, "37K EUR" (in Figure 5.1b) refers to "36900" in a

cell with row header "German MSRP" (in a rotated table). Such *approximate mentions* are frequent.

- *Aggregate quantities* that appear in text in forms such as "total of 123 patients" (Figure 5.1a) are not necessarily present in any table cell, but simply correspond to a column total. In such cases, the text mention should be aligned with all cells of the respective column to be summed up.
- Other forms of *calculated quantities* like maximum values, differences, change rates, etc., require alignments of text phrases like "least affordable" (Figure 5.1b), "up $70 million Cdn" (Figure 5.1c), "increased by 1.5%" (Figure 5.1c) etc. with a set of cells, typically in the same row or column.

### 5.1.3 Proposed Solution

**Our Approach:** For aligning quantity mentions in text with cells in tables, we have developed a full-fledged system called *BriQ* (for "*Bri*dging *Q*uantities in tables and text"). The core of *BriQ* is a hybrid algorithm for mapping mentions onto cells, by first learning a supervised classifier that accepts or drops mention-cell candidate pairs. The classifier not only serves to prune the search space, but also yields a prior for additional unsupervised steps based on random walks over appropriately weighted candidate graphs. The latter steps harness joint inference over the full alignment of all mentions in a document and all candidate cells in one or more tables within the document. To minimize dependence on hard-to-obtain training data and to cope with larger scale, the joint inference is unsupervised.

Our methods pay particular attention to the challenges of aggregated (e.g., column totals) and calculated quantities (e.g., change rates). We do this by carefully generating candidates in the form of "virtual cells," standing for cell combinations such as table columns or same-row cell pairs. For example, a virtual cell is generated for a column total even if the table itself does not explicitly show the total. We devise various techniques to prune the number of such virtual candidate cells, to ensure computational tractability and to control spurious matches.

**Contributions:** Salient points of this paper are:

- We introduce and formalize the novel problem of *quantity alignment* for Web pages that contain text and one or more ad-hoc tables.
- We present the *BriQ* system[1], including a two-stage algorithm for computing alignments, with a trained classifier as a prior and unsupervised, random-walk-based, techniques for global inference.
- Comprehensive experiments, with a large collection of Web tables and high-quality ground-truth annotations, demonstrate the practical viability of the *BriQ* method and its superior performance over two baselines.

[1] code and dataset available at: `https://www.mpi-inf.mpg.de/briq/`

## 5.2 Related Work

**Web Tables:** Schema-less ad-hoc tables embedded in Web pages have first been brought to the database research agenda by the seminal work of [CHW$^+$08, CHM11, MAAH09]. The focus of this work was on enabling search engines to include tables as results of keyword queries. Follow-up work tackled various forms of light-weight data integration, like matching names in table headers against queries, matching attributes of different tables with each other, and inferring approximate schemas (e.g., [LSC10, PS12, VHM$^+$11, LB17]).

**Entity Linking:** Mapping names of people, places, products, etc. onto canonicalized entities in a knowledge base has received great attention in the literature; a recent survey is given by [SWH15]. This work has mostly focused on surface names in text documents. The most notable exceptions that addressed names in tables (in combination with mapping column headers) are [LSC10, BND15, RLB15]. Their methods for entity linking vary from context-similarity-based rankings and simple classifiers to advanced forms of probabilistic graphical models for joint inference over a set of mentions.

**Quantity Extraction:** Recent work has addressed the task of recognizing quantities in text and extracting them as proper mentions (including units, reference range, etc.) [SC14, SA15, NUPP16, IRW16, MMM$^+$16, RVR15, SPM17, AS18]. These methods are based on pattern matching and/or machine learning models like Conditional Random Fields. However, only [SC14, IRW16] go beyond mere extraction and aim to canonicalize quantity mentions by linking them to a knowledge base of measures and units. In doing this, they rely on an explicit – in their cases small and manually crafted – knowledge base, though. This approach is limited in scope and does not scale to the wide diversity of quantities in large collections of Web tables. The BriQ approach, on the other hand, does not require an explicit knowledge base and copes with the full scope of possible inputs.

**Coreference Resolution in NLP:** A very different domain with resemblance to our problem of quantity alignment is the task of coreference resolution in natural language processing (NLP). Given a text document with entity names as well as underspecified expressions like pronouns ("he", "she", "her" etc.) and common noun phrases (e.g., "the lead singer", "the founder of Apple" etc.), the task is to compute equivalence classes of coreferences. For example, pronouns should be linked to a name in the same or a preceding sentence. State-of-the-art methods for this problem are mostly based on rules and/or machine-learning techniques for clustering or classification (e.g., [HK09, LCP$^+$13, DK14, CM16b, Ng17]). None of these considers mentions of quantities, though.

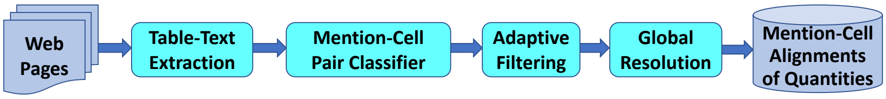

Web Pages
Table-Text Extraction
Mention-Cell Pair Classifier
Adaptive Filtering
Global Resolution
Mention-Cell Alignments of Quantities

Figure 5.2: *BriQ* System Architecture

## 5.3 System Overview

### 5.3.1 Computational Model

The *BriQ* method takes the following inputs:

- A piece of text, like a (part of a) web page, with a set of $m$ text mentions of quantities $X = \{x_i : i = 1, \ldots, m\}$.
- A table $q$ with $r$ rows and $c$ columns and a set of $n$ mentions of quantities $T = \{t_j : j = 1, \ldots, n\}$.

**Text mentions** include terms containing numbers or numerals such as "123 patients", "37K EUR", "1.5%" or "twenty pounds". To focus on informative quantities, we eliminate date/time, headings (such as "Section 1.1"), phone numbers and references (such as "[2]", "Win10").

**Table mentions** include two types of quantities. The first are explicit **single-cell mentions**, such as '36900' in Table 5.1c, second row, third column. Given a table with $r$ rows and $c$ columns we have at most $r \cdot c$ single-cell quantity mentions. The second type of are **composite quantity mentions** (or *virtual-cell* mentions), computed as an aggregation of one or more table cells, such as '123', the sum for the fourth column in Table 5.1a.

We consider a broad range of **aggregate functions** that take two or more table cells as input and produce a single quantity:

- *Sum*: given $q$ quantities, $\text{sum}(y_1, .., y_q) = \sum_{i=1}^{q} y_i$
- *Difference*: given 2 quantities, $\text{diff}(a, b) = a - b$
- *Percentage*: given 2 quantities, $\text{pct}(a, b) = \frac{a}{b} \cdot 100\%$
- *Change Ratio*: given 2 quantities, $\text{ratio}(a, b) = \frac{a-b}{a}$
- *Average*: given $q$ quantities, $\text{avg}(y_1, ..., y_q) = \frac{\sum_{i=1}^{q} y_i}{q}$
- *Max* or *Min*: given $q$ quantities, $\max_{i=1}^{q} y_i$ or $\min_{i=1}^{q} y_i$

These composite quantities may be present in a table already, but we also consider them if they are not explicit as the surrounding text may still refer to totals, diffs, etc. Hence the notion of *virtual-cell mentions*. In our experience, aggregates almost always refer to cells in the same row or column. More precisely, sum, average, min, and max tend to be computed for an entire row or column, resulting in $O(r + c)$ composite quantity candidates in the table. Since difference, percentage, and change ratio aggregate two values in a row or column, there are $O(\binom{r}{2} + \binom{c}{2})$ candidates for them. This leads to a quadratic (in table size) search space for the alignments, which is prohibitive for large tables. We will present adaptive filtering techniques for carefully pruning this search space.

Note that this model can be generalized by considering aggregations over other subsets of table cells, and even cells in different tables. For example, the text in Figure 5.1c could possibly refer to "the total income of the last two years," which is the sum of two cells (in the 2013 and 2012 columns) rather than a row total. With this generalization, the search space of the alignment problem would further increase, becoming exponential in table size already when arbitrary subsets of cells in a row or column are considered. The BriQ framework can handle this extended setting as well, and we studied it experimentally. It turned out, however, that such sophisticated cases are very rare, and hence did not have any impact on the overall quality of the BriQ outputs. For run-time efficiency, we consider only the case where sums and averages are restricted to entire rows or entire columns or two cells in the same row or same column, leaving the rare cases for future work.

The BriQ framework can handle a broad range of aggregation functions. However, in our experiments we only consider aggregations that appeared in at least 5% of the tables, because we need a sufficient number of examples to train and evaluate our models. Therefore, we only consider the following four aggregations in our experiments: sum, difference, percentage and change ratio.

For aligning quantity mentions between text and table, we aim to compute as output a subset of mention pairs $\langle x_i, t_j \rangle$ where $x_i \in X$ is a text mention and $t_j \in T$ is a table mention, including virtual cells for composite quantities. These pairs should denote the same quantity with high confidence. For the examples in Figure 5.1, the algorithm output should include the following pairs:

- ⟨ "total of 123", sum('35','38','34','11','5') ⟩,
- ⟨ "least affordable option with 37K EUR", '36900' ⟩ ,
- ⟨ "increased by 1.5%", ratio('890','876') ⟩.

BriQ also returns the locations of the mentions, which we omitted here for the sake of presentation. Note that alignments include approximate values such as "37K EUR" and composite quantities that are not explicitly present in the table, such as 'ratio('890','876').' The alignment would ideally be a total mapping, covering all text mentions in the input. However, realistic cases may contain numbers in text that do not refer to any table—so we compute a partial mapping.

### 5.3.2 BriQ Architecture

Figure 5.2 gives a pictorial overview of the *BriQ* system architecture. In the following, we outline each of the shown components.

### Table-Text Extraction

This module takes as input a web page and splits it into coherent segments, which we refer to as *documents*. Each document consists of a sequence of paragraphs and one or more tables to which the text refers. For each document, quantity mentions are extracted from the text and the tables, using regular expressions. Virtual cells—for aggregated quantities—are automatically generated by considering: (i) all rows and columns for totals; and (ii) all pairs of cells in the same row or column for difference, percentage, and change ratio.

### Mention-Cell Pair Classification

This module first computes features for each text mention and each table mention by analyzing surrounding context. Also, similarity-based features are computed for each pair of text mention and table cell that could be a candidate pair for alignment. We use manually annotated web pages with ground-truth alignment to train a binary classifier that accepts or rejects candidate pairs.

The classifier operates *locally* in the sense that it predicts the alignment confidence for each mention-cell pair in isolation, i.e., it does not consider several mention-cell pairs together for joint inference. It serves two purposes: First, it enables the subsequent filtering step, which significantly reduces the number of candidate pairs. This is essential for achieving acceptable running time in the more expensive global resolution step. Second, it provides a prior for that global resolution step.

### Adaptive Filtering

This stage filters the classifier's output to arrive at a sufficiently small set of candidate pairs that the subsequent global resolution can handle. The filtering uses the confidence scores of the classifier, but also considers more sophisticated measures to adapt to the specifics of different situations.

### Global Resolution

This module takes as input the candidate mention-pairs from the classifier and outputs the final alignment of quantities between text and tables. It uses the classifier confidence values as prior weights, and employs global inference methods such as random walks over graphs to resolve the alignments.

## 5.4 Table-Text Extraction

Web pages, such as Wikipedia articles or product test reports, can be very long and cover a variety of thematic aspects, along with several tables. We therefore pre-process and split them into *coherent documents*. Since paragraphs form a natural unit in text for discussing a specific aspect, we use them as atomic building blocks. More precisely, we define a coherent document to be a paragraph together with all "related" tables from the

same Web page. Related tables are identified by computing pairwise similarities between all paragraphs and all tables in the page, and then selecting those with similarity above a threshold. We consider tokens in the entire content of the table including column headers and captions. Note that a paragraph may have more than one related table, and a table might be related to multiple paragraphs.
For each document, we extract all quantity mentions from both text and tables, using regular expression matching (e.g., '`\d+\s*\p{Currency_Symbol}`' for monetary values). Quantities are extracted from text as follows: first we identify and remove complex quantities that involve multiple parts, such as '5 ± 1 km per hour'. Then, we extract simple quantities, such as '$500 million' and '1.34%'. This order ensures that complex quantities are not erroneously split into several matches. For tables, we employ the same procedure and attempt to extract a single quantity mention per cell, together with its unit (if present). In addition, we also attempt to extract information about the unit from each row and column header, footer, and the caption. We normalize quantity mentions; for example '0.5 million' transformed to '500 000'.

## 5.5 Mention-Pair Classification

This stage of the *BriQ* system applies supervised learning to predict if a text mention does refer to a table mention so that they should be aligned. This binary classifier performs local resolution in the sense that it makes a prediction for an *individual* mention-pair, not taking into account dependencies between predictions made for *different* mention-pairs. Such couplings will be considered by the global resolution later, at much higher computational costs, however. The confidence scores of the classifier serve as prior weights for the joint inference at the global resolution stage.

### 5.5.1 Classification Algorithms

We use a **Random Forest (RF)** classifier for this purpose. RFs are among the most powerful classifiers that are not prone to overfitting. An RF classifier consists of an ensemble of decision trees, each trained on an independent bootstrap sample of the training data. The final prediction for an input is obtained based on the majority vote of the individual trees, returning the fraction of votes for the "related" class as the probability of the mention-pair being related. It has been shown that RFs yield well calibrated probabilities [NMC05, CNM06], which is important for our usage of RF outputs fed into the global resolution stage.

### 5.5.2 Features

We judiciously designed a variety of features that capture information a human reader would use in order to determine if text mention $x$ and table cell $t$ denote the same quantity. The alternative—automatic representation learning, e.g., with Deep Learning—was not viable for our problem due to the limited amount of labeled data and the high cost for obtaining it (see Section 5.8). Overall, we believe that the complexity of our problem

setting is better served by modeling informative features rather than solely relying on end-to-end learning with limited training data.
The most obvious basic feature is **surface form similarity**, $\mathrm{f}_1(x, t)$. We adopted the Jaro-Winkler distance measure to compute the string similarity between the surface form of the text mention against the table mention. We use Jaro-Winkler because it emphasizes a match at the beginning of the string, which is desirable when comparing quantity mentions. For example, a quantity mention "26.7$" in the text is closer to "26.65$" than to "29.75$".

### Context Features

**Local context word overlap**, $\mathrm{f}_2(x, t)$, measures the similarity between the local contexts of a pair of text and table mention. A window of $n$ words preceding and following the text mention is considered; for the table mention it is the full row and the full column content. The feature value is defined as the weighted overlap coefficient between the two bags of words. That is, we assign a weight to each word relative to its position. We use the following formula to compute the weight of a word $e$ at distance $d$ from the text mention:

$$\text{weight}(e) = 1 - \left( \frac{d}{\text{stepSize}} \cdot \text{stepWeight} \right),$$

where stepWeight is the discounted weight at each stepSize away from the text mention. Then, we compute the overlap coefficient using these weights. We tune $n$, stepSize, and stepWeight on the withheld validation dataset.

**Global context word overlap**, $\mathrm{f}_3(x, t)$, is similar to $\mathrm{f}_2(x, t)$, but uses the entire paragraph as the context of the text mention; and the entire table content as the context of the table mention.

**Local context phrase overlap**, $\mathrm{f}_4(x, t)$, measures the similarity between the *noun phrases* in the local context of text and table mention. The local context of the text mention is the sentence in which the text mention occurs; and for the table mention it is the full row and the full column content. For example, the noun phrase "segment profit' in Figure 5.3.

**Global context phrase overlap**, $\mathrm{f}_5(x, t)$, is defined analogously, but considers noun phrases in the entire paragraph as the global context of the text mention; and the noun phrases in the entire table as the global context of the table mention.

### Quantity Features

**Relative difference between normalized quantity values**, $\mathrm{f}_6(x, t) = \frac{|x-t|}{max(x,t)}$, reflects the numeric distance between mentions. Here, $x$ and $t$ denote the numerical values of the respective mentions, after normalization. In Figure 5.1 the normalized value of mention '37K EUR' is 37000.

**Unnormalized relative difference between quantities**, $f_7(x, t)$, is the relative difference of the values *without* normalization. For example, the unnormalized value of mention '37K EUR' is 37.

**Unit match**, $f_8(x, t)$, is a four-valued categorical feature that captures the degree to which the quantity units match. A *strong match* occurs when both mentions have a specified unit and these units match; a *weak match* when both mentions have no specified units; a *weak mismatch* when only one mention has a specified unit; and a *strong mismatch* when both mentions have a specified unit and these units do not match.

**Scale difference**, $f_9(x, t)$, is the difference in the orders of magnitude between two quantities. For example, the scale difference of '37000' and '37' is 3 (powers of ten).

**Precision difference**, $f_{10}(x, t)$, captures the difference in the number of digits after the decimal point. For example the precision difference of '1.5' and '1.543' is 2.

**Approximation indicator**, $f_{11}(x, t)$, reflects if the text mention is accompanied by a modifier indicating an approximation. This categorical feature can take on values 'approximate', 'exact', 'upper bound', and 'lower bound'. These are derived from text cues like "ca.", "about", "nearly", "more than", etc.

**Aggregate function match**, $f_{12}(x, t)$, is the degree to which the aggregate function for computing the value of the cell or virtual cell matches the kind of aggregation for the text mention as inferred from text cues. We implement this by looking up the words around the text mention in a dictionary that maps words to names of aggregate functions. (We set the neighborhood size by default to five words; but this could also be tuned on the validation data.) Analogous to the unit-match feature, there are four possible values: *strong match*, *weak match*, *weak mismatch*, and *strong mismatch.* For example in Figure 5.1(a) the inferred aggregation of mention 'total of 123 patients' is *sum* and it has a strong match with the aggregation of the virtual cell carrying the sum of the last column; and it has a strong mismatch with the virtual cell carrying the average of the last column.

## 5.6 Adaptive Filtering

As discussed in Section 5.3, it is essential for performance to significantly reduce the number of mention-pair candidates for global resolution, typically from 1000s of candidates to 100s for tractability of global inference algorithms. An obvious approach for the necessary filtering would be to use the classifier's confidence scores: we could retain only candidates above a certain threshold, or we could keep a certain number of highest-scoring candidates. While superficially appealing, it is rather rigid and disregards the need to handle different kinds of quantity mentions in a more flexible way, e.g., simple quantities vs. aggregate quantities. Hence we devised an *adaptive filtering* strategy as follows. First we develop a *text mention tagger* to predict the aggregation function for each text mention, or tag the mention as a single-cell match. Then, we prune mention-pairs based on this tagger's outcome. In a second step, we further prune mention-pairs based on *value difference* and *unit agreement*. Finally, we sort mention-pairs according to classifier scores, and select top-$k$ mention-pairs for each quantity mention based on *mention type* and *score distribution*.

### 5.6.1 Text-Mention Tagger

We tag text mentions, based on local features, with one of the following labels: *difference*, *sum*, *change ratio*, *percentage*, or *single cell*. Each of the four aggregation labels is associated with a small list of manually compiled cue words, such as "total, summed, overall, together" for *sum*, and analogous lists for the other tags. Likewise, words like "around, about, ca., approximately, nearly, almost" are considered as indicators for mention values being approximate. Observing the presence of such cue words in the proximity of a text mention is used for the following features that the tagger considers:

- *Approximation Indicator:* A categorical feature that specifies an approximation indicator accompanying the mention. The indicator is inferred from the immediate context of the text mention, where the immediate context is a window of 10 words around the text mention. The approximation indicator can take one of the following values: *approximate*, *exact*, *upper bound*, *lower bound*, and *none*.

- *Aggregation Function Features:* For each aggregation function we compute the count of supporting cue words in the mention context under the following scopes:

  1. *Immediate Context:* contains the tokens occurring within a window of 10 words around the text mention.
  2. *Local Context:* contains the tokens occurring in the same sentence with the text mention.
  3. *Global Context:* contains the tokens occurring in the same paragraph with the text mention.

- *Scale:* numerical value indicating the order of magnitude of the text mention.

- *Precision:* numerical value indicating the number of digits after the decimal point.

- *Unit:* a categorical feature that specifies the unit associated with the mention. The following is the list of units we consider: dollar, euro, percent, pound, and unknown unit.

- *Exact Match in Table(s):* the number of table mentions that exactly matches the surface form of the text mention. This number is summed up over all tables associated with the document.

We train the tagger, as a simple classifier, with a small labeled dataset, withheld from all other components and experiments. The tagger achieves high precision for the four kinds of aggregation functions. We intentionally optimize for high precision, at the expense of lower recall: the tagger sometimes confuses text mentions that match single cells with aggregates, incorrectly tagging them as *sum* or *diff* etc. However, this is not a problem as we can prune mention-pairs conservatively, by avoiding to eliminate single-cell matches at this stage. We use the tagger for the following pruning heuristics for mention-pairs:

- We keep all mention-pairs for single-cell mentions in tables.
- We prune aggregate mention-pairs if the aggregation function for the virtual cell does not match the predicted tag.

So this pruning step typically discards mention-pairs for all but one aggregation-function virtual cell, but keeps all mention-pairs with single cells. Further pruning steps for the single-cell cases are presented next.

### 5.6.2 Mention-Pair Pruning

**Pruning based on Value Difference and Unit Mismatch:** Based on the confidence scores returned by the mention-pair classifier, we prune mention-pairs whose numeric values differ by more than a threshold $v$ if the classifier score is less than $p$. We tune the values of $v$ and $p$ on the withheld validation dataset. In addition, for mentions with specified units, we prune mention-pairs that disagree in unit.

After these pruning steps, we select the top-$k$ candidate pairs for each text mention by the following criteria:

- **Mention Type:** We determine the mention type based on its surface form, context and the table mentions it potentially pairs with. A text mention can be *exact* (12.374), *approximate* (12.4) or *truncated* (12.3). First we rely on the context to determine the type of the quantity mention, by extracting quantity modifiers, such as ‘approximately’, ‘exactly’, and ‘about’. If the context is insufficient to determine the mention type, we compare the surface form of the mention to that of potential table mentions with high confidence returned by the classifier. Then, we determine the mention type by majority vote. For example, if most of the high-confidence potential table mentions exactly match the text mention, then the text mention is exact. For exact mentions we pick the top $k_{exact}$ mention-pairs and for approximate and truncated mentions we pick the top $k_{approx}$ mention-pairs, where $k_{exact}$ and $k_{approx}$ are tunable parameters.
- **Distribution Entropy:** We consider the distribution of confidence scores returned by the classifier for the pairs with the same text mention. Sometimes, this distribution can be so skewed that only few candidates need to be kept, whereas in other cases a large number of candidates could be near-ties and should all be kept. To reflect this intuition, we compute the entropy of the distribution, and adjust $k$

for the top-$k$ candidates in proportion to the entropy. We set a specific threshold for the entropy value, and for distributions with entropy falling below this threshold, we pick the top $k_s$ mention-pairs, otherwise we pick the top $k_l$ mention-pairs, where $k_s$ and $k_l$ are tunable parameters.

## 5.7 Global Resolution

The need for joint inference over candidate pairs for multiple text mentions arises due to dependencies among mentions, which need to be harnessed to resolve ambiguties. Consider the example in Figure 5.3. The text mentions “11%” and “13.3%” have exact matches in both of the shown tables, and local-resolution algorithms cannot infer the proper alignment. However, when considering these two mentions jointly with “60 bps” and “5%”, it becomes clear that all of these refer to the first table.

We have devised an unsupervised algorithm for this kind of global resolution. The algorithm encodes dependencies among mentions into a graph and uses random walks to infer the best joint alignment. We also considered an alternative algorithm based on constraint reasoning with Integer Linear Programming (ILP) and experimented with it, but that approach did not scale sufficiently well.

Sales were up 5% on both a reported and organic basis, compared with the second quarter of 2012. Segment profit was up 11% and segment margins increased 60 bps to 13.3% primarily driven by strong productivity and volume leverage.

Table 1: Transportation Systems

| ($ Millions) | 2Q 2012 | 2Q 2013 | % Change |
|---|---|---|---|
| Sales | 900 | 947 | 5% |
| Segment Profit | 114 | 126 | 11% |
| Segment Margin | 12.7% | 13.3% | 60 bps |

Table 2: Automation & Control

| ($ Millions) | 2Q 2012 | 2Q 2013 | % Change |
|---|---|---|---|
| Sales | 3,962 | 4,065 | 3% |
| Segment Profit | 525 | 585 | 11% |
| Segment Margin | 13.3% | 14.4% | 110 bps |

Figure 5.3: Example with Coupled Quantities

A human reader who glances a text mention and wants to identify to which table cell it refers, would first consider some matching values, including approximate or aggregate matches. These are candidate pairs, which we encode as edges in a graph, using the classifier’s confidence scores as prior edge weights. In case of ambiguity, the human user would then spot neighboring quantities in either text or table to assess the possible options and refine the hypothesis space of viable pairs. This would include looking at other quantities in textual proximity as well as other table cells in the same row or

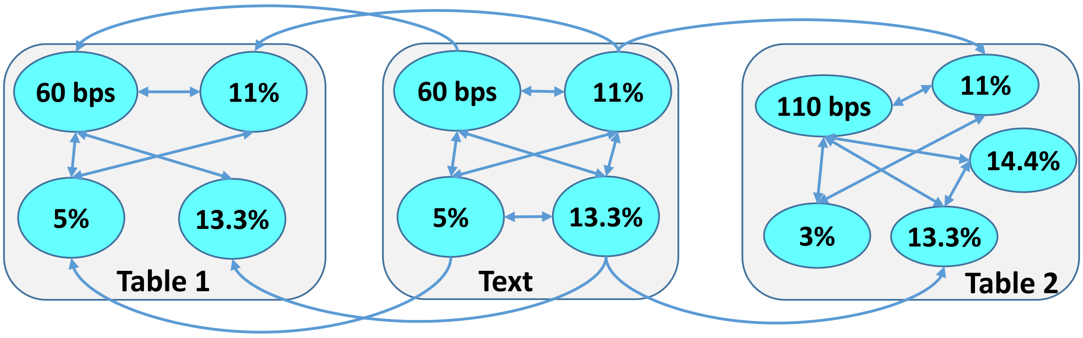


Figure 5.4: Fragment of the Graph for Figure 5.3

column. This intuition of human inference is cast into dependency edges between such context-related mentions, in both text and table—with weights based on relatedness strengths. Finally, the "strongest paths" connecting a text mention with table mention candidates determine the best alignment. We cast this intuition into a random walk over the weighted graph.

### 5.7.1 Graph Construction

We construct an undirected edge-weighted graph $G = (V, E)$ for each document:

- The node set $V$ consists of all quantity mentions in the document's text and tables.
- The edge set $E$ consists of three kinds of edges connecting related nodes: text-text edges, table-table edges, and text-table edges as explained below.

**Text-text edges:** There is an edge for each pair of text quantity mentions that are within a certain proximity or have similar surface forms. Edge weights ($W_{xx}$) are computed based on the following linear combination of proximity and string similarity:

$$W_{xx}(x_1, x_2) = \lambda_1 \mathrm{f}_{\mathrm{prox}}(x_1, x_2) + \lambda_2 \mathrm{f}_{\mathrm{strsim}}(x_1, x_2).$$

The hyperparameters $\lambda_1$ and $\lambda_2$ are tuned using grid search on the withheld validation dataset. We define $\mathrm{f}_{\mathrm{prox}}(x_1, x_2)$ as the number of tokens separating the two mentions, divided by the length of the document. String similarity $\mathrm{f}_{\mathrm{strsim}}(x_1, x_2)$ is defined as the Jaro-Winkler distance as described in Section 5.5.2.

**Table-table edges:** There is an edge for each pair of table quantity mentions in the same row or the same column of the same table. Edge weights $W_{tt}$ are set uniformly for each pair of table mentions sharing the same row or the same column.

**Text-table edges:** There is an edge for each pair of text and table mention that is kept by the adaptive filtering stage. Edge weights ($W_{xt}$) are set to the confidence scores returned by the classifier. This can be viewed as an informed prior for the global resolution stage.

After this initial graph construction, all edge weights are normalized to obtain a stochastic graph, via dividing each node's outgoing weights by the total weight of these edges.

### 5.7.2 Graph Algorithm

**Random walk with restart (RWR):** Random walks have been widely used for ranking and alignment tasks over graphs (e.g. [TFP06, LMC11, PJN13]), the most famous case being PageRank. In our setting, we employ random walks with restart: starting from a text mention, the graph is stochastically traversed, with a certain probability of jumping back to the initial node. This technique is also known as topic-specific or personalized PageRank [Hav03]. It approximates the *stationary visiting probabilities* $\pi(t|x)$ of table-mention node $t$ for walks starting from a text mention $x$. Our implementation iterates RWRs for each text mention until the estimated visiting probabilities of the candidate table mentions change by less than a specified convergence bound. This way we can rank the candidate table mentions $t$ for the text mention $x$. Finally, this information is combined with the prior scores $\sigma(t|x)$ of the previous-stage classifier, leading to the overall scoring:

$$\text{OverallScore}(t|x) = \alpha \cdot \pi(t|x) + \beta \cdot \sigma(t|x), \tag{5.1}$$

with hyper-parameters $\alpha$ and $\beta$ (which are tuned on the validation data).

**Alignment decisions:** The RWR from text mention $x$ computes $\pi(t|x)$ for each table mention $t$. Pair $\langle x, t^* \rangle$ forms an alignment if and only if (i) $t^*$ is the table mention with the highest overall score $\text{OverallScore}(t^*|x)$, and (ii) its overall score $\text{OverallScore}(t^*|x)$ exceeds a tunable threshold $\epsilon$. Interestingly, making an alignment decision adds knowledge, and we propose to exploit that by updating the graph. In particular, after identifying an alignment $\langle x, t^* \rangle$, $x$ cannot have alignments with any other table mention, and hence we modify the graph by removing all edges $(x, t)$ for any $t \neq t^*$ (if no alignment is found for $x$, then all text-table edges adjacent to $x$ are removed.) This way the next RWR for another text mention is able to leverage the new alignment information for improved results. This introduces a new issue: the order in which text mentions are processed. We discuss our approach to this next.

**Entropy-based ordering:** Note that a correct alignment decision will improve knowledge for future RWR executions, but an incorrect alignment decision can be harmful. Hence one intuitively should make decisions for the easier text mentions first, and then factor this information into the later decisions on the harder cases. To quantify the difficulty of aligning a text mention, we use the entropy of the classifier's confidence scores (see Section 5.6). High entropy, close to uniform scores, means that there are several candidates among the table mentions that are not easy to distinguish. Low entropy, with highly skewed scores, indicates that there is one strongly preferred candidate—with the extreme case of having exactly one candidate only. Thus we process text mentions in order of increasing entropy. Once an alignment is resolved for a text mention, only this text-table edge is kept and all edges to other table-mention candidates are removed.
Pseudo-code for the overall graph algorithm is given in Algorithm 1.

**Algorithm 1** Graph-based global resolution

```
Data: undirected edge-weighted graph $G = (V, E)$; set $C$ of mention-pair candidates
      $(x, t) \in E$ with prior confidence scores $\sigma(x, t)$
Result: subset $A \subseteq C$ of pairs for final alignment
$A := \emptyset$
for each text mention $x$ with $\exists t : (x, t) \in C$ do
    normalize $\{\sigma(x, t) : (x, t) \in C\}$ to a probability distribution  compute its entropy
     H(x)
end
for each $x$ in increasing order of H(x) do
    run RWR from $x$ to compute stationary probabilities $\pi(t|x)$ for all $t$ with $(x, t) \in C$
    OverallScore$(t|x) = \alpha \cdot \pi(t|x) + \beta \cdot \sigma(t|x)$  let $t^* := argmax_t$ OverallScore$(t|x)$
    if  OverallScore$(t^*|x) > \epsilon$ then
        add $(x, t^*)$ to $A$  delete edges $(x, t)$ for all $t \neq t^*$ from $G$
    end
    else
        delete edges $(x, t)$ for all $t$ from $G$
    end
end
```

## 5.8 Experimental Setup

### 5.8.1 Data

To evaluate BriQ, we use the Dresden Web Table Corpus (DWTC) which comprises about 125 Million tables extracted from 3.6 Billion web pages in the Common Crawl of July 2014 [ETBL15]. We compiled two datasets:

- **tableS**: a small annotated corpus from 495 web pages with complete assessment of ground-truth alignments, used to evaluate precision and recall of our method, and

- **tableL**: a large set from 1.5 million web pages, used to perform run-time measurements and demonstrate scalability of our method.

To construct the larger tableL corpus, we filtered the DWTC collection for web pages that meet a variety of criteria: English language, table(s) containing numerical cells, numerical mentions in text, overlap of tokens between table(s) and text. The resulting 1.5 million pages mostly fall under five major topics: finance, environment, health, politics, and sports (as determined by simple surface cues, and validated by manual sampling).

The tableS corpus is constructed from tableL by randomly selecting 505 pages and having them manually annotated by 8 hired annotators, all being non-CS students. We refrained from using mturk-like crowdsourcing for this purpose, as the annotation required fairly sophisticated guidelines and very thorough inspection of web pages; crowd workers would be unlikely to meet this quality assurance. In total, the 8 annotators spent about 130 person-hours on judging text-table mention pairs, and classifying them by their type: exact-match with single cell, sum, average, percentage, difference, ratio, minimum, maximum, unrelated, or other.

The inter-annotator agreement, with Fleiss' Kappa [Fle71] being 0.6854, was substantial. All mention pairs confirmed by at least two annotators were kept, resulting in a final tableS corpus of 495 pages corresponding to 1,598 documents with 1,703 tables and 7,468 distinct text mentions of quantities.

### 5.8.2 Classifier Training

The tableS dataset was randomly split into disjoint training (80%), test (10%) and validation sets (10%). For each ground-truth mention pair in the training data (serving as positive samples), we automatically generated 5 negative samples by picking the table cells with the highest similarity to the positive sample (i.e., approximately the same values and similar context). These included many virtual cells for aggregate values, making the task very challenging. Table 5.1 gives a break-down of positive and negative samples by mention type.

We counter the label imbalance (#pos $\ll$ #neg) by giving different weights to the positive and negative labels in the classifiers' loss functions [JS02, LBB+12]. These weights are inversely proportional to the ratio of the positive or negative labels in the dataset. The loss function is optimized for the area under the ROC curve, to ensure that neither precision nor recall could be neglected.

Table 5.1: Classifier training data.

| type | #pos |
|---|---|
| single-cell | 4376 |
| sum | 267 |
| percent | 115 |
| diff. | 134 |
| ratio | 141 |
| total | 5039 |

| type | #neg |
|---|---|
| single-cell | 3315 |
| sum | 9300 |
| percent | 4995 |
| diff. | 7924 |
| ratio | 5002 |
| total | 39767 |

### 5.8.3 Metrics, Tuning and Testing

The traditional classifier performance metrics like accuracy and error rate are not informative in our setting with high imbalance between the positive and negative class. Therefore, we use precision, recall and F1 as major metrics to evaluate the BriQ system. For tuning hyper-parameters, we use the withheld validation set of the annotated tableS corpus (10%). We use grid search to choose the best values for the hyper-parameters, for the classifiers as well as for the graph-based algorithm.

For testing classifiers, we use the withheld part of the annotated tableS corpus (10%). We apply the learned models on all possible mention pairs between text and table (i.e., not just limited to the negative samples generated for training.) Overall, the test set has 687,321 mention pairs out of which only 0.1% are correct. The global resolution algorithm is tested with the outputs of the classifier and the adaptive filtering stage, typically reducing the size by two orders of magnitude.

### 5.8.4 Baselines

We compare BriQ against the following two baselines:

- *Classifier-only (RF)*: the Random Forest algorithm deployed in the first stage of BriQ, trained the same way as BriQ. For each text mention, the cell of the classifier's top-ranked mention-pair is chosen as output.
- *Random-Walk-only (RWR)*: a graph-based algorithm similar to the one used in the second stage of BriQ. The algorithm uses all features that are available to BriQ (see Section 5.5.2). However, as there are no prior probabilities computed from the first stage, these features are combined using uniform weights and then normalized to graph-traversal probabilities. Also, there is no pruning of any mention-pairs, making this baseline fairly expensive while still being an interesting comparison point.

We also considered an additional baseline derived from our earlier work on linking quantities to a knowledge base (QKB) [IRW16]. Given a candidate mention-pair, we map both the text mention and the table cell to the QKB, this way normalizing them. Then we compare the two mentions if they are the same (i.e., link to the same QKB entry with exact-matching values). While this takes care of unit matching, it is limited to the units registered in the QKB and does not nearly cover all the diverse units in our large-scale input data. Moreover, the test can work only if the values of the two normalized mentions match exactly. For approximate matches where one text mention could be mapped to different single or virtual cells, the approach is unsuitable. Since approximate matches are very frequent in our test data, we did not pursue this possible baseline any further.

## 5.9 Experimental Results

### 5.9.1 Alignment Quality

We conducted experiments with three variations of text mentions, with increasing difficulty:

- **Original text mentions,** as given in the document. This is the main experiment.
- **Truncated text mentions,** where we removed the least significant digit of each original text mention. For example, 6746, 2.74, 0.19 became 6740, 2.7, and 0.1. This is meant as an additional test of robustness, making all test cases more difficult.

- **Rounded text mentions,** where we numerically rounded the least significant digit of each text mention. For example, 6746, 2.74, 0.19 became 6750, 2.7, and 0.2. This is meant as a stress test, with the additional challenge of making surface-form similarity less informative.

**Original mentions:** Table 5.2 shows the results for the original, truncated and rounded mentions. For the original mentions, BriQ outperforms both baselines, RF and RWR, by a large margin, regarding both precision and recall. BriQ achieved an F1 score of more than 70%, which is remarkably high given the noisy nature of the real-life data and the difficulty of the alignment problem.

Table 5.2: Results for *original, truncated and rounded* text mentions.

| | Original | | | Truncated | | | Rounded | | |
|---|---|---|---|---|---|---|---|---|---|
| | RF | RWR | BriQ | RF | RWR | BriQ | RF | RWR | BriQ |
| recall | 0.43 | 0.52 | 0.68 | 0.27 | 0.42 | 0.58 | 0.13 | 0.34 | 0.49 |
| prec. | 0.37 | 0.53 | 0.79 | 0.25 | 0.44 | 0.63 | 0.10 | 0.35 | 0.52 |
| F1 | 0.40 | 0.53 | 0.73 | 0.26 | 0.43 | 0.60 | 0.11 | 0.34 | 0.51 |

**Truncated and rounded mentions:** As expected, the results for truncated and rounded mentions in Table 5.2 show a drop in quality, and the decrease is more pronounced for rounded mentions. In both of these situations, BriQ has the best results. For truncated mentions, BriQ still achieves fairly good quality, with an F1 score of ca. 60%. For rounded mentions, it achieves decent quality, with an F1 score of ca. 51%. In contrast, the two baselines degrade strongly. Especially, the RF classifier alone is not competitive at all, demonstrating our insight that the quantity alignment problem cannot be solved solely by supervised end-to-end machine learning.

**Results by Mention Type:** Tables 5.3, 5.4 and 5.5 break down the results by aggregation type: sum, difference, percentage, change ratio and single-cell match. BriQ clearly outperforms RF and RWR on all mention types and RWR outperforms RF on all types except for single-cell. As expected, BriQ has the best F1 score, 79%, on text mentions that refer to a *single* table cell. For sum and difference, BriQ achieved fairly good F1 scores of 72% and 43%, respectively. For the remaining two cases—percentage and ratio—all methods dropped substantially in output quality. The reason is that these cases are rather infrequent, so that the classifier gave them very low prior scores, a bias effect that the global resolution could not fully compensate.

Table 5.3: Results by mention type for original mentions, using RF.

| | sum | diff. | percent | change ratio | single-cell |
|---|---|---|---|---|---|
| recall | 0.00 | 0.27 | 0.03 | 0.06 | 0.48 |
| prec. | 0.00 | 0.04 | 0.02 | 0.01 | 0.70 |
| F1 | 0.00 | 0.06 | 0.03 | 0.02 | 0.57 |

**Effectiveness of Adaptive Filtering:** The adaptive filtering is crucial for BriQ to reduce the input size of the global resolution stage. Table 5.6 shows the selectivity of

Table 5.4: Results by mention type for original mentions, using RWR.

| | sum | diff. | percent | change ratio | single-cell |
|---|---|---|---|---|---|
| recall | 0.61 | 0.33 | 0.09 | 0.18 | 0.57 |
| prec. | 0.52 | 0.22 | 0.43 | 0.27 | 0.57 |
| F1 | 0.56 | 0.26 | 0.15 | 0.21 | 0.57 |

Table 5.5: Results by mention type for original mentions, using BriQ.

| | sum | diff. | percent | change ratio | single-cell |
|---|---|---|---|---|---|
| recall | 0.74 | 0.62 | 0.10 | 0.20 | 0.75 |
| prec. | 0.71 | 0.33 | 0.75 | 0.30 | 0.84 |
| F1 | 0.72 | 0.43 | 0.17 | 0.24 | 0.79 |

our filters (i.e., the ratio of retained mention pairs to all mention pairs that the classifier dealt with) and the recall after the filters. These numbers clearly demonstrate the enormous gains of the filtering stage. Conversely, the near-optimal recall numbers in the table show that we rarely make false-negative errors: BriQ effectively avoids erroneously dismissing good candidates from the mention-pair space.

Table 5.6: Selectivity and recall after filtering.

| type | selectivity | recall |
|---|---|---|
| sum | 0.01 | 1.00 |
| difference | 0.01 | 0.87 |
| percentage | < 0.01 | 0.91 |
| change ratio | < 0.01 | 0.88 |
| single-cell | 0.04 | 0.91 |
| overall | 0.01 | 0.91 |

### 5.9.2 Ablation Study

We studied the influence of different feature groups on the two baselines and BriQ. We divide our feature space into three feature groups:

- **surface form similarity**.
- **context features**, including local and global word overlap, local and global noun phrases overlap, aggregate function match, and approximate indicator.
- **quantity features**, including relative value difference, unnormalized value difference, unit match, precision difference, and scale difference.

For the ablation study, we carried out three experiments, each corresponding to one feature group left out, thus training, tuning and testing the three models end-to-end on the remaining features. Table 5.7 shows the F1 score, precision and recall of the three experiments in comparison with the full-feature model. The results underline the robustness of BriQ in comparison to the other baselines. Although BriQ's recall is affected by leaving out some features, its precision is stable. Leaving out context features leads to the highest degradation in BriQ's performance. Interestingly, leaving out the quantity features resulted in improvements of the RF classifier. The reason is that, without these features, the classifier has fewer virtual cells to consider (i.e., approximately matching values from aggregation of several table cells), making it easier to get the frequent single-cell cases right. However, BriQ still outperformed the RF classifier by a large margin.

Table 5.7: Ablation Study: Recall, Precision and F1 score

| | Recall | | | Precision | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | RF | RWR | BriQ | RF | RWR | BriQ | RF | RWR | BriQ |
| all features | 0.43 | 0.52 | 0.68 | 0.37 | 0.53 | 0.79 | 0.40 | 0.53 | 0.73 |
| w/o surf. sim. | 0.37 | 0.36 | 0.65 | 0.33 | 0.39 | 0.77 | 0.35 | 0.37 | 0.70 |
| w/o context | 0.43 | 0.38 | 0.59 | 0.34 | 0.44 | 0.77 | 0.38 | 0.41 | 0.67 |
| w/o quantity | 0.43 | 0.31 | 0.61 | 0.54 | 0.35 | 0.77 | 0.48 | 0.33 | 0.68 |

### 5.9.3 Run-Time Results

BriQ is implemented in PySpark using Python, NetworkX, and SciPy libraries for the graph algorithm. For the RF classifier, we use R with the caret package, integrated into BriQ by the rpy2 library. All experiments were run on a Spark cluster with 10 executors, each with 6 cores and 30GB of memory, and with 50GB of driver memory. Training and tuning takes about 10 hours (on a very large dataset), with the grid search for the best hyper-parameters being the major factor (as it is often the case in machine learning). This is a *one-time* pre-processing effort.

To measure the run-time performance of BriQ for processing documents, we use the *tableL* dataset of about 1.5 million web pages. Table 5.8 shows the throughput of BriQ in terms of completed documents per minute, broken down into different thematic domains (e.g., quantities in finance are different in nature from quantities in sports). The throughput numbers clearly indicate that BriQ is practically viable at large scale. Moreover, it is 30 time faster than the RWR baseline that has a throughput of 76 documents per minute.

Table 5.9 gives more statistics for each of these domains. We see that documents on sports led to a large number of virtual cells for aggregated values, incurring higher load and hence resulting in lower throughput than for the other domains.

Table 5.8: BriQ throughput by domain.

| | pages | documents | mentions | #docs/min |
|---|---|---|---|---|
| environment | 118,724 | 986,180 | 3,062,943 | 2,935 |
| finance | 325,853 | 3,374,175 | 10,596,979 | 5,029 |
| health | 102,132 | 879,388 | 1,930,975 | 4,604 |
| politics | 128,318 | 2,762,873 | 4,123,800 | 6,223 |
| sports | 527,263 | 2,173,832 | 7,393,225 | 863 |
| others | 309,292 | 3,141,865 | 6,796,835 | 2,588 |
| total | 1,511,582 | 13,318,313 | 33,904,757 | 2,478 |

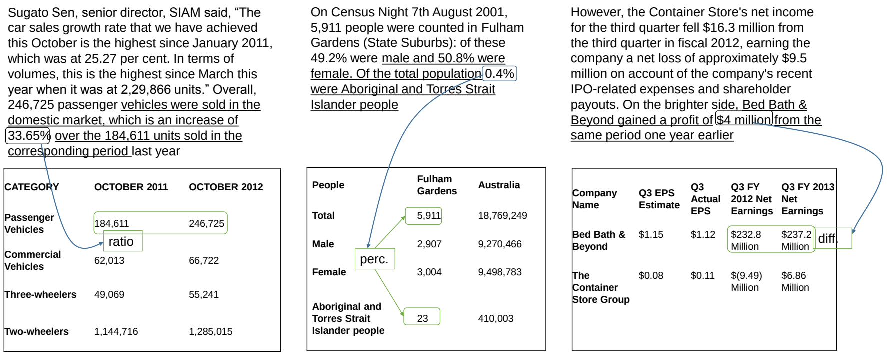
Table 5.9: Table statistics by domain.

| | rows | columns | single cells | virtual cells |
|---|---|---|---|---|
| environment | 7 | 4 | 21 | 243 |
| finance | 7 | 4 | 16 | 142 |
| health | 3 | 2 | 4 | 26 |
| politics | 8 | 3 | 17 | 137 |
| sports | 8 | 6 | 35 | 523 |
| others | 7 | 4 | 21 | 252 |
| average | 7 | 4 | 19 | 220 |

## 5.10 Discussion

**Anecdotal examples:** Figure 5.5 shows three alignments computed by BriQ. Examples (a) and (b) illustrate the ability to detect and align change rates and percentages to the correct cell pairs. In example (c), BriQ is even able to discover the approximate difference between two cells and align it properly.

**Typical error cases:** Figure 5.6 shows some of the typical errors made by BriQ. The first case is in examples (a) and (b), having same-value collisions with several cells in the tables. In (a) the value '3.2' exists in two cells in the same row with very similar context. As the immediate context of the quantity '3.2' in the text, underlined, does not contain any words related to the columns, BriQ fails to identify the correct alignment. In (b) the immediate context of the quantity '\$50' contains both words 'wholesale' and 'retail'. Moreover, the quantity '\$100' is closer to the incorrectly aligned cell '\$50'. So BriQ fails here because of high ambiguity.
The third example (c) illustrates the case where the immediate context of the text mention '\$7.32 billion' has a single-word overlap with the table context, "August". In addition the scale of the quantity (i.e., billion) is missing in the table. Such cases are extremely difficult to deal with, since neither the quantity features nor context features can help in finding the correct alignment.



Sugato Sen, senior director, SIAM said, “The car sales growth rate that we have achieved this October is the highest since January 2011, which was at 25.27 per cent. In terms of volumes, this is the highest since March this year when it was at 2,29,866 units.” Overall, 246,725 passenger vehicles were sold in the domestic market, which is an increase of 33.65% over the 184,611 units sold in the corresponding period last year

| CATEGORY | OCTOBER 2011 | OCTOBER 2012 |
|---|---|---|
| Passenger Vehicles | 184,611 | 246,725 |
| Commercial Vehicles | 62,013 | 66,722 |
| Three-wheelers | 49,069 | 55,241 |
| Two-wheelers | 1,144,716 | 1,285,015 |

a) Detected change ratio.

On Census Night 7th August 2001, 5,911 people were counted in Fulham Gardens (State Suburbs): of these 49.2% were male and 50.8% were female. Of the total population 0.4% were Aboriginal and Torres Strait Islander people

| People | Fulham Gardens | Australia |
|---|---|---|
| Total | 5,911 | 18,769,249 |
| Male | 2,907 | 9,270,466 |
| Female | 3,004 | 9,498,783 |
| Aboriginal and Torres Strait Islander people | 23 | 410,003 |

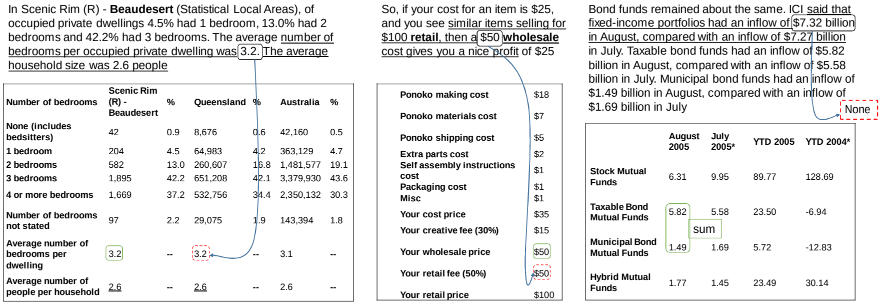
b) Detected percentage

However, the Container Store's net income for the third quarter fell $16.3 million from the third quarter in fiscal 2012, earning the company a net loss of approximately $9.5 million on account of the company's recent IPO-related expenses and shareholder payouts. On the brighter side, Bed Bath & Beyond gained a profit of $4 million from the same period one year earlier

| Company Name | Q3 EPS Estimate | Q3 Actual EPS | Q3 FY 2012 Net Earnings | Q3 FY 2013 Net Earnings |
|---|---|---|---|---|
| Bed Bath & Beyond | $1.15 | $1.12 | $232.8 Million | $237.2 Million |
| The Container Store Group | $0.08 | $0.11 | $(9.49) Million | $6.86 Million |

b) Detected difference

Figure 5.5: Examples of alignments discovered by BriQ



In Scenic Rim (R) - **Beaudesert** (Statistical Local Areas), of occupied private dwellings 4.5% had 1 bedroom, 13.0% had 2 bedrooms and 42.2% had 3 bedrooms. The average number of bedrooms per occupied private dwelling was 3.2. The average household size was 2.6 people

| Number of bedrooms | Scenic Rim (R) - Beaudesert | % | Queensland | % | Australia | % |
|---|---|---|---|---|---|---|
| None (includes bedsitters) | 42 | 0.9 | 8,676 | 0.6 | 42,160 | 0.5 |
| 1 bedroom | 204 | 4.5 | 64,983 | 4.2 | 363,129 | 4.7 |
| 2 bedrooms | 582 | 13.0 | 260,607 | 16.8 | 1,481,577 | 19.1 |
| 3 bedrooms | 1,895 | 42.2 | 651,208 | 42.1 | 3,379,930 | 43.6 |
| 4 or more bedrooms | 1,669 | 37.2 | 532,756 | 34.4 | 2,350,132 | 30.3 |
| Number of bedrooms not stated | 97 | 2.2 | 29,075 | 1.9 | 143,394 | 1.8 |
| Average number of bedrooms per dwelling | 3.2 | -- | 3.2 | -- | 3.1 | -- |
| Average number of people per household | 2.6 | -- | 2.6 | -- | 2.6 | -- |

a) Wrong alignment

So, if your cost for an item is $25, and you see similar items selling for $100 **retail**, then a $50 **wholesale** cost gives you a nice profit of $25

| | |
|---|---|
| Ponoko making cost | $18 |
| Ponoko materials cost | $7 |
| Ponoko shipping cost | $5 |
| Extra parts cost | $2 |
| Self assembly instructions cost | $1 |
| Packaging cost | $1 |
| Misc | $1 |
| Your cost price | $35 |
| Your creative fee (30%) | $15 |
| Your wholesale price | $50 |
| Your retail fee (50%) | $50 |
| Your retail price | $100 |

b) Wrong alignment

Bond funds remained about the same. ICI said that fixed-income portfolios had an inflow of $7.32 billion in August, compared with an inflow of $7.27 billion in July. Taxable bond funds had an inflow of $5.82 billion in August, compared with an inflow of $5.58 billion in July. Municipal bond funds had an inflow of $1.49 billion in August, compared with an inflow of $1.69 billion in July

| | August 2005 | July 2005* | YTD 2005 | YTD 2004* |
|---|---|---|---|---|
| Stock Mutual Funds | 6.31 | 9.95 | 89.77 | 128.69 |
| Taxable Bond Mutual Funds | 5.82 | 5.58 | 23.50 | -6.94 |
| Municipal Bond Mutual Funds | 1.49 | 1.69 | 5.72 | -12.83 |
| Hybrid Mutual Funds | 1.77 | 1.45 | 23.49 | 30.14 |

c) Undetected alignment

Figure 5.6: Examples of errors made by BriQ

## 5.11 Summary

We have introduced the new problem of aligning quantities between text and tables. Our methodology combines supervised classification based on local contexts, adaptive filtering techniques for computational tractability, and joint inference methods for global resolution. Comprehensive experiments with ad-hoc web tables show that all stages of this pipeline are essential, and together can achieve good precision and recall at affordable computational cost.

As for future work, we plan to investigate this problem also in the context of enterprise content (e.g., spreadsheets in documents) and specialized domains such as material science or biomedical documents.

Quantity alignment is an important step towards semantically understanding numbers in unstructured and semi-structured content. This in turn can open up the path towards next-generation search engines that can handle queries about quantities, such as Internet companies with annual income above 5 Mio. USD, electric cars with energy consumption below 100 MPGe (or equivalently, ca. 21 kWh/100km), or clinical trials with a daily anti-coagulant dosage above 30 mg. All these examples are way beyond the scope of today's search engines; quantity understanding would bring them closer to feasibility.

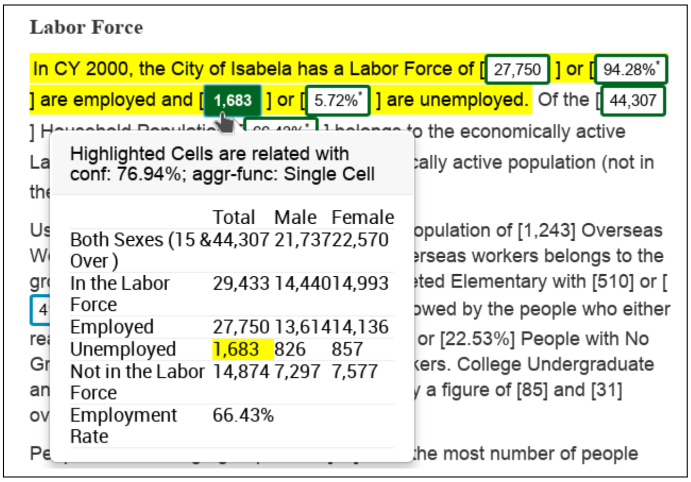
# 6 ExQuisiTe: Explaining Quantities in Text

## 6.1 Introduction

### 6.1.1 Motivation

The Web contains a wealth of pages with embedded tables, and reports with spreadsheets are abundant in enterprises. Such documents, with financial or statistical data, are challenging to read, as they are often packed with numbers and tables. For example, in a financial report, a reader can stumble upon a statement like "..overall revenues were up 21 percent year-over-year... ", giving rise to questions such as: "What was the revenue of the previous year?" or "Which particular product or sector contributed to this increase?". In such cases, the table(s) accompanying the text can provide answers. However, long documents contain several tables, and table cells are referenced at many spots throughout the report. Moreover, many textual references round or truncate numbers, or refer to aggregates such as row or column totals, which are not explicitly given in the table(s). Therefore, it is tedious work to navigate between text and tables to answer the reader's questions.
Generally, what a reader would desire is an easy and seamless way of drilling down from text passages to the relevant table cells for additional detail, and zooming out from tables to the relevant sentences that explain the numbers.

### 6.1.2 Contribution



**Labor Force**

In CY 2000, the City of Isabela has a Labor Force of [ 27,750 ] or [ 94.28%* ] are employed and [ 1,683 ] or [ 5.72%* ] are unemployed. Of the [ 44,307 ] H... ] belongs to the economically active

La ... cally active population (not in

the

Us ... opulation of [1,243] Overseas

W ... erseas workers belongs to the

gr ... eted Elementary with [510] or [

4 ... owed by the people who either

re ... or [22.53%] People with No

Gr ... kers. College Undergraduate

an ... a figure of [85] and [31]

ov

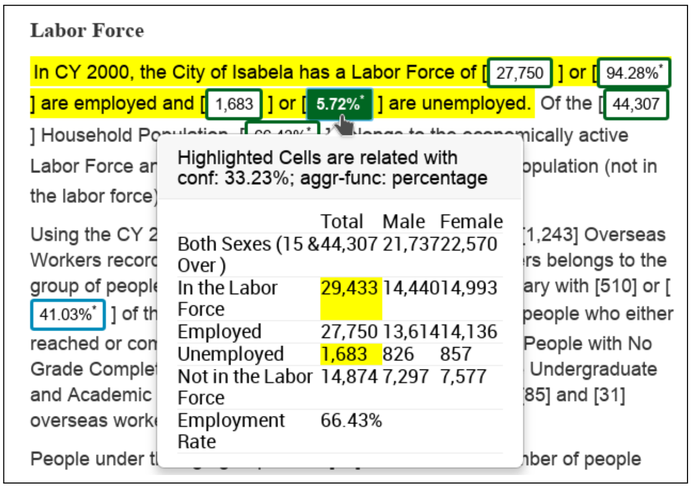
Pe ... he most number of people

Highlighted Cells are related with
conf: 76.94%; aggr-func: Single Cell

| | Total | Male | Female |
|---|---|---|---|
| Both Sexes (15 & Over ) | 44,307 | 21,737 | 22,570 |
| In the Labor Force | 29,433 | 14,440 | 14,993 |
| Employed | 27,750 | 13,614 | 14,136 |
| Unemployed | 1,683 | 826 | 857 |
| Not in the Labor Force | 14,874 | 7,297 | 7,577 |
| Employment Rate | 66.43% | | |

Figure 6.1: Simple quantity reference.

To address the above desiderata, we propose *ExQuisiTe*, a system that identifies relations between quantities in text and tables. ExQuisiTe automatically detects these relations and generates an easy-to-read document where numbers in text are linked to their source tables and respective cells. It identifies simple mentions of single-cell table quantities as well as mentions of aggregate quantities. For example, in Figure 6.1 the mention "1,683" in the text refers to a simple quantity in the table; and in Figure 6.2 the mention "5.72%" refers to an aggregate quantity (percentage) in the same table.



**Labor Force**

In CY 2000, the City of Isabela has a Labor Force of [ 27,750 ] or [ 94.28%* ] are employed and [ 1,683 ] or [ 5.72%* ] are unemployed. Of the [ 44,307 ] Household Po... opulation (not in the labor force)

Using the CY 2... [1,243] Overseas Workers record... rs belongs to the group of people... ary with [510] or [ 41.03%* ] of th... people who either reached or con... People with No Grade Complet... Undergraduate and Academic ... 85] and [31] overseas work...

People under t... nber of people

Highlighted Cells are related with
conf: 33.23%; aggr-func: percentage

| | Total | Male | Female |
|---|---|---|---|
| Both Sexes (15 & Over ) | 44,307 | 21,737 | 22,570 |
| In the Labor Force | 29,433 | 14,440 | 14,993 |
| Employed | 27,750 | 13,614 | 14,136 |
| Unemployed | 1,683 | 826 | 857 |
| Not in the Labor Force | 14,874 | 7,297 | 7,577 |
| Employment Rate | 66.43% | | |

Figure 6.2: Aggregate quantity reference.

Furthermore, ExQuisiTe can guide *Extractive Text Summarization (ETS)* systems by emphasizing sentences with aggregate quantities. Current summarization systems [NM12, GG17] do not include table data, and ExQuisiTe opens the opportunity for them to harness table data. Once ExQuisiTe identifies references of simple and aggregate table quantities in the text, it can suggest sentences with aggregations to be included in the summary generated by the ETS algorithm.

For example, in Figure 6.2 the highlighted sentence covers more cells in the table than the other sentences. It contains more aggregate mentions, and hence it provides a better summary, with judicious consideration of the numbers in the tables.

ExQuisiTe is base on the BriQ algorithm explained in Chapter 5, and consists of four configurable stages: (i) Document Extraction, (ii) Local Resolution, (iii) Global Resolution, and (iv) Markup and Summary Generation. The first stage extracts text segments and their possible related tables using string similarity measures. The second stage identifies potential alignments between quantity mentions in text and tables based on local features. Then, the third stage collectively aligns quantities in the text to their relevant quantities in tables. Finally, the system generates markup for the document with the inferred alignments and selects important sentences for summarization.

The code of ExQuisiTe as well as all the annotated data used for training is available on the project web page [1]. Our main contributions are:

[1] https://www.mpi-inf.mpg.de/briq/

- an end-to-end system for quantity alignment,
- a system that generates salient suggestions for a downstream ETS method,
- an open-source efficient pipeline that can be flexibly configured on a Spark cluster for online document processing.

## 6.2 Computational Model

Our algorithm handles the following inputs:

- a piece of text, like a (part of a) web page, with a set of $m$ text mentions of quantities $X = \{x_i : i = 1, \dots, m\}$,
- a table $q$ with $r$ rows and $c$ columns and a set of $n$ mentions of quantities $T = \{t_j : j = 1, \dots, n\}$.

*Text mentions* include terms containing numbers or numerals such as "123 patients", "37K EUR", "1.5%" or "twenty pounds".
*Table mentions* include two types of quantities. The first are *simple mentions*, such as '1,683' in Figure 6.1. Given a table with $r$ rows and $c$ columns we have at most $r \cdot c$ single-cell quantity mentions. The second type is *aggregate mentions*, computed as an aggregation of one or more table cells, such as '5.72%' in Figure 6.2.
In this demo we consider the following *aggregate functions*: average, sum, difference, percentage, and change ratio.
For aligning quantity mentions between text and tables, we aim to compute as output a subset of mention pairs $\langle x_i, t_j \rangle$ where $x_i \in X$ is a text mention and $t_j \in T$ is a table mention, including aggregate quantities. These pairs should denote the same quantity with high confidence.

## 6.3 System Components

### 6.3.1 Document Extraction

Long web pages can cover a variety of thematic aspects, along with several tables. Therefore, we decompose the input web page into coherent segments which we refer to as *documents*. We define a coherent document to be a paragraph together with all "related" tables from the same web page. Each document can be processed independently from the other documents. Hence, we can leverage a distributed computing framework, Spark, for online page processing.
This module first decomposes the input web page into paragraphs, then recognizes related tables for each paragraph using pairwise similarities between all paragraphs and all tables in the web page.

### Quantity Extraction

For each document, quantity mentions are extracted from the text and the tables, using regular expressions. Our method pays particular attention to the challenges of aggregated quantities (e.g., column totals). Therefore, we generate table candidates as combinations of cells with an associated aggregate function. For example, we generate aggregate mention for a column total even if the table does not explicitly show the total. Aggregate quantities are automatically generated by considering (i) all rows and columns for totals and averages and (ii) all pairs of cells in the same row or column for difference, percentage, and change ratio. We prune the aggregate quantity candidate, to ensure computational tractability and to control spurious matches.

## 6.3.2 Local Resolution

This module first computes features for each text mention and each table mention by analyzing the surrounding context. Then, it computes similarity-based features for each pair of text mention and table mention including aggregate quantities. After that, it uses a binary classifier that accepts or rejects candidate mention-pairs. This binary classifier assigns a confidence score to each mention-pair, and we use this score in the following steps. At the end of this module, we filter the candidate mention-pairs according to their confidence score and other measures which we will explain later.

### Mention-Pair Classification

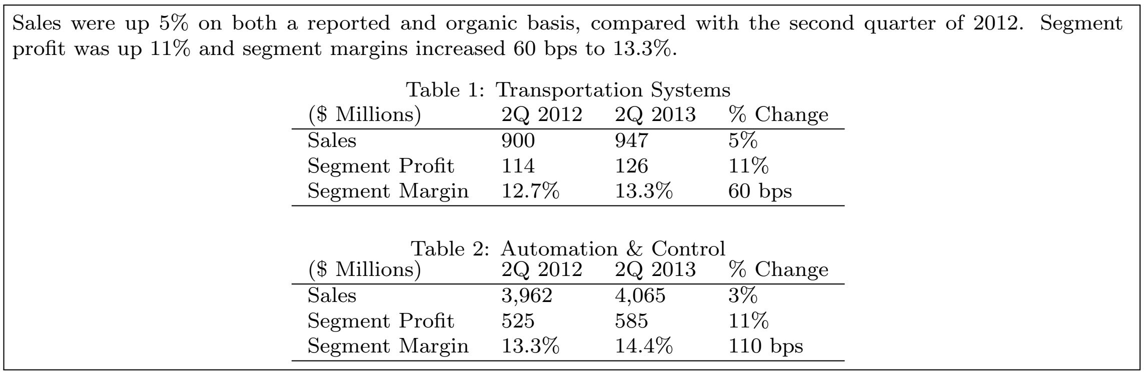
We use manually annotated web pages with ground-truth alignment to train a *Random Forest (RF)* classifier. The classifier operates *locally* in the sense that it predicts the alignment confidence for each mention-cell pair in isolation, It serves two purposes: First, it enables the subsequent filtering step, which significantly reduces the number of candidate pairs for achieving an acceptable running time in the global resolution step. Second, it provides a prior for that global resolution step.

### Classifier Feature

For the mention-pair classifier, we designed a variety of features that capture information a human reader would use in order to determine if text mention $x$ and table cell $t$ denote the same quantity. This includes *surface form similarities*, *context features*, and *quantity features.* For more details refer to Chapter 5

### Adaptive Filtering

This stage reduces the number of mention-pair candidates from 1000s of candidates to 100s for tractability of global inference algorithms. We design the *adaptive filtering* algorithm to work in two stages. In the first stage, we develop a *text mention tagger* to predict the aggregation function for each text mention or tag the mention as a single-cell match. Then, we prune mention-pairs based on this tagger's outcome. In the second stage, we prune mention-pairs based on *value difference* and *unit mismatch.* Finally, we



Sales were up 5% on both a reported and organic basis, compared with the second quarter of 2012. Segment profit was up 11% and segment margins increased 60 bps to 13.3%.

Table 1: Transportation Systems

| ($ Millions) | 2Q 2012 | 2Q 2013 | % Change |
|---|---|---|---|
| Sales | 900 | 947 | 5% |
| Segment Profit | 114 | 126 | 11% |
| Segment Margin | 12.7% | 13.3% | 60 bps |

Table 2: Automation & Control

| ($ Millions) | 2Q 2012 | 2Q 2013 | % Change |
|---|---|---|---|
| Sales | 3,962 | 4,065 | 3% |
| Segment Profit | 525 | 585 | 11% |
| Segment Margin | 13.3% | 14.4% | 110 bps |

Figure 6.3: Example with Coupled Quantities

sort mention-pairs according to classifier scores and select top-$k$ mention-pairs for each quantity mention based on *mention type* and *score distribution.* For more details refer to Chapter 5.

### 6.3.3 Global Resolution

This module takes as input the candidate mention-pairs from the classifier and outputs the alignment of quantity mentions. We harness dependencies among mentions to resolve ambiguities. Consider the example in Figure 6.3. The text mentions "11%" and "13.3%" have exact matches in both of the shown tables, and local-resolution algorithms cannot infer the proper alignment. However, when considering these two mentions jointly with "60 bps" and "5%", it becomes clear that all of these refer to the first table.
We devised an unsupervised algorithm for this kind of global resolution. The algorithm encodes dependencies among mentions into a graph and uses *Random Walks with Restarts(RWRs)* to infer the best joint alignment.

#### Graph Construction

We construct an undirected weighted graph $G = (V, E)$ for each document:

- The node set $V$ consists of all quantity mentions in the document's text and tables.
- The edge set $E$ consists of three kinds of edges connecting related nodes: text-text edges, table-table edges, and text-table edges as explained below.

*(i) Text-text edges:* connects each pair of text quantity mentions that are within a certain proximity or have similar surface forms. Edge weights are computed based on a linear combination of proximity and string similarity. *(ii) Table-table edges:* connects each pair of table quantity mentions in the same row or the same column of the same table, and edge weights are set uniformly. *(iii) Text-table edges:* connects each pair of text and table mention that is kept by the adaptive filtering stage, and edge weights are set to the confidence scores returned by the classifier.

After this initial graph construction, all edge weights are normalized to obtain a stochastic graph, via dividing each node's outgoing weights by the total weight of these edges.

#### Graph Algorithm

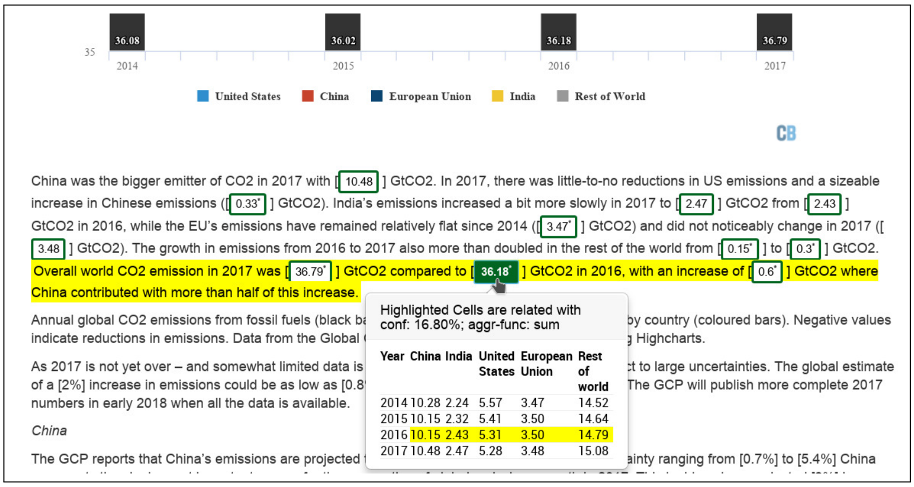
In our setting, we employ random walks with restart: starting from a text mention, the graph is stochastically traversed, with a certain probability of jumping back to the initial node. This technique is also known as topic-specific or personalized PageRank [Hav03]. Our implementation iterates RWRs for each text mention until the estimated visiting probabilities of the candidate table mentions change by less than a specified convergence bound. This way we obtain a ranked list of table mentions for each text mention $x$.
*Alignment decisions:* The RWR from text mention $x$ computes the stationary probability $\pi(t|x)$ for each table mention $t$. Pair $\langle x, t^* \rangle$ forms an alignment if and only if (i) $t^*$ is the table mention with the highest overall score, and (ii) its overall score exceeds the defined confidence threshold. We then exploit the alignment decisions to update the graph, such that after identifying an alignment $\langle x, t^* \rangle$, $x$ we modify the graph by removing all edges $(x, t)$ for any $t \neq t^*$.(If no alignment is found for $x$, then all text-table edges adjacent to $x$ are removed.)

### 6.3.4 Markup and Summarization

This module integrates the output of our system with the content of the web page and displays the results to the user in the form of an HTML page. This module is also responsible for analyzing the aligned quantities and highlighting the important sentences for the summarization engine. It estimates the importance of a sentence based on its coverage of table cells.
We define the *coverage of a quantity mention* in the text to be the number of individual single-cells it refers to. For mentions referring to an aggregate table quantity such as the sum of a column, we include all the individual single-cells in this column. Then, we compute the *coverage of a sentence* as the sum of the coverage of its mentions.
For each table, we extract all the sentences that reference it. Then, we compute a score for each sentence based on its coverage. After that, we highlight the sentence with the maximum score in the generated HTML. For example in Figure 6.4 even though the second sentence has more simple quantity references to the table than the last sentence, the latter provides a better summary of the table. The last sentence discusses the overall CO2 emission in the world and has the highest coverage of table cells, while the second sentence only discusses the emission of India and the EU.

## 6.4 Experimental Results

We trained and evaluated our system on a manually annotated corpus of 495 web pages. The F1 score of our system is 79% for the simple quantity mentions, and 40% for the aggregate mentions. We carried out a run time analysis on a Spark cluster with 10 executors, each with 6 cores and 30GB of memory, and with 50GB of driver memory.



China was the bigger emitter of CO2 in 2017 with [ 10.48 ] GtCO2. In 2017, there was little-to-no reductions in US emissions and a sizeable increase in Chinese emissions ([ 0.33* ] GtCO2). India's emissions increased a bit more slowly in 2017 to [ 2.47 ] GtCO2 from [ 2.43 ] GtCO2 in 2016, while the EU's emissions have remained relatively flat since 2014 ([ 3.47* ] GtCO2) and did not noticeably change in 2017 ([ 3.48 ] GtCO2). The growth in emissions from 2016 to 2017 also more than doubled in the rest of the world from [ 0.15* ] to [ 0.3* ] GtCO2. Overall world CO2 emission in 2017 was [ 36.79* ] GtCO2 compared to [ 36.18* ] GtCO2 in 2016, with an increase of [ 0.6* ] GtCO2 where China contributed with more than half of this increase.

Highlighted Cells are related with
conf: 16.80%; aggr-func: sum

| Year | China | India | United States | European Union | Rest of world |
|---|---|---|---|---|---|
| 2014 | 10.28 | 2.24 | 5.57 | 3.47 | 14.52 |
| 2015 | 10.15 | 2.32 | 5.41 | 3.50 | 14.64 |
| 2016 | 10.15 | 2.43 | 5.31 | 3.50 | 14.79 |
| 2017 | 10.48 | 2.47 | 5.28 | 3.48 | 15.08 |

Annual global CO2 emissions from fossil fuels (black ba by country (coloured bars). Negative values indicate reductions in emissions. Data from the Global g Highcharts.

As 2017 is not yet over – and somewhat limited data is ct to large uncertainties. The global estimate of a [2%] increase in emissions could be as low as [0.8 The GCP will publish more complete 2017 numbers in early 2018 when all the data is available.

*China*

The GCP reports that China's emissions are projected ainty ranging from [0.7%] to [5.4%] China

Figure 6.4: The figure shows the results of the system as described in Section 6.5

The throughput of our system is 2.5K documents per minute. For full evaluation results refer to Chapter 5.

## 6.5 Demo Overview

In this demonstration, we will show how ExQuisiTe aligns quantity mentions in web pages. ExQuisiTe is web-based therefore it only requires a modern web browser. It has two main views: the first view is for configuring the system and selecting the type of input document, and the second view is for displaying the alignment results.

### Configuration and Input

ExQuisiTe gives the user control over the settings of the different components. The user can select the type of input she wants to process. Currently, we support HTML input given as a valid page URL or a file containing the HTML content.
For the global resolution the user can choose either (i) Random Walk With Restart (RWR) or (ii) No Global Resolution. RWR is the default option. The second option deactivates the global resolution module and uses only the outcome of the classifier. It uses the confidence value given by the classifier to select the highest-confidence table mention for each text mention.
The user can adjust the threshold for the final confidence score: mapping the text mention to a table mention or to a NIL. In the case of local resolution only, this threshold is applied to the classifier's confidence score. In the case of RWR global resolution, the threshold is applied to the RWR outcome. Finally, there is an option to turn adaptive filtering on or off.

### Results Display and Interactive Exploration

The system processes the document according to the user's configuration. Then, it displays the results embedded in the original HTML document as shown in Figure 6.4. The text quantity mentions appear between square bracket. Each text mention is hyperlinked to its aligned table mention using a colored button with the mention's surface form as the hypertext. The color of the button is determined by the table, such that each table is assigned a color and all its related mentions in the text are assigned the same color.

The system marks the aggregate quantity mentions with a '*' superscript. When a text mention is clicked, the system displays a pop-up with the table. This pop-up includes—in addition to the table—the confidence of the alignment, the type of alignment, and the related cells in the table. The type of alignment can take one of the following values: single-cell, average, sum, difference, percentage, and change ratio.

The system marks the salient sentences for downstream summarization with yellow background. For each table, the system marks the sentence with the highest coverage. In Figure 6.4, the last sentence has the highest coverage of table cells.

## 6.6 Related Work

Although information extraction research has targeted Web tables, no prior work has examined the relation between mentions in the text and tables within a document. Quantity annotation has been addressed in [SC14, IRW16], but these methods rely on external knowledge bases, linking table cells to entries in the knowledge base. Further methods focused on named entities, by annotating table cells with entities and classes from the knowledge base [LSC10, RLB15, BND15, RB17, GRE$^+$17]. Table data fusion for search and schema inference was studied in [VHM$^+$11, YGCC12, ZC13, LB17].

Our work differs from all these prior works in two main aspects: (i) we do not rely on any external knowledge base, and (ii) we handle approximate and aggregated quantities mentions which do not have exact matches.

# 7 Lessons Learned and Outlook

This chapter summarizes the contributions of this thesis and discusses possible future directions to pursue.

## 7.1 Contributions

### 7.1.1 Semantic Representation of Quantities

The first contribution of this thesis is the `Quantity Knowledge Base (QKB)`. The QKB provides a taxonomy that provides a semantic representation of quantity mentions as a triple of unit, measure, and value. The QKB includes conversion rules to the international system of units (SI). When we began our research in quantity understanding, there was limited support for quantities in `Wikipedia` and in other knowledge bases, such as `Yago` or `DBpedia`. Therefore, it was essential for us to construct the QKB for anchoring quantity mentions. However, now `Wikipedia` provides comprehensive lists of SI base units and SI derived units.
We make the QKB available for download to the community. It can be extended to include units of specialized domain, e.g. material science.

### 7.1.2 Canonicalization of Quantity Mentions

The second contribution of this thesis is the `Equity` framework. Equity disambiguates table cells and headers to quantities, entities, concepts, and classes that reside in a knowledge base. Equity disambiguates mentions of quantities to the QKB, mentions of entities and classes to Yago, and mentions of concepts to Wikipedia pages. In Equity, we model the disambiguation problem using an MRF model. We distantly train the MRF model using relatedness measures from a knowledge base. Then, we cast the disambiguation problem into an inference problem over a graph and we employ a Random Walks algorithm to solve this inference task. Our algorithm incorporates cues from the text and the table to jointly disambiguate the mentions.
A limitation of Equity lies within the coverage of the knowledge base. That is quite evident in the limited data to estimate the relatedness between specific measures and units or entities and concepts. However, adding new sources of data can overcome this limitation.
We make the source code of Equity available for the research community. It can be used with specialized knowledge bases to disambiguate mentions of quantities in specific domains of interest.

### 7.1.3 Alignment of Quantity Mentions in Tables and Text

The third contribution of this thesis is introducing the quantity alignment problem, and proposing `BriQ` to solve it. We define the quantity alignment problem as computing bidirectional links between textual mentions of quantities and the corresponding table cells. The aim of finding these alignments is to support advanced content summarization and to facilitate navigation between the explanation of quantities in text and the details in tables. We also propose `ExQuisiTe`: a system that identifies relations between quantities in text and tables. ExQuisiTe automatically detects these relations and generates an easy-to-read document where numbers in the text are linked to their source tables and respective cells.

We publish the source code of both BriQ and ExQuisiTe as well as the training dataset for the research community.

## 7.2 Outlook

We believe that the following are important research directions to pursue further.

#### Quantity aggregation across multiple documents

We proposed Equity to anchor mentions of quantities, entities, concepts, and classes to a knowledge base. However, Equity only processes single documents. It does not handle multiple documents with potentially overlapping contents. An interesting question arises here: how can anchoring mentions to a knowledge base help in data fusion tasks? Canonicalizing mentions of quantities, entities, concepts, and classes to a knowledge base is expected to aid data fusion tasks. Tables on the web contain valuable data, but each table holds a certain view of the data. Fusing tables across multiple documents aids in constructing a full view of the data.

Another question is how grouping multiple documents can facilitate the disambiguation of mentions? We can examine the effect of adding textual cues from multiple documents on the disambiguation task. Also, we can devise collective disambiguation algorithms that can process multiple tables from multiple documents at a time. Then, we may examine the effect of disambiguating tables collectively.

#### Generating natural language descriptions of web tables

BriQ aligns quantity mentions in text and tables. We can use BriQ to align quantities in documents from a specific domain, then study these alignments to understand table quantities that are mentioned in the text. Using the data produced by the alignment task, we can develop Natural Language Generation (NLG) algorithms that are capable of producing a textual description of tabular data in a specific domain. The challenges in this task are: (i) How to identify important entities? (ii) How to identify important quantities? (iii) How to use aggregations and approximations in the text?

### Downstream Applications: smart editor and text summarization system

Our work facilitates multiple downstream applications. One straightforward application would be a smart report editor that is capable of suggesting quantity mentions to add to the text. Such an editor would hyperlink these mentions to their corresponding mentions in tables. It can also suggest adding extra tables to support the content of the report. Another class of applications that can benefit from the work in this thesis are text summarization systems. Using the relations inferred by ExQuisiTe would potentially enrich the generated summary and enhance comprehensibility.

# Bibliography

[AB95] Chinatsu Aone and Scott William Bennett. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL 1995)*, pages 122–129. Association for Computational Linguistics, 1995.

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web and the 2nd Asian Semantic Web Conference (ISWC 2007/ASWC 2007)*, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[Ait02] James S Aitken. Learning information extraction rules: An inductive logic programming approach. In *ECAI*, pages 355–359, 2002.

[ARHB+93] Douglas Appelt, Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson. Fastus: A finite-state processor for information extraction from real-world text. pages 1172–1178, 01 1993.

[AS18] Omar Alonso and Thibault Sellam. Quantitative information extraction from social data. pages 1005–1008, 2018.

[BCS+07] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*, pages 2670–2676, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[BEP+08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, pages 1247–1250, New York, NY, USA, 2008. ACM.

[BKR11] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Markov Random Fields for Vision and Image Processing.* The MIT Press, 2011.

[BND15] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel: Entity linking in web tables. In Marcelo Arenas, Oscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *Proceedings of The 14th International Semantic Web Conference (ISWC 2015)*, pages 425–441. Springer International Publishing, 2015.

[BP06] Razvan Bunescu and Marius Paşca. Using encyclopedic knowledge for named entity disambiguation. In *11th conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[CBD94] Dennis Connolly, John D. Burger, and David S. Day. A machine learning approach to anaphoric reference. In *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP 1994)*, pages 133–144. ACL, 1994.

[CBK$^+$10] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 1306–1313. AAAI Press, 2010.

[Cha07] Soumen Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pages 571–580, New York, NY, USA, 2007. ACM.

[CHK09] Michael J. Cafarella, Alon Halevy, and Nodira Khoussainova. Data integration for the relational web. *Proceedings of the VLDB Endowment (PVLDB 2009)*, 2(1):1090–1101, August 2009.

[CHM11] Michael J. Cafarella, Alon Halevy, and Jayant Madhavan. Structured data on the web. *Communications of the ACM CACM 2011*, 54(2):72–79, February 2011.

[CHW$^+$08] Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: Exploring the power of tables on the web. *Proceedings of the VLDB Endowment (PVLDB 2008)*, 1(1):538–549, August 2008.

[CHZ$^+$08] Michael J Cafarella, Alon Y Halevy, Yang Zhang, Daisy Zhe Wang, and Eugene Wu. Uncovering the relational web. June 2008.

[CM99] Mary Elaine Califf and Raymond J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence (AAAI 1999 / AAI 1999)*, AAAI '99/IAAI '99, pages 328–334, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[CM12] Angel X. Chang and Christopher Manning. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3735–3740. European Language Resources Association (ELRA), 2012.

[CM15] Kevin Clark and Christopher D. Manning. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL 2015)*, pages 1405–1415. Association for Computational Linguistics, 2015.

[CM16a] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2256–2262. Association for Computational Linguistics, 2016.

[CM16b] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, volume 1, pages 643–653. Association for Computational Linguistics, 2016.

[CMBT02] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. Gate: An architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, pages 168–175, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[CNM06] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. pages 161–168, 2006.

[Coh10] William W Cohen. Graph walks and graphical models. Technical report, 2010.

[CSMA16] Angel Chang, Valentin I. Spitkovsky, Christopher D. Manning, and Eneko Agirre. A comparison of named-entity disambiguation and word sense disambiguation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[Cuc07] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empir-*

*ical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, 2007.

[Cuc14] Silviu Cucerzan. Name entities made obvious: The participation in the erd 2014 evaluation. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation (ERD 2014)*, pages 95–100, New York, NY, USA, 2014. ACM.

[DCG13] Luciano Del Corro and Rainer Gemulla. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web (WWW 2013)*, pages 355–366, New York, NY, USA, 2013. ACM.

[DGH$^+$14] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014)*, pages 601–610, New York, NY, USA, 2014. ACM.

[DK14] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics (TACL 2014)*, 2:477–490, 2014.

[ERD] ERD 2014: Acm international workshop on entity recognition & disambiguation, co-located with sigir 2014.

[ESW18] Patrick Ernst, Amy Siu, and Gerhard Weikum. Highlife: Higher-arity fact harvesting. In *Proceedings of the 27th International Conference on World Wide Web (WWW 2018)*, pages 1013–1022, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[ETBL15] Julian Eberius, Maik Thiele, Katrin Braunschweig, and Wolfgang Lehner. Top-k entity augmentation using consistent set covering. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management (SSDBM 2015)*, pages 8:1–8:12, New York, NY, USA, 2015. ACM.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[Fle71] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 1971.

[GB14] Zhaochen Guo and Denilson Barbosa. Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014)*, pages 499–508, New York, NY, USA, 2014. ACM.

[GCK13] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 1020–1030. Association for Computational Linguistics, 2013.

[GG17] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, Jan 2017.

[Gra06] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[GRE$^+$17] Anna Lisa Gentile, Petar Ristoski, Steffen Eckel, Dominique Ritze, and Heiko Paulheim. Entity matching on web tables: a table embeddings approach for blocking. In *Proceedings of the 20th International Conference on Extending Database Technology (EDBT 2017), Venice, Italy, March 21-24, 2017*, pages 510–513, Konstanz, 2017. OpenProceedings. Online-Ressource.

[GS09] Rahul Gupta and Sunita Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proceedings of the VLDB Endowment (PVLDB 2009)*, 2(1):289–300, August 2009.

[Hav02] Taher H Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web (WWW 2002)*, pages 517–526. ACM, 2002.

[Hav03] T. H. Haveliwala. Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering (TKDE 2003)*, 15(4):784–796, July 2003.

[HK09] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 1152–1161. Association for Computational Linguistics, 2009.

[HYB$^+$11] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[IAYW] Yusra Ibrahim, Mohamed Amir Yosef, and Gerhard Weikum. Aida-social: Entity linking on the social stream. In *Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR 2014)*.

[IR<sup>+</sup>19] Yusra Ibrahim, Mirek Riedewald, , Demetrios Zeinalipour-Yazti, and Gerhard Weikum. Bridging quantities in tables and text. *Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE 2019)*, 2019.

[IRW16] Yusra Ibrahim, Mirek Riedewald, and Gerhard Weikum. Making sense of entities and quantities in web tables. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Managemen (CIKM 2016)*, pages 1703–1712, New York, NY, USA, 2016. ACM.

[IW19] Yusra Ibrahim and Gerhard Weikum. Exquisite: Explaining quantities in text. *Proceedings of The Web Conference. (WWW 2019)*, 2019.

[JM09] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall, Pearson Education International, 2. ed., pearson international edition edition, 2009.

[JS02] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, October 2002.

[KFB09] Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques.* The MIT press, 2009.

[KSRC09] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2009)*, pages 457–466. ACM, 2009.

[LB17] Oliver Lehmberg and Christian Bizer. Stitching web tables for improving matching quality. *Proceedings of the VLDB Endowment (PVLDB 2017)*, 10(11):1502–1513, August 2017.

[LBB<sup>+</sup>12] Steve Lawrence, Ian Burns, Andrew Back, Ah Chung Tsoi, and C. Lee Giles. *Neural Network Classification and Prior Class Probabilities*, pages 295–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[LBS<sup>+</sup>16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 260–270. Association for Computational Linguistics, 2016.

[LCP$^+$13] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, December 2013.

[Len95] Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of ACM*, 38(11):33–38, November 1995.

[LIJ$^+$04] Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. Association for Computational Linguistics, 2004.

[LM06] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings.* Princeton University Press, 2006.

[LMC11] Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 529–539, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[LSC10] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *The Proceedings of the VLDB Endowment (PVLDB 2010)*, 3(1-2):1338–1347, September 2010.

[LSRP15] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics (TACL 2015)*, 3:503–515, 2015.

[MAAH09] Jayant Madhavan, Loredana Afanasiev, Lyublena Antova, and Alon Y. Halevy. Harnessing the deep web: Present and future. In *Proceedings of The fourth biennial Conference on Innovative Data Systems Research (CIDR 2009)*, 2009.

[MC07] Rada Mihalcea and Andras Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM 2007)*, pages 233–242, New York, NY, USA, 2007. ACM.

[MFJ13] Varish Mulwad, Tim Finin, and Anupam Joshi. Semantic message passing for generating linked data from tables. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proceedings of The 12th International Semantic Web Conference (ISWC 2013)*, pages 363–378, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[MGMR02] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002)*, pages 117–128, Feb 2002.

[Mil98] George Miller. *WordNet: An electronic lexical database.* The MIT press, 1998.

[MJGSB11] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS 2011)*, pages 1–8, New York, NY, USA, 2011. ACM.

[ML95] Joseph F. McCarthy and Wendy G. Lehnert. Using decision trees for conference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, volume 2, pages 1050–1055, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[ML03] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning (CONLL 2003) at NAACL-HLT 2003 - Volume 4*, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[MMM$^+$16] Aman Madaan, Ashish Mittal, Mausam, Ganesh Ramakrishnan, and Sunita Sarawagi. Numerical relation extraction with minimal supervision. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 2764–2771. AAAI Press, 2016.

[MRN14] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics (TACL 2014)*, 2:231–244, 2014.

[MS01] Marina Maila and Jianbo Shi. A random walks view of spectral segmentation. In *AI and STATISTICS (AISTATS) 2001*, January 2001.

[MSB$^+$12] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL*

*2012)*, pages 523–534, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[MSB13] Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 447–457. Association for Computational Linguistics, 2013.

[Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012.

[MW08] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, pages 509–518, New York, NY, USA, 2008. ACM.

[NC02] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, pages 104–111, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[Ng10] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1396–1411. Association for Computational Linguistics, 2010.

[Ng17] Vincent Ng. Machine learning for entity coreference resolution: A retrospective look at two decades of research, 2017.

[NM12] Ani Nenkova and Kathleen McKeown. *A Survey of Text Summarization Techniques*, pages 43–76. Springer US, Boston, MA, 2012.

[NMC05] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning (ICML 2005)*, pages 625–632, New York, NY, USA, 2005. ACM.

[NMTG16] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, Jan 2016.

[Nor98] James R Norris. *Markov chains.* Number 2. Cambridge university press, 1998.

[NTW16] Dat Nguyen, Martin Theobald, and Gerhard Weikum. J-nerd: Joint named entity recognition and disambiguation with rich linguistic features. *Transactions of the Association for Computational Linguistics (TACL 2016)*, 4:215–229, 2016.

[NUPP16] Sebastian Neumaier, Jürgen Umbrich, Josiane Xavier Parreira, and Axel Polleres. Multi-level semantic labelling of numerical values. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *Proceedings of the 2016 International Semantic Web Conference (ISWC 2016)*, pages 428–445, Cham, 2016. Springer International Publishing.

[NZRS12] Feng Niu, Ce Zhang, Christopher Re, and Jude W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. In Marco Brambilla, Stefano Ceri, Tim Furche, and Georg Gottlob, editors, *Proceedings of the Second International Workshop on Searching and Integrating New Web Data Sources (VLDS 2012)*, volume 884 of *CEUR Workshop Proceedings*, pages 25–28. CEUR-WS.org, August 2012.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[PF14] Francesco Piccinno and Paolo Ferragina. From tagme to wat: A new entity annotator. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation (ERD 2014)*, pages 55–62, New York, NY, USA, 2014. ACM.

[PHG15] Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, pages 238–243. Association for Computational Linguistics (ACL), 2015.

[PJN13] Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, volume 1, pages 1341–1351. Association for Computational Linguistics, 2013.

[PS12] Rakesh Pimplikar and Sunita Sarawagi. Answering table queries on the web using column keywords. *Proceedings of the VLDB Endowment (PVLDB 2012)*, 5(10):908–919, June 2012.

[RB17] Dominique Ritze and Christian Bizer. Matching web tables to dbpedia - a feature utility study. In *Proceedings of the 20th International Conference on Extending Database Technology (EDBT 2017), Venice, Italy, March 21-24, 2017*, pages 210–221, Konstanz, 2017. OpenProceedings.

[Ril93] Ellen Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI 1993)*, pages 811–816. AAAI Press, 1993.

[RLB15] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics (WIMS 2015)*, pages 10:1–10:6, New York, NY, USA, 2015. ACM.

[RLR+10] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 492–501, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[RMKS15] S.K. Ramnandan, Amol Mittal, Craig A. Knoblock, and Pedro Szekely. Assigning semantic labels to data sources. In Fabien Gandon, Marta Sabou, Harald Sack, Claudia d'Amato, Philippe Cudré-Mauroux, and Antoine Zimmermann, editors, *Proceedings of the 12th European Semantic Web Conference on The Semantic Web (ESWC 2015). Latest Advances and New Domains - Volume 9088*, pages 403–417, Cham, 2015. Springer International Publishing.

[RN09] Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, volume 2, pages 968–977, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[RRDA11] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT 2011)*, volume 1, pages 1375–1384, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[RVR15] Subhro Roy, Tim Vieira, and Dan Roth. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics (TACL 2015)*, 3:1–13, 2015.

[SA15] Thibault Sellam and Omar Alonso. Raimond: Quantitative data extraction from twitter to describe events. In *Proceedings of the 15th International Conference on Engineering the Web in the Big Data Era (ICWE 2015) - Volume 9114*, pages 251–268, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[Sar08] Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, March 2008.

[SC05] Sunita Sarawagi and William W Cohen. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems (NIPS 2005)*, pages 1185–1192, 2005.

[SC14] Sunita Sarawagi and Soumen Chakrabarti. Open-domain quantity queries on web tables: Annotation, response, and consensus models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014)*, pages 711–720, New York, NY, USA, 2014. ACM.

[SDNR07] Warren Shen, AnHai Doan, Jeffrey F. Naughton, and Raghu Ramakrishnan. Declarative information extraction using datalog with embedded extraction predicates. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB 2007)*, pages 1033–1044. VLDB Endowment, 2007.

[SG15] Jannik Strötgen and Michael Gertz. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 541–547. Association for Computational Linguistics, 2015.

[SG16] Jannik Strötgen and Michael Gertz. *Domain-Sensitive Temporal Tagging*, volume 9. Morgan & Claypool Publishers, 2016.

[SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pages 697–706, New York, NY, USA, 2007. ACM.

[Sod99] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272, Feb 1999.

[SP03] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL 2003) - Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.

[SPM17] Swarnadeep Saha, Harinder Pal, and Mausam. Bootstrapping for numerical open ie. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 317–323. Association for Computational Linguistics, 2017.

[SWH15] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering (TKDE 2015)*, 27(2):443–460, Feb 2015.

[TFP06] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM 2006)*, pages 613–622. IEEE, 2006.

[TMN04] Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. Learning random walk models for inducing word dependency distributions. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004)*, pages 103–, New York, NY, USA, 2004. ACM.

[URNN+15] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. Gerbil: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web (WWW 2015)*, pages 1133–1143, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[VHM+11] Petros Venetis, Alon Halevy, Jayant Madhavan, Marius Paşca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment (PVLDB 2011)*, 4(9):528–538, June 2011.

[WMLC15] William Yang Wang, Kathryn Mazaitis, Ni Lao, and William W. Cohen. Efficient inference and learning in a large knowledge base. *Machine Learning*, 100(1):101–126, Jul 2015.

[YBE+12] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. Natural language questions for the web of data. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL 2012)*, pages 379–390, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[YGCC12] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (SIGMOD 2012)*, pages 97–108, New York, NY, USA, 2012. ACM.

[YSL+08] Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu, and Sheng Li. An entity-mention model for coreference resolution with inductive logic programming. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 843–851, June 2008.

[YSZT04] Xiaofeng Yangy, Jian Su, Guodong Zhou, and Chew Lim Tan. An np-cluster based approach to coreference resolution. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004*, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[ZC13] Meihui Zhang and Kaushik Chakrabarti. Infogather+: Semantic matching and annotation of numeric and time-varying attributes in web tables. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD 2013)*, pages 145–156, New York, NY, USA, 2013. ACM.

[ZLR05] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, language technologies institute, 2005.

[ZSG16] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Doser - a knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *proceedings of the 15th International Semantic Web Conference (ISWC 2016). Latest Advances and New Domains*, pages 182–198, Cham, Germany, 2016. Springer International Publishing.