Foundational Research
Propelled by Text Analytics

Benny Kimelfeld
LogicBlox
Preamble

• Myself:
  – Ph.D. @ HebrewU (DB uncertainty + search)
  – IBM Almaden (DB theory, IR, Text Analytics)
  – LogicBlox (ML in DB, Prob. Programming)
  – Technion IL (Associate Prof., next year)

• This talk:
  ▪ Infrastructure for text analytics
    + DB theory, formal languages, NLP, data mining, computational complexity, …
Outline

Text Analytics in the Big Data Era

• Information Extraction Systems & Formalism
• Foundational Research Challenges
• Summary
Text Analytics Matters

Some important applications are based on the analysis of text-centric data; for example:

**Semantic Search**
Semantic understanding & indexing of content to better match user's intent

**Life-Science Mining**
Extract knowledge bases from scientific publications

**Log Analysis**
Summarize, visualize and analyze logs produced by machines

**e-Commerce**
Comparison Shopping extracts & compares inventory from online sources

**CRM / BI**
Monitor customer’s social-media activity for sentiment & business leads
Database Management Systems

• Old news: Data management is involved!
  – Data semantics, query/analysis semantics, storage, query evaluation, indices, consistency, transactions, backup, privacy, recovery, …
  – From-scratch engineering is highly challenging

• Motivation to the concept of a general-purpose *Database Management System*
  – Most notably: relational model (pioneered by Edgar F. Codd in 1969) and SQL
## “Big Data” Phenomena

<table>
<thead>
<tr>
<th><strong>Past:</strong></th>
<th><strong>Present:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Proprietary data in orgs. (enterprises, governments, …)</td>
<td>Proliferation of publically open data sources (Web, social, …)</td>
</tr>
<tr>
<td>Data structured/controlled by admins, e-forms, software, …</td>
<td>Uncontrolled data from humans’ free text, heterogeneous kbs, …</td>
</tr>
<tr>
<td>Massive-data analyses incurred high machinery/personnel cost</td>
<td>Business models (cloud, crowd, opensource) facilitate analyses</td>
</tr>
<tr>
<td>Analyses by specialized teams of heavily trained experts</td>
<td>Analyses by a wide community featuring a wide range of skills</td>
</tr>
</tbody>
</table>
“By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions.”

“We need dev. & management systems to facilitate value extraction from Big Data by a wide range of users / skills”
“Information Extraction (IE) is the name given to any process which selectively structures and combines data which is found, explicitly stated or implied, in one or more texts. The final output of the extraction process varies; in every case, however, it can be transformed so as to populate some type of database.”

J. Cowie and Y. Wilks., Handbook of Natural Language Processing, 2000

“In short: data-in-text → data-in-db
(unstructured) (structured)

“Information extraction is the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making the information more suitable for information processing tasks.”

Popular Classes of IE Tasks

- Named Entity Recognition

From September 1936 to July 1938, Turing spent most of his time studying under Church at Princeton University. In June 1938, he obtained his PhD from Princeton.
Popular Classes of IE Tasks

- Named Entity Recognition
- Relation Extraction

From September 1936 to July 1938, Turing spent most of his time studying under Church at Princeton University. In June 1938, he obtained his PhD from Princeton.
Popular Classes of IE Tasks

- Named Entity Recognition
- Relation Extraction
- Event Extraction

From September 1936 to July 1938, Turing spent most of his time studying under Church at Princeton University. In June 1938, he obtained his PhD from Princeton.
Popular Classes of IE Tasks

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal IE
Popular Classes of IE Tasks

- Named Entity Recognition
- Relation Extraction
- Event Extraction
- Temporal IE
- Coreference Resolution

From September 1936 to July 1938, Turing spent most of his time studying under Church at Princeton University. In June 1938, he obtained his PhD from Princeton.
IE Paradigms: Rules & Statistics

- Rule-based
- ML classification
- Probabilistic graphical models
- Soft logic

“[…] rules are effective, interpretable, and are easy to customize by non-experts to cope with errors.”

Gupta & Manning, CONLL’14

- 54 industrial vendors (Who’s Who in Text Analytics, 2012)

[Chiticariu, Li, Reiss, EMNLP’ 13]
Outline

• Text Analytics in the Big Data Era
• Information Extraction Systems & Formalism
• Foundational Research Challenges
• Summary
Xlog: Datalog for IE

[Shen, Doan, Naughton, Ramakrishnan, VLDB 2007]

- Extension of (non-recursive) Datalog
- Use case: DBLife (db research kb: dblife.cs.wisc.edu)
- Data types: string, document, span
  - Focus on single-document programs
- “Procedural predicates” (p-predicates) are user-defined functions that produce relations over spans
  - Example: sentence(doc, span)
- Query-plan optimization

Kaspersky Lab CEO Eugene Kaspersky said Intel CEO Paul Otellini and the Intel board had no idea what they were in for when the company announced it was acquiring McAfee on August 19, 2010.

Same string, different spans
Xlog Example

[Shen, Doan, Naughton, Ramakrishnan, VLDB 2007]

people(d, personMention) :- docs(d), personPatterns(personPattern),
match(d, personPattern, personMention)

conferences(d, conferenceMention) :- docs(d), confPatterns(confPattern),
match(d, confPattern, conferenceMention).

chairType(d, chairType, chairPosition) :- docs(d), chairTypePatterns(chairTypePattern),
match(d, chairTypePattern, chairType),
match(d, "(?i)(vice\W+)?(co-)?chair", chairPosition),
isBefore(chairType, chairPosition),
distChar(chairType, chairPosition) < 20.

chair(d, personMention, conferenceMention, chairPosition, chairType) :-
people(d, personMention), conferences(d, conferenceMention),
chairType(d, chairType, chairPosition),
isBefore(conferenceMention, chairType),
isBefore(chairPosition, personMention),
distChar(chairPosition, personMention) < 20.

Figure 3: A sample Xlog program in our experiments.

“Declarative Information Extraction using Datalog with Embedded Extraction Predicates”
Instaread: Datalog + NLP

[Hoffmann, 2012]

- Datalog syntax
  - Types: string, span

- Built in collection of p-predicates
  - Various types of built-in regex formulas

\[
\text{killed}(a, c) \iff \text{next}(a, b) \land \text{next}(b, c) \land \text{token}(b, \text{"killed"}) \\
\land \text{capitalized}(a) \land \text{capitalized}(b)
\]

- Linguistic: deep parsing, coreference resolution, named-entity extractor

For tractability, we require rules to be in safe domain-relational calculus [154]. See section 4.6 for details.
IBM SystemT: SQL for IE

• Engine for AQL: SQL-like declarative IE lang.
  – AQL = Annotation Query Language

• SystemT = AQL + Runtime + Dev. Tooling
  – [Chiticariu et al., ACL 2010]: position SystemT as a high-quality and high-efficiency IE solution
  – System and IDE demos in ACL 2011, SIGMOD 2011

• Commercial product, high academic presence
  – Integration on public financial records [Hernández et al., EDBT’13, Balakrishnan et al. SIGMOD’10], NER [Chiticariu et al. EMNLP’10, ACL’10, Nagesh et al. EMNLP’12, Roy et al. SIGMOD’13], IR [Zhu et al. WWW’10, K et al. SIGIR’12, CIKM’12], sentiment analysis [Hu et al., Interact’13], social media [Sindhwani et al., IBM Journal 2011]
SystemT’s AQL Example

```
cREATE VIEW Caps AS
EXTRACT REGEX /[A-Z](\w|-)+/ ON D.Text AS name FROM Document D;

CREATE VIEW Last AS
EXTRACT DICTIONARY LastGaz ON D.Text AS name FROM Document D;

CREATE VIEW CapsLast AS
SELECT CombineSpans(C.name, L.name) AS name
FROM Caps C, Last L
WHERE FollowsTok(C.name, L.name, 0, 0);

... REGEX + JOIN WITH PREVIOUS VIEWS...

CREATE VIEW PersonAll AS
(SELECT R.name FROM FirstLast R) UNION ALL ...

CREATE VIEW Person AS SELECT * FROM PersonAll R
CONSOLIDATE ON R.name USING 'ContainedWithin';

OUTPUT VIEW Person;
```

[Chiticariu, Krishnamurthy, Li, Raghavan, Reiss, Vaithyanathan, ACL 2010]
Formal Framework

• Repeated concept: Extend a relational query language with text transducers (p-predicates, usually regex formulas)

• Research challenge: theoretical underpinnings of this combined document/relation model

• Expressive power
  – Query-plan optimization: Can we rewrite an operator via “easier” building blocks?
  – System extensions: Can we express a new operation using existing ones, or prove impossibility?

• Next: a formal framework
  – With Fagin, Reiss, Vansummeren, PODS’13, JACM
Kaspersky Lab CEO Eugene Kaspersky said Intel CEO Paul Otellini and the Intel board had no idea what they were in for when the company announced it was acquiring McAfee on August 19, 2010.
Document Spanners

**Document Spanner**: a function that maps every doc. (string) into a relation over the doc.’s spans

More formally:
- Finite alphabet $\Sigma$ of *symbols*
- A spanner maps each doc. $d \in \Sigma^*$ into a relation over the spans $[i,j)$ of $d$
- The relation has a **fixed signature** (set of attributes)
  - The attributes come from an infinite domain of *variables* $x, y, z, \ldots$

Kaspersky Lab CEO Eugene Kaspersky said Intel CEO Paul Otellini and the Intel board had no idea what they were in for when the company announced it was acquiring McAfee on August 19, 2010.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,14)</td>
<td>[30,36)</td>
<td>[1,36)</td>
</tr>
<tr>
<td>[42,47)</td>
<td>[52,65)</td>
<td>[42,65)</td>
</tr>
<tr>
<td>[102,110)</td>
<td>[115,125)</td>
<td>[102,125)</td>
</tr>
</tbody>
</table>

*Document d*

*Relation over the spans of d*
Spanners as Regex Formulas

- Regular expression with embedded variables
  \[ \gamma ::= \emptyset \mid \varepsilon \mid \sigma \mid \gamma \lor \gamma \mid \gamma \cdot \gamma \mid \gamma^* \mid x\{\gamma\} \]

- Examples:
  - .* \(x\{d\d\d\d\}\).*
  - .* in \(w\{\text{Alabama} \mid \text{Alaska} \mid \text{Arizona} \mid \ldots\}\).*
  - (.\(.*z\{[A-Z][a-z]^*\}, y\{[A-Z][a-z]^*\}\}).*\) | ...

- Restriction: each “evaluation” (parse tree) assigns one span to each variable (see [Fagin et al., PODS’13])

Representation system for spanners
Spanners as Datalog w/ Regex

- Non-recursive Datalog (NR-Datalog)
- Operate over a document (not a relational db)

Token(x) := \((\varepsilon | \cdot \_ ) x[[\text{a-zA-Z}]+] ( ((,V\_ ) \cdot \_ ) | \varepsilon ) \]
State(x) := Token(x), [\cdot \_ x\{\text{Georgia | Virginia | Washington}\}.\_]
Cap1st(x) := Token(x), [\cdot \_ x\{\text{[A-Z]}\}.\_].\_]
CommaSp(x,y,z) := [\cdot \_ z\{\cdot \_ \} ,\_ y\{\cdot \_ \}].\_]
Loc(z) := CommaSp(x,y,z), Cap1st(x), State(y)

RETURN(x,z) := Cap1st(x), [\cdot \_ x\{\cdot \_ \}_\text{from}_z\{\cdot \_ \}].\_], Loc(z)

Carter_from_Plains,_Georgia,_Washington_from_Westmoreland,_Virginia

Another representation system for spanners

<table>
<thead>
<tr>
<th>x</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,7) Carter</td>
<td>[13,28) Plains,_Georgia</td>
</tr>
</tbody>
</table>
Spanners as Automata

- In an accepting run, each variable opens and later closes exactly once
  ⇒ Each accepting run defines an assignment to the variables

- Nondeterministic ⇒ multiple accepting runs ⇒ multiple tuples

Another representation system for spanners
Study of Expressive Power

Spanners definable by var-stack automata

Spanners definable by var-set automata

Spanners definable by regex formulas

Spanners definable by NR Datalog w/ regex formulas

---

Token(x) := \([ \epsilon \mid \_ \_ \_ \] x{[a-zA-Z]*} ( ((,\_\_\_ \_ \_ \_ \_ \_\_\_\_) \mid \epsilon ) \)
State(x) := Token(x) , [.* x{Georgia|Virginia|Washington}.]*
Cap1st(x) := Token(x) , [.* x{[A-Z].*}.]*
CommaSp(x,y,z) := [.* z{.*},_ y{.*}).]*
Loc(z) := CommaSp(x,y,z) , Cap1st(x) , State(y)
RETURN(x,z) := Cap1st(x) , [.*x{.*}_from_z{.*}).*] , Loc(z)
Consequences

• Connections between Datalog+regex spanners and other language formalisms
  – Classic string relations [Berstel 79]
  – Graph queries (CRPQs) [Cruz et al. 87]

• Extension with string equality & difference
  – Expressiveness / closure properties

• Principles for cleaning inconsistencies
  – Follow up work [PODS’14]
  – Next…
Outline

• Text Analytics in the Big Data Era

• Information Extraction Systems & Formalism

Foundational Research Challenges

• Summary
Next, highlight 3 lines of foundational research that were motivated by our work on text analytics:

1. Database inconsistency w/ repair priorities
2. Frequent subgraph mining
3. Update propagation
Cleaning IE Inconsistencies

• Extractors may produce inconsistent results
  – Data artifacts
  – Developer limitations

• Rather than repairing the existing extractors, common practice is to clean (intermediate) results
  – SystemT “consolidators” [Chiticariu et al. 10]
  – GATE/JAPE “controls” [Cunningham 02]
  – Implicit in other rule systems, e.g., WHISK [Soderland 99]
  – POSIX regex disambiguation [Fowler 03]
SystemT Consolidators

```
cREATE VIEW Caps AS
EXTRACT REGEX /\b[A-Z](\w+\d+\w*)/ ON D.text AS name FROM Document D;

CREATE VIEW Last AS
EXTRACT DICTIONARY LastGaz ON D.text AS name FROM Document D;

CREATE VIEW CapsLast AS
SELECT CombineSpans(C.name, L.name) AS name
FROM Caps C, Last L
WHERE FollowsTok(C.name, L.name, 0, 0);
...

CREATE VIEW PersonAll AS
  (SELECT R.name FROM FirstLast R) UNION ALL ...
  ... UNION ALL (SELECT R.name FROM CapsLast R);

CREATE VIEW Person AS SELECT * FROM PersonAll R
CONSOLIDATE ON R.name USING 'ContainedWithin';

OUTPUT VIEW Person;
```

[Chiticariu, Krishnamurthy, Li, Raghavan, Reiss, Vaithyanathan, ACL 2010]
Five GATE/JAPE Controls

Sequence 12345 and sequence 12.

Document

Spanner

.* x{\d\d+} .*
Cleaning via Prioritized Repairs

- Problem: existing policies are ad-hoc; how to expose a language for user declaration?
- [Fagin, K, Reiss, Vansummeren 2014]: spanner formalism for declarative cleaning
- Key: prioritized repairs [Staworko, et al. 12]
- Idea: Extend extraction programs with
  - Denial constraints: which facts are in conflict?
  - Priority declarations: preference between facts
- Captures SystemT, GATE, WHISK, POSIX, …
- We are now trying to improve our understanding of prioritized repairs…
## Prioritized Repairs: Definition

<table>
<thead>
<tr>
<th>Database</th>
<th>Denial Constraints</th>
<th>Priority Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of facts</td>
<td>Which sets of facts cannot co-exist?</td>
<td>Binary “is preferred to” relation</td>
</tr>
</tbody>
</table>

### Inconsistent Database Instance

- **[Arenas, Bertossi, Chomicki 99]**: Inconsistent DB represents a set of (equally likely) “repairs”
  - *Then we can ask for the “possible” or “consistent” query answers*
- **[Staworko, Chomicki, Marcinkowski 12]** add priorities:
- Let \(A\) and \(B\) be two consistent subsets of the database
- Say that \(A\) *improves* \(B\) if we can obtain \(A\) from \(B\) by a “profitable” exchange of facts (*precision later…*)
- A *repair* is a consistent subset that cannot be improved
A improves B if we get A from B by removing tuples & adding tuple; each removed preferred to by some added

<table>
<thead>
<tr>
<th>professor</th>
<th>university</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monica</td>
<td>ubiobio</td>
<td>Concepción</td>
</tr>
<tr>
<td>Monica</td>
<td>carleton</td>
<td>Ottawa</td>
</tr>
<tr>
<td>Jorge</td>
<td>uchile</td>
<td>Santiago</td>
</tr>
<tr>
<td>Jorge</td>
<td>ubiobio</td>
<td>Santiago</td>
</tr>
<tr>
<td>Pablo</td>
<td>uchile</td>
<td>Santiago</td>
</tr>
</tbody>
</table>

Violated constraints (*functional dependencies*):

- professor → university, city ("key constraint")
- university → city

“Ordinary” repairs [Arenas et al. 99]

<table>
<thead>
<tr>
<th>professor</th>
<th>university</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monica</td>
<td>ubiobio</td>
<td>Concepción</td>
</tr>
<tr>
<td></td>
<td>carleton</td>
<td>Ottawa</td>
</tr>
<tr>
<td>Jorge</td>
<td>uchile</td>
<td>Santiago</td>
</tr>
<tr>
<td></td>
<td>ubiobio</td>
<td>Santiago</td>
</tr>
<tr>
<td>Pablo</td>
<td>uchile</td>
<td>Santiago</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>professor</th>
<th>university</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monica</td>
<td>ubiobio</td>
<td>Concepción</td>
</tr>
<tr>
<td></td>
<td>carleton</td>
<td>Ottawa</td>
</tr>
<tr>
<td>Jorge</td>
<td>uchile</td>
<td>Santiago</td>
</tr>
<tr>
<td></td>
<td>ubiobio</td>
<td>Santiago</td>
</tr>
<tr>
<td>Pablo</td>
<td>uchile</td>
<td>Santiago</td>
</tr>
</tbody>
</table>

Tuple priority → some repairs can be discarded [Staworko et al.]
Complexity of Testing Improvability

Can a consistent subset be improved?

Theorem:

- In the case of a *single functional dependency* or *two keys* per relation, improvability can be tested in polynomial time.

- In any other combination of FDs, the problem is NP-complete.

Recent work (unpublished) w/ Fagin & Kolaitis.

<table>
<thead>
<tr>
<th>university</th>
<th>faculty</th>
<th>dean</th>
</tr>
</thead>
<tbody>
<tr>
<td>UChile</td>
<td>Economics</td>
<td>Agosin</td>
</tr>
<tr>
<td>Technion</td>
<td>CS</td>
<td>Yavneh</td>
</tr>
<tr>
<td>Stanford</td>
<td>Law</td>
<td>Magill</td>
</tr>
</tbody>
</table>
1. Apply dependency parsing

I want to buy my advisor a gift.

I really want to buy a gift to my advisor.

I want to buy a gift to the secretary and to my advisor.

[Zhang, Baldwin, Ho, K, Li, ACL13]: Restoring grammar in social media, sms, etc.
IE with Recurring Patterns

1. Apply dependency parsing
   - I want to buy my advisor a gift.

2. Find freq. recurring patterns
   - I really want to buy a gift to my advisor.

   - I want to buy a gift to the secretary and to my advisor.

   [Zhang, Baldwin, Ho, K, Li, ACL13]: Restoring grammar in social media, SMS, etc.
Maximal Frequent Subgraphs

<table>
<thead>
<tr>
<th>τ = 3</th>
<th>$g_1$</th>
<th>$g_2$</th>
<th>$g_3$</th>
<th>$g_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1" alt="Graph $g_1$" /></td>
<td><img src="image2" alt="Graph $g_2$" /></td>
<td><img src="image3" alt="Graph $g_3$" /></td>
<td><img src="image4" alt="Graph $g_4$" /></td>
</tr>
</tbody>
</table>
Complexity Study

• Naturally, there has been a lot of work on this problem
  – SPIN [Huan et al. 04], MARGIN [Thomas et al. 10], …

• But little was known about the computational complexity

• Studied: impact of assumptions on comp. complexity
  – Graph properties (e.g., trees, treewidth, etc.)
  – Label repeatability
  – Bounded #results desired
  – Bounded threshold

• This work led to novel complexity results and a new methodologies for mining maximal subgraphs
  – [K & Kolaitis, ACM PODS’13, ACM TODS]

• Next, some complexity nuggets →
Complexity Nuggets

• **Good news:** If labels do not repeat in each input graph, then there are PTime solutions when
  – The threshold is bounded; or
  – Graphs are trees & few results are desired

• In general graphs w/o label repetition, you can find 2 results in PTime
  – **Bad news:** But finding 3rd is NP-hard!
  – **Bad news:** And if labels repeat and graphs are trees, then finding 2nd is already NP-hard!
    • Even for a bounded threshold
Improving Dictionaries w/ Feedback

- **text fragments** (sentences, tables, rows, …)
- **company occurrences**
- **address occurrences**

- **IE**

**join**

auto. suggest a “good” fix to the IE program

“good” = small effect on other results

IBM, Armonk | Apple, Cupertino | Yahoo!, Cupertino | IBM, San Jose | Google
View Updates

- View-update problem: Translate an update on a view to an update on the base relations
- Deletion propagation as a special case
  - Update is delete(a set of view tuples)
- Motivation:
  - Classic: database/view maintenance
    - DB access only through views, hidden join keys, etc.
  - Debugging
    - [K&al.12]: deletion propagation for debugging text extractors
  - Database causality [Meliou&al.10]
    - Intuition: good propagation provides a good explanation of why we have the tuples to begin with
- [Bertossi, Salimi 14]: “Unifying Causality, Diagnosis, Repairs and View-Updates in Databases”
Example: File Access

Access\((u, f)\) :− UserGroup\((u, g)\), GroupFile\((g, f)\)

Delete source rows, s.t. \textcolor{blue}{Emma} won’t access \textcolor{blue}{a.txt}. But, maintain \textbf{maximum access permissions}!
Example: File Access

Access\((u,f)\) :– UserGroup\((u,g)\), GroupFile\((g,f)\)

Delete source rows, s.t. Emma won’t access a.txt. But, maintain maximum access permissions!
Example: File Access

[Cui&Widom01; Buneman&al.02]

<table>
<thead>
<tr>
<th>Access</th>
<th>UserGroup</th>
<th>GroupFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>file</td>
<td></td>
</tr>
<tr>
<td>Emma</td>
<td>a.txt</td>
<td></td>
</tr>
<tr>
<td>Emma</td>
<td>b.txt</td>
<td></td>
</tr>
<tr>
<td>Olivia</td>
<td>a.txt</td>
<td></td>
</tr>
<tr>
<td>Olivia</td>
<td>b.txt</td>
<td></td>
</tr>
<tr>
<td>Jacob</td>
<td>a.txt</td>
<td></td>
</tr>
<tr>
<td>Jacob</td>
<td>b.txt</td>
<td></td>
</tr>
</tbody>
</table>

Access(u,f) :- UserGroup(u,g), GroupFile(g,f)

Delete source rows, s.t. Emma won’t access a.txt.
But, maintain maximum access permissions!

Decision variant is NP-complete [Buneman et al. 02]
Trichotomy in Complexity

[K, Vondrak, Williams, Woodruff, PODS11, PODS12, TODS12, VLDB14]

Fix a schema (w/ fds) and a CQ w/o self joins

What is the complexity of finding a solution with a minimal side effect?

We have established a precise (easily testable) criterion that partition all cases into 3 categories:

1. The problem is solvable in PTime, and even via a straightforward algorithm [Buneman et al. 2001]

2. The problem is NP-hard, but constant-ratio approximable in PTime (ILP relaxation)

3. The problem is inapproximable for every ratio
Outline

• Text Analytics in the Big Data Era
• Information Extraction Systems & Formalism
• Foundational Research Challenges

Summary
Summary

- Text analytics & IE
- Rule systems for IE
- A formal framework for rules, relating IE to traditional DB concepts such as Datalog
- Research directions motivated by IE
  - Prioritized repairs
  - Graph mining
  - Update propagation