EDA 2015

Foundations of Database Systems for Text Analytics

Benny Kimelfeld

Faculty of Computer Science

Technion, Israel

Technion
Israel Institute of Technology

Outline

- Text Analytics in Modern Applications
- Information Extraction Systems & Formalism
- Foundational Research Challenges
- Conclusions and Outlook

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

e-Commerce

Comparison Shopping extracts & compares inventory from online sources

CRM / BI

Monitor customer's social-media activity for sentiment & business leads

Life-Science Mining

Extract knowledge bases from scientific publications

Log Analysis

Summarize, visualize and analyze logs produced by machines

Core Task: Information Extraction (IE)

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

*"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**."*

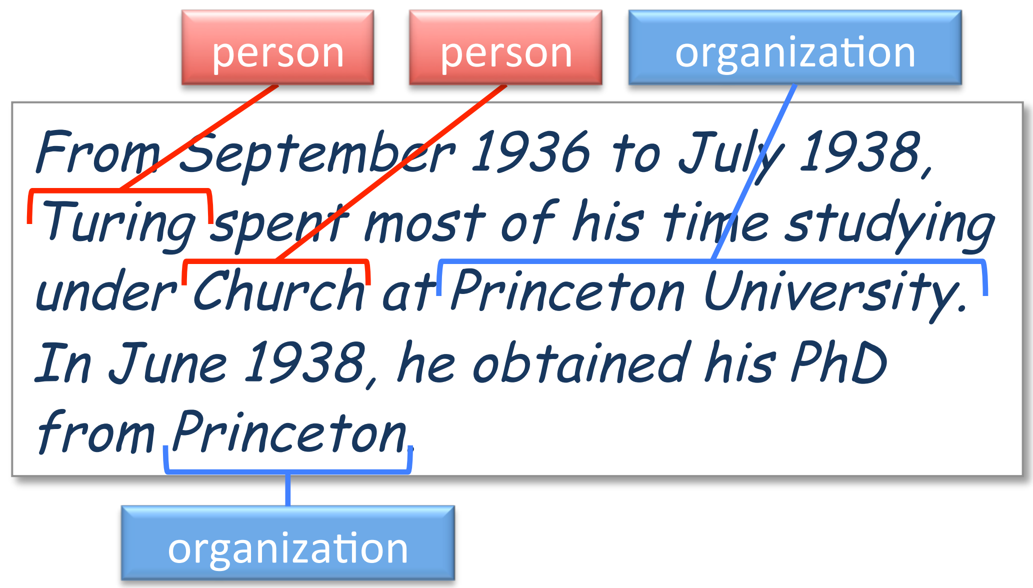
J. Cowie and Y. Wilks., *Handbook of Natural Language Processing*, 2000

*"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."*

M. F. Moens, Information Extraction: Algorithms and Prospects in a Retrieval Context, 2006

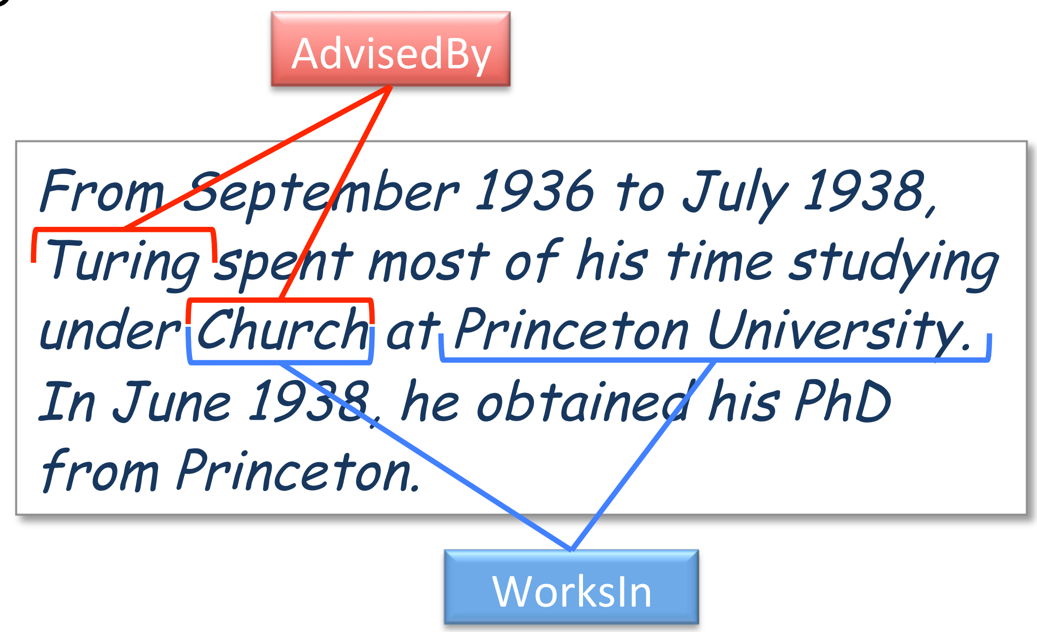
Popular Classes of IE Tasks

- Named Entity Recognition



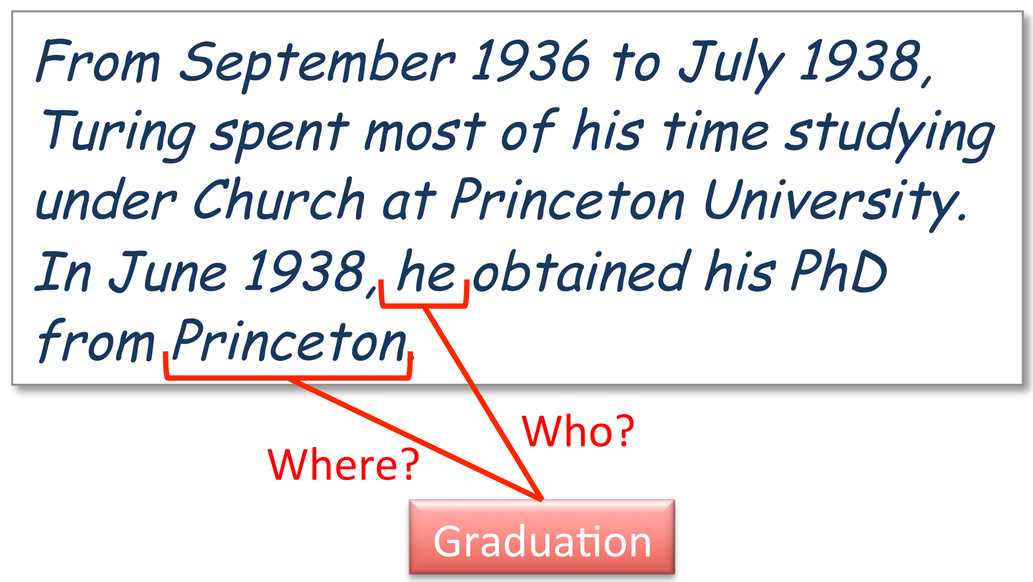
Popular Classes of IE Tasks

- Named Entity Recognition
- Relation Extraction



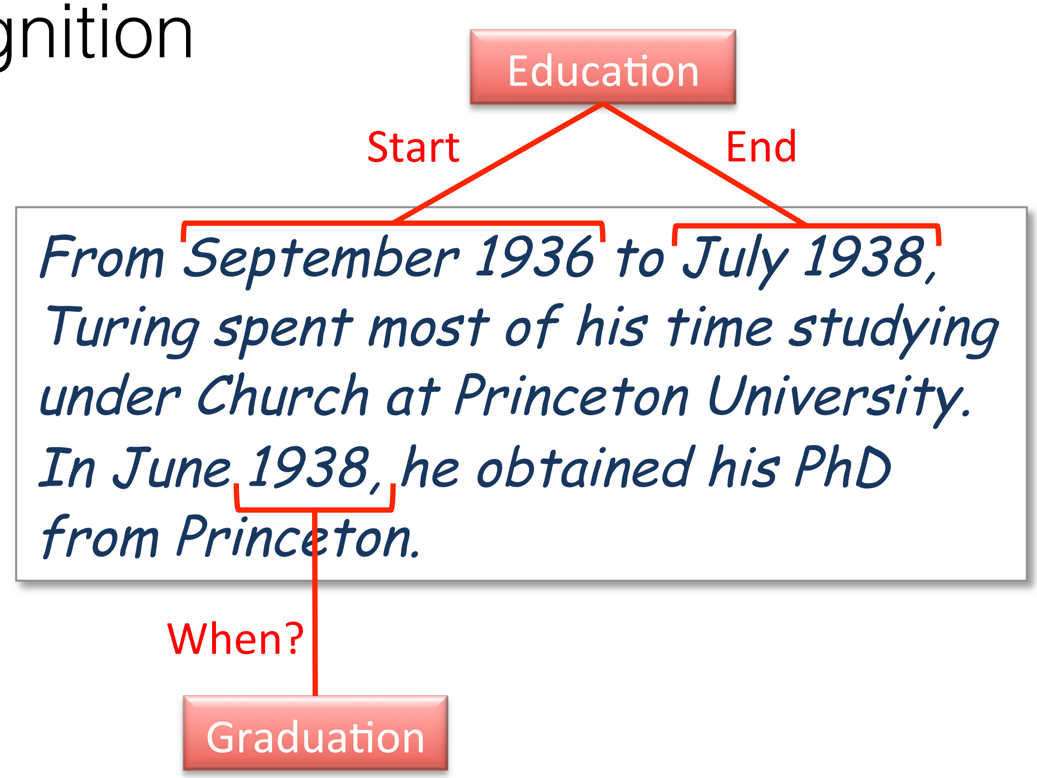
Popular Classes of IE Tasks

- Named Entity Recognition
- Relation Extraction
- Event Extraction



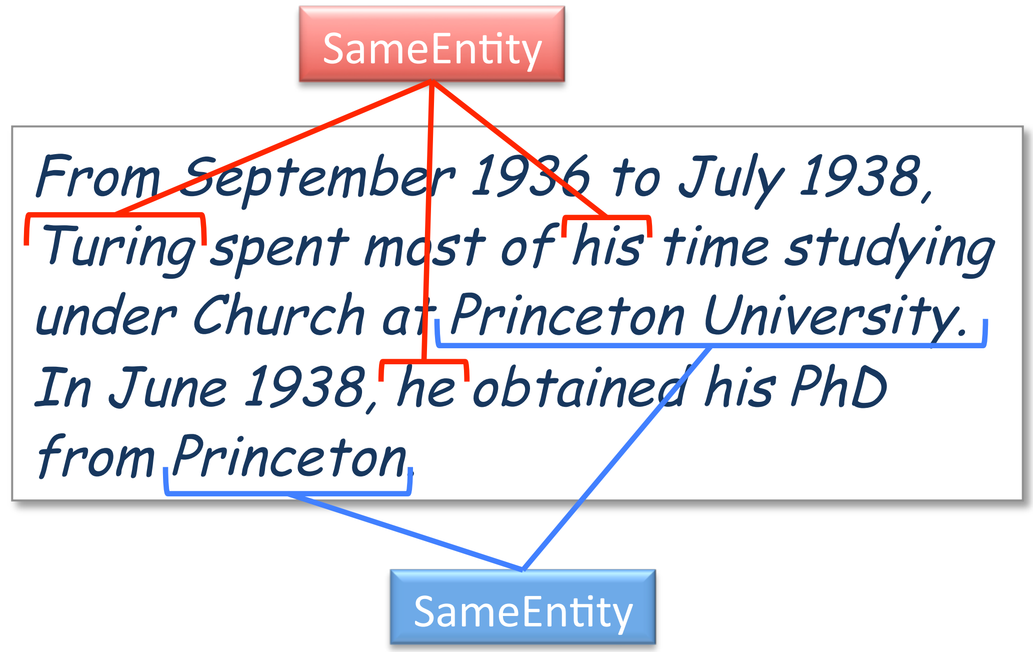
Popular Classes of IE Tasks

- 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



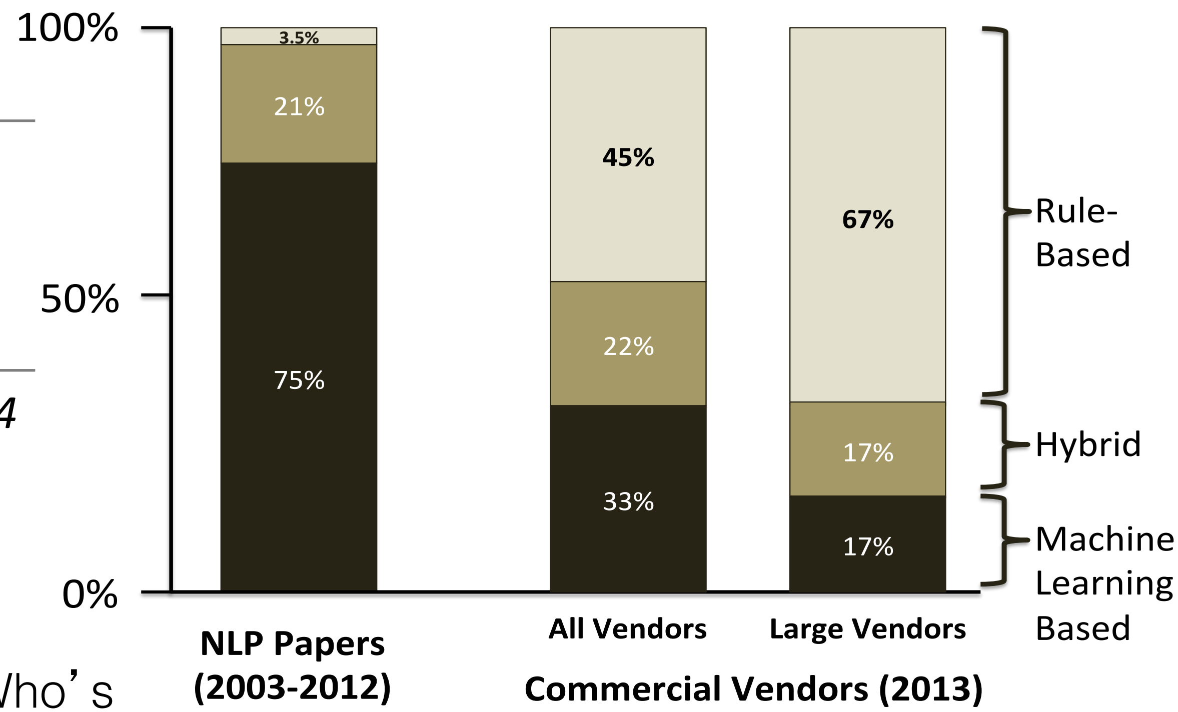
IE Paradigms: Rules & Statistics

+ NLP
- Rules
- 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

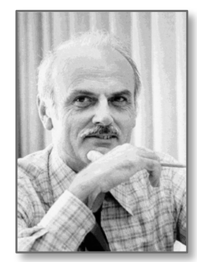
- EMNLP, ACL, NAACL, 2003-2012
- 54 industrial vendors (Who's Who in Text Analytics, 2012)



[Chiticariu, Li, Reiss, EMNLP'13]

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

Past:	*Present:*
Proprietary data in orgs. (enterprises, governments, …)	Proliferation of publically open data sources (Web, social, …)
Data structured/controlled by admins, e-forms, software, …	Uncontrolled data from humans' free text, heterogeneous kbs, …
Massive-data analyses incurred high machinery/personnel cost	Business models (cloud, crowd, opensource) facilitate analyses
Analyses by specialized teams of heavily trained experts	Analyses by a wide community featuring a wide range of skills

> "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."

"Big data: The next frontier for innovation, competition, and productivity"
McKinsey Report, May 2011

We need dev. & management systems to facilitate value extraction from *Big Data* by a wide range of users / skills

Outline

- Text Analytics in Modern Applications
- Information Extraction Systems & Formalism
- Foundational Research Challenges
- Conclusions and Outlook

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

Span [42,47)

```
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]

Regex. (string)

Unary regex formula

Binary regex formula

```
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

$$\mathtt{killed}(\mathrm{a},\mathrm{c}) \Leftarrow \boxed{\mathtt{next}(\mathrm{a},\mathrm{b})} \wedge \boxed{\mathtt{next}(\mathrm{b},\mathrm{c})} \wedge \boxed{\mathtt{token}(\mathrm{b},\text{‘}\mathtt{killed}\text{’})}$$
$$\wedge \boxed{\mathtt{capitalized}(\mathrm{a})} \wedge \boxed{\mathtt{capitalized}(\mathrm{b})}$$

Binary regex formulas

Unary regex formulas

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

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'15

Terminology

```
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.
```

Span [52,65)

Document

Company	CEO	CompanyCEO
[1,14) (Kaspersky Lab)	[19,36) (Eugene Kaspersky)	[1,36)
[42,47) (Intel)	[52,65) (Paul Otellini)	[42,65)

*Relation over **spans** from the document*

Document Spanners

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

More formally:

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

```
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 d

x	y	z
[1,14)	[30,36)	[1,36)
[42,47)	[52,65)	[42,65)
[102,110)	[115,125)	[102,125)

Relation over the spans of d

Spanners as Datalog w/ Regex

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

Rep. of Spanners

Token(x)	:=	[(ε | .*_) x{[a-zA-Z]$^+$} (((,V_) .*) | ε)]
State(x)	:=	Token(x) , [.* x{Georgia|Virginia|Washington}.*]
Cap1st(x)	:=	Token(x) , [.* x{[A-Z].*}.*]
CommaSp(x,y,z)	:=	[.* z{x{.*} ,_ y{.*}}.*]
Loc(z)	:=	CommaSp(x,y,z) , Cap1st(x) , State(y)
RETURN(x,z)	:=	Cap1st(x) , [.*x{.*}_from_z{.*}.*}] , Loc(z)

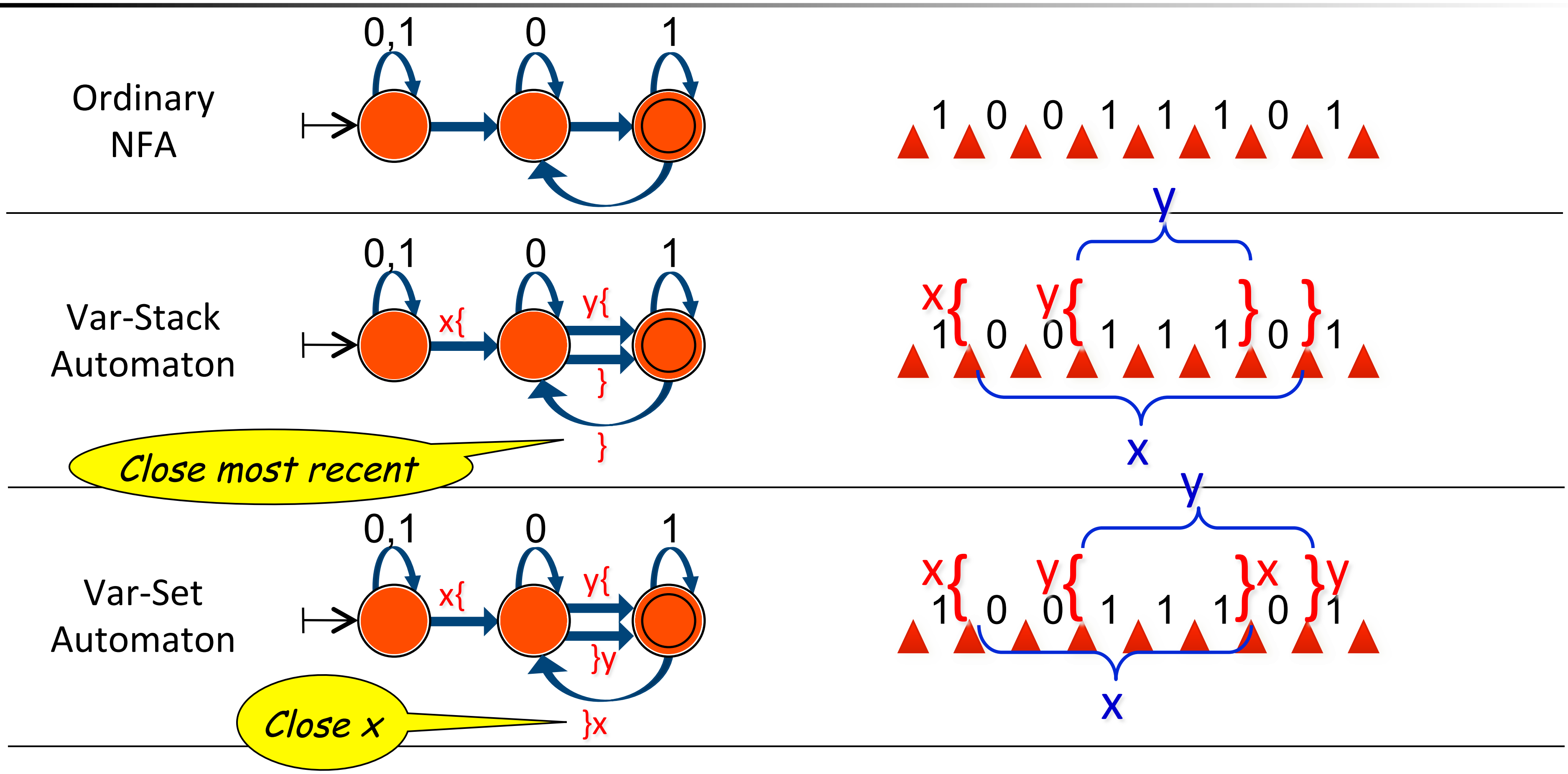
Rep. of Spanners

Query goal

Carter_from_Plains,_Georgia,_Washington _from_Westmoreland,_Virginia

x	z
[1,7) *Carter*	[13,28) *Plains,_Georgia*
[30,40) *Washington*	[46,69) *Westmoreland,_Virginia*

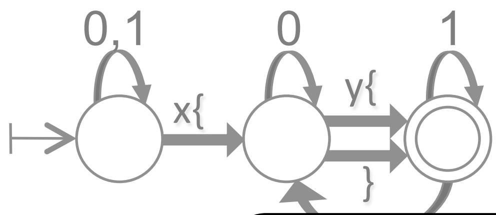
Spanners as Automata



- 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



.*y{x{.*}_from_z{.*}.*}

Spanners definable by var-stack automata = Spanners definable by regex formulas

Spanners definable by Datalog (NR) w/ regex formulas = Spanners definable by var-set automata = Spanners definable by Rel. Algebra over regex formulas

Token(x) := [(ε | .*_) x{[a-zA-Z]+} (((,V_) .*) | ε)]

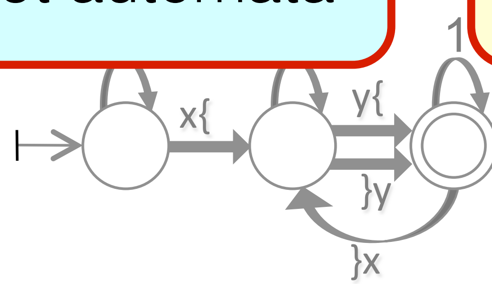
State(x) := Token(x) , [.* x{Georgia|Virginia|Washington}.*]

Cap1st(x) := Token(x) , [.* x{[A-Z].*}.*]

aSp(x,y,z) := [.* z{x{.*} ,_ y{.*}}.*]

Loc(z) := CommaSp(x,y,z) , Cap1st(x) , State(y)

TURN(x,z) := Cap1st(x) , [.*x{.*}_from_z{.*}.*}] , Loc(z)



Join ⋈

Union ∪

Product ×

Projection π

Selection ς

Difference -

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]
  - (Later in the talk …)

IBM SystemT: SQL for IE

```
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);
...
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;
```

Unary regex formulas

regex + join w/ previous views

union

projection

Cleaning

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

SystemT Research

- 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]

Outline

- Text Analytics in Modern Applications
- Information Extraction Systems & Formalism
- Foundational Research Challenges
- Conclusions and Outlook

Propelled Research

- Next, highlight 2 lines of foundational research motivated by text analytics:
  - Cleaning inconsistency w/ prioritized repairs
    - [Fagin, K, Reiss, Vansummeren 2014]
    - [Fagin, K, Kolaitis, PODS'15]
  - Frequent subgraph mining
    - [K, Kolaitis, PODS'13, TODS'14]
- Not covered:
  - Update propagation
    - [K+, VLDB'13, TODS'12, PODS'12, PODS'11]
  - Querying Markov sequences
    - [PODS'08, JACM'14]

Cleaning IE Inconsistencies

- Extractors may produce inconsistent results
  - Data artifacts
  - Developer limitations

Person1

Martin Luther King Jr.

Person2 Address1

- 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]

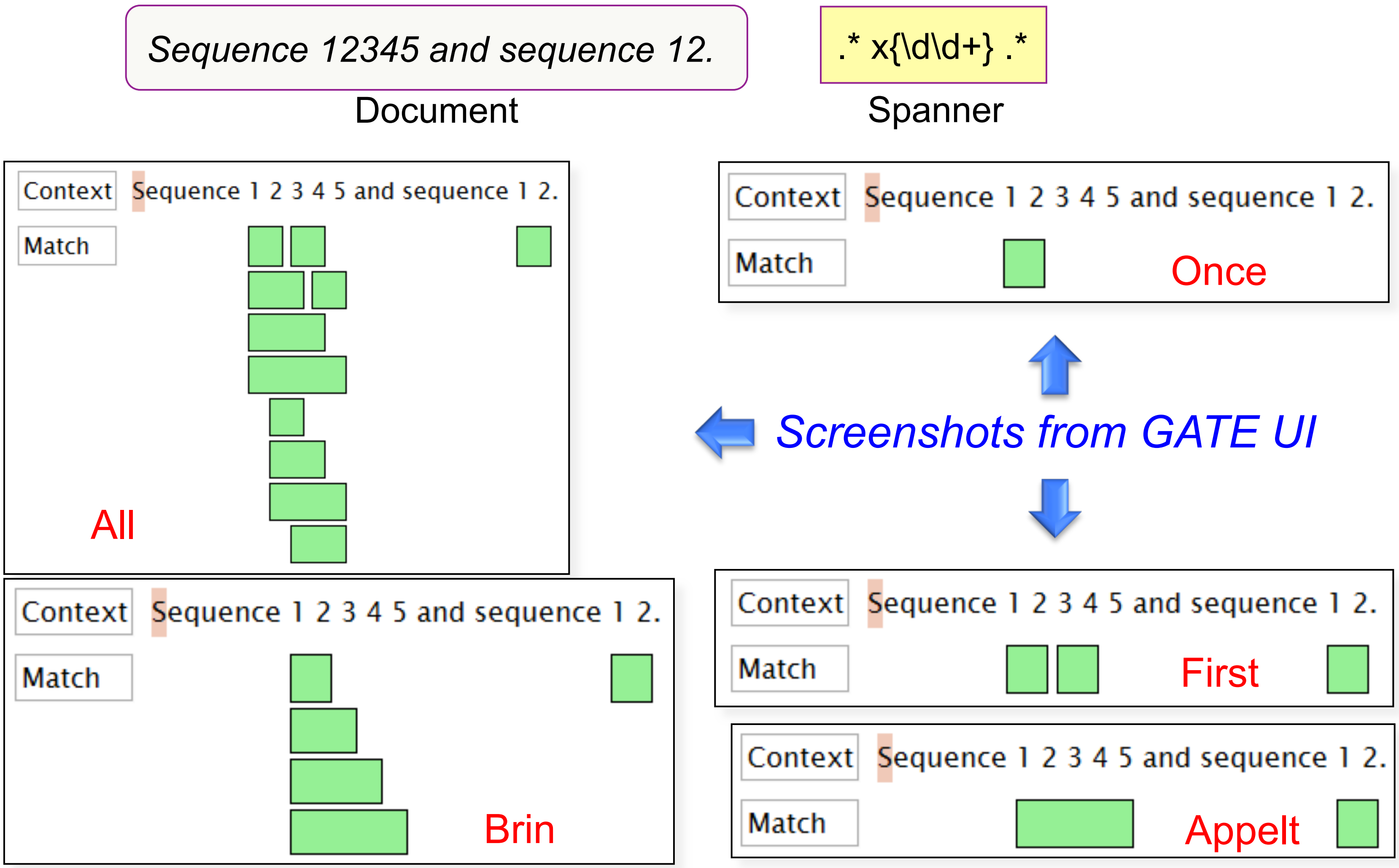
SystemT Consolidators

```
create view Caps as
extract regex /[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);

...

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;
```

Other policies built in

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

Five GATE/JAPE Controls



Declarative Cleaning

- Problem: existing policies are ad-hoc; how to expose a language for user declaration?
- [Fagin, K, Reiss, Vansummeren, PODS14]: spanner formalism for declarative cleaning
  - Captures SystemT, GATE, WHISK, POSIX, …
  - *Can state rules like:*

x and y are overlapping spans	$\rightarrow$	**not** [*Person(x) & Location(y)*]
x and y are separated by "and/or"	$\rightarrow$	**not** [*Person(x) & Location(y)*]
y strictly contains x	$\rightarrow$	*Prefer Person(y) to Person(x)*
true	$\rightarrow$	*Prefer Location(y) to Person(x)*

Prioritized Repairs: Definition

Database	Denial Constraints	Priority Relation
Collection of facts	Which sets of facts cannot co-exist?	Binary "is preferred to" relation

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:
  - Improve a consistent DB subsets by "profitable" exchanges of facts, again and again until impossible
  - A *preferred repair* is a subset that cannot be improved

Example

professor	university	city
Monica	ubiobio	Concepción
Monica	carleton	Ottawa
Jorge	uchile	Santiago
Jorge	ubiobio	Santiago
Pablo	uchile	Santiago

Violated constraints (*functional dependencies*):

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

"Ordinary" repairs

professor	university	city
Monica	ubiobio	Concepción
~~Monica~~	~~carleton~~	~~Ottawa~~
Jorge	uchile	Santiago
~~Jorge~~	~~ubiobio~~	~~Santiago~~
Pablo	uchile	Santiago

professor	university	city
~~Monica~~	~~ubiobio~~	~~Concepción~~
Monica	carleton	Ottawa
~~Jorge~~	~~uchile~~	~~Santiago~~
Jorge	ubiobio	Santiago
Pablo	uchile	Santiago

Tuple priority → some repairs can be discarded

Complexity of Testing Improvability

[Fagin, K, Kolaitis, PODS'15]

Can a consistent subset be improved?

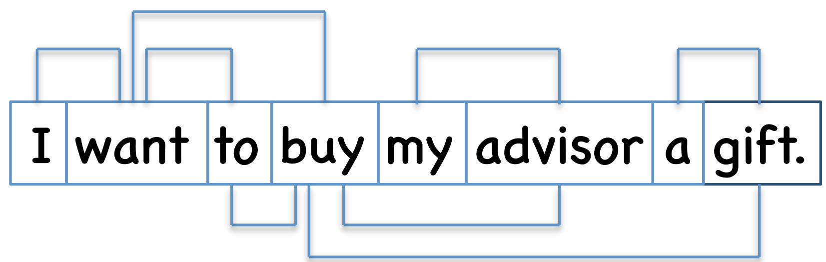
- In the case of a *single functional dependency* or *two keys* per relation, improvability can be tested in polynomial time
- In **any other combination of FDs**, the problem is NP-complete!

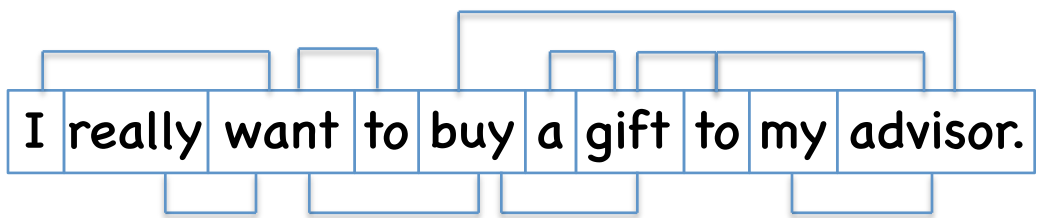
two keys

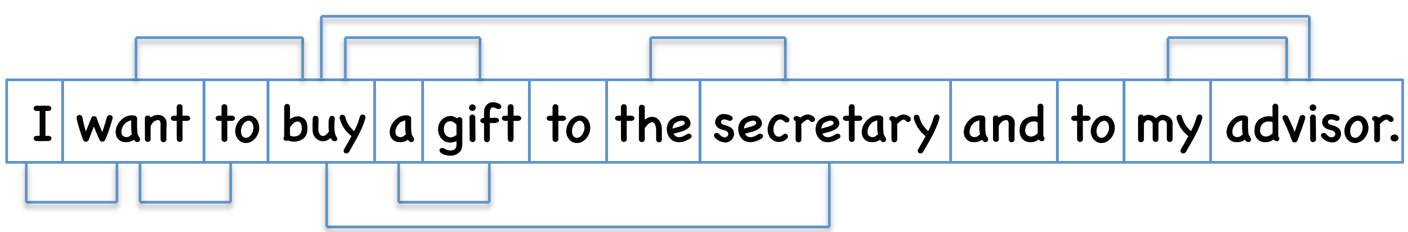
university	faculty	dean
UChile	Economics	Agosin
Technion	CS	Yavneh
Stanford	Law	Magill

IE with Recurring Patterns

1. Apply dependency parsing



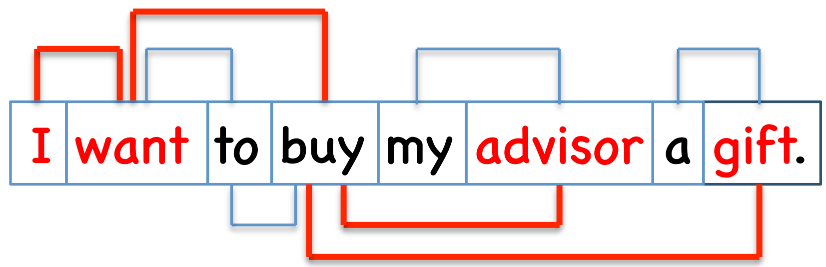


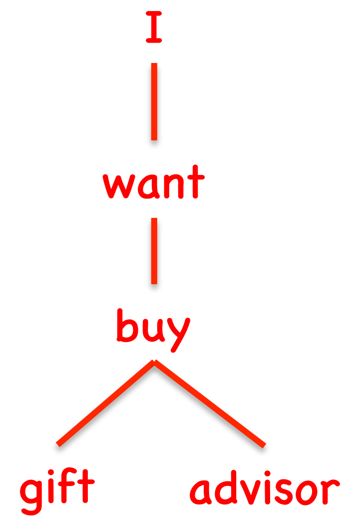


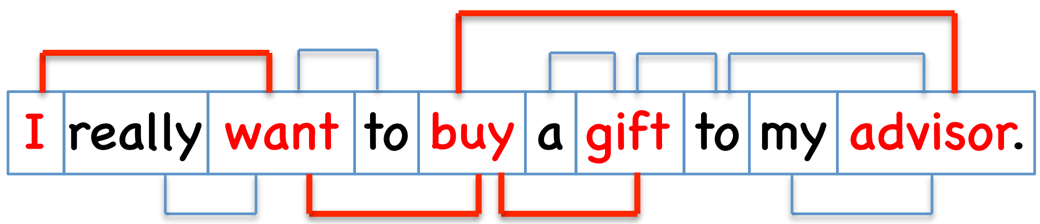
[Zhang, Baldwin, Ho, **K**, Li, ACL13]: Restoring grammar in social media, sms, etc.

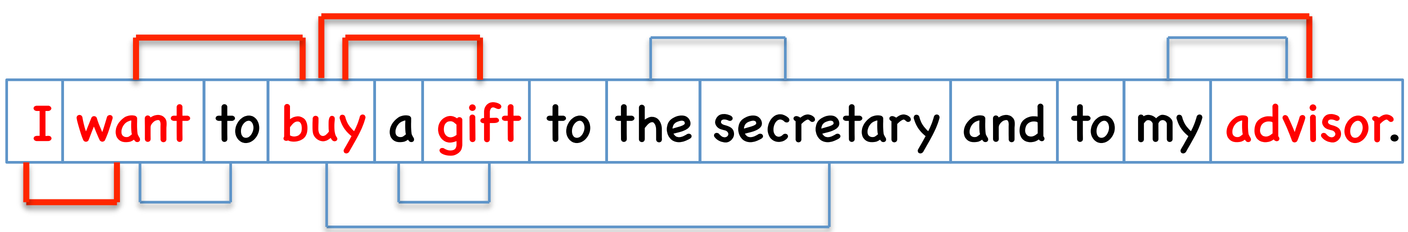
IE with Recurring Patterns

1. Apply dependency parsing
2. Find freq. recurring patterns



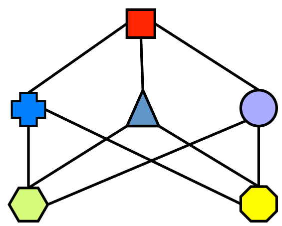
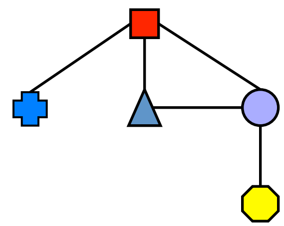
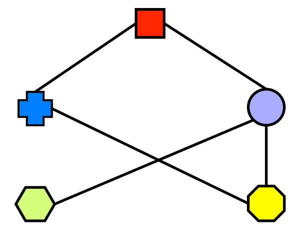
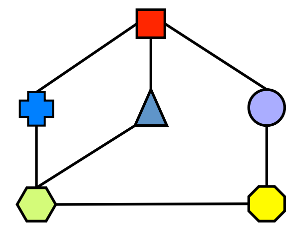






[Zhang, Baldwin, Ho, **K**, Li, ACL13]: Restoring grammar in social media, sms, etc.

Maximal Frequent Subgraphs

$\tau = 3$			
g_1	g_2	g_3	g_4
			

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
  – [Kolaitis, K, PODS'13, TODS'14]
- Solved open problems on graph-mining complexity
- Established a novel approach to graph mining, based on enumeration with hereditary properties
  – [Cohen, K, Sagiv, JCSS'08]

Outline

- Text Analytics in Modern Applications
- Information Extraction Systems & Formalism
- Foundational Research Challenges
- Conclusions and Outlook

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

Outlook: DB w/ Proper Text Support

- Structured + text data & query model
  - Elegant and useful marriage
  - Based on spanners
  - Gracefully incorporate generic NLP solvers
- Underspecification
  - Balance automation & control: from full specification by experts to feature generation for nonexperienced
  - ***Maximally realize the potential of every developer!***
- In-model uncertainty
  - Well-defined & intuitive probability model w/ practical execution cost for principled recall/precision control





Thank you!

***PS** looking for grads and postdocs to build next-generation DBs in Haifa…*