15.3 Knowledge Harvesting

Automatic construction of large knowledge bases about entities, classes, relations from Web sources (incl. Wikipedia) using pattern matching, statistical learning & consistency reasoning.
Web of Open Linked Data and Knowledge

> 50 Bio. subject-predicate-object triples from > 1000 sources

+ Web tables

[Image of a web of data and knowledge resources including DBpedia, Yago, Freebase, WordNet, SUMO, BabelNet, and other linked data sources.]

Knowlede Bases on the Web

> 50 Bio. subject-predicate-object triples from > 1000 sources

- 10M entities in 350K classes
- 180M facts for 100 relations
- 100 languages
- 95% accuracy

- 4M entities in 250 classes
- 500M facts for 6000 properties
- live updates

- 600M entities in 15000 topics
- 20B facts

- 40M entities in 15000 topics
- 1B facts for 4000 properties
- core of Google Knowledge Graph

Knowlede Bases on the Web

> 50 Bio. subject-predicate-object triples from > 1000 sources

**Taxonomic knowledge**

- Bob_Dylan type songwriter
- Bob_Dylan type civil_rights_activist
- songwriter subclassOf artist
- Bob_Dylan composed Hurricane
- Hurricane isAbout Rubin_Carter
- Bob_Dylan marriedTo Sara_Lownds
- validDuring [Sep-1965, June-1977]

**Factual knowledge**

- Bob_Dylan knownAs „voice of a generation“
- Steve_Jobs „was big fan of“ Bob_Dylan
- Bob_Dylan „briefly dated“ Joan_Baez

**Temporal knowledge**

- validDuring [Sep-1965, June-1977]

**Terminological knowledge**

- evidence & belief knowledge
Knowledge Base (aka. Knowledge Graph): a Pragmatic Definition

Comprehensive and semantically organized

**machine-readable** collection of
universally relevant or domain-specific
**entities, classes, and**
**SPO facts** (attributes, relations)

plus **spatial** and **temporal** dimensions
plus **commonsense** properties and rules
plus **contexts** of entities and facts
   (textual & visual witnesses, descriptors, statistics)
plus …..
Some Publicly Available Knowledge Bases

YAGO: yago-knowledge.org
Dbpedia: dbpedia.org
Freebase: freebase.com
Wikidata: www.wikidata.org
Entitycube: entitycube.research.microsoft.com
                      renlifang.msra.cn
NELL: rtw.ml.cmu.edu
DeepDive: deepdive.stanford.edu
Probase: research.microsoft.com/en-us/projects/probase/
KnowItAll / ReVerb: openie.cs.washington.edu
                      reverb.cs.washington.edu
BabelNet: babelnet.org
WikiNet: www.h-its.org/english/research/nlp/download/
ConceptNet: conceptnet5.media.mit.edu
WordNet: wordnet.princeton.edu
Linked Open Data: linkeddata.org
### Example: DBpedia

#### About: **Steve Jobs**
An Entity of Type: [agent](http://dbpedia.org/page/Steve_Jobs), from Named Graph: [http://dbpedia.org](http://dbpedia.org), within Data Space: [dbpedia.org](http://dbpedia.org)

Steven Paul Jobs (ˈdʒɒps/; February 24, 1955 – October 5, 2011) was an American entrepreneur, marketer, and inventor, who was the cofounder, chairman, and CEO of Apple Inc.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo:activeYearsEndYear</td>
<td>2011-01-01 [xsd: date]</td>
</tr>
<tr>
<td>dbo:activeYearsStartYear</td>
<td>1974-01-01 [xsd: date]</td>
</tr>
<tr>
<td>dbo:alias</td>
<td>Jobs, Steven Paul</td>
</tr>
<tr>
<td>dbo:almaMater</td>
<td>db:Reed_College</td>
</tr>
<tr>
<td>dbo:birthDate</td>
<td>1955-02-24 [xsd: date]</td>
</tr>
<tr>
<td>dbo:birthName</td>
<td>Steven Paul Jobs</td>
</tr>
<tr>
<td>dbo:birthPlace</td>
<td>db:California</td>
</tr>
<tr>
<td></td>
<td>db:San_Francisco</td>
</tr>
<tr>
<td>dbo:birthYear</td>
<td>1955-01-01 [xsd: date]</td>
</tr>
<tr>
<td>dbo:board</td>
<td>db:Apple_Inc.</td>
</tr>
<tr>
<td></td>
<td>db:The_Walt_Disney_Company</td>
</tr>
<tr>
<td>dbo:child</td>
<td>db:Lisa_Brennan-Jobs</td>
</tr>
<tr>
<td>dbo:deathDate</td>
<td>2011-10-05 [xsd: date]</td>
</tr>
<tr>
<td>dbo:deathPlace</td>
<td>db:California</td>
</tr>
<tr>
<td>dbo:deathYear</td>
<td>2011-01-01 [xsd: date]</td>
</tr>
<tr>
<td>dbo:networth</td>
<td>8.3E9</td>
</tr>
<tr>
<td>dbo:occupation</td>
<td>db:Pikar</td>
</tr>
<tr>
<td></td>
<td>db:Apple_Inc.</td>
</tr>
<tr>
<td></td>
<td>db:NeXT</td>
</tr>
<tr>
<td></td>
<td>db:Steve_Jobs__1</td>
</tr>
<tr>
<td></td>
<td>db:Steve_Jobs__2</td>
</tr>
<tr>
<td></td>
<td>db:Steve_Jobs__3</td>
</tr>
<tr>
<td></td>
<td>db:Steve_Jobs__4</td>
</tr>
<tr>
<td></td>
<td>db:Steve_Jobs__5</td>
</tr>
<tr>
<td></td>
<td>db:Steve_Jobs__6</td>
</tr>
<tr>
<td>dbo:partner</td>
<td>db:Chrisann_Erennan</td>
</tr>
<tr>
<td>dbo:relative</td>
<td>db:Mona_Simpson</td>
</tr>
<tr>
<td>dbo:religion</td>
<td>db:Lutheranism</td>
</tr>
<tr>
<td></td>
<td>db:Zen</td>
</tr>
<tr>
<td>dbo:residence</td>
<td>db:California</td>
</tr>
</tbody>
</table>

[http://dbpedia.org/page/Steve_Jobs](http://dbpedia.org/page/Steve_Jobs)
## Example: Wikidata

### David Bowie (Q5383)

<table>
<thead>
<tr>
<th>Occupation</th>
<th>Editor Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>painter</td>
<td>edit</td>
</tr>
<tr>
<td>singer-songwriter</td>
<td>edit</td>
</tr>
<tr>
<td>guitarist</td>
<td>edit</td>
</tr>
<tr>
<td>saxophonist</td>
<td>edit</td>
</tr>
<tr>
<td>composer</td>
<td>edit</td>
</tr>
<tr>
<td>film actor</td>
<td>edit</td>
</tr>
</tbody>
</table>
**Example: NELL**

50 Mio. SPO assertions, 2.5 Mio high confidence

Recently-Learned Facts

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>bresaola is a visualizable thing</td>
<td>922</td>
<td>05-may-2015</td>
<td>96.4</td>
</tr>
<tr>
<td>francis_denwent_wood is a visual artist</td>
<td>922</td>
<td>05-may-2015</td>
<td>99.9</td>
</tr>
<tr>
<td>frank_g is an Australian person</td>
<td>922</td>
<td>05-may-2015</td>
<td>92.2</td>
</tr>
<tr>
<td>g_protein_coupled_receptor_124 is a protein</td>
<td>922</td>
<td>05-may-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>n_butyl_benzyl_phthalate is a chemical</td>
<td>922</td>
<td>05-may-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>chicken001 eat potatoes</td>
<td>926</td>
<td>20-may-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>bioinformatics is an academic program at the university college</td>
<td>922</td>
<td>05-may-2015</td>
<td>93.8</td>
</tr>
<tr>
<td>samuel_j_palmisano is the CEO of ibm</td>
<td>926</td>
<td>20-may-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>national001 is a company that has an office in the country czech_republic</td>
<td>922</td>
<td>05-may-2015</td>
<td>99.2</td>
</tr>
<tr>
<td>the companies dc and fox_news_channel compete with eachother</td>
<td>922</td>
<td>05-may-2015</td>
<td>98.4</td>
</tr>
</tbody>
</table>

http://rtw.ml.cmu.edu/rtw/kbbrowser/
Example: NELL
50 Mio. SPO assertions, 2.5 Mio high confidence

nick_cave (musician)
literal strings: NICK CAVE, nick cave, Nick cave, Nick Cave

Help NELL Learn!
NELL wants to know if this belief is correct.
If it is or ever was, click thumbs-up. Otherwise, click thumbs-down.

- nick_cave is a musician

categories
- musician (98.7%)
  - MBL @865 (98.9%) on 25-aug-2014 [ Promotion of celebrity:nick_cave musicaninmusicartist
musicartist:bad_seeds ]
  - SEAL @623 (57.5%) on 10-aug-2012 [ 1 ] using nick_cave

NELL has only weak evidence for items listed in grey
- visualartist
  - SEAL @221 (50.0%) on 18-mar-2011 [ 1 ] using nick_cave
- personaustralia
  - SEAL @628 (65.7%) on 26-aug-2012 [ 1 ] using nick_cave
- celebrity
  - SEAL @347 (75.0%) on 13-jul-2011 [ 1 2 ] using nick_cave

relations
NELL has only weak evidence for items listed in grey
- agentcollaborateswitha
  - http://rtw.ml.cmu.edu/rtw/kbbrowser/
**Example: NELL**

50 Mio. SPO assertions, 2.5 Mio high confidence

### NELL Knowledge Base Browser

- touristattractions such as touristattraction
- celebrities such as celebrity
- archaeasuchasarchae
- agriculturalproducts including agriculture
- plantincludeplant
- actorsuchasactor
- arachnids such as arachnids
- mammals such as mammal
- architectsuchasarchitects
- companyeconomicsector
- trophywonbycoaches
- agentinvolvedwithitem
- wineryproduce such as wine
- agentworkedondrug
- productsuchasproducttype
- productsuchasproduct
- automakerproduce such as model
- mlauthorsofsoftware
- musicianplaysinstrument
- animaldevelopedisease
- countrylanguage
- issueofpoliticsbill
- universityoperateinlanguage
- iteminvolvedwithagent
- drugworkedonbyagent
- wineproducedbywinery
- msoftwrauthor
- produceby
- automodelproducedbymaker
- typeproducedby
- instrumentplayedbymusician
- doateof
- dateatwhichexistsitem
- dateevent
- dateofsportsgame
- dateofmeetingeventtitle

### musicianplaysinstrument

*(relation: domain musician, range musicinstrument)*

Specifies that a musical instrument is played by a particular musician

See metadata for musicianplaysinstrument

712 instances, 1 page

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>adam, drums</td>
<td>799</td>
<td>27-dec-2013</td>
<td>100.0</td>
</tr>
<tr>
<td>adam, guitar</td>
<td>799</td>
<td>27-dec-2013</td>
<td>100.0</td>
</tr>
<tr>
<td>bach, piano</td>
<td>551</td>
<td>19-apr-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>bach, violin</td>
<td>551</td>
<td>19-apr-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>barbar, violin</td>
<td>598</td>
<td>21-jun-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>bb_king, guitar</td>
<td>680</td>
<td>09-jan-2013</td>
<td>100.0</td>
</tr>
<tr>
<td>beethoven, piano</td>
<td>853</td>
<td>11-jul-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>beethoven, violin</td>
<td>853</td>
<td>11-jul-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>ben_harper, guitar</td>
<td>820</td>
<td>08-mar-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>billie_joe_armstrong, guitar</td>
<td>818</td>
<td>03-mar-2014</td>
<td>100.0</td>
</tr>
<tr>
<td>brahms, piano</td>
<td>592</td>
<td>13-jun-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>brahms, violin</td>
<td>503</td>
<td>06-feb-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>buddy_guy, guitar</td>
<td>684</td>
<td>01-dec-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>b_b_king, guitar</td>
<td>406</td>
<td>08-sep-2011</td>
<td>(Seed) 100.0</td>
</tr>
<tr>
<td>charlie, guitar</td>
<td>724</td>
<td>12-apr-2013</td>
<td>100.0</td>
</tr>
<tr>
<td>chopin, piano</td>
<td>683</td>
<td>15-jan-2013</td>
<td>100.0</td>
</tr>
<tr>
<td>copland, piano</td>
<td>665</td>
<td>05-dec-2012</td>
<td>100.0</td>
</tr>
<tr>
<td>david, bass</td>
<td>904</td>
<td>20-feb-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>david, drums</td>
<td>904</td>
<td>20-feb-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>david, guitar</td>
<td>904</td>
<td>20-feb-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>david, keyboards</td>
<td>904</td>
<td>20-feb-2015</td>
<td>100.0</td>
</tr>
<tr>
<td>earl_scruggs, banjo</td>
<td>904</td>
<td>20-feb-2015</td>
<td>100.0</td>
</tr>
</tbody>
</table>

[http://rtw.ml.cmu.edu/rtw/kbbrowser/](http://rtw.ml.cmu.edu/rtw/kbbrowser/)
Knowledge for Intelligent Applications

Enabling technology for:

- **disambiguation**
  in written & spoken natural language
- **deep reasoning**
  (e.g. QA to win quiz game)
- **machine reading**
  (e.g. to summarize book or corpus)
- **semantic search**
  in terms of entities&relations (not keywords&pages)
- **entity-level linkage**
  for Big Data & Deep Text analytics
15.3.1 Harvesting Unary Predicates with Patterns

Which **entity types** *(classes, unary predicates)* are there?

- scientists, doctoral students, computer scientists, …
- female humans, male humans, married humans, …

Which **subsumptions** should hold
(subclass/superclass, hyponym/hypernym, inclusion dependencies)?

- subclassOf (computer scientists, scientists)
- subclassOf (physicists, scientists),
- subclassOf (scientists, humans), …

Which **individual entities** belong to which classes?

- InstanceOf (Jim Gray computer scientists),
- InstanceOf (Barbara Liskov, computer scientists),
- InstanceOf (Barbara Liskov, female humans),
- InstanceOf (Steve Jobs, male humans),
- InstanceOf (Steve Jobs, entrepreneurs), … …
Hearst Patterns

Goal: find **instances of classes** (and/or: find subclasses of classes)

Hearst specified **lexico-syntactic patterns** for type relationship:
- X such as Y; X like Y;
- X and other Y; X including Y;
- X, especially Y;

Find such patterns in text: //better with POS tagging
- companies such as Apple
- Google, Microsoft and other companies
- Internet companies like Amazon and Facebook
- Chinese cities including Kunming and Shangri-La
- computer pioneers like the late Steve Jobs
- computer pioneers and other scientists
- lakes including the surrounding Hangzhou hills

Derive type(Y,X)
- type(Apple, company), type(Google, company), ...
- or as unary predicates: company(Apple), …
Doubly-anchored patterns

[Kozareva/Hovy 2010, Dalvi et al. 2012]

Goal:
find instances of classes

Start with a set of seeds:
companies = {Microsoft, Google}

Parse Web documents and find the pattern
W, Y and Z

If two of three placeholders match seeds, harvest the third:

Google, Microsoft and Amazon  \[\rightarrow\]  type(Amazon, company)
Cherry, Apple, and Banana  \[\rightarrow\]  --- (no output)
Set Completion from Tables

[Kozareva/Hovy 2010, Dalvi et al. 2012]

Goal: find instances of classes

Start with a set of seeds:

\[
\text{cities} = \{\text{Paris, Shanghai, Brisbane}\}
\]

Parse Web documents and find tables

If at least two seeds appear in a column, harvest the others:

\[
\begin{align*}
\text{type}(\text{Berlin, city}) \\
\text{type}(\text{London, city})
\end{align*}
\]
Set Completion Example 1

Automatically create sets of items from a few examples.

Enter a few items from a set of things. (example)
Next, press Large Set or Small Set and we'll try to predict other items in the set.

- Stanford
- Princeton
- Penn State
- 
- 
- 
### Set Completion Example 1

<table>
<thead>
<tr>
<th>Predicted Items</th>
<th>georgetown</th>
</tr>
</thead>
<tbody>
<tr>
<td>penn state</td>
<td>michigan</td>
</tr>
<tr>
<td>stanford</td>
<td>arizona</td>
</tr>
<tr>
<td>princeton</td>
<td>washington</td>
</tr>
<tr>
<td>ucla</td>
<td>dartmouth</td>
</tr>
<tr>
<td>harvard</td>
<td>oregon</td>
</tr>
<tr>
<td>mit</td>
<td>nyu</td>
</tr>
<tr>
<td>usc</td>
<td>california</td>
</tr>
<tr>
<td>yale</td>
<td>brown</td>
</tr>
<tr>
<td>columbia</td>
<td>chicago</td>
</tr>
<tr>
<td>cornell</td>
<td>northwestern</td>
</tr>
<tr>
<td>berkeley</td>
<td>caltech</td>
</tr>
<tr>
<td>duke</td>
<td>virginia</td>
</tr>
<tr>
<td></td>
<td>penn</td>
</tr>
</tbody>
</table>
Set Completion Example 2

Discuss  Terms of Use

Automatically create sets of items from a few examples.

Enter a few items from a set of things. (example)
Next, press Large Set or Small Set and we'll try to predict other items in the set.

- Pushkin
- Tolstoy
- Pasternak

(clear all)

Large Set  Small Set (15 items or fewer)

Examples:
green, purple, red  chicken dance, macarena, ymca  alexander, gladiator, troy  hilary duff, kelly clarkson  more...

http://labs.google.com/sets

IRDM  WS 2015 1-60
### Predicted Items

<table>
<thead>
<tr>
<th>john steinbeck</th>
</tr>
</thead>
<tbody>
<tr>
<td>russia literature</td>
</tr>
<tr>
<td>lermontov</td>
</tr>
<tr>
<td>stephen king</td>
</tr>
<tr>
<td>cs lewis</td>
</tr>
<tr>
<td>madame bovary</td>
</tr>
<tr>
<td>bible</td>
</tr>
<tr>
<td>the idiot</td>
</tr>
<tr>
<td>mark twain</td>
</tr>
<tr>
<td>mikhail bulgakov</td>
</tr>
<tr>
<td>fyodor dostoevsky</td>
</tr>
<tr>
<td>nikola gogol</td>
</tr>
<tr>
<td>susanna tamaro</td>
</tr>
<tr>
<td>edward said</td>
</tr>
<tr>
<td>dirty dancing</td>
</tr>
<tr>
<td>albert camus</td>
</tr>
<tr>
<td>shakespeare</td>
</tr>
<tr>
<td>romance novel</td>
</tr>
<tr>
<td>jack london</td>
</tr>
<tr>
<td>george orwell</td>
</tr>
<tr>
<td>fiction</td>
</tr>
<tr>
<td>authors</td>
</tr>
</tbody>
</table>

| tolstoy               |
| pushkin              |
| leo tolstoy          |
| anna karenina        |
| gogol                |
| drama                |
| danielle steel       |
| dostoevsky           |
| maxim gorky          |
| russia               |
| fyodor dostoevsky    |
| anton chekhov        |
| ivan turgenev        |
| paulo coelho         |
| dan brown            |
| ernest hemingway     |
| dostojevski          |
| alexander pushkin    |
Extracting instances from lists & tables

State-of-the-Art Approach (e.g. SEAL):
• Start with seeds: a few class instances
• Find lists, tables, text snippets (“for example: …“), … that contain one or more seeds
• Extract candidates: noun phrases from vicinity
• Gather co-occurrence stats (seed&cand, cand&className pairs)
• Rank candidates
  • point-wise mutual information, …
  • random walk (PR-style)
  on seed-cand graph

Caveats:
Precision drops for classes with sparse statistics
Harvested items are names, not entities
Canonicalization (de-duplication) unsolved

\[
\text{PMI (x,y) = } \log \frac{P(x,y)}{P(x)P(y)}
\]
15.3.2 Harvesting Binary Predicates
with Seeds and Constraints

Which **instances** (pairs of individual entities) are there for given **binary relations** with specific **type signatures**?

- hasAdvisor (JimGray, MikeHarrison)
- graduatedAt (JimGray, Berkeley)
- graduatedAt (Chris Manning, Stanford)
- hasWonPrize (JimGray, TuringAward)
- hasWonPrize (VintCerf, TuringAward)
- bornOn (JohnLennon, 9-Oct-1940)
- diedOn (JohnLennon, 8-Dec-1980)
- marriedTo (JohnLennon, YokoOno)

Which additional & interesting **relation types** are there between given classes of entities?

- attendedSchool(x,y), competedWith(x,y), nominatedForPrize(x,y), …
- divorcedFrom(x,y), affairWith(x,y), …
- assassinated(x,y), rescued(x,y), admired(x,y), …
<table>
<thead>
<tr>
<th>Composed</th>
<th>Appeared In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Dylan wrote the song <em>Knockin’ on Heaven’s Door</em></td>
<td>knockedIn (<em>Knockin’ on Heaven’s Door</em>, <em>Billy the Kid</em>)</td>
</tr>
<tr>
<td>Lisa Gerrard wrote many haunting pieces, including <em>Now You Are Free</em></td>
<td>appearedIn (<em>Now You Are Free</em>, <em>Gladiator</em>)</td>
</tr>
<tr>
<td>Morricone’s masterpieces include the <em>Ecstasy of Gold</em></td>
<td></td>
</tr>
<tr>
<td>Dylan’s song <em>Hurricane</em> was covered by Ani DiFranco</td>
<td></td>
</tr>
<tr>
<td>Strauss’s famous work was used in 2001, titled <em>Also sprach Zarathustra</em></td>
<td></td>
</tr>
<tr>
<td>Frank Zappa performed a jazz version of Rota’s <em>Godfather Waltz</em></td>
<td></td>
</tr>
<tr>
<td>Hallelujah, originally by Cohen, was covered in many movies, including <em>Shrek</em></td>
<td></td>
</tr>
</tbody>
</table>

**Pattern-based Gathering** (statistical evidence) + **Constraint-aware Reasoning** (logical consistency)
Pattern-based Harvesting: Fact-Pattern Duality

Task populate relation *composed* starting with *seed facts*

---

**Facts & Fact Candidates**

(Dylan, Knockin)

(Gerrard, Now)

(Dylan, Hurricane)

(Morricone, Ecstasy)

(Zappa, Godfather)

(Mann, Buddenbrooks)

(Gabriel, Biko)

(Puebla, Che Guevara)

(Mezrich, Zuckerberg)

(Jobs, Apple)

(Newton, Gravity)

---

**Patterns**

X wrote the song Y

X wrote … including Y

X covered the story of Y

X has favorite movie Y

X is famous for Y

---

• good for recall
• noisy, drifting
• not robust enough for high precision

Improving Pattern Precision or Recall

• Statistics for confidence:
  occurrence frequency with seed pairs
  distinct number of pairs seen

• Negative seeds for confusabale relations:
  capitalOf(city,country) → X is the largest city of Y
  **pos. seeds:** (Paris, France), (Rome, Italy), (New Delhi, India), ...
  **neg. seeds:** (Sydney, Australia), (Istanbul, Turkey), ...

• Generalized patterns with wildcards and POS tags:
  hasAdvisor(student,prof) → X met his celebrated advisor Y
  → X * PRP ADJ advisor Y

• Dependency parsing for complex sentences:
Statistics for Pattern Quality Assessment

**Support of pattern p:**

\[
\frac{\# \text{ occurrences of } p \text{ with seeds } (e1,e2)}{\# \text{ occurrences of all patterns with seeds}}
\]

**Confidence of pattern p:**

\[
\frac{\# \text{ occurrences of } p \text{ with seeds } (e1,e2)}{\# \text{ occurrences of } p}
\]

**Confidence of fact candidate \((e1,e2):\)**

\[
\sum_p \text{freq}(e1,p,e2) \times \text{conf}(p) / \sum_p \text{freq}(e1,p,e2)
\]

or: \(\text{PMI}(e1,e2) = \log \frac{\text{freq}(e1,e2)}{\text{freq}(e1) \times \text{freq}(e2)}\)

- gathering can be iterated,
- can promote best facts to additional seeds for next round
Negative Seeds for Improved Precision
(Ravichandran 2002; Suchanek 2006; ...)

Problem: Some patterns have high support, but poor precision:

- X is the largest city of Y for isCapitalOf (X,Y)
- joint work of X and Y for hasAdvisor (X,Y)

Idea: Use positive and negative seeds:

- **pos. seeds:** (Paris, France), (Rome, Italy), (New Delhi, India), ...
- **neg. seeds:** (Sydney, Australia), (Istanbul, Turkey), ...

Compute the confidence of a pattern as:

\[
\frac{\text{# occurrences of } p \text{ with pos. seeds}}{\text{# occurrences of } p \text{ with pos. seeds or neg. seeds}}
\]

- can promote best facts to additional seeds for next round
- can promote rejected facts to additional counter-seeds
- works more robustly with few seeds & counter-seeds
Problem: Some patterns are too narrow and thus have small recall:

X and his celebrated advisor Y
X carried out his doctoral research in math under the supervision of Y
X received his PhD degree in the CS dept at Y
X obtained his PhD degree in math at Y

Idea: generalize patterns to n-grams, allow POS tags

X \{ his doctoral research, under the supervision of \} Y
X \{ PRP ADJ advisor \} Y
X \{ PRP doctoral research, IN DET supervision of \} Y

Compute match quality of pattern \( p \) with sentence \( q \) by Jaccard:

\[
\frac{|\{n\text{-grams} \in p\} \cap \{n\text{-grams} \in q\}|}{|\{n\text{-grams} \in p\} \cup \{n\text{-grams} \in q\}|}
\]

=> Covers more sentences, increases recall
Constrained Reasoning for Logical Consistency

Use knowledge (consistency constraints) for joint reasoning on hypotheses and pruning of false candidates.

### Hypotheses:
- composed (Dylan, Hurricane)
- composed (Morricone, Ecstasy)
- composed (Zappa, Godfather)
- composed (Rota, Godfather)
- composed (Gabriel, Biko)
- composed (Mann, Buddenbrooks)
- composed (Jobs, Apple)
- composed (Newton, Gravity)

### Constraints:
- \( \forall x, y: \text{composed}(x,y) \implies \text{type}(x)=\text{musician} \)
- \( \forall x, y: \text{composed}(x,y) \implies \text{type}(y)=\text{song} \)
- \( \forall x, y, z: \text{composed}(x,y) \land \text{appearedIn}(y,z) \implies \text{wroteSoundtrackFor}(x,z) \)
- \( \forall x, y, t, b, e: \text{composed}(x,y) \land \text{composedInYear}(y,t) \land \text{bornInYear}(x,b) \land \text{diedInYear}(x,e) \implies b < t \leq e \)
- \( \forall x, y, w: \text{composed}(x,y) \land \text{composed}(w,y) \implies x = w \)
- \( \forall x, y: \text{sings}(x,y) \land \text{type}(x,\text{singer-songwriter}) \implies \text{composed}(x,y) \)

Consistent subset(s) of hypotheses ("possible world(s)“, "truth“) → Weighted MaxSat solver for set of logical clauses → max a posteriori (MAP) for probabilistic factor graph
**Grounding** of formulas produces **clauses** (propositional logic: disjunctions of positive or negative literals) connecting patterns, facts, hypotheses, constraints

Ex.: composed(Gabriel,Biko); \( \neg \) composed(Gabriel,Biko) \lor \text{type}(Gabriel,musician);

composed(Mandela,Biko); \( \neg \) composed(Mandela,Biko) \lor \text{type}(Mandela,musician);

\( \neg \) composed(Gabriel,Biko) \lor \neg \text{appearedIn}(Biko,CryForFreedom) \lor \text{wroteSoundtrack}(Gabriel,CryForFreedom);

\( \neg \) composed(Gabriel,Biko) \lor \neg \neg \text{composed}(Mandela,Biko) \lor \text{False}; \ldots

**Treat hypotheses** (literals) **as variables**, facts as constants:

\( A; \neg A \lor B; C; \neg C \lor D; \neg A \lor \neg E \lor F; \neg A \lor \neg C; \ldots \)

**Clauses are weighted by pattern statistics and rule confidence**

**Solve weighted Max-Sat problem:**

assign truth values to variables s.t.

total weight of satisfied clauses is max!

\( \rightarrow \) NP-hard, but good approximation algorithms
Markov Logic Networks (MLN’s)

Map logical constraints & fact candidates into probabilistic graph model: Markov Random Field (MRF)

\[
\text{spouse}(x,y) \land \text{diff}(y,z) \Rightarrow \neg \text{spouse}(x,z)
\]
\[
\text{spouse}(x,y) \land \text{diff}(w,y) \Rightarrow \neg \text{spouse}(w,y)
\]
\[
\text{spouse}(x,y) \Rightarrow \text{female}(x)
\]
\[
\text{spouse}(x,y) \Rightarrow \text{male}(y)
\]

Variety of algorithms for joint inference:
- Gibbs sampling, other MCMC, belief propagation, …
- MAP inference equivalent to Weighted MaxSat

RVs coupled by MRF edge if they appear in same clause

MRF assumption:
\[
P[X_i|X_1...X_n] = P[X_i|N(X_i)]
\]
MRF: Markovian Probabilistic Graphical Model

Network of discrete random variables (often binary)

Markov assumption: \( P[X_1 | X_2, X_3 \ldots X_n] = P[X_1 | \text{Neighbors}(X_1)] \)

Hammersley-Clifford Theorem:
\[
P[X_1 X_2 \ldots] = \frac{1}{Z} \prod_c \Phi_c(X_i X_j \ldots \in c)
\]

...over all cliques \( c \)

or as log-linear model:
\[
P[X_1 X_2 \ldots] = \frac{1}{Z} \exp \left( \sum_c w_c f_c(X_i X_j \ldots \in c) \right)
\]

...weights \( w \) features \( f \)

Inference for \( X_i \)'s by Monte Carlo sampling, belief propagation, etc.

Parameter learning by non-convex optimization
Related Alternative Probabilistic Models

Constrained Conditional Models  [Roth et al.]
log-linear classifiers with constraint-violation penalty
mapped into Integer Linear Programs

Factor Graphs with Imperative Variable Coordination  [A. McCallum et al.]
RV’s share “factors“ (joint feature functions)
generalizes MRF, BN, CRF;  inference via advanced MCMC
flexible coupling & constraining of RV’s

Probabilistic Soft Logic (PSL)  [L. Getoor et al.]
gains MAP efficiency by continuous RV’s (degree of truth)
so far KB has explicit model:

• canonicalized entities
• relations with type signatures \(<\text{entity1}, \text{relation}, \text{entity2}>\)

\(<\text{CarlaBruni marriedTo NicolasSarkozy}> \in \text{Person} \times \text{R} \times \text{Person}\)

\(<\text{NataliePortman wonAward AcademyAward}> \in \text{Person} \times \text{R} \times \text{Prize}\)

Open and Dynamic Knowledge Harvesting:
would like to discover new entities and new relation types
\(<\text{name1}, \text{phrase}, \text{name2}>\)

*Madame Bruni* in *her happy marriage with the French president* ...
*The first lady* had *a passionate affair with Stones singer Mick* ...
*Natalie* was *honored by the Oscar* ...
*Bonham Carter* was *disappointed that her nomination for the Oscar* ...
Example: ReVerb

Open Information Extraction

307 answers from 1015 sentences (cached)

Whitney Houston, Jermaine Jackson (7)
John McCain, a lobbyist (5)
Bill Clinton, Monica Lewinsky (5)
Jesus, Mary Magdalene (5)
Suzanne Coleman, Bill Clinton (3)
her mother, Tiger Woods (3)
the medias, Barack Obama (3)
Newt Gingrich, House (3)
Thomas Jefferson, Sally Hemings (3)
Saddam Hussein, Samira Shahbandar (3)
Suzanne Coleman Reportedly, Bill Clinton (3)
his wife, George Foreman (2)
Clementine Churchill, Baroness Spencer-Churchill, Terence Phillip (2)
the extraterrestrial, Hillary Rodham Clinton (2)
the unmentioned, John F. Kennedy (2)

http://openie.cs.washington.edu
http://openie.allenai.org
Consider all verbal phrases as potential relations and all noun phrases as arguments

Problem 1: incoherent extractions
   “New York City has a population of 8 Mio” → <New York City, has, 8 Mio>
   “Hero is a movie by Zhang Yimou” → <Hero, is, Zhang Yimou>

Problem 2: uninformative extractions
   “Gold has an atomic weight of 196” → <Gold, has, atomic weight>
   “Faust made a deal with the devil” → <Faust, made, a deal>

Problem 3: over-specific extractions
   “Hero is the most colorful movie by Zhang Yimou”
   → <..., is the most colorful movie by, ...>

Solution:
• regular expressions over POS tags:
  VB DET N PREP; VB (N | ADJ | ADV | PRN | DET)* PREP; etc.
• relation phrase must have # distinct arg pairs > threshold
### Mining Paraphrases of Relations

<table>
<thead>
<tr>
<th>composed (musician, song)</th>
<th>covered (musician, song)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dylan wrote his song Knockin’ on Heaven’s Door, a cover song by the Dead Morricone’s masterpiece is the Ecstasy of Gold, covered by Yo-Yo Ma.</td>
<td>Nina Simone’s singing of Don’t Explain revived Holiday’s old song.</td>
</tr>
<tr>
<td>Amy’s souly interpretation of Cupid, a classic piece of Sam Cooke.</td>
<td>Cale performed Hallelujah written by L. Cohen.</td>
</tr>
</tbody>
</table>

**covered by:**
- (Amy, Cupid), (Ma, Ecstasy), (Nina, Don’t),
- (Cat, Don’t), (Cale, Hallelujah), ...

**voice in version of:**
- (Amy, Cupid), (Sam, Cupid), (Nina, Don’t),
- (Cat, Don’t), (Cale, Hallelujah), ...

**performed:**
- (Amy, Cupid), (Amy, Black), (Nina, Don’t),
- (Cale, Hallelujah), (Dylan, Knockin"

**frequent sequence mining for relational phrases**

**support sets of entity pairs for paraphrases**

**clustering for "synsets"**

**covered (musician, song):**
cover song, interpretation of, singing of, voice in … version , ...

**composed (musician, song):**
wrote song, classic piece of, ‘s old song, written by, composition of, …
WordNet-style dictionary/taxonomy for relational phrases based on SOL patterns (syntactic-lexical-ontological)

Relational phrases are typed

\(<\text{person}\> \text{ graduated from } <\text{university}\>
\<\text{singer}\> \text{ covered } <\text{song}\>
\<\text{book}\> \text{ covered } <\text{event}\>

Relational phrases can be synonymous

“graduated from” ⇔ “obtained degree in * from”
“and PRP ADJ advisor” ⇔ “under the supervision of”

One relational phrase can subsume another

“wife of” ⇒ “spouse of”
PATTY: Pattern Taxonomy for Relations


350 000 SOL patterns with 4 Mio. instances accessible at: www.mpi-inf.mpg.de/yago-naga/patty

In particular, Rossdale’s forced, stream of consciousness dismissed by some as an imitation singer, Kurt Cobain.

lead singer;
- Paramore, Hayley Williams
- All (band), Dave Smalley
- Alabama (band), Randy Owen
- Clutch (band), Neil Fallon
- Nirvana (band), Kurt Cobain
- Los Bravos, Mike Kogle
- Twisted Sister, Dee Snider
15.3.4 Harvesting Commonsense by Patterns and Logical & Statistical Inference

Assertions about general concepts (not individual entities) and their attributes and relations

hasProperty (circle, round), hasProperty (lake, round)
hasProperty (coffee, strong)
hasAbility (bird, fly), hasAbility (human, make jokes)
hasColor (cherry, red), hasTaste (cherry, juicy), hasShape (cherry, round)
smallerThan (cherry, apple), largerThan (cherry, pea)
partOf (pedal, bike), partOf (nose, human), visualPartOf (nose, human)
locatedAt (bike, park), locatedAt (coffee, cup),
usedFor (cherry, ice cream), usedFor (book, learn),
happensAtTime (traffic jam, rush hour), happensAtLocation (traffic jam, street)
Every child knows that

apples are green, red, round, juicy, …
but not fast, funny, verbose, …

pots and pans are in the kitchen or cupboard, on the stove, …
but not in in the bedroom, in your pocket, in the sky, …

children usually live with their parents

But: commonsense is rarely stated explicitly
Plus: web and social media have reporting bias

rich family: 27.8 Mio on Bing
poor family: 3.5 Mio on Bing

singers: 22.8 Mio on Bing
workers: 14.5 Mio on Bing
Example: ConceptNet

many inputs incl. WordNet, Verbosity game, etc.

ConceptNet 5:
3.9 Mio concepts
12.5 Mio. edges

http://conceptnet5.media.mit.edu/
Example: WebChild

https://gate.d5.mpi-inf.mpg.de/webchild/

a general name for alcoholic beverages made by fermenting a cereal (or mixture of cereals) flavored with hops
Pattern-Based Harvesting of Commonsense Properties
(N. Tandon et al.: AAAI 2011)

Approach: Start with seed facts for

- apple hasProperty round
- dog hasAbility bark
- plate hasLocation table

Find patterns that express these relations, such as

- X is very Y, X can Y, X put in/on Y, …

Apply these patterns to find more facts.

Problem: noise and sparseness of data
Solution: harness Web-scale n-gram corpora

→ 5-grams + frequencies

Confidence score: PMI (X,Y), PMI (p,(XY)), support(X,Y), …

are features for regression model
Commonsense with SPO Properties
[N. Tandon et al.: WSDM’14]

Who looks hot? What tastes hot? What is hot? What feels hot?

→ 4 Mio sense-disambiguated SPO triples for predicates:
  hasProperty, hasColor, hasShape, hasTaste, hasAppearance,
  isPartOf, hasAbility, hasEmotion, …

• pattern learning with seeds: high recall
• semisupervised label propagation: good precision
• integer linear program: sense disambiguation, high precision

https://gate.d5.mpi-inf.mpg.de/webchild/
Visual Commonsense

**ImageNet:** populate WordNet classes with many photos
[J. Deng et al.: CVPR‘09]
http://www.image-net.org

**NEIL:** infer instances of partOf occursAt, inScene relations
[X. Chen et al.: ICCV‘13]
http://www.neil-kb.com/

**Mountain bike, all-terrain bike, off-roader**
A bicycle with a sturdy frame and fat tires, originally designed for riding in mountainous country

**NEIL: Never Ending Image Learner**
Crash, I See, I Learn.

**STATISTICS**
- 1,310,143 images
- 31,492 image relations
- 4,995 visual relationships

**Bicycle**
- object: sport
- occursAt: park
- partOf: bike
- pedal
- inScene

**Clusters Discovered**
- bicycle
- footbike
- track
- road
- mountain bike
- off-road
- bike
- mountain bike
- off-road
- bike

**Relationships Discovered**
- inScene
- partOf
- occursAt
- pedal

**How:**
crowdsourcing for seeds, distantly supervised classifiers, object recognition (bounding boxes) in computer vision
Activity knowledge from movie & TV scripts, aligned with visual scenes

→ 0.5 Mio activity types with attributes: location, time, participants, prev/next

Refined part-whole relations from web & books text and image tags

→ 6.7 Mio sense-disambiguated triples for physicalPartOf, visualPartOf, hasCardinality, memberOf, substanceOf
Challenge: Commonsense Rules

Horn clauses:
can be learned by Inductive Logic Programming

\[
\forall x, m, c: \text{type}(x, \text{child}) \land \text{mother}(x, m) \land \text{livesIn}(m, t) \Rightarrow \text{livesIn}(x, t)
\]

\[
\forall x, m, f: \text{type}(x, \text{child}) \land \text{mother}(x, m) \land \text{spouse}(m, f) \Rightarrow \text{father}(x, f)
\]

Advance rules beyond Horn clauses:
specified by human experts

\[
\forall x: \text{type}(x, \text{spider}) \Rightarrow \text{numLegs}(x) = 8
\]

\[
\forall x: \text{type}(x, \text{animal}) \land \text{hasLegs}(x) \Rightarrow \text{even}(\text{numLegs}(x))
\]

\[
\forall x: \text{human}(x) \Rightarrow (\exists y: \text{mother}(x, y) \land \exists z: \text{father}(x, z))
\]

\[
\forall x: \text{human}(x) \Rightarrow (\text{male}(x) \lor \text{female}(x))
\]
Additional Literature for 15.3

- F.M. Suchanek, G. Weikum: Knowledge harvesting in the big-data era, SIGMOD 2013
- M. Hearst: Automatic Acquisition of Hyponyms from Large Text Corpora. COLING 1992
- E. Agichtein. Snowball: extracting relations from large plain-text collections, ACM DL 2000
- F. Suchanek et al.: SOFIE: a self-organizing framework for information extraction, WWW ‘09
- M. Mintz et al.: Distant supervision for relation extraction without labeled data. ACL 2009
- N. Nakashole: Scalable knowledge harvesting with high precision and high recall, WSDM ‘11
- A. Fader et al.: Identifying Relations for Open Information Extraction, EMNLP 2011
- Mausam et al.: Open Language Learning for Information Extraction, EMNLP 2012
- N. Nakashole: PATTY: A Taxonomy of Relational Patterns with Semantic Types, EMNLP’12
- R. Speer, C. Havasi: Representing General Relational Knowledge in ConceptNet 5, LREC’12
- N. Tandon et al.: Deriving a Web-Scale Common Sense Fact Database, AAAI 2011
Summary of Chapter 15

• Information Extraction lifts text&Web contents into structured data: entities, attributes, relations, facts and opinions

• Regex-centric rules and patterns good for homogenous Web sites

• Statistical learning of patterns (HMM, CRF/MRF, classifiers, etc.) crucial for heterogenous sources and natural-language text

• Knowledge harvesting exploits Web-scale redundancy & statistics