Knowledge Base Construction Pipelines

Christian Grant
Projects in Data Science – October 22th 2014
http://www.cise.ufl.edu/class/cis6930fa14pro/

DSR @ UF
Data Science Research
Outline

• Quick motivation and description of knowledge base construction.

• The Pipeline

• Discussion and pipeline systems
Knowledge Base Construction Pipeline
Knowledge Base Construction Pipeline

- What is a **Knowledge Base**?
Knowledge Base Construction Pipeline

• What is a Knowledge Base?

• A collection of facts (database) with an inference engine.
Knowledge Base Construction Pipeline

• What is a **Knowledge Base**?

• A collection of facts (database) with an inference engine.

• The **Knowledge Base Construction pipeline** is the series of tasks involved in creating a Knowledge Base (KB).
Get the data!

• Data sets are typically compressed in large batches of files.

• The files are:
  • Encrypted with gpg
  • Compressed .gz/.bz/.xz files
  • Hadoop sequence .sc files
  • Thrift/JSON/XML/CSV
Get the data!

• Data sets are typically compressed in large batches of files.

• The files are:
  • Encrypted with gpg
  • Compressed .gz/.bz/.xz files
  • Hadoop sequence .sc files
  • Thrift/JSON/XML/CSV
Get the data!

- Data sets are typically compressed in large batches of files.
- The files are:
  - Encrypted with gpg
  - Compressed .gz/.bz/.xz files
  - Hadoop sequence .sc files
  - Thrift/JSON/XML/CSV
Get the data!

- Data sets are typically compressed in large batches of files.
- The files are:
  - Encrypted with gpg
  - Compressed .gz/.bz/.xz files
  - Hadoop sequence .sc files
  - Thrift/JSON/XML/CSV
Get the data!

- Data sets are typically compressed in large batches of files.
- The files are:
  - Encrypted with gpg
  - Compressed .gz/.bz/.xz files
  - Hadoop sequence .sc files
  - Thrift/JSON/XML/CSV
Get the data!

- Data sets are typically compressed in large batches of files.

- The files are:
  - Encrypted with gpg
  - Compressed .gz/.bz/.xz files
  - Hadoop sequence .sc files
  - Thrift/JSON/XML/CSV
Get the data!

- Data sets are typically compressed in large batches of files.

- The files are:
  - Encrypted with gpg
  - Compressed .gz/.bz/.xz files
  - Hadoop sequence .sc files
  - Thrift/JSON/XML/CSV
Get the data!

- Data sets are typically compressed in large batches of files.

- The files are:
  - Encrypted with gpg
  - Compressed .gz/.bz/.xz files
  - Hadoop sequence .sc files
  - Thrift/JSON/XML/CSV
Sentence Segmentation

- Split a textual document into sentences.

```python
sentence = tokenize.sent_tokenize(text)
```

She stopped. She said, "Hello there," and then went on.

He's vanished! What will we do? It's up to us.

Please add 1.5 liters to the tank.
Word Tokenization

• Split a sentence into tokens

• text.split(“ “) is not always enough

• What about apostrophe, abbreviations, misspellings, URIs, different languages?

In Düsseldorf I took my hat off. But I can't put it back on.
Part-of-Speech Tagging (POS)

- Classifying word tokens into parts of speech

There is also a Brown tag set http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html
Example generated from http://nlp.stanford.edu:8080/corenlp/process
Part-of-Speech Tagging (POS)

- Classifying word tokens into parts of speech

<table>
<thead>
<tr>
<th>Penn Treebank POS tagset</th>
<th>Brown tag set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CC     Coordinating conjunction</td>
<td><a href="http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html">http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html</a></td>
</tr>
<tr>
<td>2. CD     Cardinal number</td>
<td></td>
</tr>
<tr>
<td>3. DT     Determiner</td>
<td></td>
</tr>
<tr>
<td>4. EX     Existential there</td>
<td></td>
</tr>
<tr>
<td>5. FW     Foreign word</td>
<td></td>
</tr>
<tr>
<td>6. IN     Proposition/subordinating conjunction</td>
<td></td>
</tr>
<tr>
<td>7. JJ     Adjective</td>
<td></td>
</tr>
<tr>
<td>8. JJR    Adjective, comparative</td>
<td></td>
</tr>
<tr>
<td>9. JJ$    Adjective, superlative</td>
<td></td>
</tr>
<tr>
<td>10. LS    List item marker</td>
<td></td>
</tr>
<tr>
<td>11. MD    Modal</td>
<td></td>
</tr>
<tr>
<td>12. NN    Noun, singular or mass</td>
<td></td>
</tr>
<tr>
<td>13. NNS   Noun, plural</td>
<td></td>
</tr>
<tr>
<td>14. NNP   Proper noun, singular</td>
<td></td>
</tr>
<tr>
<td>15. NNPS  Proper noun, plural</td>
<td></td>
</tr>
<tr>
<td>16. PDT   Predeterminer</td>
<td></td>
</tr>
<tr>
<td>17. POS   Possessive ending</td>
<td></td>
</tr>
<tr>
<td>18. PRP   Personal pronoun</td>
<td></td>
</tr>
<tr>
<td>19. PRP$  Possessive pronoun</td>
<td></td>
</tr>
<tr>
<td>20. RB    Adverb</td>
<td></td>
</tr>
<tr>
<td>21. RBR   Adverb, comparative</td>
<td></td>
</tr>
<tr>
<td>22. RBS   Adverb, superlative</td>
<td></td>
</tr>
<tr>
<td>23. RP    Particle</td>
<td></td>
</tr>
<tr>
<td>24. SYM   Symbol (mathematical or scientific)</td>
<td></td>
</tr>
<tr>
<td>25. TO    to</td>
<td></td>
</tr>
<tr>
<td>26. UH    Interjection</td>
<td></td>
</tr>
<tr>
<td>27. VB    Verb, base form</td>
<td></td>
</tr>
<tr>
<td>28. VBD   Verb, past tense</td>
<td></td>
</tr>
<tr>
<td>29. VBG   Verb, gerund/present participle</td>
<td></td>
</tr>
<tr>
<td>30. VBN   Verb, past participle</td>
<td></td>
</tr>
<tr>
<td>31. VBP   Verb, non-3rd ps. sing. present</td>
<td></td>
</tr>
<tr>
<td>32. VBZ   Verb, 3rd ps. sing. present</td>
<td></td>
</tr>
<tr>
<td>33. WDT   wh-determiner</td>
<td></td>
</tr>
<tr>
<td>34. WP    wh-pronoun</td>
<td></td>
</tr>
<tr>
<td>35. WPS   Possessive wh-pronoun</td>
<td></td>
</tr>
<tr>
<td>36. WRB   wh-adverb</td>
<td></td>
</tr>
<tr>
<td>37. #     Pound sign</td>
<td></td>
</tr>
<tr>
<td>38. $     Dollar sign</td>
<td></td>
</tr>
<tr>
<td>39. .     Sentence-final punctuation</td>
<td></td>
</tr>
<tr>
<td>40. ,     Comma</td>
<td></td>
</tr>
<tr>
<td>41. :     Colon, semi-colon</td>
<td></td>
</tr>
<tr>
<td>42. (     Left bracket character</td>
<td></td>
</tr>
<tr>
<td>43. )     Right bracket character</td>
<td></td>
</tr>
<tr>
<td>44. &quot;     Straight double quote</td>
<td></td>
</tr>
<tr>
<td>45. '     Left open single quote</td>
<td></td>
</tr>
<tr>
<td>46. &quot;     Left open double quote</td>
<td></td>
</tr>
<tr>
<td>47. '     Right close single quote</td>
<td></td>
</tr>
<tr>
<td>48. &quot;     Right close double quote</td>
<td></td>
</tr>
</tbody>
</table>

There is also a Brown tag set http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html
Example generated from http://nlp.stanford.edu:8080/corenlp/process
Part-of-Speech Tagging (POS)

- Classifying word tokens into parts of speech

There is also a Brown tag set [http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html](http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html)
Example generated from [http://nlp.stanford.edu:8080/corenlp/process](http://nlp.stanford.edu:8080/corenlp/process)
Named Entity Recognition (NER)

- Identify the tokens in a sentence that correspond to an Entity.

<table>
<thead>
<tr>
<th>Type</th>
<th>Tag</th>
<th>Sample Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>PER</td>
<td>Individuals, fictional characters, small groups</td>
</tr>
<tr>
<td>Organization</td>
<td>ORG</td>
<td>Companies, agencies, political parties, religious groups, sports teams</td>
</tr>
<tr>
<td>Location</td>
<td>LOC</td>
<td>Physical extents, mountains, lakes seas</td>
</tr>
<tr>
<td>Geo-Political Entity</td>
<td>GPE</td>
<td>Countries, states, provinces, counties</td>
</tr>
<tr>
<td>Facility</td>
<td>FAC</td>
<td>Bridges, buildings, airports</td>
</tr>
<tr>
<td>Vehicles</td>
<td>VEH</td>
<td>Planes, trains, and automobiles</td>
</tr>
</tbody>
</table>

Notice the difference the second won identifies names that are multiple tokens. In this case we need chunking.
Named Entity Recognition (NER)

- Identify the tokens in a sentence that correspond to an Entity.

<table>
<thead>
<tr>
<th>Type</th>
<th>Tag</th>
<th>Sample Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>PER</td>
<td>Individuals, fictional characters, small groups</td>
</tr>
<tr>
<td>Organization</td>
<td>ORG</td>
<td>Companies, agencies, political parties, religious groups, sports teams</td>
</tr>
<tr>
<td>Location</td>
<td>LOC</td>
<td>Physical extents, mountains, lakes seas</td>
</tr>
<tr>
<td>Geo-Political Entity</td>
<td>GPE</td>
<td>Countries, states, provinces, counties</td>
</tr>
<tr>
<td>Facility</td>
<td>FAC</td>
<td>Bridges, buildings, airports</td>
</tr>
<tr>
<td>Vehicles</td>
<td>VEH</td>
<td>Planes, trains, and automobiles</td>
</tr>
</tbody>
</table>

My favorite class is Projects in data science taught by Dr. Wang and Dr. Ranka.

Notice the difference the second word identifies names that are multiple tokens. In this case we need chunking.
Chunking

- Identify sequences of non-overlapping labels

Chunking

• Identify sequences of non-overlapping labels

Chunking

- Identify sequences of non-overlapping labels

Chunking

- Identify sequences of non-overlapping labels

Chunking

- Identify sequences of non-overlapping labels

NP - Chunking

Chunking

- Identify sequences of non-overlapping labels

NP - Chunking

dog - DT, JJ, NN

cat - DT, NN

El presidente de la Duma (cámara baja), Guennadi Selezniov, calificó de "claramente apa
del Kremlin para Chechenia, Serguéi Yastrzhembski.

Destacados representantes del Parlamento y la prensa rusos criticaron hoy el "belicism
ha definido como posible blanco de su lucha antiterrorista.

Chunking

• IOB Representation

• Every token is In a chunk or Out of a chunk.

• Distinguish the Beginnings of chunks.

<table>
<thead>
<tr>
<th>We</th>
<th>saw</th>
<th>the</th>
<th>yellow</th>
<th>dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRP</td>
<td>VBD</td>
<td>DT</td>
<td>JJ</td>
<td>NN</td>
</tr>
<tr>
<td>B-NP</td>
<td>O</td>
<td>B-NP</td>
<td>I-NP</td>
<td>I-NP</td>
</tr>
</tbody>
</table>

MADden

http://dl.acm.org/citation.cfm?doid=2396761.2398746
MADden: Query-Driven Statistical Text Analytics

Give me sentiment comments about Revis

Query 1

Compare players tebow and revis by the twitter sentiment over dates from
10/22/2011 to 10/27/2011

and return 5 results.

Query 2

Return all the named entity tags from the text

Kim began his career in psychology, graduating from UF with a master’s degree in clinical psychology in 1971 and a doctorate in the same subject in 1974. While at UF, he met his wife, Katrine, who also earned her doctorate in clinical psychology at UF.

Query 4

http://dl.acm.org/citation.cfm?doid=2396761.2398746
MADden

Statistical

Query

The Query

```sql
-- select termnum, term from cgrant_ne_chunk('Kim began his career in psychology, graduating from UF with a master's degree in clinical psychology in 1971 and a doctorate in the same subject in 1974. While at UF, he met his wife, Katrine, who also earned her doctorate in clinical psychology at UF. He worked in the mental health field for six years, first as an intern and later at community mental health centers and in a private practice in Kentucky that he owned with his wife. He also was a full-time faculty member at Bellarmine University in Louisville for six years', true) where tag = 'NE';
```

4.608199261261 sec

Answer

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Kim</td>
</tr>
<tr>
<td>32</td>
<td>UF</td>
</tr>
<tr>
<td>39</td>
<td>Katrine</td>
</tr>
<tr>
<td>79</td>
<td>Kentucky</td>
</tr>
<tr>
<td>94</td>
<td>Bellarmine</td>
</tr>
<tr>
<td>94</td>
<td>University</td>
</tr>
<tr>
<td>96</td>
<td>Louisville</td>
</tr>
</tbody>
</table>

Query Plan

Function Scan on cgrant_ne_chunk (cost=0.25..12.75 rows=5 width=36)
Filter: (tag = 'NE':text)
Dependency Parsing

• A graph depicting the relationship between a word (head) and its dependents.

• Starts with a verb and finds the related subject and object.

• Useful in understanding phrases

• Similar to chunking

• Very close to semantic relationships

• Link grammar is the most notable implementation (in AbiWord)

Varies between Subject–Verb–Object languages and SOV languages
Example slides from http://stp.linqfil.uu.se/~nivre/docs/ACLslides.pdf
http://www.abisource.com/projects/link-grammar/
http://en.wikipedia.org/wiki/Link_grammar
had --
something happened with news, economic news
it had an effect, little effect
little effect on markets, financial markets
Dependency Parsing

Economic news had little effect on financial markets.

had --
something happened with news, economic news
it had an effect, little effect
little effect on markets, financial markets
something happened with news, economic news
it had an effect, little effect
little effect on markets, financial markets
Dependency Parsing

Economic news had little effect on financial markets.

had --
something happened with news, economic news
it had an effect, little effect
little effect on markets, financial markets
had --
something happened with news, economic news
it had an effect, little effect
little effect on markets, financial markets
had --
something happened with news, economic news
it had an effect, little effect
little effect on markets, financial markets
something happened with news, economic news it had an effect, little effect little effect on markets, financial markets
Word-Sense Disambiguation

• Classifying the meaning of a word among many possible interpretations.

• Classification can be done in a myriad of ways.

• Still an open NLP problem

• I like bass!
Co-Reference Resolution

- Determining the mentions in a document that correspond to the same entity.

“I voted for Nader because he was most aligned with my values,” she said.

http://www.bart-coref.org/
Co-Reference Resolution

- Determining the mentions in a document that correspond to the same entity.

“I voted for Nader because he was most aligned with my values,” she said.
Co-Reference Resolution

- Determining the mentions in a document that correspond to the same entity.

"I voted for Nader because he was most aligned with my values," she said.
Cross-Document Entity Resolution

- Take coreference chains from *across documents* and match the ones that correspond to the same real world entity.

- A type of clustering problem.

- Use the features from the document.
Entity Resolution using Graphical Models

- Score the current state by scoring the factors.
- Find the best score by proposing new arrangements (MCMC)
- Order the proposals so we find the arrangement we want to know first (Query-driven MCMC)
Document Indexing

- Index document along with parts of speech and other features.
- It helps for fast searching, lookup, and statistics computation.
- Using Lucene/Solr, Sphinx fast retrieval
- In-database full-text indexing with PostgreSQL/MySQL
GPText: In-database text analytics

- Text analytics inside Greenplum (MPP Database)
- Parallel loading of files
- Distributed indexes and statistical computation.

http://dl.acm.org/citation.cfm?id=2486767.2486774
Relation Extraction

- Structuring phrases into a *subject* – *predicate* – *object* format.
- SVO triples are very close to logic expressions statements.
- Some Regex, build an ML model
- OLLIE uses a trained ML model build over a dependency grammar.

*Bill Gates works at Microsoft.*
\[ \rightarrow \text{Works-at(Bill Gates, Microsoft)} \]

*UF is located in Gainesville.*
\[ \rightarrow \text{Located-In(UF, Gainesville)} \]

[Links to additional resources]
Streaming Fact Extraction

- Given a set of *entities*, a set of *slots*, and 5 TBs of compressed documents.

- Put the items through the pipeline and extract facts.
Streaming Fact Extraction

- Given a set of entities, a set of slots, and 5 TBs of compressed documents.

- Put the items through the pipeline and extract facts.
Streaming Fact Extraction

- Given a set of entities, a set of slots, and 5 TBs of compressed documents.
- Put the items through the pipeline and extract facts.

“Boris Berezovsky, who made his fortune in Russia in the 1990s, passed away March 2013.”
Streaming Fact Extraction

- Given a set of entities, a set of slots, and 5 TBs of compressed documents.
- Put the items through the pipeline and extract facts.

“Boris Berezovsky, who made his fortune in Russia in the 1990s, passed away March 2013.”

_DiedOn_(Boris Berezovsky, March 2013)
_LivedIn_(Boris Berezovsky, Russia)

Knowledge Base Construction

- Collect the generated relations into a DB.
- Define or Mine for rules.
- **Ground** the relations to find the expressed facts and their probabilities.
- See ProbKB by Yang and Daisy

<table>
<thead>
<tr>
<th>Schema</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>pSimHard(perl, per2)</td>
<td>coOccurs(‘Ullman’, ‘Stanford Univ’)</td>
</tr>
<tr>
<td>pSimSoft(perl, per2)</td>
<td>coOccurs(‘Jeff Ullman’, ‘Stanford’)</td>
</tr>
<tr>
<td>oSimHard(orgl, org2)</td>
<td>coOccurs(‘Gray’, ‘San Jose Lab’)</td>
</tr>
<tr>
<td>pSimSoft(orgl, org2)</td>
<td>coOccurs(‘J. Gray’, ‘IBM San Jose’)</td>
</tr>
<tr>
<td>coOccurs(per, org)</td>
<td>coOccurs(‘Mike’, ‘UC-Berkeley’)</td>
</tr>
<tr>
<td>homepage(per, page)</td>
<td>coOccurs(‘Mike’, ‘UCB’)</td>
</tr>
<tr>
<td>oMention(page, org)</td>
<td>faculty(‘MIT’, ‘Chomsky’)</td>
</tr>
<tr>
<td>faculty(per, org)</td>
<td>homepage(‘Joe’, ‘Doc201’)</td>
</tr>
<tr>
<td>+affil(per, org)</td>
<td>oMention(‘Doc201’, ‘IBM’)</td>
</tr>
<tr>
<td>+oCoref(orgl, org2)</td>
<td></td>
</tr>
<tr>
<td>+pCoref(perl, per2)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>weight</th>
<th>rule</th>
<th>evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+\infty$</td>
<td>pCoref($p, p$)</td>
<td>coOccurs(‘Ullman’, ‘Stanford Univ’)</td>
</tr>
<tr>
<td>$+\infty$</td>
<td>pCoref($p1, p2$) =&gt; pCoref($p2, p1$)</td>
<td>coOccurs(‘Jeff Ullman’, ‘Stanford’)</td>
</tr>
<tr>
<td>$+\infty$</td>
<td>pCoref($x, y$), pCoref($y, z$) =&gt; pCoref($x, z$)</td>
<td>coOccurs(‘Gray’, ‘San Jose Lab’)</td>
</tr>
<tr>
<td>6</td>
<td>pSimHard($p1, p2$) =&gt; pCoref($p1, p2$)</td>
<td>coOccurs(‘J. Gray’, ‘IBM San Jose’)</td>
</tr>
<tr>
<td>2</td>
<td>pSimSoft($p1, p2$) =&gt; pCoref($p1, p2$)</td>
<td>coOccurs(‘Mike’, ‘UC-Berkeley’)</td>
</tr>
<tr>
<td>$+\infty$</td>
<td>faculty($o, p$) =&gt; affil($p, o$)</td>
<td>coOccurs(‘Mike’, ‘UCB’)</td>
</tr>
<tr>
<td>8</td>
<td>homepage($p, d$), oMention($d, o$) =&gt; affil($p, o$)</td>
<td>faculty(‘MIT’, ‘Chomsky’)</td>
</tr>
<tr>
<td>3</td>
<td>coOccurs($p, o1$), oCoref($o1, o2$) =&gt; affil($p, o2$)</td>
<td>homepage(‘Joe’, ‘Doc201’)</td>
</tr>
<tr>
<td>4</td>
<td>coOccurs($p1, o$), pCoref($p1, p2$) =&gt; affil($p2, o$)</td>
<td>oMention(‘Doc201’, ‘IBM’)</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Popular Tools

• NLTK (Python)
• StanfordNLP/OpenNLP (Java)
• Dlib (C++)
• ScalaNLP (Scala)
Discussion
Discussion

- Batch vs Streaming
Discussion

• Batch vs Streaming

• Corpus Exploration vs Continuous Querying
Discussion

- Batch vs Streaming
- Corpus Exploration vs Continuous Querying
- Speed vs Accuracy
Discussion

• Batch vs Streaming
• Corpus Exploration vs Continuous Querying
• Speed vs Accuracy
• Serial/\textbf{Joint} vs Parallelization
Other Projects

- OpenIE — UW knowitall project
- NELL — CMU Never-Ending Language Learning
- GATE — Univ. Sheffield General Architecture for Text Engineering
- UIMA/SystemT — IBM
- Purple Sox — Yahoo
- DeepDive — Stanford
- TweetNLP — CMU Twitter parser
- Many others!!

http://www.ark.cs.cmu.edu/TweetNLP/
Other pipeline tasks

- Word normalization: ttyl => talk to you later, 2morrow => tomorrow
- Digit normalization: 2 => two, 17 => Seventeen, 40 => DIGIT
- Stop word elimination: a, the, and is
- N-gram generation
- Lemmatization
- Sentiment analysis
- Event Detection
Thank you

- http://www.christangrant.com
- http://dsr.cise.ufl.edu
State-of-the-art
