Software for Text Data Mining

Steven Abney

Dept. of Linguistics

2013 Oct 14

Text processing pipeline

- **Acquisition**
  - Conversion: PDF to plain text, OCR, screen scraping
  - Preprocessing: tokenization, lowercasing, stopword elimination, stemming, synonym expansion, POS tagging
- **Extraction**
  - Entities: chunking, named entities, other entities, coreference resolution, multi-reference merging
  - Relations: parsing, relation extraction
- **Analysis**
  - Feature extraction, creation of instance matrix, tf-idf weighting, feature selection
  - Stats, machine learning, clustering, classification

Example
Preprocessing

- Plain text

```
On the Commodity Exchange in New York, gold settled at
$367.30 an ounce, up 20 cents.  Estimated volume was a
light 2.4 million ounces.
```

- Sentence segmentation `nltk.sent_tokenize(txt)`

```
On the Commodity Exchange in New York, gold settled at $367.30
Estimated volume was a light 2.4 million ounces.
```

- Tokenization `nltk.word_tokenize(sent)`

```
in New York , gold settled at $ 367.30 an ounce , up 20 cents .
```

- POS tagging `nltk.pos_tag(toks)`

```
in  New  York  ,  gold  settled  at  $  367.30  an  ounce  ,
IN  NNP  NNP   ,  NN    VBD      IN  $  CD      DT  NN     ,
```

Example
Extraction

- Named entities `nltk.ne_chunk(tagged)`

  On the Commodity Exchange (ORGANIZATION) in New York (GPE) , gold settled at

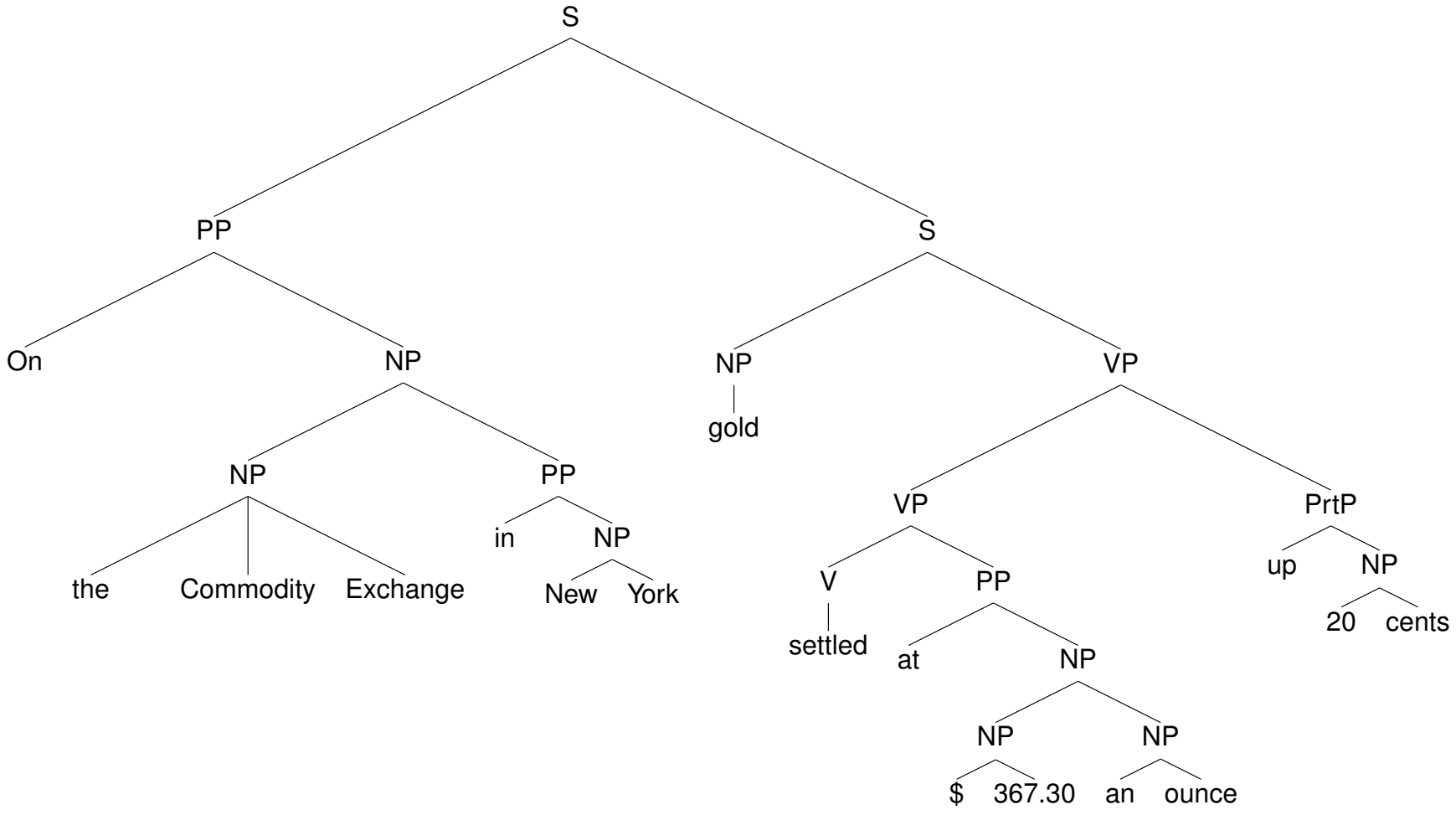
- Chunks, other entities

```
P           On
NP(Org)     the Commodity Exchange
P           in
NP(GPE)     New York
NP          gold
VP          settled
P           at
NP(Money)   $ 367.30
NP(Unit)    an ounce
```

- Relations `nltk.sem.extract_rels(doc)`

```
gold    settled at    $ 367.30
```

Example
Parse tree



Example
Semantic dependencies

$$
\begin{bmatrix}
\text{settle} & \\
\textit{subj:} & \text{gold} \\
\text{on:} & \begin{bmatrix} \text{Commodity Exchange} \\ \text{in:} \quad \text{New York} \end{bmatrix} \\
\text{at:} & \begin{bmatrix} \textit{money amount} & \\ \textit{unit:} & \$ \\ \textit{qty:} & 367.30 \\ \textit{NP-mod:} & \text{ounce} \end{bmatrix} \\
\text{up:} & \begin{bmatrix} \textit{money amount} & \\ \textit{unit:} & \text{cent} \\ \textit{qty:} & 20 \end{bmatrix}
\end{bmatrix}
$$

NLTK (`nltk.org`)

General toolkit, Python

- Texts: concordancing, vocabulary, stemming, distrib sim, bigrams, collocations
- Data: pronouncing dict, stopwords, gazetteer lists, names, propbank, roget's, verbnet, wordnet, text corpora
- Python provides: string processing, regex, charset conversion
- Preprocessing: feed parser, screen scraping, various stemmers, tokenization, sentence segmentation, POS tagging
- Extraction: named entities, facilities for building variety of chunkers, parsers, interpreters, relation extractors
- Learning: Naive Bayes, decision tree, loglinear models, confusion matrices

OpenNLP (opennlp.apache.org)

General toolkit, Java

- More industrial-strength than NLTK
- Preprocessing: sentence segmentation, tokenization, POS tagging
- Entity recognition: name finder, chunker, coreference resolution
- Parser
- Learning: document classifier, loglinear models
- Has models for multiple languages

Stanford NLP Software

Software collection, Java

- `www-nlp.stanford.edu/software/`
- Tokenization; word segmentation for Arabic and Chinese; POS tagging for English, Arabic, Chinese, French, German
- Named entity recognizer, temporal tagger
- Parser, conversion to dependencies, biomedical event parser
- Text classification (loglinear)
- Topic modeling

Preprocessing

- OCR
  - Adobe Acrobat (commercial)
  - ABBYY (`finereader.abbyy.com`, commercial)
  - Omnipage (commercial)
  - Ocropus (`code.google.com/p/ocropus`, open-source)
- Screen scraping (plain text from HTML)
  - Tika (`https://tika.apache.org`)
  - `www.crummy.com/software/BeautifulSoup`
- Plain text from PDF
  - PDF Box (`http://pdfbox.apache.org/`)
  - `tm` (`http://tm.r-forge.r-project.org/`): also stopwords, stemming, etc.
- Plain text from Microsoft formats
  - `javax.swing.text.rtf`, `RTFEditorKit`
  - POI (`http://poi.apache.org/`: MS Office files

Entity extraction

- OpenNLP, Stanford NE extractor, NLTK NE extractor
- GATE (`http://gate.ac.uk/`): plain text conversion, tokenization, gazetteer, sentence segmentation, POS tagging, entity extraction, biomedical entities, coreference
- LingPipe (`alias-i.com/lingpipe`): POS tagging, named entity recognition, doc classification, Naive Bayes, conditional random fields, latent dirichlet allocation
- MinorThird (`http://teamcohen.github.io/MinorThird/`): facilities for training entity extractors and text classifiers

Relation extraction

- Reverb (`reverb.cs.washington.edu`)
  - Sample of output:

bilberry	also contain	vitamin c	0.94124
cabbage	also contain significant amount of	vitamin a	0.92608
folic acid	also play an important role in	hair loss prevention	0.91184

Parsers

- Charniak: `ftp://ftp.cs.brown.edu/pub/nlparser`
- Collins: `http://people.csail.mit.edu/mcollins/PARSER.tar.gz`. Bikel: Java version.
- Berkeley parser: `http://code.google.com/p/berkeleyparser`
- Epic: `https://github.com/dlwh/epic/`
- Stanford: `http://nlp.stanford.edu/software/lex-parser.shtml`
- Malt: `www.maltparser.org`
- MST: `http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html`

Machine learning

- Weka: `http://www.cs.waikato.ac.nz/ml/weka/`
- Mallet: `http://mallet.cs.umass.edu/`
- SciKit-Learn: `http://scikit-learn.org/stable/`
- R packages: `hclust` hierarchical clustering, `lda` latent dirichlet allocation, RWeka, MCMCPack