COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Cognitive Computation Group

Natural Language Processing Tutorial
May 26 & 27, 2011

http://cogcomp.cs.illinois.edu

MULTIMODAL INFORMATION ACCESS & SYNTHESIS mias

So why aren't words enough?

- Depends on the application… more advanced task may require more sophisticated patterns to separate classes
- Sparsity of features
  - Many words/sequences of words may not occur very often
  - This means a learned classifier may not generalize well
  - More abstract representation can help
- Ambiguity of words – e.g. "terminal", "moving"
  - Additional information may help
- Meaning encoded in structure – e.g.

  "Matthew Smith, the Maverick's last hope…"
- NLP annotation tools generally abstract over underlying words so that features generalize better

Outline

- Annotators
  - POS, Chunk, NER, Coreference, SRL
- Comparators
  - Overview
  - Instances: WNSim, NESim
- Curator
  - Overview
  - Installing and running
  - Current services

ANNOTATORS

Available from CCG

- Tokenization/Sentence Splitting
- Part Of Speech
- Chunking
- Named Entity Recognition
- Coreference
- Semantic Role Labeling

Tokenization and Sentence Segmentation

- Given a document, find the sentence and token boundaries

> The police chased Mr. Smith of Pink Forest, Fla. all the way to Bethesda, where he lived. Smith had escaped after a shoot-out at his workplace, Machinery Inc.

- Why?
  - Word counts may be important features
  - Words may themselves be the object you want to classify
  - "lived." and "lived" should give the same information
  - different analyses need to align if you want to leverage multiple annotators from different sources/tasks

Tokenization and Sentence Segmentation ctd.

- Believe it or not, this is an open problem
- No agreed standard for token-level segmentation
  - e.g. "American-led" vs. "American - led"?
  - e.g. "$ 32 M" vs "$32 M" and "$32M"?
- Different tasks may use different standards
- No wildly successful sentence segmenter exists (see the excerpts in news aggregators for some nice errors)
- Noisier text (e.g. online consumer reviews) => poorer performance (for reasons like inconsistent capitalization)
- LBJ distribution includes the Illinois tokenizer and sentence segmenter

Part of Speech (POS)

- Allows simple abstraction for pattern detection

POS	DT	NN	VBD	PP	DT	JJ	NN
Word	The	boy	stood	on	the	burning	deck

POS	DT	NN	VBD	PP	DT	JJ	NN
Word	A	boy	rode	on	a	red	bicycle

- Disambiguate a target, e.g.

  "make (a cake)" vs. "make (of car)"

- Specify more abstract patterns,

  e.g. Noun Phrase: (DT JJ* NN)

- Specify context in abstract way
  - e.g. "DT boy VBX" for "actions boys do"
  - This expression will catch "a boy cried", "some boy ran", …

Chunking

- Identifies phrase-level constituents in sentences

  [NP Boris] [ADVP regretfully] [VP told] [NP his wife] [SBAR that] [NP their child] [VP could not attend] [NP night school] [PP without] [NP permission] .

- Useful for filtering: identify e.g. only noun phrases, or only verb phrases
  - Groups modifiers with heads
  - Useful for e.g. Mention Detection
- Used as source of features, e.g. distance (abstracts away determiners, adjectives, for example), sequence,…
  - More efficient to compute than full syntactic parse
  - Applications in e.g. Information Extraction – getting (simple) information about concepts of interest from text documents

Named Entity Recognition

- Identifies and classifies strings of characters representing proper nouns

  [PER Neil A. Armstrong] , the 38-year-old civilian commander, radioed to earth and the mission control room here: "**[LOC Houston]** , **[ORG Tranquility]** Base here; the Eagle has landed."

- Useful for filtering documents
  - "I need to find news articles about organizations in which Bill Gates might be involved…"
- Disambiguate tokens: "Chicago" (team) vs. "Chicago" (city)
- Source of abstract features
  - E.g. "Verbs that appear with entities that are Organizations"
  - E.g. "Documents that have a high proportion of Organizations"

Coreference

- Identify all phrases that refer to each entity of interest – i.e., group mentions of concepts

  [Neil A. Armstrong] , **[the 38-year-old civilian commander]**, radioed to **[earth]**. **[He]** said the famous words, "**[the Eagle]** has landed"."

- The Named Entity recognizer only gets us part-way…
- …if we ask, "what actions did Neil Armstrong perform?", we will miss many instances (e.g. "He said…")
- Coreference resolver **abstracts over different ways of referring to the same person**
  - Useful in feature extraction, information extraction

Semantic Role Labeler

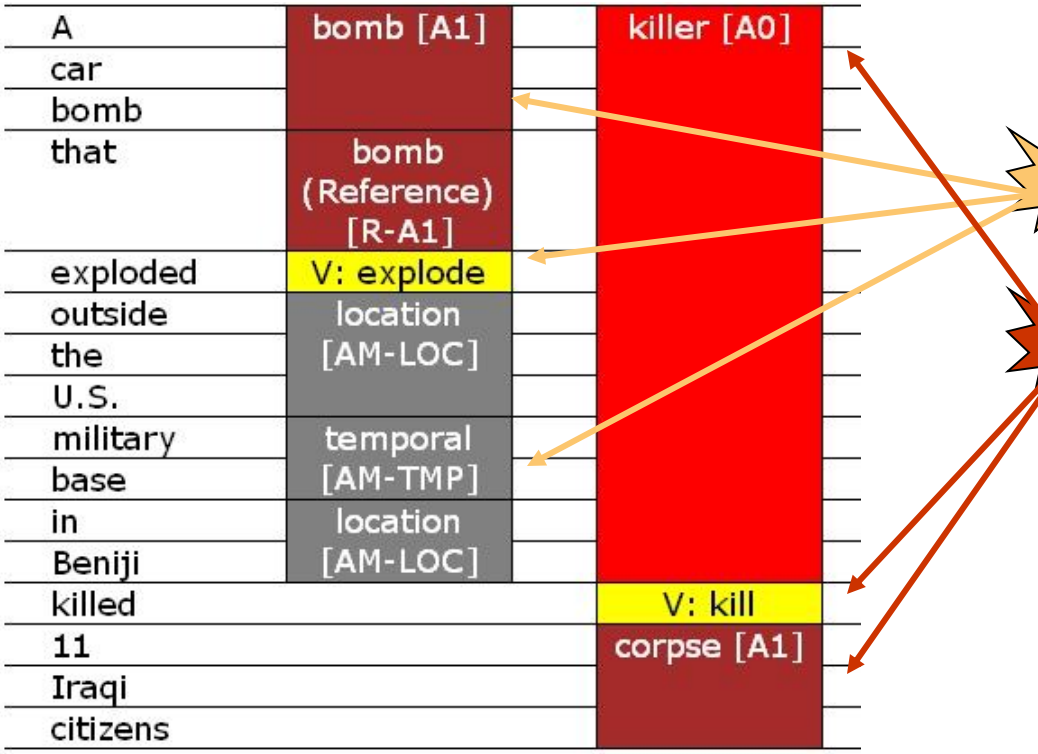
Semantic Role Labeling Output

Input Text:

A car bomb that exploded outside the U.S. military base in Beniji killed 11 Iraqi citizens.

Result: Complete!

⊞ General Explanation of Argument Labels



- SRL reveals relations and arguments in the sentence (where relations are expressed as verbs)
- Cannot abstract over variability of expressing the relations – e.g. kill vs. murder vs. slay…

COMPARATORS

So you want to compare some text….

- **How similar are two words? Two strings? Two paragraphs?**
  - Depends on what they are
  - String edit distance is usually a weak measure
  - … think about coreference resolution…

String 1	String 2	Norm. edit sim.
Shiite	Shi' 'ite	0.667
Mr. Smith	Mrs. Smith	0.900
Wilbur T. Gobsmack	Mr. Gobsmack	0.611
Frigid	Cold	0.167
Wealth	Wreath	0.667
Paris	France	0.167

- **Solution: specialized metrics**

WNSim

- **Generate table mapping terms linked in WordNet ontology**
  - Synonymy, Hypernymy, Meronymy
- **Score reflects distance (up to 3 edges, undirected – e.g. via lowest common subsumer)**
- **Score is symmetric**

String 1	String 2	WNSim distance
Shiite	Shi' 'ite	0
Mr. Smith	Mrs. Smith	0
Wilbur T. Gobsmack	Mr. Gobsmack	0
Frigid	Cold	1
Wealth	Wreath	0
Paris	France	0

Using WNSim (present)

- Install and run the WNSim code (see software page)
  - Sets up an xmlrpc server
  - Expects xmlrpc ‘struct’ data structure (analogous to Dictionary)

```
STRUCT { FIRST_STRING: aString;
         SECOND_STRING anotherString }
```

  - Returns another xmlrpc data structure:

```
STRUCT {  SCORE: aDouble; REASON: aString }
```

- USE: call and cache (reduce network latency overhead)

NESim

- **Set of entity-type-specific measures**
  - Acronyms, Prefix/Title rules, distance metric
- **Score reflects similarity based on type information**
- **Score is asymmetric**

String 1	String 2	Norm. edit distance
Shiite	Shi' 'ite	0.922
Joan Smith	John Smith	0
Wilbur T. Gobsmack	Mr. Gobsmack	0.95
Frigid	Cold	0
Wealth	Wreath	0.900
Paris	France	0.411

Using NESim (present)

- Either: Install and run the WNSim code (…, …)
  - Sets up an xmlrpc server
  - Expects xmlrpc 'struct' data structure (analogous to Dictionary)

```
STRUCT { FIRST_STRING: aString;
         SECOND_STRING anotherString }
```

  - Returns another xmlrpc data structure:

```
STRUCT {  SCORE: aDouble; REASON: aString }
```

- USE: call and cache (reduce network latency overhead)
- OR put jar on classpath, call programmatically
  - CompareNames()

Using NESim (cont'd)

- Strings have optional extra information – type, context

  [<Type>#]<original string>[#<start offset>#<end offset]

- NESim will use specialized resources depending on the type
  - Rules/gazetteers for People's names
  - Acronyms for Organizations
- NESim can use context to help determine similarity

CURATOR

Big NLP

- We introduced a lot of tools, some of them quite sophisticated
- The more complex, the bigger the memory requirement
  - NER: 1G; Coref: 1G; SRL: 4G ….
- If you use tools from different sources, they may be…
  - In different languages
  - Using different data structures
- If you run a lot of experiments on a single corpus, it would be nice to cache the results
  - …and for your colleagues, nice if they can access that cache.
- Curator is our solution to these problems.

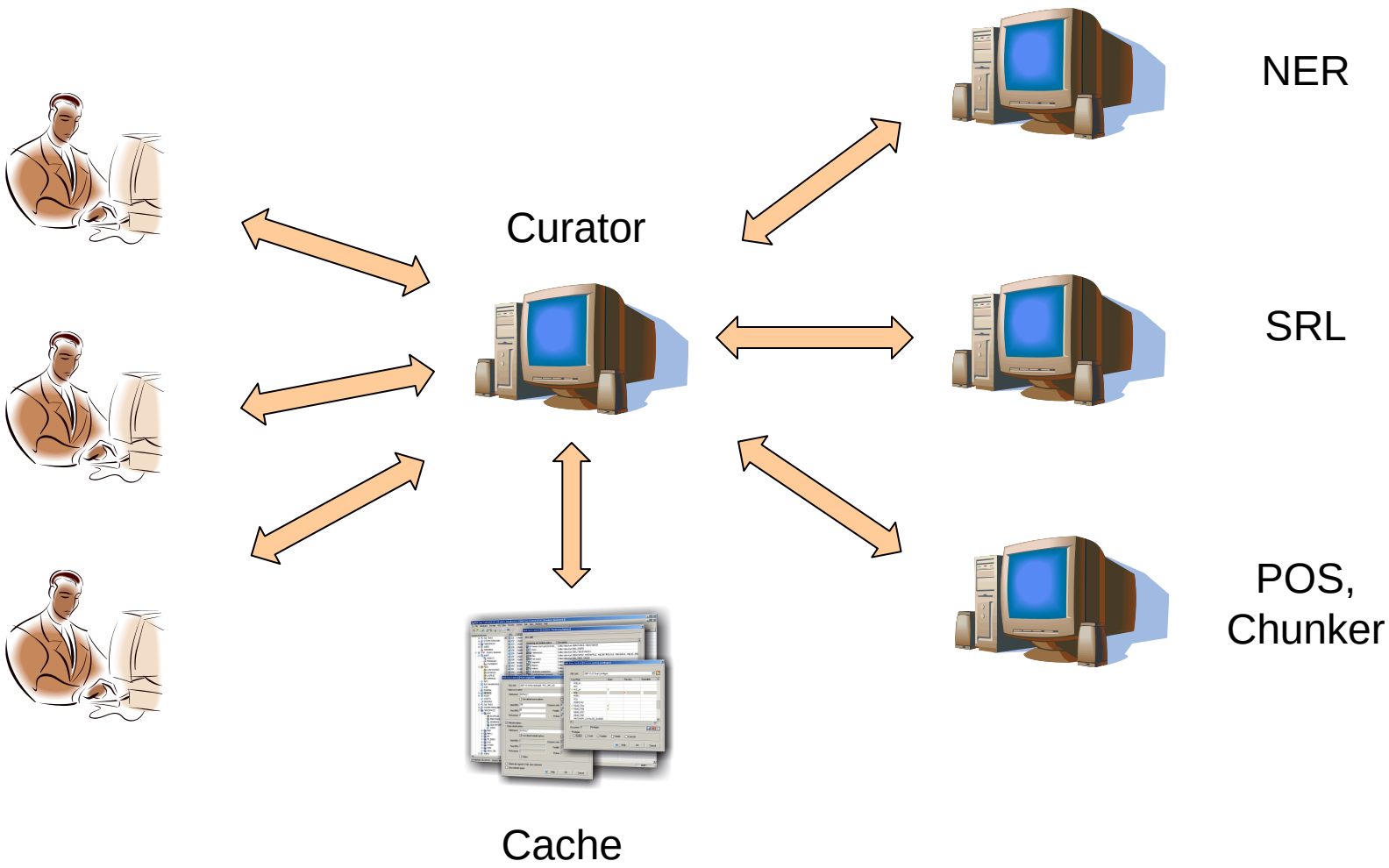
Curator

- Supports distributed NLP resources
  - Central point of contact
  - Single set of interfaces
  - Code generation in many languages (using Thrift)
- Programmatic interface
  - Defines set of common data structures used for interaction
- Caches processed data
- Enables highly configurable NLP pipeline

Overhead:

- Annotation is all at the level of character offsets: Normalization/mapping to token level required
- Need to wrap tools to provide requisite data structures

Curator



Using Curator for Flexible NLP Pipeline

- http://cogcomp.cs.illinois.edu/curator/demo/

- Setting up:
  - Install Curator Server instance
  - Install components (Annotators)
  - Update configuration files

- Use:
  - Use libraries provided: curatorClient.provide() method
  - Access Record field indicated by Component documentation/configuration

Record Data Structure

```
struct Record {
  /** how to identify this record. */
   1: required string identifier,
   2: required string rawText,
   3: required map<string, base.Labeling> labelViews,
   4: required map<string, base.Clustering> clusterViews,
   5: required map<string, base.Forest> parseViews,
   6: required map<string, base.View> views,
   7: required bool whitespaced,
}
```

- rawText contains original text span
- Annotators populate one of the <abc>Views
  - Key is specified in configuration files

Annotator Example: Parser

- Will populate a View, named ‘charniak’
- Curator will expect a Parser interface from the annotator
- Client will expect prerequisites to be provided in other Record fields
  - Specified via Curator server’s annotator configuration file:

```
<annotator>
  <type>parser</type>
  <field>charniak</field>
  <host>mycharniakhost.uiuc.edu:8087</host>
  <requirements>sentences:tokens:pos</requirements>
</annotator>
```