State-of-the-art Parsing
State-of-the-art Parsers

- Unlabeled attachment score: fraction of words with correct parent

- Labeled attachment score: have to label each edge correctly (but this isn’t that hard — noun before verb -> nsubj in most contexts)

- 2005: Eisner algorithm graph-based parser was SOTA (~91 UAS)

- 2010: Better graph-based parsers using “parent annotation” (~93 UAS)

- 2012: Transition-based Maltparser achieved good results (~90 UAS)

- 2014: Stanford neural dependency parser (Chen and Manning) got 92 UAS with transition-based neural model

- 2016: Improvements to Chen and Manning
Stanford Dependency Parser

- Feedforward neural network on top of feature vector extracted from stack and buffer

1st in stack  2nd in stack  1st in buf  ...  POS of leftmost child of 1st in stack  ...

Stack
- ROOT has_VBZ good JJ
- He_PRP nsubj

Buffer
- control_NN ...

Chen and Manning (2014)
Stanford Dependency Parser

**Softmax layer:**
\[ p = \text{softmax}(W_2h) \]

**Hidden layer:**
\[ h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3 \]

**Input layer:** \([x^w, x^t, x^l] \]

---

Configuration

**Stack**
- ROOT
- has_VBZ
- good_JJ
- He_PRP
- nsubj

**Buffer**
- control_NN
- ...

Chen and Manning (2014)
<table>
<thead>
<tr>
<th>Parser</th>
<th>Dev UAS</th>
<th>Dev LAS</th>
<th>Test UAS</th>
<th>Test LAS</th>
<th>Speed (sent/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>90.2</td>
<td>87.8</td>
<td>89.4</td>
<td>87.3</td>
<td>26</td>
</tr>
<tr>
<td>eager</td>
<td>89.8</td>
<td>87.4</td>
<td>89.6</td>
<td>87.4</td>
<td>34</td>
</tr>
<tr>
<td>Malt:sp</td>
<td>89.8</td>
<td>87.2</td>
<td>89.3</td>
<td>86.9</td>
<td>469</td>
</tr>
<tr>
<td>Malt:eager</td>
<td>89.6</td>
<td>86.9</td>
<td>89.4</td>
<td>86.8</td>
<td>448</td>
</tr>
<tr>
<td>MSTParser</td>
<td>91.4</td>
<td>88.1</td>
<td>90.7</td>
<td>87.6</td>
<td>10</td>
</tr>
<tr>
<td>Our parser</td>
<td>92.0</td>
<td>89.7</td>
<td>91.8</td>
<td>89.6</td>
<td>654</td>
</tr>
</tbody>
</table>

- MSTParser: “graph-based” parser (like CKY) from 2005 — so Chen+Manning’s parser isn’t much better but is much faster!
Parsey McParseFace (a.k.a. SyntaxNet)

- Close to state-of-the-art, released by Google publicly
- 94.61 UAS on the Penn Treebank using a transition-based system
  - Additional data harvested via “tri-training”, form of self-training
- Same feature set as Chen and Manning (2014), Google fine-tuned it

https://github.com/tensorflow/models/tree/master/research/syntaxnet

Andor et al. (2016)
AllenNLP

- Reimplementation of graph-based, state-of-the-art parser
- Some fancy tricks we haven’t discussed yet
- Very nice and usable web demo

https://demo.allennlp.org/dependency-parsing
Other languages

- Annotate dependencies with the same representation in many languages

English

1. The dog was chased by the cat

Bulgarian

2. Кучето се преследваше от котката

Czech

3. Pes byl honěn kočkou

Swiss

4. Hunden jagades av katten

http://universaldependencies.org/
Semantic Role Labeling
Performing event

- Subject: Lady Gaga
- Object: a concert
- Audience: students

Same event described but the representation looks different
VerbNet

<table>
<thead>
<tr>
<th>Frames</th>
<th>NP V NP</th>
<th>NP V</th>
<th>NP V NP PP,beneficiary</th>
</tr>
</thead>
</table>
| EXAMPLE | "Sandy sang a song."
| SYNTAX  | Agent V Theme |
| SEMANTICS | perform(during(E), Agent, Theme) |
| EXAMPLE | "Sandy sang."
| SYNTAX  | Agent V |
| SEMANTICS | perform(during(E), Agent, ?Theme) |
| EXAMPLE | "Sandy sang a song for me."
| SYNTAX  | Agent V Theme {for} Beneficiary |
| SEMANTICS | perform(during(E), Agent, Theme) benefit(E, Beneficiary) |

- Defines the semantics of verbs, arguments for every verb in English
Semantic Roles

- “Postprocessing” layer on top of dependency parsing that exposes useful information, canonicalizes across grammatical constructions

- Related to theta roles in linguistics

- Agent (~ subject), patient/theme (~ object), goal (~ indirect object)
  
  \[
  \text{ARG0} \quad \text{ARG1} \quad \text{ARG2+ (semantics vary)}
  \]
Semantic Role Labeling

- Identify predicate, disambiguate it, identify that predicate’s arguments
- Verb roles from Propbank (Palmer et al., 2005)

Quicken:

**Arg0-PAG:** cause of speed-up
**Arg1-PPT:** thing becoming faster (vnrole: 45.4-patient)
**Arg2-EXT:** EXT
**Arg3-DIR:** old speed
**Arg4-PRD:** new speed

Figure from He et al. (2017)
Question and several answer candidates

Q: Who discovered prions?

AC1: In 1997, Stanley B. Prusiner, a scientist in the United States, discovered prions...

AC2: Prions were researched by...

Score by matching expected answer phrase (EAP) against answer candidate (AC)

Shen and Lapata (2007)
More on SRL

- Even complex neural network models for SRL benefit from dependency information

- Emma Strubell from UMass Amherst: “Neural Network Architectures for Fast and Robust NLP”
  - Tuesday, 11am GDC main auditorium
  - Includes discussion of work on neural SRL system
Relation Extraction
Tim Cook is the CEO of Apple.

Apple CEO Tim Cook said that...

Apple shares have taken a beating, much to the chagrin of its CEO, Tim Cook.

Cook’s tenure as CEO of Apple ...

Wozniak’s desire to be CEO ...
Relation Extraction

- Extract entity-relation-entity triples from a fixed inventory
  
  
  
  Located_In

  Nationality

  During the war in [Iraq], [American journalists] were sometimes caught in the line of fire

- Use NER-like system to identify entity spans, classify relations between entity pairs with a classifier

- Systems can be feature-based or neural, look at surface words, dependency path features, semantic roles

- Problem: limited data for scaling to big ontologies

ACE (2003-2005)
Lots of relations in our knowledge base already (e.g., 23,000 film-director relations); use these to bootstrap more training data.

If two entities in a relation appear in the same sentence, assume the sentence expresses the relation:

- Director

**[Steven Spielberg]'s film [Saving Private Ryan] is loosely based on the brothers’ story**

- **Allison** co-produced the Academy Award-winning [Saving Private Ryan], directed by [Steven Spielberg]
Distant Supervision

- Learn decently accurate classifiers for ~100 Freebase relations
- Could be used to crawl the web and expand our knowledge base

<table>
<thead>
<tr>
<th>Relation name</th>
<th>100 instances</th>
<th></th>
<th></th>
<th>1000 instances</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Syn</td>
<td>Lex</td>
<td>Both</td>
<td>Syn</td>
<td>Lex</td>
<td>Both</td>
</tr>
<tr>
<td>/film/director/film</td>
<td>0.49</td>
<td>0.43</td>
<td>0.44</td>
<td>0.49</td>
<td>0.41</td>
<td>0.46</td>
</tr>
<tr>
<td>/film/writer/film</td>
<td>0.70</td>
<td>0.60</td>
<td>0.65</td>
<td>0.71</td>
<td>0.61</td>
<td>0.69</td>
</tr>
<tr>
<td>/geography/river/basin_countries</td>
<td>0.65</td>
<td>0.64</td>
<td>0.67</td>
<td>0.73</td>
<td>0.71</td>
<td>0.64</td>
</tr>
<tr>
<td>/location/country/administrative_divisions</td>
<td>0.68</td>
<td>0.59</td>
<td>0.70</td>
<td>0.72</td>
<td>0.68</td>
<td>0.72</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>0.81</td>
<td>0.89</td>
<td>0.84</td>
<td>0.85</td>
<td>0.83</td>
<td>0.84</td>
</tr>
<tr>
<td>/location/us_county/country_seat</td>
<td>0.51</td>
<td>0.51</td>
<td>0.53</td>
<td>0.47</td>
<td>0.57</td>
<td>0.42</td>
</tr>
<tr>
<td>/music/artist/origin</td>
<td>0.64</td>
<td>0.66</td>
<td>0.71</td>
<td>0.61</td>
<td>0.63</td>
<td>0.60</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>0.80</td>
<td>0.79</td>
<td>0.81</td>
<td>0.80</td>
<td>0.81</td>
<td>0.78</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>0.61</td>
<td>0.70</td>
<td>0.72</td>
<td>0.56</td>
<td>0.61</td>
<td>0.63</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>0.78</td>
<td>0.77</td>
<td>0.78</td>
<td>0.88</td>
<td>0.85</td>
<td>0.91</td>
</tr>
<tr>
<td>Average</td>
<td>0.67</td>
<td>0.66</td>
<td>0.69</td>
<td>0.68</td>
<td>0.67</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Mintz et al. (2009)
Open IE
Open Information Extraction

- “Open”ness — want to be able to extract all kinds of information from open-domain text

- Acquire commonsense knowledge just from “reading” about it, but need to process lots of text (“machine reading”)

- Typically no fixed relation inventory
Extract positive examples of (e, r, e) triples via parsing and heuristics

Train a Naive Bayes classifier to filter triples from raw text: uses features on POS tags, lexical features, stopwords, etc.

Barack Obama, 44th president of the United States, was born on August 4, 1961 in Honolulu

=> Barack Obama, was born in, Honolulu

80x faster than running a parser (which was slow in 2007...)

Use multiple instances of extractions to assign probability to a relation

Banko et al. (2007)
Exploiting Redundancy

- 9M web pages / 133M sentences
- 2.2 tuples extracted per sentence, filter based on probabilities
- Concrete: definitely true
  Abstract: possibly true but underspecified
- Hard to evaluate: can assess precision of extracted facts, but how do we know recall?

Banko et al. (2007)
More constraints: open relations have to begin with verb, end with preposition, be contiguous (e.g., was born on)

Extract more meaningful relations, particularly with light verbs

<table>
<thead>
<tr>
<th>is</th>
<th>is an album by, is the author of, is a city in</th>
</tr>
</thead>
<tbody>
<tr>
<td>has</td>
<td>has a population of, has a Ph.D. in, has a cameo in</td>
</tr>
<tr>
<td>made</td>
<td>made a deal with, made a promise to</td>
</tr>
<tr>
<td>took</td>
<td>took place in, took control over, took advantage of</td>
</tr>
<tr>
<td>gave</td>
<td>gave birth to, gave a talk at, gave new meaning to</td>
</tr>
<tr>
<td>got</td>
<td>got tickets to, got a deal on, got funding from</td>
</tr>
</tbody>
</table>
For each verb, identify the longest sequence of words following the verb that satisfy a POS regex (V.*P) and which satisfy heuristic lexical constraints on specificity

Find the nearest arguments on either side of the relation

Annotators labeled relations in 500 documents to assess recall

Fader et al. (2011)
(a) **CCG parse** builds an underspecified semantic representation of the sentence.

\[
\begin{array}{c}
\begin{array}{c}
N/N \\
\lambda f \lambda x. f(x) \land former(x) \\
\lambda x \cdot municipalities(x)
\end{array}
\end{array} \quad \begin{array}{c}
\begin{array}{c}
N \\
\lambda x \cdot municipalities(x)
\end{array}
\end{array} \quad \begin{array}{c}
\begin{array}{c}
N \backslash N/NP \\
\lambda f \lambda x \lambda y. f(y) \land in(y, x) \\
\text{Brandenburg}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\lambda x \cdot former(x) \land municipalities(x)
\end{array}
\end{array} \quad \begin{array}{c}
\begin{array}{c}
\lambda f \lambda x \cdot municipalities(x)
\end{array}
\end{array} \quad \begin{array}{c}
\begin{array}{c}
N \backslash N \\
\lambda f \lambda y. f(y) \land in(y, Brandenburg)
\end{array}
\end{array}
\]

\[
l_0 = \lambda x \cdot former(x) \land municipalities(x) \land in(x, Brandenburg)
\]

(b) **Constant matches** replace underspecified constants with Freebase concepts

\[
l_0 = \lambda x \cdot former(x) \land municipalities(x) \land in(x, Brandenburg)
\]

\[
l_1 = \lambda x \cdot former(x) \land municipalities(x) \land in(x, Brandenburg)
\]

\[
l_2 = \lambda x \cdot former(x) \land municipalities(x) \land location.containedby(x, Brandenburg)
\]

\[
l_3 = \lambda x \cdot former(x) \land OpenRel(x, Municipality) \land location.containedby(x, Brandenburg)
\]

\[
l_4 = \lambda x \cdot OpenType(x) \land \boxed{OpenRel(x, Municipality)} \land location.containedby(x, Brandenburg)
\]

Choi et al. (2015)
Takeaways

- SRL/AMR: handle a bunch of phenomena, but more or less like syntax++ in terms of what they represent
- Relation extraction: can collect data with distant supervision, use this to expand knowledge bases
- Open IE: extracts lots of things, but hard to know how good or useful they are
  - Can combine with standard question answering
  - Add new facts to knowledge bases
- Many, many applications and techniques
Roadmap
Roadmap

- Classification: conventional and neural, word representations (3 weeks)
  - Linear and neural classification
  - How to build effective word vectors
- Text analysis: tagging, parsing, information extraction (3.5 weeks)
  - Structured models for sequences, trees (HMMs, PCFGs), as well as unstructured approaches (transition-based parsing)
  - Lots of NLP tasks can be formulated as tagging
Applications of Tagging

- Extract product occurrences in cybercrime forums, but not everything that looks like a product is a product

<table>
<thead>
<tr>
<th>TITLE: [ buy ] Backconnect bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODY: Looking for a solid backconnect bot . If you know of anyone who codes them please let me know</td>
</tr>
</tbody>
</table>

(a) File 0-initiator4856

<table>
<thead>
<tr>
<th>TITLE: Exploit cleaning ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODY: Have some Exploits i need fud .</td>
</tr>
</tbody>
</table>

(b) File 0-initiator10815

Not a product in this context

Portnoff et al. (2017), Durrett et al. (2017)
Roadmap

- Classification: conventional and neural, word representations (3 weeks)
  - Linear and neural classification
  - How to build effective word vectors
- Text analysis: tagging, parsing, information extraction (3.5 weeks)
  - Structured models for sequences, trees (HMMs, PCFGs), as well as unstructured approaches (transition-based parsing)
  - Missing: structured neural models. These are a bit beyond this class but we’ll see one way to do this after spring break
- Generation, applications: language modeling, machine translation, dialogue (4 weeks)
- Other applications: question answering, TBD (3 weeks)