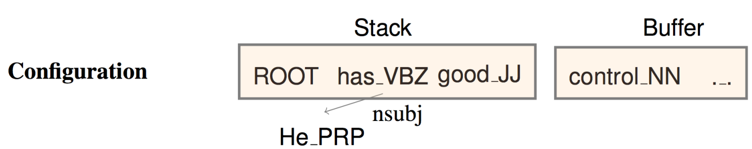
State-of-the-art Parsing

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



Chen and Manning (2014)

Stanford Dependency Parser

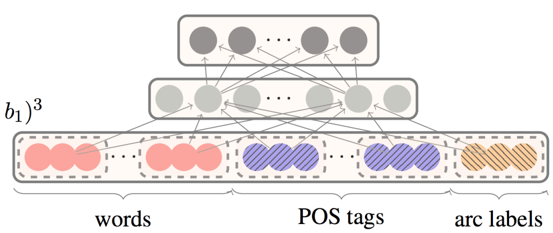
Softmax layer:

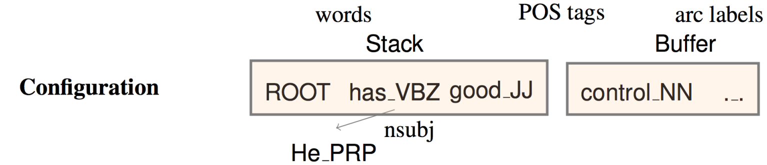
$p = \texttt{softmax}(W_2 h)$

Hidden layer:

$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$

Input layer: $[x^w, x^t, x^l]$





Chen and Manning (2014)

Stanford Dependency Parser

Parser	Dev		Test		Speed
	UAS	LAS	UAS	LAS	(sent/s)
standard	90.2	87.8	89.4	87.3	26
eager	89.8	87.4	89.6	87.4	34
Malt:sp	89.8	87.2	89.3	86.9	469
Malt:eager	89.6	86.9	89.4	86.8	448
MSTParser	91.4	88.1	90.7	87.6	10
Our parser	**92.0**	**89.7**	**91.8**	**89.6**	**654**

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

Chen and Manning (2014)

Parsey McParseFace (a.k.a. SyntaxNet)

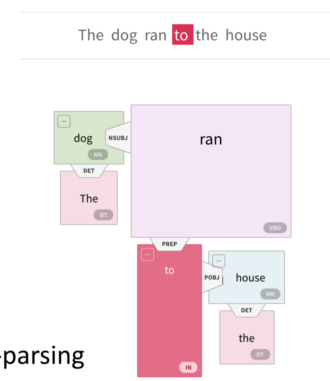
- Close to state-of-the-art, released by Google publicly
- 94.61 UAS on the Penn Treebank using a transition-based system
  - Additional data harvested via “tri-training”, form of self-training
- Same feature set as Chen and Manning (2014), Google fine-tuned it

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

Andor et al. (2016)

AllenNLP

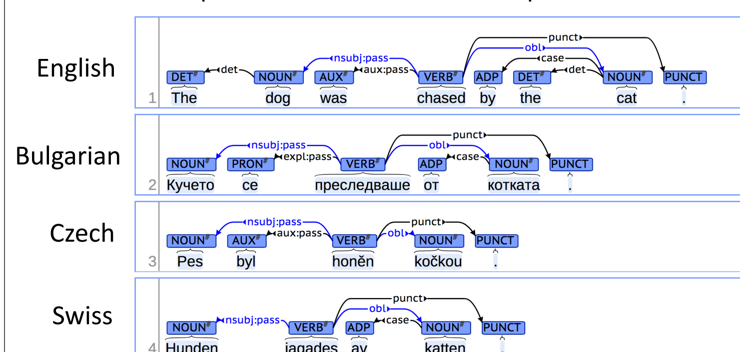
- Reimplementation of graph-based, state-of-the-art parser
- Some fancy tricks we haven’t discussed yet
- Very nice and usable web demo



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

Other languages

- Annotate dependencies with the same representation in many languages

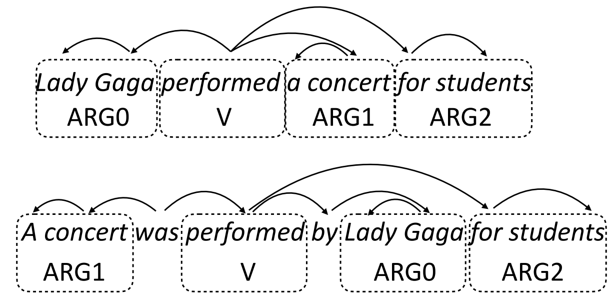


http://universaldependencies.org/

Semantic Role Labeling

Semantic Role Labeling

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



- Same event described but the representation looks different

VerbNet

percentile.n	(Grouping)
perception.n	(Grouping)
perch	...n Placing), (Grouping)
percolate	
perfect	
perforate	
perform	
performance.n	
perfume	
perfuse	
peril.n	

Frames

NP V NP
- EXAMPLE "Sandy sang a song."
- SYNTAX Agent V Theme
- SEMANTICS perform(during(E), Agent, Theme)

NP V
- EXAMPLE "Sandy sang."
- SYNTAX Agent V
- SEMANTICS perform(during(E), Agent, ?Theme)

NP V NP PP.beneficiary
- 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

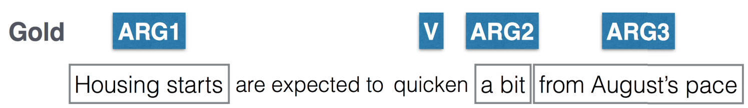
verbs.colorado.edu

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)
  ARG0 ARG1 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: *causer 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)

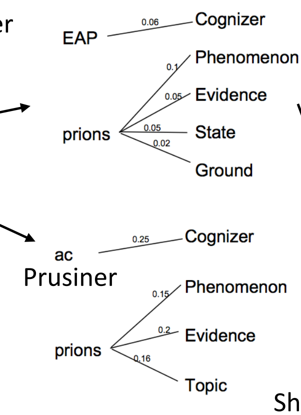
SRL for QA

- 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

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)

Distant Supervision

- 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]

Director

Mintz et al. (2009)

Distant Supervision

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

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

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

TextRunner

- 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

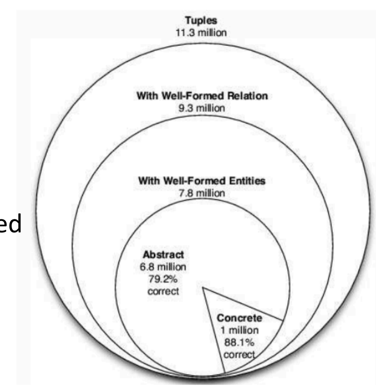
=> Barack_Obama, was born in, Honolulu

- 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)

ReVerb

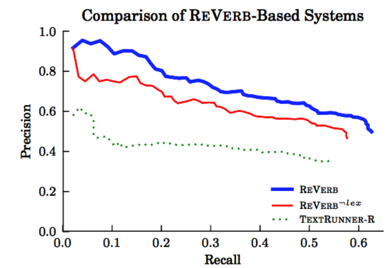
- More constraints: open relations have to begin with verb, end with preposition, be contiguous (e.g., *was born on*)
- Extract more meaningful relations, particularly with light verbs

is	is an album by, is the author of, is a city in
has	has a population of, has a Ph.D. in, has a cameo in
made	made a deal with, made a promise to
took	took place in, took control over, took advantage of
gave	gave birth to, gave a talk at, gave new meaning to
got	got tickets to, got a deal on, got funding from

Fader et al. (2011)

ReVerb

- 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)

QA from Open IE

(a) **CCG parse** builds an underspecified semantic representation of the sentence.

Former	municipalities	in	Brandenburgh
N/N	N	$N\backslash N/NP$	NP
$\lambda f \lambda x.f(x) \wedge former(x)$	$\lambda x.municipalities(x)$	$\lambda f \lambda x \lambda y.f(y) \wedge in(y,x)$	$Brandenburg$

$$N: \lambda x.former(x) \wedge municipalities(x) \qquad N\backslash N: \lambda f \lambda y.f(y) \wedge in(y, Brandenburg)$$

$$N: l_0 = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, Brandenburg)$$

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

$\mathbf{l}_0 = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, Brandenburg)$

$\mathbf{l}_1 = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, \mathtt{Brandenburg})$

$\mathbf{l}_2 = \lambda x.former(x) \wedge municipalities(x) \wedge \mathtt{location.containedby}(x, \mathtt{Brandenburg})$

$\mathbf{l}_3 = \lambda x.former(x) \wedge \mathtt{OpenRel}(x, \mathtt{Municipality}) \wedge \mathtt{location.containedby}(x, \mathtt{Brandenburg})$

$\mathbf{l}_4 = \lambda x.\mathtt{OpenType}(x) \wedge \boxed{\mathtt{OpenRel}(x, \mathtt{Municipality})} \wedge \mathtt{location.containedby}(x, \mathtt{Brandenburg})$

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

TITLE: [buy] Backconnect bot
BODY: Looking for a solid backconnect bot .
If you know of anyone who codes them please let me know

(a) File 0-initiator4856

TITLE: Exploit cleaning ?
BODY: Have some Exploits i need fud .

(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)