Information Extraction

Wei Xu

(many slides from Greg Durrett, Luheng He, Emma Strubell)
This Lecture

- How do we represent information for information extraction?
- Semantic role labeling / abstract meaning representation
- Relation extraction
- Slot filling
- Open Information Extraction
IE: The Big Picture

- How do we represent information? What do we extract?
  - Semantic roles
  - Abstract meaning representation
  - Slot fillers
  - Entity-relation-entity triples (fixed ontology or open)
Semantic Role Labeling
Semantic Role Labeling

- Find out 5W in text — “who did what to whom, when and where”
- Identify predicate, disambiguate it, identify that predicate’s arguments

Figure from He et al. (2017)
Semantic Role Labeling

The robot *broke* my favorite mug with a wrench.

My mug *broke* into pieces immediately.

Figure from He et al. (2017)
The robot *broke* my favorite mug with a wrench.

My mug *broke* into pieces immediately.
The robot *broke* my favorite mug with a wrench.

My mug *broke* into pieces immediately.

Figure from He et al. (2017)
The robot *broke* my favorite mug with a wrench.

My mug *broke* into pieces immediately.

Frame: *break.01*

<table>
<thead>
<tr>
<th>role</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG0</td>
<td>breaker</td>
</tr>
<tr>
<td>ARG1</td>
<td>thing broken</td>
</tr>
<tr>
<td>ARG2</td>
<td>instrument</td>
</tr>
<tr>
<td>ARG3</td>
<td>pieces (final state)</td>
</tr>
<tr>
<td>ARG4</td>
<td>broken away from what?</td>
</tr>
</tbody>
</table>
The Proposition Bank (PropBank)

Core roles:
Verb-specific roles (ARG0-ARG5) defined in frame files

Frame: break.01

<table>
<thead>
<tr>
<th>role</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG0</td>
<td>breaker</td>
</tr>
<tr>
<td>ARG1</td>
<td>thing broken</td>
</tr>
<tr>
<td>ARG2</td>
<td>instrument</td>
</tr>
</tbody>
</table>

Frame: buy.01

<table>
<thead>
<tr>
<th>role</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG0</td>
<td>buyer</td>
</tr>
<tr>
<td>ARG1</td>
<td>thing bought</td>
</tr>
<tr>
<td>ARG2</td>
<td>seller</td>
</tr>
<tr>
<td>ARG3</td>
<td>price paid</td>
</tr>
<tr>
<td>ARG4</td>
<td>benefactive</td>
</tr>
</tbody>
</table>

Adjunct roles:
(ARGM-) shared across verbs

<table>
<thead>
<tr>
<th>role</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP</td>
<td>temporal</td>
</tr>
<tr>
<td>LOC</td>
<td>location</td>
</tr>
<tr>
<td>MNR</td>
<td>manner</td>
</tr>
<tr>
<td>DIR</td>
<td>direction</td>
</tr>
<tr>
<td>CAU</td>
<td>cause</td>
</tr>
<tr>
<td>PRP</td>
<td>purpose</td>
</tr>
</tbody>
</table>

Annotated on top of the Penn Treebank Syntax

Figure from He et al. (2017)
Syntax vs. Semantics

Figure 1.2: Syntax and semantics are closely related. The phrase-syntactic tree is shown in brown above the sentence. Semantic role labeling (SRL) structures from PropBank (Palmer et al., 2005) are shown alongside, in green, blue and magenta. Under SRL, words in the sentence that indicate stand-alone events are selected as predicates. These are shown as highlighted leaf nodes—“encouraging”, “told” and “left”. Each predicate is disambiguated to its relevant sense shown above it. Arguments to the predicates are are annotated on top of syntactic nodes, with the role labels color-coded by the predicate. SRL substructures (predicates, arguments) thus fully overlap with phrase-syntactic nodes.

Figure from Swayamdipta (2019)
In 1950 Alan M. Turing published "Computing machinery and intelligence" in Mind, in which he proposed that machines could be tested for intelligence using questions and answers.

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>published</td>
<td>Who published something?</td>
<td>Alan M. Turing</td>
</tr>
<tr>
<td></td>
<td>What was published?</td>
<td>&quot;Computing Machinery and Intelligence&quot;</td>
</tr>
<tr>
<td></td>
<td>When was something published?</td>
<td>In 1950</td>
</tr>
<tr>
<td>proposed</td>
<td>Who proposed something?</td>
<td>Alan M. Turing</td>
</tr>
<tr>
<td></td>
<td>What did someone propose?</td>
<td>that machines could be tested for intelligent using questions and answers</td>
</tr>
<tr>
<td></td>
<td>When did someone propose something?</td>
<td>In 1950</td>
</tr>
<tr>
<td>tested</td>
<td>What can be tested?</td>
<td>machines</td>
</tr>
<tr>
<td></td>
<td>What can something be tested for?</td>
<td>intelligence</td>
</tr>
<tr>
<td></td>
<td>How can something be tested?</td>
<td>using questions and answers</td>
</tr>
<tr>
<td>using</td>
<td>What was being used?</td>
<td>questions and answers</td>
</tr>
<tr>
<td></td>
<td>Why was something being used?</td>
<td>tested for intelligence</td>
</tr>
</tbody>
</table>

Figure from FitzGerald et al. (2018)
SRL Systems

(syntax-based)

Pipeline Systems

sentence, predicate
syntactic features
argument id.
candidate argument spans
labeling
labeled arguments
ILP/DP
prediction

End-to-end Systems

sentence, predicate
context window features
Deep BiLSTM + CRF layer
BIO sequence
Viterbi
prediction

He et al. (2017)

sentence, predicate
Deep BiLSTM
BIO sequence
Hard constraints
prediction

Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

Collobert et al., 2011
Zhou and Xu, 2015
Wang et al., 2015

He et al., 2017

Figure from He et al. (2017)
Identify predicates (*love*) using a classifier (not shown)

Identify ARG0, ARG1, etc. as a tagging task with a BiLSTM conditioned on *love*

Other systems incorporate syntax, joint predicate-argument finding
Residual / Highway Connections

Non-linearity
\[ F(c_{t-1}, h_{t-1}, x_t) \]

output to the next layer
\[ h_t \]

shortcut \[ x_t \]

new output:
\[ h_t + x_t \]

residual net

\[ r_t \circ h_t + (1 - r_t) \circ x_t \]
gated highway network:
\[ r_t = \sigma(f(h_{t-1}, x_t)) \]

References:
Deep Residual Networks, Kaiming He, ICML 2016 Tutorial
Training Very Deep Networks, Srivastava et al., 2015

Figure from He et al. (2017)
10 Years of PropBank SRL

Figure from Strubell et al. (2018)
Can we combine the two approaches — incorporate syntactic information into neural networks?

Multi-task learning with related tasks, e.g., part-of-speech tagging, dependency parsing ...

Syntactically-informed self-attention: use the Transformer to encore the sentence; in one head, token attends to its likely syntactic parents; in next layer, tokens observe all other parents.
Recall: Transformer (multi-head self-attention)
Recall: Transformer (multi-head self-attention)

Figure from Strubell et al. (2018)
Syntactically-Informed Self-Attention

[Dozat and Manning 2017]
Linguistically-Informed Self-Attention

Figure from Strubell et al. (2018)
Linguistically-Informed Self-Attention
Figure from Strubell et al. (2018)
Linguistically-Informed Self-Attention

Figure from Strubell et al. (2018)
Layer J

Multi-head self-attention + feed forward

Layer p

Syntactically-informed self-attention

Layer r

Multi-head self-attention + feed forward

Layer 1

Multi-head self-attention + feed forward

Nobel committee awards Strickland who advanced optics

Figure from Strubell et al. (2018)
Linguistically-Informed Self-Attention

Figure from Strubell et al. (2018)
Linguistically-Informed Self-Attention

Figure from Strubell et al. (2018)
Linguistically-Informed Self-Attention

Bilinear

Layer J
Multi-head self-attention + feed forward

Layer p
Syntactically-informed self-attention

Layer r
Multi-head self-attention + feed forward

Nobel committee awards Strickland who advanced optics
Linguistically-Informed Self-Attention

Layer J
Multi-head self-attention + feed forward

Layer p
Syntactically-informed self-attention

Layer r
Multi-head self-attention + feed forward

Nobel committee awards Strickland who advanced optics
Linguistically-Informed Self-Attention

Layer r
Multi-head self-attention + feed forward

Layer p
Syntactically-informed self-attention

Layer J
Multi-head self-attention + feed forward

Bilinear predicates args

Nobel committee awards Strickland who advanced optics
Linguistically-Informed Self-Attention

Layer $r$: Multi-head self-attention + feed forward

Layer $p$: Syntactically-informed self-attention

Layer $J$: Multi-head self-attention + feed forward

Bilinear

predicates

args

Nobel committee awards Strickland who advanced optics
Linguistically-Informed Self-Attention

B-ARG₀  I-ARG₀  B-V  B-ARG₁  O  O  O  O

Bilinear predicates args

Layer J

Multi-head self-attention + feed forward

Layer p

Syntactically-informed self-attention

Layer r

Multi-head self-attention + feed forward

Nobel committee awards Strickland who advanced optics
Linguistically-Informed Self-Attention

Bilinear

predicates

args

Layer J

Multi-head self-attention + feed forward

Layer p

Syntactically-informed self-attention

Layer r

Multi-head self-attention + feed forward

Nobel committee awards Strickland who advanced optics
Linguistically-Informed Self-Attention

Layer r
Multi-head self-attention + feed forward

Layer p
Syntactically-informed self-attention

Layer J
Multi-head self-attention + feed forward

Bilinear

predicates

args

B-ARG0
O

I-ARG0
O

B-V
O

B-ARG1
B-ARG0

O
B-R-ARG0

O
B-V

B-ARG1

SRL

labeled
parse

preds

pos

NNP

NN

VBZ/PRED

NNP

WP

VBN/PRED

NN

Nobel

committee

awards

Strickland

who

advanced

optics
## Linguistically-Informed Self-Attention

<table>
<thead>
<tr>
<th></th>
<th>GloVe in-domain (dev)</th>
<th>ELMo in-domain (dev)</th>
</tr>
</thead>
<tbody>
<tr>
<td>He et al. 2017</td>
<td>81.5</td>
<td>---</td>
</tr>
<tr>
<td>He et al. 2018</td>
<td>81.6</td>
<td>85.3</td>
</tr>
<tr>
<td>SA</td>
<td>82.39</td>
<td>85.26</td>
</tr>
<tr>
<td><strong>LISA</strong></td>
<td><strong>82.24</strong></td>
<td><strong>85.35</strong></td>
</tr>
<tr>
<td>+D&amp;M</td>
<td>83.58</td>
<td>85.17</td>
</tr>
<tr>
<td>+Gold</td>
<td>86.81</td>
<td>87.63</td>
</tr>
</tbody>
</table>

Strubell et al. (2018)
Why SRL is difficult? or NLP in general

- Syntactic Alternation

The robot *plays* piano.

ARG0: player

ARG2: instrument

The cafe is *playing* my favorite song.

ARG0: player

ARG1: thing performed

ARG1: thing performed

ARGM-MNR

The music *plays* softly.
Why SRL is difficult? or NLP in general

- Prepositional Phrase (PP) Attachment

  I  eat  [pasta]  [with delight].

<table>
<thead>
<tr>
<th>ARG0</th>
<th>ARG1</th>
<th>ARG-MNRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>eater</td>
<td>meal</td>
<td>manner</td>
</tr>
</tbody>
</table>

  I  eat  [pasta with broccoli].

<table>
<thead>
<tr>
<th>ARG0</th>
<th>ARG1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eater</td>
<td>meal</td>
</tr>
</tbody>
</table>
Why SRL is difficult? or NLP in general

- Long Dependencies

We flew to Chicago.

ARG1: passenger
ARGM-GOL: goal

We remember the nice view flying to Chicago.

ARG1: passenger
ARGM-GOL: goal

We remember John and Mary flying to Chicago.

ARG1: passenger
ARGM-GOL: goal

Syntactic Alternation
PP Attachment
Long-range Dependencies

‣ Long Dependencies

Why SRL is difficult? or NLP in general
Why SRL is difficult? or NLP in general

- Even harder for out-of-domain data

“Dip chicken breasts into eggs to coat”

Active, Ser133-phosphorylated CREB effects transcription of CRE-dependent genes via interaction with the 265-kDa …
Abstract Meaning Representation
Abstract Meaning Representation

- Graph-structured annotation
- Nodes are variables labeled by concepts; edges are semantic relations

"Bob ate four cakes that he bought."

(x2 / eat-01
 :ARG0 (x1 / person
 :name (n / name
 :op1 "Bob")
 :wiki "Bob_X")
 :ARG1 (x4 / cake
 :quant 4
 :ARG1-of (x7 / buy-01
 :ARG0 x1)))

Figure from Gruzitis et al. (2014)
Abstract Meaning Representation

- Superset of SRL: full sentence analyses, contains coreference and multi-word expressions as well
- F1 scores in the 60s: hard!
- So comprehensive that it’s hard to predict, but still doesn’t handle tense or some other things...

Banarescu et al. (2014)
Summarization with AMR

- Merge AMRs across multiple sentences
- Summarization = subgraph extraction

Sentence A: I saw Joe’s dog, which was running in the garden.
Sentence B: The dog was chasing a cat.

Summary: Joe’s dog was chasing a cat in the garden.

Liu et al. (2015)
Relation Extraction
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, syntactic features (dependency paths), semantic roles
- Problem: limited data for scaling to big ontologies

ACE (2003-2005)
Syntactic patterns especially for finding hypernym-hyponym pairs ("is a" relations)

- $Y$ is a $X$  
  - Berlin is a city

- $X$ such as [list]  
  - cities such as Berlin, Paris, and London.

- other $X$ including $Y$  
  - other cities including Berlin

- Totally unsupervised way of harvesting world knowledge for tasks like parsing and coreference (Bansal and Klein, 2011-2012)

Hearst (1992)
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

[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>1000 instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Syn</td>
<td>Lex</td>
</tr>
<tr>
<td>/film/director/film</td>
<td>0.49</td>
<td>0.43</td>
</tr>
<tr>
<td>/film/writer/film</td>
<td>0.70</td>
<td>0.60</td>
</tr>
<tr>
<td>/geography/river/basin_countries</td>
<td>0.65</td>
<td>0.64</td>
</tr>
<tr>
<td>/location/country/administrative.divisions</td>
<td>0.68</td>
<td>0.59</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>0.81</td>
<td>0.89</td>
</tr>
<tr>
<td>/location/us_county/county_seat</td>
<td>0.51</td>
<td>0.51</td>
</tr>
<tr>
<td>/music/artist/origin</td>
<td>0.64</td>
<td>0.66</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>0.80</td>
<td>0.79</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>0.61</td>
<td>0.70</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>0.78</td>
<td>0.77</td>
</tr>
<tr>
<td>Average</td>
<td>0.67</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Mintz et al. (2009)
Inherently have noise in training data, need special methods (e.g., multi-instance learning) to handle false positives AND false negatives.

---

**Freebase**

<table>
<thead>
<tr>
<th>Entity 1</th>
<th>Entity 2</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thailand</td>
<td>Bangkok</td>
<td>/location/country/capital</td>
</tr>
</tbody>
</table>

**Sentences mentioning the two entities:**

1. *Bangkok* is the most populous city of *Thailand*.

2. *Bangkok* grew rapidly during the 1960s through the 1980s and now exerts a significant impact among *Thailand*’s politics, economy, education, media and modern society.

3. The nation of *Thailand* is about to get its very first visit ever from a president this weekend, President Obama, so the American Embassy in *Bangkok* is understandably very excited right now.
Instead of labels on each individual instance, the learner only observes labels on bags of instances.

- **Negative Bags**: A bag is labeled negative, if all the examples in it are negative.
- **Positive Bags**: A bag is labeled positive, if there is at least one positive example.

Xu et al. (2013), Pershina et al. (2014), Tabassum (2016)
Multi-instance Learning

Figure 1: Plate diagram of Guided DS

Xu et al. (2013), Pershina et al. (2014), Tabassum (2016)
Slot Filling
A massive earthquake of magnitude 7.3 struck Iraq on Sunday, 103 kms southeast of the city of As-Sulaymaniyah, the US Geological Survey said, reports Reuters. US Geological Survey initially said the quake was of a magnitude 7.2, before revising it to 7.3.

**Speaker:** Alan Clark

**Title:** “Gender Roles in the Holy Roman Empire”

**Location:** Allagher Center Main Auditorium

This talk will discuss...

Freitag and McCallum (2000)
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?
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>
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)
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
- Slot filling: tied to a specific ontology, but gives fine-grained information
- 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