What is natural language understanding?
Humans are the only example
The Imitation Game (1950)

"Can machines think?"
The Imitation Game (1950)

"Can machines think?"
"Can machines think?"

Q: Please write me a sonnet on the subject of the Forth Bridge.
A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.
A: (Pause about 30 seconds and then give as answer) 105621.
"Can machines think?"

Q: Please write me a sonnet on the subject of the Forth Bridge.
A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.
A: (Pause about 30 seconds and then give as answer) 105621.

- Behavioral test
- ...of intelligence, not just natural language understanding
William Wilkinson’s "An Account of the Principalities of Wallachia and Moldavia" inspired this author’s most famous novel.
Siri

What can I help you with?

“What's the best Japanese restaurant in San Francisco”

I found a number of Japanese restaurants... 24 of them are in San Francisco. I've sorted them by rating:

Market Street
Sushi Zone
3.5 stars
690 reviews
Google

how many people live in lille

Web  Maps  News  Shopping  Images  More  Search tools

About 14,200,000 results (0.54 seconds)

227,560 (2010)
Lille, Population
Representations for natural language understanding?
Word vectors?
Word vectors?
The boy wants to go to New York City.
Cynthia sold the bike to Bob for $200
Logical forms?

What is the largest city in California?

\[
\text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{population}(x))
\]
Why ICML?

Opportunity for transfer of ideas between ML and NLP
Why ICML?

Opportunity for transfer of ideas between ML and NLP

- mid-1970s: **HMMs** for speech recognition $\Rightarrow$ probabilistic models
Why ICML?

Opportunity for transfer of ideas between ML and NLP

- mid-1970s: **HMMs** for speech recognition $\Rightarrow$ probabilistic models
- early 2000s: **conditional random fields** for part-of-speech tagging $\Rightarrow$ structured prediction
Why ICML?

Opportunity for transfer of ideas between ML and NLP

- mid-1970s: **HMMs** for speech recognition ⇒ probabilistic models
- early 2000s: **conditional random fields** for part-of-speech tagging ⇒ structured prediction
- early 2000s: **Latent Dirichlet Allocation** for modeling text documents ⇒ topic modeling
Why ICML?

Opportunity for transfer of ideas between ML and NLP

- mid-1970s: HMMs for speech recognition $\implies$ probabilistic models
- early 2000s: conditional random fields for part-of-speech tagging $\implies$ structured prediction
- early 2000s: Latent Dirichlet Allocation for modeling text documents $\implies$ topic modeling
- mid 2010s: sequence-to-sequence models for machine translation $\implies$ neural networks with memory/state
Why ICML?

Opportunity for transfer of ideas between ML and NLP

- mid-1970s: **HMMs** for speech recognition $\Rightarrow$ probabilistic models
- early 2000s: **conditional random fields** for part-of-speech tagging $\Rightarrow$ structured prediction
- early 2000s: **Latent Dirichlet Allocation** for modeling text documents $\Rightarrow$ topic modeling
- mid 2010s: **sequence-to-sequence models** for machine translation $\Rightarrow$ neural networks with memory/state
- now: ??? for natural language understanding
Goals of this tutorial

- Provide **intuitions** about natural language
Goals of this tutorial

• Provide **intuitions** about natural language

• Describe current **state-of-the-art** methods
Goals of this tutorial

• Provide intuitions about natural language

• Describe current state-of-the-art methods

• Propose challenges / opportunities
Tips

What to expect:

• A lot of tutorial is about thinking about the phenomena in language

• Minimal details on methods and empirical results
Tips

What to expect:

• A lot of tutorial is about thinking about the phenomena in language

• Minimal details on methods and empirical results

What to look for:

• Challenging machine learning problems: representation learning, structured prediction

• Think about the **end-to-end** problem and decide what phenomena to focus on, which ones to punt on, which ones are bulldozed by ML
Outline

Properties of language

- Distributional semantics
- Frame semantics
- Model-theoretic semantics

Reflections
Levels of linguistic analyses

natural language utterance
Levels of linguistic analyses

Syntax: what is grammatical?

natural language utterance
Levels of linguistic analyses

**Semantics**: what does it mean?

**Syntax**: what is grammatical?

natural language utterance
Levels of linguistic analyses

**Pragmatics**: what does it do?

**Semantics**: what does it mean?

**Syntax**: what is grammatical?

*natural language utterance*
Analogy with programming languages

Syntax: no compiler errors

Semantics: no implementation bugs

Pragmatics: implemented the right algorithm
Analogy with programming languages

**Syntax:** no compiler errors

**Semantics:** no implementation bugs

**Pragmatics:** implemented the right algorithm

Different **syntax**, same **semantics** (5):

\[ 2 + 3 \Leftrightarrow 3 + 2 \]
Analogy with programming languages

Syntax: no compiler errors

Semantics: no implementation bugs

Pragmatics: implemented the right algorithm

Different syntax, same semantics (5):

\[ 2 + 3 \iff 3 + 2 \]

Same syntax, different semantics (1 and 1.5):

\[ 3 / 2 \text{ (Python 2.7)} \not\iff 3 / 2 \text{ (Python 3)} \]
Analogy with programming languages

**Syntax:** no compiler errors

**Semantics:** no implementation bugs

**Pragmatics:** implemented the right algorithm

Different syntax, same semantics (5):

\[ 2 + 3 \Leftrightarrow 3 + 2 \]

Same syntax, different semantics (1 and 1.5):

\[ 3 / 2 \text{ (Python 2.7)} \not\Leftrightarrow 3 / 2 \text{ (Python 3)} \]

Good semantics, bad pragmatics:

correct implementation of deep neural network for estimating coin flip prob.
Syntax

Dependency parse tree:

```
DT  det
 The

NN  nsubj
 boy

VBZ  xcomp
 wants

TO  mark
 to

VB  TO
 go

TO
 to

NNP
 New

NNP  nmod
 York

NNP
 City
```

The boy wants to go to New York City.
Syntax

Dependency parse tree:

Parts of speech:

- **NN**: common noun
- **NNP**: proper noun
- **VBZ**: verb, 3rd person singular
Syntax

Dependency parse tree:

Parts of speech:
- NN: common noun
- NNP: proper noun
- VBZ: verb, 3rd person singular

Dependency relations:
- nsubj: subject (nominal)
- nmod: modifier (nominal)
Prepositional attachment ambiguity

*I ate some dessert with a fork.*
Prepositional attachment ambiguity

*I ate some dessert with a fork.*
Prepositional attachment ambiguity

*I ate some dessert with a fork.*

![Tree diagram showing two possible interpretations of the sentence: one with the prepositional phrase attached to the verb and the other with it attached to the noun phrase.](image)
Prepositional attachment ambiguity

I ate some dessert with a fork.

S

NP
I
VP
V
ate
NP
some dessert
PP
with a fork
S
NP
I
VP
V
ate
NP
some dessert
PP
with a fork
Prepositional attachment ambiguity

I ate some dessert with a fork.

Both are grammatical; is syntax enough to disambiguate?
Semantics

Meaning
This is the tree of life.

Lexical semantics: what words mean

Compositional semantics: how meaning gets combined
What’s a word?

light
What’s a word?

*light*

**Multi-word expressions:** meaning unit beyond a word

*light bulb*
What’s a word?

*light*

**Multi-word expressions**: meaning unit beyond a word

*light bulb*

**Morphology**: meaning unit within a word

*light*  *lighten*  *lightening*  *relight*
What’s a word?

light

Multi-word expressions: meaning unit beyond a word

light bulb

Morphology: meaning unit within a word

light lighten lightening relight

Polysemy: one word has multiple meanings (word senses)

- The **light** was filtered through a soft glass window.
- He stepped into the **light**.
- This lamp **lights** up the room.
- The load is not **light**.
Synonymy

Words:

confusing
Synonymy

Words:

confusing unclear perplexing mystifying
Synonymy

Words:

confusing  unclear  perplexing  mystifying

Sentences:

I have fond memories of my childhood.
I reflect on my childhood with a certain fondness.
I enjoy thinking back to when I was a kid.
Synonymy

Words:
confusing unclear perplexing mystifying

Sentences:
I have fond memories of my childhood.
I reflect on my childhood with a certain fondness.
I enjoy thinking back to when I was a kid.

Beware: no true equivalence due to subtle differences in meaning; think distance metric
Synonymy

Words:
confusing unclear perplexing mystifying

Sentences:
I have fond memories of my childhood.
I reflect on my childhood with a certain fondness.
I enjoy thinking back to when I was a kid.

Beware: no true equivalence due to subtle differences in meaning; think distance metric

But there’s more to meaning than similarity...
Other lexical relations

**Hyponymy (is-a):**

a cat is a mammal
Other lexical relations

**Hyponymy (is-a):**

a cat is a mammal

**Meronomy (has-a):**

a cat has a tail
Other lexical relations

Hyponymy (is-a):

a cat is a mammal

Meronomy (has-a):

a cat has a tail

Useful for entailment:

I am giving an NLP tutorial at ICML.
⇒
I am speaking at a conference.
Compositional semantics

Two ideas: **model theory** and **compositionality**

Model theory: sentences refer to the world

*Block 2 is blue.*
Compositional semantics

Two ideas: **model theory** and **compositionality**

Model theory: sentences refer to the world

*Block 2 is blue.*
Compositional semantics

Two ideas: model theory and compositionality

Model theory: sentences refer to the world

Block 2 is blue.

Compositionality: meaning of whole is meaning of parts

The [block left of the red block] is blue.
Quantifiers

Universal and existential quantification:

**Every** block is blue.

**Some** block is blue.
Quantifiers

Universal and existential quantification:

**Every** block is blue.

```
   1  2  3  4
```

**Some** block is blue.

```
   1  2  3  4
```

Quantifier scope ambiguity:

**Every** non-blue block is next to **some** blue block.

```
   1  2  3  4
```
Quantifiers

Universal and existential quantification:

Every block is blue.

\begin{center}
\begin{tabular}{cccc}
1 & 2 & 3 & 4
\end{tabular}
\end{center}

Some block is blue.

\begin{center}
\begin{tabular}{cccc}
1 & 2 & 3 & 4
\end{tabular}
\end{center}

Quantifier scope ambiguity:

Every non-blue block is next to some blue block.

\begin{center}
\begin{tabular}{cccc}
1 & 2 & 3 & 4
\end{tabular}
\end{center}

Every non-blue block is next to some blue block.

\begin{center}
\begin{tabular}{ccc}
1 & 2 & 3
\end{tabular}
\end{center}
Multiple possible worlds

Modality:

*Block 2 must be blue. Block 1 can be red.*
Multiple possible worlds

Modality:

*Block 2 must be blue. Block 1 can be red.*

Beliefs:

Clark Kent  Superman
Multiple possible worlds

Modality:

Block 2 **must** be blue. Block 1 **can** be red.

Beliefs:

*Lois believes* Superman is a hero.

*Lois believes* Clark Kent is a hero.
Anaphora

The dog chased the cat, which ran up a tree. It waited at the top.

The dog chased the cat, which ran up a tree. It waited at the top.
The **dog** chased the **cat**, which ran up a tree. **It** waited at the top.

The **dog** chased the **cat**, which ran up a tree. **It** waited at the bottom.
The dog chased the cat, which ran up a tree. It waited at the top.

The dog chased the cat, which ran up a tree. It waited at the bottom.

"The Winograd Schema Challenge" (Levesque, 2011)

- Easy for humans, can’t use surface-level patterns
Pragmatics

Conversational implicature: new material suggested (not logically implied) by sentence

- A: *What on earth has happened to the roast beef?*

  B: *The dog is looking very happy.*
Conversational implicature: new material suggested (not logically implied) by sentence

- A: What on earth has happened to the roast beef?

  B: The dog is looking very happy.

- Implicature: The dog at the roast beef.
**Pragmatics**

**Conversational implicature**: new material *suggested* (not logically implied) by sentence

- A: *What on earth has happened to the roast beef?*

  *B: The dog is looking very happy.*

- Implicature: *The dog at the roast beef.*

**Presupposition**: background assumption independent of truth of sentence

- *I have stopped eating meat.*
Pragmatics

Conversational implicature: new material suggested (not logically implied) by sentence

- A: What on earth has happened to the roast beef?

  B: The dog is looking very happy.

- Implicature: The dog at the roast beef.

Presupposition: background assumption independent of truth of sentence

- I have stopped eating meat.

- Presupposition: I once was eating meat.
Pragmatics

Semantics: what does it mean \textit{literally}? 

Pragmatics: what is the speaker really conveying?
Pragmatics

Semantics: what does it mean **literally**?

Pragmatics: what is the speaker really conveying?

- Underlying principle (Grice, 1975): language is cooperative game between speaker and listener

- Implicatures and presuppositions depend on people and context and involves soft inference (machine learning opportunities here!)
Vagueness, ambiguity, uncertainty

Vagueness: does not specify full information

*I had a late lunch.*
Vagueness, ambiguity, uncertainty

Vagueness: does not specify full information

I had a late lunch.

Ambiguity: more than one possible (precise) interpretations

One morning I shot an elephant in my pajamas.
Vagueness, ambiguity, uncertainty

Vagueness: does not specify full information

*I had a *late* lunch.*

Ambiguity: more than one possible (precise) interpretations

*One morning I shot an elephant *in my pajamas.*

*How he got in my pajamas, I don’t know.* — Groucho Marx
Vagueness, ambiguity, uncertainty

**Vagueness**: does not specify full information

*I had a late lunch.*

**Ambiguity**: more than one possible (precise) interpretations

*One morning I shot an elephant in my pajamas.*

*How he got in my pajamas, I don’t know.* — Groucho Marx

**Uncertainty**: due to an imperfect statistical model

*The witness was being contumacious.*
Summary so far

- **Analyses**: syntax, semantics, pragmatics

- **Lexical semantics**: synonymy, hyponymy/meronymy

- **Compositional semantics**: model theory, compositionality

- **Challenges**: polysemy, vagueness, ambiguity, uncertainty
Outline

Properties of language

**Distributional semantics**

Frame semantics

Model-theoretic semantics

Reflections
Distributional semantics: warmup

The new design has _____ lines.

Let’s try to keep the kitchen _____.

I forgot to _____ out the cabinet.
Distributional semantics: warmup

The new design has _____ lines.

Let’s try to keep the kitchen _____.

I forgot to _____ out the cabinet.

What does _____ mean?
Distributional semantics

*The new design has _____ lines.*

Observation: **context** can tell us a lot about word meaning

Context: local window around a word occurrence (for now)
Distributional semantics

The new design has ____ lines.

Observation: context can tell us a lot about word meaning

Context: local window around a word occurrence (for now)

Roots in linguistics:

- **Distributional hypothesis**: Semantically similar words occur in similar contexts [Harris, 1954]
- ”You shall know a word by the company it keeps.” [Firth, 1957]
Distributional semantics

*The new design has _____ lines.*

**Observation:** *context* can tell us a lot about word meaning

**Context:** local window around a word occurrence (for now)

**Roots in linguistics:**
- **Distributional hypothesis:** Semantically similar words occur in similar contexts [Harris, 1954]
- ”You shall know a word by the company it keeps.” [Firth, 1957]
- Contrast: Chomsky’s generative grammar (lots of hidden prior structure, no data)
Distributional semantics

The new design has _____ lines.

Observation: **context** can tell us a lot about word meaning

Context: local window around a word occurrence (for now)

Roots in linguistics:
- **Distributional hypothesis**: Semantically similar words occur in similar contexts [Harris, 1954]
- ”You shall know a word by the company it keeps.” [Firth, 1957]
- Contrast: Chomsky’s generative grammar (lots of hidden prior structure, no data)

Upshot: **data-driven!**
1. Form a **word-context matrix** of counts (data)

context $c$

| word $w$ | $N$ |
General recipe

1. Form a **word-context matrix** of counts (data)

\[
\begin{array}{c|c|c}
\text{context } c & \text{word } w & N \\
\end{array}
\]

2. Perform **dimensionality reduction** (generalize)

\[
\text{word } w \Theta \Rightarrow \text{word vectors } \theta_w \in \mathbb{R}^d
\]
Latent semantic analysis

Data:

Doc1: *Cats have tails.*
Doc2: *Dogs have tails.*
Latent semantic analysis

Data:

Doc1: *Cats have tails.*
Doc2: *Dogs have tails.*

Matrix: contexts = documents that word appear in

<table>
<thead>
<tr>
<th></th>
<th>Doc1</th>
<th>Doc2</th>
</tr>
</thead>
<tbody>
<tr>
<td>cats</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>dogs</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>have</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>tails</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

[Deerwater/Dumais/Furnas/Landauer/Harshman, 1990]
Latent semantic analysis

Dimensionality reduction: $\text{SVD}$

$$\text{word } w \quad N \quad \approx \quad \Theta \quad S \quad V^\top$$

[Deerwater/Dumais/Furnas/Landauer/Harshman, 1990]
Latent semantic analysis

Dimensionality reduction: SVD

\[
\text{document } c \\
\begin{pmatrix}
\text{word } w
\end{pmatrix}
\approx \Theta
\begin{pmatrix}
S \\
V^T
\end{pmatrix}
\]

- Used for information retrieval
- Match query to documents in latent space rather than on keywords

[Deerwater/Dumais/Furnas/Landauer/Harshman, 1990]
Unsupervised part-of-speech induction

Data:

*Cats have tails.*

*Dogs have tails.*
Unsupervised part-of-speech induction

Data:

*Cats have tails.*

*Dogs have tails.*

Matrix: contexts = words on left, words on right

<table>
<thead>
<tr>
<th></th>
<th>cats_L</th>
<th>dogs_L</th>
<th>tails_R</th>
<th>have_L</th>
<th>have_R</th>
</tr>
</thead>
<tbody>
<tr>
<td>cats</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>dogs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>have</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>tails</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Dimensionality reduction: SVD
Effect of context

Suppose *Barack Obama* always appear together (a *collocation*).
Effect of context

Suppose *Barack Obama* always appear together (a collocation).

Global context (document):

- same context $\Rightarrow \theta_{\text{Barack}}$ close to $\theta_{\text{Obama}}$
- more ”semantic”
Effect of context

Suppose *Barack Obama* always appear together (a collocation).

**Global context (document):**

- same context $\Rightarrow \theta_{Barack}$ close to $\theta_{Obama}$
- more "semantic"

**Local context (neighbors):**

- different context $\Rightarrow \theta_{Barack}$ far from $\theta_{Obama}$
- more "syntactic"
Skip-gram model with negative sampling

Data:

*Cats and dogs have tails.*
Skip-gram model with negative sampling

Data:

*Cats and dogs have tails.*

Form matrix: contexts = words in a window

<table>
<thead>
<tr>
<th></th>
<th>cats</th>
<th>and</th>
<th>dogs</th>
<th>have</th>
<th>tails</th>
</tr>
</thead>
<tbody>
<tr>
<td>cats</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>and</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>dogs</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>have</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>tails</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

[Mikolov/Sutskever/Chen/Corrado/Dean, 2013 (word2vec)]
Skip-gram model with negative sampling

Dimensionality reduction: logistic regression with SGD

[Mikolov/Sutskever/Chen/Corrado/Dean, 2013 (word2vec)]
Skip-gram model with negative sampling

Dimensionality reduction: logistic regression with SGD

Model: predict good \((w, c)\) using logistic regression

\[
p_\theta(g = 1 \mid w, c) = \left(1 + \exp(\theta_w \cdot \beta_c)\right)^{-1}
\]
Skip-gram model with negative sampling

Dimensionality reduction: logistic regression with SGD

Model: predict good \((w, c)\) using logistic regression

\[
p_\theta(g = 1 \mid w, c) = (1 + \exp(\theta_w \cdot \beta_c))^{-1}
\]

Positives: \((w, c)\) from data

Negatives: \((w, c')\) for irrelevant \(c'\) (\(k\) times more)

\[](cats, AI) −(cats, linguistics) −(cats, statistics)
Skip-gram model with negative sampling

Data distribution:

\[ \hat{p}(w, c) \propto N(w, c) \]

Objective:

\[
\max_{\theta, \beta} \sum_{w,c} \hat{p}(w, c) \log p(g = 1 \mid w, c) + \\
\sum_{w,c'} \hat{p}(w) \hat{p}(c') \log p(g = 0 \mid w, c')
\]

[Levy/Goldberg, 2014]
Skip-gram model with negative sampling

Data distribution:
\[ \hat{p}(w, c) \propto N(w, c) \]

Objective:
\[
\max_{\theta, \beta} \sum_{w, c} \hat{p}(w, c) \log p(g = 1 \mid w, c) + 
\]
\[
k \sum_{w, c'} \hat{p}(w)\hat{p}(c') \log p(g = 0 \mid w, c')
\]

If no dimensionality reduction:
\[
\theta_w \cdot \beta_c = \log \left( \frac{\hat{p}(w,c)}{\hat{p}(w)\hat{p}(c)} \right) = \text{PMI}(w, c)
\]

[Levy/Goldberg, 2014]
2D visualization of word vectors
2D visualization of word vectors
Nearest neighbors

**cherish**

(words)
- adore
- love
- admire
- embrace
- rejoice

(contexts)
- cherish
- both
- love
- pride
- thy

quasi-synonyms
Nearest neighbors

**cherish**

(words)
- adore
- love
- admire
- embrace
- rejoice

(contexts)
- cherish
- both
- love
- pride
- thy

quasi-synonyms

**tiger**

(words)
- leopard
- dhole
- warthog
- rhinoceros
- lion

(contexts)
- tiger
- leopard
- panthera
- woods
- puma

co-hyponyms
Nearest neighbors

**cherish**
- *words*
  - adore
  - love
  - admire
  - embrace
  - rejoice
- *contexts*
  - cherish
  - both
  - love
  - pride
  - thy

**tiger**
- *words*
  - leopard
  - dhole
  - warthog
  - rhinoceros
  - lion
- *contexts*
  - tiger
  - leopard
  - panthera
  - woods
  - puma

**good**
- *words*
  - bad
  - decent
  - excellent
  - lousy
  - nice
- *contexts*
  - faith
  - natured
  - luck
  - riddance
  - both

 quasi-synonyms  co-hyponyms  includes antonyms
Nearest neighbors

**cherish**
- (words)
  - adore
  - love
  - admire
  - embrace
  - rejoice
- (contexts)
  - cherish
  - both
  - love
  - pride
  - thy

**tiger**
- (words)
  - leopard
  - dhole
  - warthog
  - rhinoceros
  - lion
- (contexts)
  - tiger
  - leopard
  - panthera
  - woods
  - puma

**good**
- (words)
  - bad
  - decent
  - excellent
  - lousy
  - nice
- (contexts)
  - faith
  - natured
  - luck
  - riddance
  - both

quasi-synonyms  co-hyponyms  includes antonyms

Many things under **semantic similarity**!
Analogies

Differences in context vectors capture relations:

$$\theta_{\text{king}} - \theta_{\text{man}} \approx \theta_{\text{queen}} - \theta_{\text{woman}} \quad (\text{gender})$$

[Mikolov/Yih/Zweig, 2013; Levy/Goldberg, 2014]
**Analogies**

Differences in context vectors capture relations:

\[ \theta_{\text{king}} - \theta_{\text{man}} \approx \theta_{\text{queen}} - \theta_{\text{woman}} \] (gender)

\[ \theta_{\text{france}} - \theta_{\text{french}} \approx \theta_{\text{mexico}} - \theta_{\text{spanish}} \] (language)

\[ \theta_{\text{car}} - \theta_{\text{cars}} \approx \theta_{\text{apple}} - \theta_{\text{apples}} \] (plural)

[Mikolov/Yih/Zweig, 2013; Levy/Goldberg, 2014]
Analogies

Differences in context vectors capture relations:

\[ \theta_{\text{king}} - \theta_{\text{man}} \approx \theta_{\text{queen}} - \theta_{\text{woman}} \] (gender)

\[ \theta_{\text{france}} - \theta_{\text{french}} \approx \theta_{\text{mexico}} - \theta_{\text{spanish}} \] (language)

\[ \theta_{\text{car}} - \theta_{\text{cars}} \approx \theta_{\text{apple}} - \theta_{\text{apples}} \] (plural)

Intuition:

\[ \theta_{\text{king}} - \theta_{\text{man}} \approx \theta_{\text{queen}} - \theta_{\text{woman}} \]

\[ [\text{crown,he}] - [\text{he}] \approx [\text{crown,he}] - [\text{he}] \]

Don’t need dimensionality reduction for this to work!
Other models

Multinomial models:

• HMM word clustering [Brown et al., 1992]

• Latent Dirichlet Allocation [Blei et al., 2003]
Other models

Multinomial models:

• HMM word clustering [Brown et al., 1992]
• Latent Dirichlet Allocation [Blei et al., 2003]

Neural network models:

• Multi-tasking neural network [Weston/Collobert, 2008]
Other models

Multinomial models:

- HMM word clustering [Brown et al., 1992]
- Latent Dirichlet Allocation [Blei et al., 2003]

Neural network models:

- Multi-tasking neural network [Weston/Collobert, 2008]

Recurrent/recursive models: (can embed phrases too)

- Neural language models [Bengio et al., 2003]
- Neural machine translation [Sutskever/Vinyals/Le, 2014, Cho/Merrienboer/Bahdanau/Bengio, 2014]
- Recursive neural networks [Socher/Lin/Ng/Manning, 2011]
Hearst patterns for hyponyms

*The bow lute, such as the *Bambara ndang*, is plucked...*
Hearst patterns for hyponyms

The bow lute, such as the **Bambara ndang**, is plucked...

\[\text{Bambara ndang} \text{ hyponym-of bow lute}\]
Hearst patterns for hyponyms

The bow lute, such as the **Bambara ndang**, is plucked...

⇓

**Bambara ndang** hyponym-of **bow lute**

General rules:

- $C$ such as $X$ ⇒ $[X$ hyponym-of $C]$  
- $X$ and other $C$ ⇒ $[X$ hyponym-of $C]$  
- $C$ including $X$ ⇒ $[X$ hyponym-of $C]$
Hearst patterns for hyponyms

The bow lute, such as the Bambara ndang, is plucked...

\[ \downarrow \]

\textit{Bambara ndang} hyponym-of \textit{bow lute}

General rules:

\begin{itemize}
  \item \( C \) such as \( X \) \( \Rightarrow [X \text{ hyponym-of } C] \)
  \item \( X \) and other \( C \) \( \Rightarrow [X \text{ hyponym-of } C] \)
  \item \( C \) including \( X \) \( \Rightarrow [X \text{ hyponym-of } C] \)
\end{itemize}

- **Thrust**: apply simple patterns to large web corpora
- Again, context reveals information about semantics
Hearst patterns for hyponyms

The bow lute, such as the **Bambara ndang**, is plucked...

\[\downarrow\]

**Bambara ndang** hyponym-of **bow lute**

General rules:

\begin{itemize}
  \item $C$ such as $X \Rightarrow [X$ hyponym-of $C]$
  \item $X$ and other $C \Rightarrow [X$ hyponym-of $C]$
  \item $C$ including $X \Rightarrow [X$ hyponym-of $C]$
\end{itemize}

- **Thrust**: apply simple patterns to large web corpora
- Again, context reveals information about semantics
- Can learn patterns via bootstrapping (semi-supervised learning)
Summary so far

- **Premise**: semantics = context of word/phrase
Summary so far

- **Premise**: semantics = context of word/phrase
- **Recipe**: form word-context matrix + dimensionality reduction

\[
\begin{array}{c|c}
\text{context } c & N \\
\hline
\text{word } w & \end{array}
\]
Summary so far

- **Premise**: semantics = context of word/phrase
- **Recipe**: form word-context matrix + dimensionality reduction

Pros:
- Simple models, leverage tons of raw text
- Context captures nuanced information about usage
- Word vectors useful in downstream tasks
Food for thought

What **contexts**?

- No such thing as pure unsupervised learning, representation depends on choice of context (e.g., global/local/task-specific)
- Language is not just text in isolation, context should include world/environment
Food for thought

What contexts?

- No such thing as pure unsupervised learning, representation depends on choice of context (e.g., global/local/task-specific)
- Language is not just text in isolation, context should include world/environment

What models?

- Currently very fine-grained (non-parametric idiot savants)
- Language is about speaker’s intention, not words
Food for thought

What contexts?

- No such thing as pure unsupervised learning, representation depends on choice of context (e.g., global/local/task-specific)
- Language is not just text in isolation, context should include world/environment

What models?

- Currently very fine-grained (non-parametric idiot savants)
- Language is about speaker’s intention, not words

Examples to ponder:

Cynthia sold the bike for $200.
The bike sold for $200.
Outline

Properties of language

Distributional semantics

Frame semantics

Model-theoretic semantics

Reflections
Word meaning revisited

sold
Word meaning revisited

sold

Distributional semantics: all the contexts in which sold occurs

...was sold by... ...sold me that piece of...

• Can find similar words/contexts and generalize (dimensionality reduction), but monolithic (no internal structure on word vectors)
Word meaning revisited

sold

Distributional semantics: all the contexts in which sold occurs

...was sold by... ...sold me that piece of...

- Can find similar words/contexts and generalize (dimensionality reduction), but monolithic (no internal structure on word vectors)

Frame semantics: meaning given by a frame, a stereotypical situation

Commercial transaction

SELLER: ?
BUYER: ?
GOODS: ?
PRICE: ?
More subtle frames

I spent three hours on land this afternoon.

I spent three hours on the ground this afternoon.
More subtle frames

*I spent three hours on *land* this afternoon.*

*I spent three hours on the *ground* this afternoon.*
Two properties of frames

Prototypical: don’t need to handle all the cases

widow
Two properties of frames

Prototypical: don’t need to handle all the cases

widow

• Frame: woman marries one man, man dies

[Fillmore, 1977; Langacker, 1987]
Two properties of frames

Prototypical: don’t need to handle all the cases

*widow*

- Frame: woman marries one man, man dies
- What if a woman has 3 husbands, 2 of which died?
Two properties of frames

**Prototypical:** don’t need to handle all the cases

*widow*

- **Frame:** woman marries one man, man dies
- What if a woman has 3 husbands, 2 of which died?

**Profiling:** highlight one aspect

- *sell* is seller-centric, *buy* is buyer-centric

*Cynthia sold the bike (to Bob).*

*Bob bought the bike (from Cynthia).*
Two properties of frames

Prototypical: don’t need to handle all the cases

*widow*

- Frame: woman marries one man, man dies
- What if a woman has 3 husbands, 2 of which died?

Profiling: highlight one aspect

- *sell* is seller-centric, *buy* is buyer-centric

  *Cynthia sold the bike (to Bob).*
  *Bob bought the bike (from Cynthia).*

- *rob* highlights person, *steal* highlights goods

  *Cynthia robbed Bob (of the bike).*
  *Cynthia stole the bike (from Bob).*
A story

Joe went to a restaurant. Joe ordered a hamburger. When the hamburger came, it was burnt to a crisp. Joe stormed out without paying.
Joe went to a restaurant. Joe ordered a hamburger. When the hamburger came, it was burnt to a crisp. Joe stormed out without paying.

- Need background knowledge to really understand
- Schank and Abelson developed notion of a script which captures this knowledge
- Same idea as frame, but tailored for event sequences
A story

Joe went to a restaurant. Joe ordered a hamburger. When the hamburger came, it was burnt to a crisp. Joe stormed out without paying.

- Need background knowledge to really understand
- Schank and Abelson developed notion of a script which captures this knowledge
- Same idea as frame, but tailored for event sequences

Restaurant script (simplified):

**Entering**: S PTRANS S into restaurant, S PTRANS S to table
**Ordering**: S PTRANS< menu to S, waiter PTRANS to table, S MTRANS< 'I want food' to waiter
**Eating**: waiter PTRANS food to S, S INGEST food
**Exiting**: waiter PTRANS to S, waiter ATRANS check to S, S ATRANS money to waiter, S PTRANS out of restaurant

[Schank/Abelson, 1977]
Cynthia sold the bike for $200.
Cynthia sold the bike for $200.

Commercial transaction

SELLER : Cynthia
GOODS : the bike
PRICE : $200
From syntax to semantics

Dependency parse tree:
From syntax to semantics

Extraction rules:

\[
\begin{align*}
sold \ nsubj \ X & \Rightarrow \text{SELLER}:X \\
sold \ dobj \ X & \Rightarrow \text{GOODS}:X \\
\text{sold} \ nmod:for \ X & \Rightarrow \text{PRICE}:X
\end{align*}
\]

Dependency parse tree:
From syntax to semantics

Extraction rules:

- \( sold \ nsubj \ X \Rightarrow SELLER:X \)
- \( sold \ dobj \ X \Rightarrow GOODS:X \)
- \( sold \ nmod:for \ X \Rightarrow PRICE:X \)

Dependency parse tree:

Commercial transaction
SELLER : Cynthia
GOODS : the bike
PRICE : $200
From syntax to semantics

Extraction rules:

\[ \text{sold nsubj } X \Rightarrow \text{SELLER:}X \]
\[ \text{sold dobj } X \Rightarrow \text{GOODS:}X \]
\[ \text{sold nmod:for } X \Rightarrow \text{GOODS:}X \]

Dependency structure:
From syntax to semantics

Extraction rules:

\( sold \ nsubj \ X \Rightarrow SELLER:X \)
\( sold \ dobj \ X \Rightarrow GOODS:X \)
\( sold \ nmod:for \ X \Rightarrow GOODS:X \)

Dependency structure:

Commercial transaction
SELLER: the bike
PRICE: $200
From syntax to semantics

Commercial transaction

SELLER: Cynthia
BUYER: Bob
GOODS: the bike
PRICE: $200
From syntax to semantics

Commercial transaction

SELLER : Cynthia
BUYER : Bob
GOODS : the bike
PRICE : $200

Many **syntactic alternations** with different arguments/verbs:

*Cynthia sold the bike to Bob for $200.*
*The bike sold for $200.*
From syntax to semantics

Commercial transaction

SELLER : Cynthia
BUYER : Bob
GOODS : the bike
PRICE : $200

Many syntactic alternations with different arguments/verbs:

Cynthia sold the bike to Bob for $200.
The bike sold for $200.
Bob bought the bike from Cynthia.
The bike was bought by Bob.
The bike was bought for $200.
The bike was bought for $200 by Bob.
From syntax to semantics

Commercial transaction

SELLER : Cynthia
BUYER : Bob
GOODS : the bike
PRICE : $200

Many syntactic alternations with different arguments/verbs:

*Cynthia sold the bike to Bob for $200.*
*The bike sold for $200.*
*Bob bought the bike from Cynthia.*
*The bike was bought by Bob.*
*The bike was bought for $200.*
*The bike was bought for $200 by Bob.*

Goal: syntactic positions ⇒ semantic roles
Historical developments

Linguistics:

- **Case grammar [Fillmore, 1968]:** introduced idea of deep semantic roles (agents, themes, patients) which are tied to surface syntax (subjects, objects)
Historical developments

**Linguistics:**
- Case grammar [Fillmore, 1968]: introduced idea of deep semantic roles (agents, themes, patients) which are tied to surface syntax (subjects, objects)

**AI / cognitive science:**
- Frames [Minsky, 1975]: ”a data-structure for representing a stereotyped situation, like...a child’s birthday party”
Historical developments

**Linguistics:**

- **Case grammar [Fillmore, 1968]:** introduced idea of deep semantic roles (agents, themes, patients) which are tied to surface syntax (subjects, objects)

**AI / cognitive science:**

- **Frames [Minsky, 1975]:** ”a data-structure for representing a stereotyped situation, like...a child’s birthday party”
- **Scripts [Schank & Abelson, 1977]:** represent procedural knowledge (going to a restaurant)
Historical developments

**Linguistics:**
- Case grammar [Fillmore, 1968]: introduced idea of deep semantic roles (agents, themes, patients) which are tied to surface syntax (subjects, objects)

**AI / cognitive science:**
- Frames [Minsky, 1975]: ”a data-structure for representing a stereotyped situation, like...a child’s birthday party”
- Scripts [Schank & Abelson, 1977]: represent procedural knowledge (going to a restaurant)
- Frames [Fillmore, 1977]: coherent individuatable perception, memory, experience, action, or object
Historical developments

**Linguistics:**
- **Case grammar** [Fillmore, 1968]: introduced idea of deep semantic roles (agents, themes, patients) which are tied to surface syntax (subjects, objects)

**AI / cognitive science:**
- **Frames** [Minsky, 1975]: "a data-structure for representing a stereotyped situation, like...a child’s birthday party"
- **Scripts** [Schank & Abelson, 1977]: represent procedural knowledge (going to a restaurant)
- **Frames** [Fillmore, 1977]: coherent individuatable perception, memory, experience, action, or object

**NLP:**
Concrete realization: FrameNet

FrameNet [Baker/Fillmore/Lowe, 1998]:

- Centered around frames, argument labels are shared across frames

Commerce (sell)

SELLER : ?
BUYER : ?
GOODS : ?
PRICE : ?
Concrete realization: FrameNet

FrameNet [Baker/Fillmore/Lowe, 1998]:

- Centered around frames, argument labels are shared across frames

Lexical units that trigger frame:
- auction.n, auction.v
- retail.v, retailer.n
- sale.n, sell.v, seller.n
- vend.v, vendor.n

![Commerce (sell)]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Lexical Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELLER</td>
<td>auction.n, auction.v</td>
</tr>
<tr>
<td>BUYER</td>
<td>retail.v, retailer.n</td>
</tr>
<tr>
<td>GOODS</td>
<td>sale.n, sell.v, seller.n</td>
</tr>
<tr>
<td>PRICE</td>
<td>vend.v, vendor.n</td>
</tr>
</tbody>
</table>
Concrete realization: FrameNet

FrameNet [Baker/Fillmore/Lowe, 1998]:

• Centered around frames, argument labels are shared across frames

Commerce (sell)

LEXICAL UNITS THAT TRIGGER FRAME:

SELLER : ?
BUYER : ?
GOODS : ?
PRICE : ?

Lexical units that trigger frame:

auction.n, auction.v
retail.v, retailer.n
sale.n, sell.v, seller.n
vend.v, vendor.n

• Abstract away from the syntax by normalizing across different lexical units

• 4K predicates
Concrete realization: PropBank

PropBank [Palmer/Gildea/Kingsbury, 2002]:
- Centered around verbs and syntax, argument labels are verb-specific

sell.01
Concrete realization: PropBank

PropBank [Palmer/Gildea/Kingsbury, 2002]:
• Centered around verbs and syntax, argument labels are verb-specific

Commerce (sell)

```
sell.01.A0 (seller) : ?
sell.01.A1 (goods) : ?
sell.01.A2 (buyer) : ?
sell.01.A3 (price) : ?
sell.01.A4 (beneficiary) : ?
```
Concrete realization: PropBank

PropBank [Palmer/Gildea/Kingsbury, 2002]:

- Centered around verbs and syntax, argument labels are verb-specific

\[
\begin{array}{|l|}
\hline
\text{Commerce (sell)} \\
\hline
\text{sell.01.A0 (seller)} & : ? \\
\text{sell.01.A1 (goods)} & : ? \\
\text{sell.01.A2 (buyer)} & : ? \\
\text{sell.01.A3 (price)} & : ? \\
\text{sell.01.A4 (beneficiary)} & : ? \\
\hline
\end{array}
\]

- Word senses tied to WordNet
- Created based on a corpus, so more popular
Semantic role labeling

Task:

Input: Cynthia sold the bike to Bob for $200
Semantic role labeling

Task:

Input:  Cynthia sold the bike to Bob for $200

Output: SELLER  PREDICATE  GOODS  BUYER  PRICE
Semantic role labeling

Task:

Input:  *Cynthia* sold *the bike* to *Bob* for $200

Output:  SELLER  PREDICATE  GOODS  BUYER  PRICE

Subtasks:

1. Frame identification (PREDICATE)

2. Argument identification (SELLER, GOODS, etc.)
Frame identification

Jane recently bought flowers from Luigi’s shop.

⇒ buy.01
Frame identification

1. Construct dependency parse, choose predicate $p$ (*bought*)

[Hermann/Das/Weston/Ganchev, 2014]

⇒ buy.01
1. Construct dependency parse, choose predicate $p$ \((bought)\)
2. Extract paths from $p$ to dependents $a$

```
Jane    recently    bought    flowers    from    Luigi's    shop
```
Frame identification

1. Construct dependency parse, choose predicate \( p \) (\textit{bought})
2. Extract paths from \( p \) to dependents \( a \)
3. Map each dependent \( a \) to vector \( v_a \) (word vectors)
1. Construct dependency parse, choose predicate $p$ (bought)
2. Extract paths from $p$ to dependents $a$
3. Map each dependent $a$ to vector $v_a$ (word vectors)
4. Compute low. dim. representation $\phi = M[v_{a1}, \ldots, v_{an}]$
1. Construct dependency parse, choose predicate $p$ (bought)
2. Extract paths from $p$ to dependents $a$
3. Map each dependent $a$ to vector $v_a$ (word vectors)
4. Compute low. dim. representation $\phi = M[v_{a_1}, \ldots, v_{a_n}]$
5. Predict score $\phi \cdot \theta_y$ for label $y$ (e.g., buy.01)
Frame identification

• Learn parameters $\{v_w\}, M, \{\theta_y\}$ from full supervision

• Vectors allow generalization across verbs and arguments
Argument identification

1. Extract candidate argument spans \( \{a\} \) (using rules)

\[ 
\text{Jane} \quad \text{recently} \quad \text{bought} \quad \text{flowers} \quad \text{from} \quad \text{Luigi’s} \quad \text{shop} 
\]

\[ 
\text{Jane} \quad \text{Luigi’s shop} \quad \text{flowers} \quad \text{flowers from Luigi’s shop} 
\]
Argument identification

1. Extract candidate argument spans \( \{a\} \) (using rules)

\( \text{Jane} \quad \text{recently} \quad \text{bought} \quad \text{flowers} \quad \text{from} \quad \text{Luigi's} \quad \text{shop} \)

\( \text{Jane} \quad \text{Luigi's shop} \quad \text{flowers} \quad \text{flowers from Luigi's shop} \)

2. Predict argument label \( y_a \) for each candidate \( a \)

\( \text{A0, A1, A2, A3, A4, A5, AA, AA-TMP, AA-LOC, } \emptyset \)
Argument identification

1. Extract candidate argument spans \( \{a\} \) (using rules)

\[ \text{Jane} \quad \text{recently} \quad \text{bought} \quad \text{flowers} \quad \text{from} \quad \text{Luigi’s} \quad \text{shop} \]

\[ \text{Jane} \quad \text{Luigi’s shop} \quad \text{flowers} \quad \text{flowers from Luigi’s shop} \]

2. Predict argument label \( y_a \) for each candidate \( a \)

\( \text{A0, A1, A2, A3, A4, A5, AA, AA-TMP, AA-LOC, } \emptyset \)

Constraints include:

- Assigned spans cannot overlap
- Each core role can be used at most once
Argument identification

1. Extract candidate argument spans \( \{a\} \) (using rules)

\[
\begin{align*}
\text{Jane} & \quad \text{recently} & \quad \text{bought} & \quad \text{flowers} & \quad \text{from} & \quad \text{Luigi’s} & \quad \text{shop} \\
A0 & \quad A2 & \quad A1 & \quad \emptyset
\end{align*}
\]

2. Predict argument label \( y_a \) for each candidate \( a \)

\[
\text{A0, A1, A2, A3, A4, A5, AA, AA-TMP, AA-LOC, } \emptyset
\]

Constraints include:

- Assigned spans cannot overlap
- Each core role can be used at most once
Argument identification

1. Extract candidate argument spans \( \{a\} \) (using rules)

- Jane
- recently
- bought
- flowers
- from
- Luigi’s
- shop

\( \{A0, A2, A1, \emptyset\} \)

2. Predict argument label \( y_a \) for each candidate \( a \)

- A0, A1, A2, A3, A4, A5, AA, AA-TMP, AA-LOC, \( \emptyset \)

Constraints include:

- Assigned spans cannot overlap
- Each core role can be used at most once

Structured prediction: ILP or dynamic programming

[Punyakanok/Roth/Yih, 2008; Tackstrom/Ganchev/Das, 2015]
A brief history

- First system (on FrameNet) [Gildea/Jurafsky, 2002]
- CoNLL shared tasks [2004, 2005]
- Use ILP to enforce constraints on arguments [Punyakanok/Roth/Yih, 2008]
- No feature engineering or parse trees [Collobert/Weston, 2008]
- Semi-supervised frame identification [Das/Smith, 2011]
- Embeddings for frame identification [Hermann/Das/Weston/Ganchev, 2014]
- Dynamic programming for some argument constraints [Tackstrom/Ganchev/Das, 2015]
Abstract meaning representation (AMR)

Semantic role labeling:

- predicate + semantic roles
Abstract meaning representation (AMR)

Semantic role labeling:

- predicate + semantic roles

Named-entity recognition:

[Banarescu et al., 2013]

Cynthia went back to Lille because she liked it.
Abstract meaning representation (AMR)

Semantic role labeling:
- predicate + semantic roles

Named-entity recognition:

Coreference resolution:

[Banarescu et al., 2013]
Abstract meaning representation (AMR)

Semantic role labeling:
- predicate + semantic roles

Named-entity recognition:

Coreference resolution:

Motivation of AMR: unify all semantic annotation

[68] Banarescu et al., 2013

Motivation of AMR: 
**unify all semantic annotation**
AMR parsing task

Input: sentence

*The boy wants to go to New York City.*

Output: graph

[Flanigan/Thomson/Carbonell/Dyer/Smith, 2014]
AMR: normalize aggressively

The soldier feared battle.
The soldier feared battle.
AMR: normalize aggressively

The soldier feared battle.
The soldier was afraid of battle.
The soldier had a fear of battle.
Battle was feared by the soldier.
Battle was what the soldier was afraid of.

[뒷기, 2013]
AMR: normalize aggressively

The soldier feared battle.
The soldier was afraid of battle.
The soldier had a fear of battle.
Battle was feared by the soldier.
Battle was what the soldier was afraid of.

• Sentence-level annotation (unlike semantic role labeling)
• Challenge: must learn an (implicit) alignment!

[Banarescu et al., 2013]
AMR parsing: extract lexicon (step 1)

- Goal: given sentence-graph training examples, extract mapping from phrases to graph fragments

*The boy wants to go to New York City.*
AMR parsing: extract lexicon (step 1)

- Goal: given sentence-graph training examples, extract mapping from phrases to graph fragments

The boy wants to go to New York City.

want-01

ARG0

ARG1

visit-01

ARG0

ARG1

boy

name

name

op1

op2

op3

“New”

“York”

“City”

wants ⇒ want-01
AMR parsing: extract lexicon (step 1)

• Goal: given sentence-graph training examples, extract mapping from phrases to graph fragments

*The boy wants to go to New York City.*

• Rule-based system (14 rules)

[Flanigan/Thomson/Carbonell/Dyer/Smith, 2014]
AMR parsing: concept labeling (step 2)

- Semi-Markov model: segment new sentence into phrases and label each with at most one **concept graph**

```
φ  boy  want-01  φ  visit-01
The  boy  wants  to  visit
```

```
“New”

“York”

“City”
```

[Flanigan/Thomson/Carbonell/Dyer/Smith, 2014]
AMR parsing: concept labeling (step 2)

- Semi-Markov model: segment new sentence into phrases and label each with at most one concept graph

- Dynamic programming for computing best labeling
AMR parsing: connect concepts (step 3)

- Build a graph over concepts satisfying constraints
  
  All concept graphs produced by labeling are used
  At most 1 edge between two nodes
  For each node, at most one instance of label
  Weakly connected

[Flanigan/Thomson/Carbonell/Dyer/Smith, 2014]
AMR parsing: connect concepts (step 3)

- Build a graph over concepts satisfying constraints
  
  All concept graphs produced by labeling are used
  At most 1 edge between two nodes
  For each node, at most one instance of label
  Weakly connected

- Algorithm: adaptation of maximum spanning tree

\[\text{Flanigan/Thomson/Carbonell/Dyer/Smith, 2014}\]
Summary so far

- **Frames**: stereotypical situations that provide rich structure for understanding
Summary so far

- **Frames**: stereotypical situations that provide rich structure for understanding

- **Semantic role labeling (FrameNet, PropBank)**: resource and task that operationalize frames

- **AMR graphs**: unified broad-coverage semantic annotation
Summary so far

- **Frames**: stereotypical situations that provide rich structure for understanding

- **Semantic role labeling (FrameNet, PropBank)**: resource and task that operationalize frames

- **AMR graphs**: unified broad-coverage semantic annotation

- **Methods**: classification (featurize a structured object), structured prediction (not a tractable structure)
Food for thought

• Both distributional semantics (DS) and frame semantics (FS) involve compression/abstraction

• Frame semantics exposes more structure, more tied to an external world, but requires more supervision
Food for thought

- Both distributional semantics (DS) and frame semantics (FS) involve compression/abstraction

- Frame semantics exposes more structure, more tied to an external world, but requires more supervision

Examples to ponder:

*Cynthia went to the bike shop yesterday.*
*Cynthia bought the cheapest bike.*
Outline

Properties of language

Distributional semantics

Frame semantics

Model-theoretic semantics

Reflections
Types of semantics

Every non-blue block is next to some blue block.
Types of semantics

Every non-blue block is next to some blue block.

Distributional semantics: block is like brick, some is like every
Types of semantics

Every non-blue block is next to some blue block.

Distributional semantics: block is like brick, some is like every

Frame semantics: is next to has two arguments, block and block
Types of semantics

Every non-blue block is next to some blue block.

Distributional semantics: block is like brick, some is like every

Frame semantics: is next to has two arguments, block and block

Model-theoretic semantics: tell the difference between

1 2 3 4 and 1 2 3 4
Model-theoretic/compositional semantics

Two ideas: model theory and compositionality

Model theory: interpretation depends on the world state

Block 2 is blue.
Two ideas: model theory and compositionality

Model theory: interpretation depends on the world state

*Block 2 is blue.*
Model-theoretic/compositional semantics

Two ideas: **model theory** and **compositionality**

**Model theory**: interpretation depends on the world state

\[
\text{Block 2 is blue.}
\]

**Compositionality**: meaning of whole is meaning of parts

\[
\text{The [block left of the red block] is blue.}
\]
Model-theoretic semantics

Framework: map natural language into \textit{logical forms}
Model-theoretic semantics

Framework: map natural language into logical forms

Factorization: understanding and knowing

What is the largest city in California?

\[ \text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{population}(x)) \]
Model-theoretic semantics

Framework: map natural language into **logical forms**

Factorization: **understanding** and **knowing**

*What is the largest city in California?*

\[
\text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{population}(x))
\]

Los Angeles
Systems

Rule-based systems:

- STUDENT for solving algebra word problems [Bobrow et al., 1968]
- LUNAR question answering system about moon rocks [Woods et al., 1972]
Systems

Rule-based systems:

- STUDENT for solving algebra word problems [Bobrow et al., 1968]
- LUNAR question answering system about moon rocks [Woods et al., 1972]

Statistical semantic parsers:

- Learn from denotations [Clarke et. al, 2010; Liang et al. 2011]
Systems

Rule-based systems:

- STUDENT for solving algebra word problems [Bobrow et al., 1968]
- LUNAR question answering system about moon rocks [Woods et al., 1972]

Statistical semantic parsers:

- Learn from denotations [Clarke et. al, 2010; Liang et al. 2011]

Applications of semantic parsing:

- Question answering on knowledge bases [Berant et al., 2013, 2014; Kwiatkowski et al., 2013; Pasupat et al., 2015]
- Identifying objects in a scene [Matuszek et. al, 2012]
- Solving algebra word problems [Kushman et. al, 2014; Hosseini et al., 2014]
Components of a semantic parser

people who have lived in Chicago

Type.Person \sqcap \text{PlacesLived}.\text{Location}.\text{Chicago} \quad \{\text{BarackObama,}...\}

Parser

Learner
Components of a semantic parser

- **Grammar** (x)
- **D**
- **Model** (θ)
- **Executor** (z)
- **Parser Learner**

**people who have lived in Chicago**

Type Person \(\sqcap\) PlacesLived Location Chicago \{BarackObama,...\}

Parser Learner
Freebase

100M entities (nodes) 1B assertions (edges)

MichelleObama
- PlacesLived
- Spouse

Event21
- Location
- Type

Event8
- Gender
- Spouse
- StartDate
- Marriage

UnitedStates
- ContainedBy

Hawaii
- Type
- ContainedBy

Chicago
- Location
- ContainedBy

BarackObama
- Type
- DateOfBirth
- Profession
- PlaceOfBirth

Person
- 1961.08.04
- Politician

City
- Type

[Bollacker, 2008; Google, 2013]
Logical forms: lambda DCS

Type.Person ⊓ PlacesLived.Location.Chicago
Logical forms: lambda DCS

$\text{Type.Person} \sqsupset \text{PlacesLived.Location.Chicago}$
Logical forms: lambda DCS

Type.Person ⊓ PlacesLived.Location.Chicago
Logical forms: lambda DCS

Type.Person ⊓ PlacesLived.Location.Chicago

Person

Type

PlacesLived

Chicago

MichelleObama

Gender

Female

1992.10.03

Event8

Marriage

UnitedStates

PlaceOfBirth

Honolulu

Event3

1961.08.04

DateOfBirth

BarackObama

Profession

Politician

City

UnitedStates

ContainedBy

Hawaii

Type

USState

Type

Event21

Location

Type

Chicago

ContainedBy

PlaceOfBirth

DateOfBirth

Type

Person

Type

Spouse

StartData
Lambda DCS

Entity
Chicago
Lambda DCS

Entity
Chicago

Join
PlaceOfBirth.Chicago
Lambda DCS

Entity
Chicago

Join
PlaceOfBirth.Chicago

Intersect
Type.Person\(\sqcap\)PlaceOfBirth.Chicago
Lambda DCS

Entity
Chicago

Join
PlaceOfBirth.Chicago

Intersect
Type.Person □ PlaceOfBirth.Chicago

Aggregation
count(Type.Person □ PlaceOfBirth.Chicago)
Lambda DCS

Entity
Chicago

Join
PlaceOfBirth.Chicago

Intersect
Type.Person ⊓ PlaceOfBirth.Chicago

Aggregation
count(Type.Person ⊓ PlaceOfBirth.Chicago)

Superlative
argmin(Type.Person ⊓ PlaceOfBirth.Chicago, DateOfBirth)
Components of a semantic parser

- **Grammar**
- **Model**
- **Executor**

Components:
- People who have lived in Chicago
- Barack Obama
- Michelle Obama
- Type Person ⊓ PlacesLived Location Chicago
- {Barack Obama, ...}

Parser Learner
Generating candidate derivations

utterance ➔ Grammar ➔ derivation 1
 derivation 2
...

87
Generating candidate derivations

A Simple Grammar

(lexicon)  

Chicago ⇒ N : Chicago

people ⇒ N : Type.Person

lived ⇒ N—N : PlacesLived.Location

(join)  

N—N : r  N : z ⇒ N : r.z

(intersect)  

N : z₁  N : z₂ ⇒ N : z₁ ⊓ z₂
Derivations

A Simple Grammar

- (lexicon) Chicago ⇒ N : Chicago
- (lexicon) people ⇒ N : Type.Person
- (lexicon) lived ⇒ N—N : PlacesLived.Location
- (join) N—N : r N : z ⇒ N : r.z
- (intersect) N : z1 N : z2 ⇒ N : z1 ∩ z2

Type.Person ∩ PlaceLived.Location.Chicago

Type.Person who PlaceLived.Location.Chicago

people have PlaceLived.Location in Chicago

lived Chicago
Derivations

A Simple Grammar

(lexicon) \textit{Chicago} \Rightarrow N : \text{Chicago}
(lexicon) \textit{people} \Rightarrow N : \text{Type.Person}
(lexicon) \textit{lived} \Rightarrow N—N : \text{PlacesLived.Location}
(join) N—N : r N : z \Rightarrow N : r.z
(intersect) N : z_1 N : z_2 \Rightarrow N : z_1 \cap z_2

Type.Person \cap \text{PlaceLived.Location.Chicago}

Type.Person \textit{who} \text{PlaceLived.Location.Chicago}

\text{lexicon

\textit{people} \textit{have} \text{PlaceLived.Location} \textit{in} \text{Chicago}

\text{lexicon}

\textit{lived} \text{Chicago}
Derivations

A Simple Grammar

(lexicon) Chicago ⇒ N : Chicago
(lexicon) people ⇒ N : Type.Person
(lexicon) lived ⇒ N—N : PlacesLived.Location
(join) N—N : r N : z ⇒ N : r.z
(intersect) N : z₁ N : z₂ ⇒ N : z₁ ∩ z₂

Type.Person ∩ PlaceLived.Location.Chicago

Type.Person who

lexicon

people

have

PlaceLived.Location

in

Chicago

lived

Chicago
Derivations

A Simple Grammar

(lexicon) Chicago ⇒ N : Chicago
(lexicon) people ⇒ N : Type.Person
(lexicon) lived ⇒ N—N : PlacesLived.Location
(join) N—N : r N : z ⇒ N : r.z
(intersect) N : z1 N : z2 ⇒ N : z1 ∩ z2

Type.Person ∩ PlaceLived.Location.Chicago

people have PlaceLived.Location in Chicago
Overapproximation via simple grammars

- Modeling correct derivations requires complex rules
Overapproximation via simple grammars

- Modeling correct derivations requires complex rules
- Simple rules generate overapproximation of good derivations
Overapproximation via simple grammars

- Modeling correct derivations requires complex rules
- Simple rules generate overapproximation of good derivations
- Hard grammar rules $\Rightarrow$ soft/overlapping features

- Hard grammar rules $\Rightarrow$ soft/overlapping features
Many possible derivations!

\( x = \text{people who have lived in Chicago} \)
Many possible derivations!

\[ x = \text{people who have lived in Chicago} \]
Many possible derivations!

\[ x = \text{people who have lived in Chicago} \]
Many possible derivations!

\[ x = \text{people who have lived in Chicago} \]
Components of a semantic parser

people who have lived in Chicago

Type.Person \sqcap \text{PlacesLived.Location.Chicago} \{\text{BarackObama, ...} \}

Parser

Learner
$x$: utterance

$d$: derivation

Feature vector $\phi(x, d) \in \mathbb{R}^F$:
\( x \): utterance

\( d \): derivation

Feature vector \( \phi(x, d) \in \mathbb{R}^F \):

- apply join 1
- skipped \( \text{IN} \) 1
- \text{lived} maps to \text{PlacesLived.Location} 1
- ... ...

Scoring function:

\[
\text{Score}_\theta(x, d) = \phi(x, d) \cdot \theta
\]
Feature vector $\phi(x, d) \in \mathbb{R}^F$:

- apply join 1
- skipped IN 1
- \textit{lived} maps to \texttt{PlacesLived.Location} 1
- ...

Scoring function:

$$\text{Score}_\theta(x, d) = \phi(x, d) \cdot \theta$$

Model:

$$p(d \mid x, D, \theta) = \frac{\exp(\text{Score}_\theta(x, d))}{\sum_{d' \in D} \exp(\text{Score}_\theta(x, d'))}$$
Components of a semantic parser

Grammar $x$, Model $\theta$, Executor $y$

people who have lived in Chicago

Parser Learner

Type.Person ⊓ PlacesLived.Location.Chicago {BarackObama,...}
Goal: given grammar and model, enumerate derivations with high score
Goal: given grammar and model, enumerate derivations with high score
**Goal:** given grammar and model, enumerate derivations with high score

**Parser**
Goal: given grammar and model, enumerate derivations with high score

Use beam search: keep $K$ derivations for each cell
Components of a semantic parser

people who have lived in Chicago
Training data for semantic parsing

Heavy supervision

- What’s Bulgaria’s capital?
  Capital.Bulgaria

- When was Walmart started?
  DateFounded.Walmart

- What movies has Tom Cruise been in?
  Type.Movie □ Starring.TomCruise

...
## Training data for semantic parsing

### Heavy supervision

<table>
<thead>
<tr>
<th>What’s Bulgaria’s capital?</th>
<th>Capital.Bulgaria</th>
</tr>
</thead>
<tbody>
<tr>
<td>When was Walmart started?</td>
<td>Date.Founded.Walmart</td>
</tr>
<tr>
<td>What movies has Tom Cruise been in?</td>
<td>Type.Movie □ Starring.TomCruise</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

### Light supervision

<table>
<thead>
<tr>
<th>What’s Bulgaria’s capital?</th>
<th>Sofia</th>
</tr>
</thead>
<tbody>
<tr>
<td>When was Walmart started?</td>
<td>1962</td>
</tr>
<tr>
<td>What movies has Tom Cruise been in?</td>
<td>TopGun, VanillaSky, ...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Training intuition

Where did Mozart tupress?

Vienna
Training intuition

*Where did Mozart tupress?*

PlaceOfBirth.WolfgangMozart
PlaceOfDeath.WolfgangMozart
PlaceOfMarriage.WolfgangMozart

*Vienna*
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

Where did Hogarth tupress?
Training intuition

*Where did Mozart tupress?*

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

*Where did Hogarth tupress?*

PlaceOfBirth.WilliamHogarth
PlaceOfDeath.WilliamHogarth
PlaceOfMarriage.WilliamHogarth

London
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

Where did Hogarth tupress?

PlaceOfBirth.WilliamHogarth ⇒ London
PlaceOfDeath.WilliamHogarth ⇒ London
PlaceOfMarriage.WilliamHogarth ⇒ Paddington

London
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienn
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

Where did Hogarth tupress?

PlaceOfBirth.WilliamHogarth ⇒ London
PlaceOfDeath.WilliamHogarth ⇒ London
PlaceOfMarriage.WilliamHogarth ⇒ Paddington

London
Training intuition

Where did Mozart tpypress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna
Vienna

Where did Hogarth tpypress?

PlaceOfBirth.WilliamHogarth ⇒ London
PlaceOfDeath.WilliamHogarth ⇒ London
PlaceOfMarriage.WilliamHogarth ⇒ Paddington
London
Summary so far

• Two ideas: model theory and compositionality, both about factorization / generalization

• Modular framework: executor, grammar, model, parser, learner

• Applications: question answering, natural language interfaces to robots, programming by natural language
Food for thought

• Learning from denotations is hard; interaction between search (parsing) and learning: one improves the other — bootstrapping; don’t have good formalism yet

• Semantic parsing works on short sentences (user to computer); distributional/frame semantics has broader coverage; how to bridge the gap?
Food for thought

- Learning from denotations is hard; interaction between search (parsing) and learning: one improves the other — bootstrapping; don’t have good formalism yet

- Semantic parsing works on short sentences (user to computer); distributional/frame semantics has broader coverage; how to bridge the gap?

- Really about end-to-end training (logical forms are means to an end), captures pragmatics

- What is the best way to produce answer (blur lines between parser and executor)?
Outline

Properties of language

Distributional semantics

Frame semantics

Model-theoretic semantics

Reflections
Three types of semantics

1. Distributional semantics:
   - Pro: Most broadly applicable, ML-friendly
   - Con: Monolithic representations
Three types of semantics

1. Distributional semantics:
   - Pro: Most broadly applicable, ML-friendly
   - Con: Monolithic representations

2. Frame semantics:
   - Pro: More structured representations
   - Con: Not full representation of world
Three types of semantics

1. Distributional semantics:
   • Pro: Most broadly applicable, ML-friendly
   • Con: Monolithic representations

2. Frame semantics:
   • Pro: More structured representations
   • Con: Not full representation of world

3. Model-theoretic semantics:
   • Pro: Full world representation, rich semantics, end-to-end
   • Con: Narrower in scope

   many opportunities for synthesis
SHRDLU [1971]
Person: Pick up a big red block.
Computer: OK.
Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don’t understand which pyramid you mean.
Person: Pick up a big red block.

Computer: OK.

Person: Grasp the pyramid.

Computer: I don’t understand which pyramid you mean.

Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.

Computer: By ”it”, I assume you mean the block which is taller than the one I am holding.
Person: Pick up a big red block.

Computer: OK.

Person: Grasp the pyramid.

Computer: I don’t understand which pyramid you mean.

Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.

Computer: By ”it”, I assume you mean the block which is taller than the one I am holding.

Computer: OK.

Person: What does the box contain?

Computer: The blue pyramid and the blue block.

Person: What is the pyramid supported by?

Computer: The box.
Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don’t understand which pyramid you mean.
Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.
Computer: By ”it”, I assume you mean the block which is taller than the one I am holding.
Computer: OK.
Person: What does the box contain?
Computer: The blue pyramid and the blue block.
Person: What is the pyramid supported by?
Computer: The box.

- End-to-end
  (syntax, semantics, dialogue, planning)
A number of people have suggested to me that large programs like the SHRDLU program for understanding natural language represent a kind of dead end in AI programming. Complex interactions between its components give the program much of its power, but at the same time they present a formidable obstacle to understanding and extending it. In order to grasp any part, it is necessary to understand how it fits with other parts, presents a dense mass, with no easy footholds. Even having written the program, I find it near the limit of what I can keep in mind at once.

— Terry Winograd (1972)
Memory networks [2014]

**Goal**: learn to do reasoning tasks **end-to-end** from scratch

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.
Where is the football? A: playground
Memory networks [2014]

**Goal**: learn to do reasoning tasks **end-to-end** from scratch

- John is in the playground.
- Bob is in the office.
- John picked up the football.
- Bob went to the kitchen.
- Where is the football? **A: playground**

- Pure learning based, so much simpler than SHRDLU (+)
- Currently using artificial data, simpler than SHRDLU (-)
Memory networks [2014]

**Goal**: learn to do reasoning tasks *end-to-end* from scratch

- John is in the playground.
- Bob is in the office.
- John picked up the football.
- Bob went to the kitchen.
- Where is the football? A: playground

- Pure learning based, so much simpler than SHRDLU (+)
- Currently using artificial data, simpler than SHRDLU (-)
- How to get **real data** and how much do we need to get to SHRDLU level?
- Can the model incorporate some **structure** without getting too complex?
The future

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child’s?
The future

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child’s?

It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc.
The future

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child’s?

It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc.

— Alan Turing (1950)
Questions?