Compositional Semantics

Tom Kwiatkowski, Nov 2015

Compositional Semantics

• Building sentence meanings out of word meanings
  • Logical compositional semantics
  • Distributional compositional semantics
• Common NLP tasks include:
  • Question answering
  • Sentiment analysis
  • Entailment
  • Summarization & Paraphrasing

Compositional Semantics

“The meaning of a sentence is a function of the meanings of its parts and their mode of syntactic composition” - Gottlob Frege

Using logic

likes(slav, caviar)

Slav
likes
caviar

In vector space

friendly but not great

This Lecture

• Model theoretic semantics and logical forms
• Parsing into logical forms
• Question answering with logical forms
• Distributional compositional semantics
• Combining word embeddings for phrase similarity
• Learning compositional models for sentiment analysis
Model Theoretic Semantics

Declarative sentences state facts about the world:
- Mario’s is a Restaurant
- Artusi and Majiko are expensive restaurants
- Artusi is vegetarian
- Artusi and Mario’s serve Italian food
- Majiko is a Sushi restaurant
- Emily is a vegetarian
- Slav likes expensive restaurants

We can ask questions of the world:
What restaurants do Emily and Slav and David all like?

Objects denote elements of the domain
Properties denote sets of elements
Relations denote sets of tuples of elements

We can use logical forms to define the contents of the model.
Artusi and Majiko are expensive restaurants

Mario's and Artusi serve Italian food

Slav likes expensive restaurants

We use all of first order predicate logic (FOPL)
Model Theoretic Semantics

Model

| Domain          | entities = { slav, emily, david, marios, italian, sushi, majiko, artusi } |
| Properties      | restaurant = { artusi, majiko, marios } |
| Properties      | expensive = { artusi, majiko } |
| Properties      | vegetarian = { artusi } |
| Relations       | likes = (slav, artusi), (slav, majiko), (emily, artusi), (david, artusi), (david, marios) |
| Relations       | cuisine = (marios, italian), (artusi, italian), (majiko, sushi) |

Emily is vegetarian
\[ \forall x. \neg \text{vegetarian}(x) \implies \neg \text{likes}(\text{emily}, x) \]

The model can’t represent every relation. FOPL helps.

Model Theoretic Semantics

Model

| Domain          | entities = { slav, emily, david, marios, italian, sushi, majiko, artusi } |
| Properties      | restaurant = { artusi, majiko, marios } |
| Properties      | expensive = { artusi, majiko } |
| Properties      | vegetarian = { artusi } |
| Relations       | likes = (slav, artusi), (slav, majiko), (emily, artusi), (david, artusi), (david, marios) |
| Relations       | cuisine = (marios, italian), (artusi, italian), (majiko, sushi) |

Question

What restaurant do slav emily, and david all like?
\[ \lambda x. \text{likes}(\text{slav}, x) \land \text{likes}(\text{emily}, x) \land \text{likes}(\text{david}, x) \]

L.F. e->t Artusi

Semantic Parsers

Semantic parsers map sentences onto logical forms that can be used to interact with an external world:

Show me flights from New York to London on a Monday morning.

\[ \lambda x. \exists y. \text{flight}(x) \land \text{from}(x, NY) \land \text{to}(x, LON) \land \text{day}(x, MON) \land \text{time}(x, y) \land y < 12.00 \]

BA 893
Learning To Compose Logical Forms

Constituency parsing
Annotation of internal nodes are logical forms

Semantic Parsers

Words can map to:

- Objects
  
  \[ \text{slav} :: \text{slav} \]

- Properties
  
  \[ \text{rest} :: \lambda x.\text{restaurant}(x) \]

- Relations
  
  \[ \text{likes} :: \lambda x\lambda y.\text{likes}(y, x) \]

- Logical operators
  
  \[ \text{all} :: \lambda f\lambda g\lambda x.f(x) \Rightarrow g(x) \]

- FOPL
  
  \[ \text{before} :: \lambda x\lambda y\exists a\exists b.\text{time}(x, a) \wedge \text{time}(y, b) \wedge a < b \]

The \(\lambda\)-calculus gives us a clean way of combining semantic components:

\[ \lambda x\lambda y.\text{likes}(y, x) \quad \text{caviar} \quad \text{slav} \]

Slav likes caviar
The \( \lambda \)-calculus gives us a clean way of combining semantic components:

\[
\lambda x\lambda y.\text{likes}(y, x) \quad \text{caviar} \quad \text{slav}
\]

\[
\lambda y.\text{likes}(y, \text{caviar}) \quad \text{slav}
\]

\[
\text{likes}(\text{slav}, \text{caviar})
\]

Slav likes caviar

A syntactic tree tells us how to order function application:

\[
\text{likes}(\text{slav}, \text{caviar})
\]

\[
\lambda y.\text{likes}(y, \text{caviar})
\]

We can also build syntactic parses and semantic analyses concurrently using:

- Dependency compositional semantics (DCS),
- Synchronous context free grammars (SCFGs).

The Combinatory Categorial Grammar (CCG) has a transparency between syntax and semantics that makes this easy:

\[
\text{likes}(\text{slav}, \text{caviar})
\]

\[
\lambda x\lambda y.\text{likes}(y, x)
\]

\[
\text{likes}
\]

\[
\lambda y.\text{likes}(y, \text{caviar})
\]

\[
\text{caviar}
\]
Learning Semantic Parsers

Semantic parsers require us to learn a lexicon and a grammar.

The analysis of the sentence is via derivation \( d \).
Learning Semantic Parsers

The analysis of the sentence is via derivation $d$

$$score(d) = \phi(d) \times \theta$$

How many people live in Seattle

Can score derivation using a linear model or a number of alternatives...

Need to learn

Learning Semantic Parsers

Do not have access to labeled semantic parses!

S : \lambda x. (x, count(\lambda y. people(y) \land \exists ev. live(y, ev) \land in(ev, seattle)))

Learning Semantic Parsers

Often do not have access to labeled logical forms!

S : \lambda x. eq(x, count(\lambda y. people(y) \land \exists ev. live(y, ev) \land in(ev, seattle)))
Learning Semantic Parsers

May have access to some expected response.

How many people live in Seattle

620,778
Question Answering against Freebase

What character does Natalie Portman play in Star Wars? : Padme Amidala

What is the population of Seattle? : 620,778

<table>
<thead>
<tr>
<th>Name</th>
<th>Year</th>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berant et.al.</td>
<td>2013</td>
<td>Dependency compositional semantics (DCS)</td>
<td>35.7</td>
</tr>
<tr>
<td>Berant et.al.</td>
<td>2014</td>
<td>DCS with paraphrasing</td>
<td>39.9</td>
</tr>
<tr>
<td>Bordes et.al.</td>
<td>2014</td>
<td>Embed relations</td>
<td>39.2</td>
</tr>
<tr>
<td>Yih et.al.</td>
<td>2015</td>
<td>Query graph generation</td>
<td>52.5</td>
</tr>
</tbody>
</table>

WebQuestions dataset

This Lecture

- Truth conditional semantics and logical forms
- Parsing into logical forms
- Question answering with logical forms
- Distributional compositional semantics
- Combining word embeddings for phrase similarity
- Learning compositional models for sentiment

Compositional Distributional Semantics

Recall from the lecture on lexical semantics, that we can represent the meaning of a word with a vector:

Compositional Distributional semantics is a discipline in NLP that models the meanings of sentences with vectors.
Can we combine vector representations of words to represent phrases?

- Multiplication and addition are very impoverished composition functions
- If we have training data and a suitable loss, maybe we can learn a better composition function $f$

$$ p = f(b, c) $$
Stanford Sentiment Treebank

- Contains movie reviews
- Every node has a sentiment label
- Can use these labels to learn a composition function

Recurrent Neural Tensor Model

\[ p = f(b, c) \]

\[ E(\theta) = \sum_i \sum_j t_{ij} \log y_{ij} + \lambda \| \theta \|^2 \]

Minimize cross entropy between predicted distribution \( y \) and target distribution \( t \):

Socher et. al. 2013
### RNTN for Sentiment Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>Root Accuracy</th>
<th>Model description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>41.0</td>
<td>Bag of words</td>
</tr>
<tr>
<td>Vector Avg</td>
<td>32.7</td>
<td>Bag of vectors</td>
</tr>
<tr>
<td>RNTN</td>
<td>45.7</td>
<td>Recurrent connections using parse tree</td>
</tr>
<tr>
<td>Forest Conv</td>
<td>51.0</td>
<td>Stacked convolutions using parse forest</td>
</tr>
</tbody>
</table>

### LSTM for Email Response

- The LSTM is a sequence-to-sequence model
- The ‘meaning’ of a sentence is a vector built by the encoder
- The output is built by the decoder from this vector
Compositional semantics focuses on building sentences meanings out of word meanings.