Semantic Roles

8 April 2021
Assignment 3 deadline extended to Monday

Assignment 4 out soon

Project details are online, including timeline

(But I'll be revising it…)
Where were we?
Let’s take a look at the results for Exercise 8!
Picking up from last time:
Word similarity
Synonymy is a binary relation – two words are either synonymous or not.

*Word similarity* (or *word distance*) is a looser metric: Two words are more similar (less distant) if they share more features of meaning.
Properly, similarity is a relation between senses:

Instead of saying “bank is similar to fund”

We say

bank\textsuperscript{1} is similar to fund\textsuperscript{3}

bank\textsuperscript{2} is similar to slope\textsuperscript{5}

But we’ll compute similarity over both words and senses.
We often distinguish *word similarity* from *word relatedness*.

*Similar* words are near-synonyms;

*Related* words can be related in any way.

E.g.,

*car* and *bicycle* are similar.

*car* and *gasoline* are related but not similar.
Two classes of similarity algorithms

Distributional algorithms

Do words have similar distributional contexts?
Distributional vector semantics, as we saw previously.

Thesaurus-based algorithms

Are words “nearby” in WordNet (or Roget’s, e.g)?
Do words have similar glosses (definitions)?
Path-based similarity

Two concepts (senses/synsets) are similar if they’re nearby in thesaurus hierarchy.

That is, there’s a short path between them.

Concepts have path 1 to themselves.
Refinements to path-based similarity

\[ \text{pathlen}(c_1, c_2) = 1 + \text{the number of edges in the shortest path in the hypernym graph between the sense nodes } c_1 \text{ and } c_2 \]

\[ \text{simpath}(c_1, c_2) = \frac{1}{\text{pathlen}(c_1, c_2)} \]

\[ \text{wordsim}(w_1, w_2) = \max_{c_1 \in \text{senses}(w_1), c_2 \in \text{senses}(w_2)} \text{simpath}(c_1, c_2) \]

*If we don’t know what senses are being used, assume it’s the most similar senses.*
Example: path-based similarity

\[
simpath(c_1, c_2) = \frac{1}{\text{pathlen}(c_1, c_2)}
\]

E.g.,

\[
simpath(\text{nickel, coin}) = \frac{1}{2} = 0.5
\]
\[
simpath(\text{fund, budget}) = \frac{1}{2} = 0.5
\]
\[
simpath(\text{nickel, currency}) = \frac{1}{4} = 0.25
\]
\[
simpath(\text{nickel, money}) = \frac{1}{6} = 0.17
\]
\[
simpath(\text{coinage, Richter scale}) = \frac{1}{6} = 0.17
\]
Problem with basic path-based similarity

Assumes each link represents a uniform distance

But *nickel* to *money* seems to use to be closer than *nickel* to *standard*

Nodes high in the hierarchy are very abstract.

Instead, we want a metric that represents the cost of each edge independently

counts words connected only through abstract nodes as less similar
Problem with basic path-based similarity

Assumes each link represents a uniform distance

But *nickel* to *money* seems to use to be closer than *nickel* to *standard*

Nodes high in the hierarchy are very abstract.

Instead, we want a metric that represents the cost of each edge independently
counts words connected only through abstract nodes as less similar
Information content similarity metrics

Let’s define $P(c)$ as the probability that a randomly selected word in a corpus is an instance of concept $c$.

Formally, there is a distinct random variable, ranging over words, associated with each concept in the hierarchy.

For a given concept, each observed noun is either
- a member of that concept with probability $P(c)$
- not a member of that concept with probability $1 - P(c)$

All words are members of the root node (*entity*):

$P(\text{root}) = 1$

The lower a node in hierarchy, the lower its probability.
Information content similarity

Train by counting in a corpus

Each instance of \textit{hill} counts toward the frequency of \textit{natural elevation}, \textit{geological formation}, \textit{entity}, etc.

Let \text{words}(c) be the set of all words that are children of node \(c\)

\[
\text{words(geological formation)} = \{\text{hill, ridge, grotto, coast, cave, shore, natural elevation}\}
\]
\[
\text{words(natural elevation)} = \{\text{hill, ridge}\}
\]

Then the probability that a word is an instance of concept \(c\) is:

\[
P(c) = \frac{\sum_{w \in \text{words}(c)} \text{count}(w)}{N}
\]
Information content similarity

WordNet hierarchy augmented with probabilities $P(c)$:
Information content and probability

The *self-information* of an event is how much we learn by knowing it.

This is also called its *surprisal* – how surprised we are to know it.

The more surprising something is, the more it tells us when it happens.
Information content and probability

We measure self-information in *bits*:

\[ I(w) = -\log_2 P(w) \]

Suppose I flip a fair coin, where \( P(\text{heads}) = 0.5 \).

How many bits of information do I learn?

\[ I(\text{heads}) = -\log_2 0.5 = -\log_2 1/2 = \log_2 2 = 1 \text{ bit} \]

Suppose I flip a biased coin, where \( P(\text{heads}) = 0.8 \).

I don’t learn as much:

\[ I(\text{heads}) = -\log_2 0.8 \approx 0.32 \text{ bits} \]
Information content: definitions

Information content of a node (concept):

$$IC(c) = -\log P(c)$$

The *Lowest common subsumer* (or *most informative subsumer*) $LCS(c_1, c_2)$ is the most informative (lowest) node in the hierarchy that is a hypernym of both $c_1$ and $c_2$. 

- 1.3 bits entity 0.395
- 5.9 bits inanimate-object 0.167
- 9.1 bits natural-object 0.0163
- 9.1 bits geological-formation 0.00176
- 15.7 bits

- 0.000113 natural-elevation shore 0.0000836
- 0.0000189 hill coast 0.0000216
Using information content for similarity

The similarity between two words is related to their common information: The more two words have in common, the more similar they are.

The Resnik method:

Measure the common information as the information content of the lowest common subsumer of the two nodes

$$\text{sim}_{\text{resnik}}(c_1, c_2) = -\log P(\text{LCS}(c_1, c_2))$$

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. IJCAI.

Problems with thesaurus-based methods

We don’t have a thesaurus for every language

Even if we do, many words are missing

They rely on hyponym info:
  Strong for nouns, but lacking for adjectives and even verbs

Alternative

  Distributional methods for word similarity
  Vector semantics!
Semantic roles
Sasha broke the window.

Pat opened the door.
Sasha broke the window.

Pat opened the door.
Sasha broke the window.

*breaker* breaker
*broken thing* broken thing

Pat opened the door.

*opener* opener
*opened thing* opened thing

**breaker / opener**
volitional actors
often animate
have direct causal responsibility
for their events

**broken thing / opened thing**
often inanimate
affected by an event
Sasha broke the window.
agent  theme

Pat opened the door.
agent  theme

agent
volitional actors
often animate
have direct causal responsibility
for their events

theme
often inanimate
affected by an event
Thematic roles are semantic generalizations over the specific roles that occur with specific verbs.

That is, takers, givers, eaters, makers, doers, and killers all have something in common.

Yes, they end in -er.

But also they’re the agents of the actions.

We can generalize across other roles as well to come up with a small set of such general roles.
Why?

XYZ corporation *bought* the stock.

They *sold* the stock to XYZ corporation.

The stock *was bought* by XYZ corporation.

The *purchase* of the stock by XYZ corporation…

The stock *purchase* by XYZ corporation…
Syntax is not enough

Mary loaded the truck with hay.

Hay was loaded onto the truck by Mary.
Predicates (*bought, sold, purchase*) represent an event.

Semantic roles express the abstract role that arguments of a predicate can take in the event.
<table>
<thead>
<tr>
<th>Thematic Role</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>The volitional causer of an event</td>
<td><em>The waiter</em> spilled the soup.</td>
</tr>
<tr>
<td>EXPERIENCER</td>
<td>The experiencer of an event</td>
<td><em>John</em> has a headache.</td>
</tr>
<tr>
<td>FORCE</td>
<td>The non-volitional causer of the event</td>
<td><em>The wind</em> blows debris from the mall into our yards.</td>
</tr>
<tr>
<td>THEME</td>
<td>The participant most directly affected by an event</td>
<td>Only after Benjamin Franklin broke <em>the ice</em>...</td>
</tr>
<tr>
<td>RESULT</td>
<td>The end product of an event</td>
<td>The city built a <em>regulation-size baseball diamond</em>...</td>
</tr>
<tr>
<td>CONTENT</td>
<td>The proposition or content of a propositional event</td>
<td>Mona asked &quot;<em>You met Mary Ann at a supermarket?</em>&quot;</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>An instrument used in an event</td>
<td>He poached catfish, stunning them <em>with a shocking device</em>...</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>The beneficiary of an event</td>
<td>Whenever Ann Callahan makes hotel reservations <em>for her boss</em>...</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The origin of the object of a transfer event</td>
<td>I flew in <em>from Boston</em>...</td>
</tr>
<tr>
<td>GOAL</td>
<td>The destination of an object of a transfer event</td>
<td>I drove <em>to Portland</em>...</td>
</tr>
</tbody>
</table>
There’s no universally agreed-upon set of roles.

Items with the “same” role (e.g., *instrument*) may not behave quite the same.

Often to formally define a thematic role it needs to be fragmented.

*Intermediary instruments* can appear as subjects:

*The cook opened the jar with the new gadget.*
*The new gadget opened the jar.*

*Enabling instruments* cannot:

*Shelly ate the sliced banana with a fork.*
*The fork ate the sliced banana.*
Questions linguists face are:

- What exactly is a role?
- What is the right set of roles?
- Are such roles universal?
Options embraced in computational linguistics include “go big” and “go small”:

PropBank uses fewer roles, but instantiated in ways that are specific to an individual predicate.

FrameNet defines many roles, which are specific to a group of predicates.
FrameNet

HYPOTHESIS: People understand things by performing mental operations on what they already know. Such knowledge can be described in terms of information packets called frames.

In FrameNet, roles are specific to frames – situations covered by a set of predicates including nouns, verbs, and adjectives.
Frame semantics

Lansky Departing left Australia to study the piano at the Royal College of Music

Student

Institution

Subject

Source

Purpose

Object
“Change position on a scale” frame

<table>
<thead>
<tr>
<th>Core Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>DIFFERENCE</td>
</tr>
<tr>
<td>FINAL_STATE</td>
</tr>
<tr>
<td>FINAL_VALUE</td>
</tr>
<tr>
<td>INITIAL_STATE</td>
</tr>
<tr>
<td>INITIAL_VALUE</td>
</tr>
<tr>
<td>ITEM</td>
</tr>
<tr>
<td>VALUE_RANGE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Some Non-Core Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>DURATION</td>
</tr>
<tr>
<td>SPEED</td>
</tr>
<tr>
<td>GROUP</td>
</tr>
</tbody>
</table>

Figure 20.3 The frame elements in the change.position.on.a.scale frame from the FrameNet Labelers Guide (Ruppenhofer et al., 2016).
“Change position on a scale” frame

Lexical units:

accelerated.a, advance.v, balloon.v, climb.v, contract.v, contraction.n, decline.n, decline.v, decrease.n, decrease.v, depressed.a, depression.n, diminish.v, dip.v, double.v, down.prep, drop.v, dwindle.v, edge.v, elevated.a, elevation.n, escalation.n, explode.v, explosion.n, fall.n, fall.v, fluctuate.v, fluctuation.n, gain.n, gain.v, grow.v, growing.a, growth.n, hike.n, increase.n, increase.v, increasingly.adv, jump.v, lower.v, move.v, mushroom.v, plummet.v, reach.v, rise.n, rise.v, rocket.v, shift.n, shift.v, skyrocket.v, slide.v, soar.v, swell.v, swing.v, triple.v, tumble.n, tumble.v
FrameNet also codes relationships between frames, like inheritance or causation.

For example, the `Cause_change_of_position_on_a_scale` frame is linked to the `Change_of_position_on_a_scale` frame by the `cause` relation.
PropBank

While roles in FrameNet are specific to frames, roles in PropBank are specific to individual verbs.
PropBank roles, in particular

fall.01

**Arg1**: Logical subject, patient, thing falling

**Arg2**: Extent, amount fallen

**Arg3**: start point

**Arg4**: end point, end state of *Arg1*
PropBank roles, in general

**Arg0**: Proto-agent
- Volitional involvement in event or state
- Sentience (and/or perception)
- Causes an event or change of state in another participant
- Movement (relative to position of another participant)

**Arg1**: Proto-patient
- Undergoes change of state
- Causally affected by another participant
- Stationary relative to movement of another participant

**Arg2**: Usually benefactive, instrument, attribute, or end state

**Arg3**: Usually start point, benefactive, instrument, or attribute

**Arg4**: the end point

**Arg5**: ???

*Not that consistent; causes a problem for labeling!*
PropBank: Modifiers/adjuncts

<table>
<thead>
<tr>
<th>TEMP</th>
<th>when?</th>
<th>yesterday evening, now</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>where?</td>
<td>at the museum, in San Francisco</td>
</tr>
<tr>
<td>DIR</td>
<td>where to/from?</td>
<td>down, to Bangkok</td>
</tr>
<tr>
<td>MNR</td>
<td>how?</td>
<td>clearly, with much enthusiasm</td>
</tr>
<tr>
<td>PRP/CAU</td>
<td>why?</td>
<td>because ..., in response to the ruling</td>
</tr>
<tr>
<td>REC</td>
<td></td>
<td>themselves, each other</td>
</tr>
<tr>
<td>ADV</td>
<td>miscellaneous</td>
<td></td>
</tr>
<tr>
<td>PRD</td>
<td>secondary predication</td>
<td>...ate the meat raw</td>
</tr>
</tbody>
</table>
NomBank

Similar to PropBank, but for nouns

Inventories of semantic roles are usually limited to verbs (agent, theme, recipient, beneficiary, path, …)

When possible, role definitions are consistent across parts of speech.

E.g., PropBank’s frame file for the verb *decide* is used for NomBank’s annotation of the noun *decision*. 
Semantic role labeling
Semantic role labeling: Task definition

Given a sentence, label each token as

- part of an argument or
- the predicate of the sentence or
- none.

Mr. Stromach wants to resume a more influential role in **running** the company.
Training/development/test sets come mostly from PropBank or FrameNet.

FrameNet:

[You]  can’t  [blame]  [the program]  [for being unable to identify it]
COGNIZER   TARGET   EVALUER   REASON

PropBank:

[The San Francisco Examiner]  issued  [a special edition]  [yesterday]
ARG0  TARGET  ARG1  ARGM-TMP
Role Identification

Classification models similar to WSD

Mr. Stromach wants to resume a more influential role in **running** the company.
Role Labeling

Sentence spans:
- Mr. Stromach
- a more influential role
- the company
- influential role
- company

Potential roles:
- ARG0
- ARG1
- ARG2
- ARG3
- ARG4
- NONE

Best matching between spans and roles
Score can come from any classifier (linear, SVM, NN)
A simple modern algorithm

```python
function SEMANTICROLELABEL(words) returns labeled tree
    parse ← PARSE(words)
    for each predicate in parse do
        for each node in parse do
            featurevector ← EXTRACTFEATURES(node, predicate, parse)
            CLASSIFYNODE(node, featurevector, parse)
```
How do we decide what’s a predicate?

If we’re just doing PropBank verbs:

Choose all verbs
Possibly removing light verbs* (from a list)

If we’re doing FrameNet (verbs, nouns, adjectives, etc.):

Choose every word that was labeled as a target in training data

* A verb that has little semantic content of its own and forms a predicate with some additional expression, which is usually a noun. Common verbs in English that can function as light verbs are do, give, have, make, and take.
Semantic Role Labeling

[Diagram showing a tree structure with labeled parts, such as S, VP, NP, VBD, NP, PP-TMP, and their corresponding roles and tokens like TARGET, ARG0, ARG1, ARGM-TMP, The, San Francisco, issued, a, special, edition, around, noon, yesterday.]
Features

Headword of constituent
Examiner

Headword POS
NNP

Voice of the clause
Active

Subcategorization of predicate
VP → VBD NP PP

Named Entity type of constituent
ORGANIZATION

First and last words of constituent
The, Examiner

Linear position, clause re: predicate
Before
Path Features

Path in the parse tree from the constituent to the predicate

Examiner: NP↑ S ↓ VP ↓ VBD
<table>
<thead>
<tr>
<th>Frequency</th>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.2%</td>
<td>VB↑VP↓PP</td>
<td>PP argument/adjunct</td>
</tr>
<tr>
<td>11.8</td>
<td>VB↑VP↑S↓NP</td>
<td>subject</td>
</tr>
<tr>
<td>10.1</td>
<td>VB↑VP↓NP</td>
<td>object</td>
</tr>
<tr>
<td>7.9</td>
<td>VB↑VP↑VP↑S↓NP</td>
<td>subject (embedded VP)</td>
</tr>
<tr>
<td>4.1</td>
<td>VB↑VP↓ADVP</td>
<td>adverbial adjunct</td>
</tr>
<tr>
<td>3.0</td>
<td>NN↑NP↑NP↓PP</td>
<td>prepositional complement of noun</td>
</tr>
<tr>
<td>1.7</td>
<td>VB↑VP↓PRT</td>
<td>adverbial particle</td>
</tr>
<tr>
<td>1.6</td>
<td>VB↑VP↑VP↑VP↑S↓NP</td>
<td>subject (embedded VP)</td>
</tr>
<tr>
<td>14.2</td>
<td></td>
<td>no matching parse constituent</td>
</tr>
<tr>
<td>31.4</td>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>
3-step version of semantic role labeling algorithm

1. **Pruning**: use simple heuristics to prune unlikely constituents

2. **Identification**: a binary classification of each node as an argument to be labeled or a NONE

3. **Classification**: a 1-of-N classification of all the constituents that were labeled as arguments by the previous stage
Why add pruning and identification steps?

Algorithm is looking at one predicate at a time

Very few of the nodes in the tree could possibly be arguments of that one predicate

Imbalance between

- positive samples (constituents that are arguments of predicate)
- negative samples (constituents that are not arguments of predicate)

Imbalanced data can be hard for many classifiers

So we prune the very unlikely constituents first, and then use a classifier to get rid of the rest
Pruning heuristics

Add sisters of the predicate, then aunts, then great-aunts, etc.

But ignore anything in a coordination structure.
Pruning heuristics

Add sisters of the predicate, then aunts, then great-aunts, etc.

But ignore anything in a coordination structure.
Pruning heuristics

Add sisters of the predicate, then aunts, then great-aunts, etc.

But ignore anything in a coordination structure.
Pruning heuristics

Add sisters of the predicate, then aunts, then great-aunts, etc.

But ignore anything in a coordination structure.
Pruning heuristics

Add sisters of the predicate, then aunts, then great-aunts, etc.

But ignore anything in a coordination structure.
Pruning heuristics

Add sisters of the predicate, then aunts, then great-aunts, etc.

But ignore anything in a coordination structure

```
NP       VP
|        |
|        |
Strikes  VBD
and
mismanagement were
```

```
CC       S
|        |
|        |
and
S
```

```
NP
|       |
|       |
Premier
Ryzhkov

VBD       VP
|        |
|        |
warned

cited
```

```
S
|       |
|       |
of tough measures
```
A common final stage: joint inference

The algorithm so far classifies everything locally

Each decision about a constituent is made independently of all others

But this can’t be right: There are lots of global or joint interactions between arguments.

WE.g., constituents in FrameNet and PropBank must be non-overlapping.

A local system may incorrectly label two overlapping constituents as arguments

PropBank does not allow multiple identical arguments

Labeling one constituent ARG0 should increase the probability of another being ARG1
How to do joint inference

Reranking

The first-stage SRL system produces multiple possible labels for each constituent.

The second stage classifier determines the best *global* label for all constituents.

Often a classifier that takes all the inputs along with other features (sequences of labels)
More complications: FrameNet

We need an extra step to find the frame
More complications: FrameNet

We need an extra step to find the frame

```plaintext
function SEMANTICROLELABEL(words) returns labeled tree

parse ← PARSE(words)
for each predicate in parse do
  for each node in parse do
    featurevector ← EXTRACTFEATURES(node, predicate, parse)
    CLASSIFYNODE(node, featurevector, parse)
```
More complications: FrameNet

We need an extra step to find the frame

```
function SEMANTICROLELABEL(words) returns labeled tree

parse ← PARSE(words)
for each predicate in parse do

    for each node in parse do
        featurevector ← EXTRACTFEATURES(node, predicate, parse)
        CLASSIFYNODE(node, featurevector, parse)
```
More complications: FrameNet

We need an extra step to find the frame

```plaintext
function SEMANTICROLELABEL(words) returns labeled tree

parse ← PARSE(words)
for each predicate in parse do
    predicatevector ← EXTRACTFRAMEFEATURES(predicate, parse)
    frame ← CLASSIFYFRAME(predicate, predicatevector)
    for each node in parse do
        featurevector ← EXTRACTFEATURES(node, predicate, parse)
        CLASSIFYNODE(node, featurevector, parse)
```
More complications: FrameNet

We need an extra step to find the frame

```
function SEMANTICROLELABEL(words) returns labeled tree

parse ← PARSE(words)
for each predicate in parse do
    predicatevector ← EXTRACTFRAMEFEATURES(predicate, parse)
    frame ← CLASSIFYFRAME(predicate, predicatevector)
for each node in parse do
    featurevector ← EXTRACTFEATURES(node, predicate, parse)
    CLASSIFYNODE(node, featurevector, parse, frame)
```
Demo

https://demo.allennlp.org/semantic-role-labeling
Summary

Semantic roles are a level of shallow semantics for representing events and their participants

Intermediate between parses and full semantics

Two common architectures, for various languages

FrameNet: frame-specific roles
PropBank: Proto-roles

Current systems extract by

Parsing a sentence
Finding predicates in the sentence
For each one, classify each parse tree constituent
Acknowledgments

The lecture incorporates material from:

Sam Bowman, New York University
Nancy Ide, Vassar College
Daniel Jurafsky, Stanford University
Jonathan May, University of Southern California