Using Frame Semantics in Natural Language Processing

Apoorv Agarwal, Daniel Bauer
Owen Rambow
Center for Computational Learning Systems
Columbia University
rambow@ccls.columbia.edu
Goal of Talk

• Claim: FrameNet is ready to be used in NLP
• But: it’s not quite like using a POS tagger!
• Show 2 case studies
  1. Use FrameNet parser (SEMAFOR) in system for extraction of social network from narrative text
  2. Use FrameNet resource (frame hierarchy) in system for generating pictures from text (“semantic grounding”)
• Draw some conclusions
Case Study 1: Extraction of Social Networks from Text

- Apoorv Agarwal
- Agarwal & Rambow 2010 (EMNLP); Agarwal et al 2013 (IJCNLP); Agarwal et al 2014 (EACL)
“Social Network”: Two Meanings

1. Set of binary social ties between people
   – Integral part of our cognition
   – Also found in other species

2. Software or website which allows people to expand and manage their social network
“Social Network”:
Two Meanings

1. Set of binary social ties between people
   – Integral part of our cognition
   – Also found in other species

2. Software or website which allows people to expand and manage their social network
Social Network Extraction

• Basic claim: social network made up of social events

• Social Event:
  – 2 people interacting: 
    *John and Mary had dinner*
  – 1 person observing another: 
    *John saw Mary across the room*

• No social event: *I know both John and Mary*
Social Events

Interaction

Bi-directional relationship

Directed relationship

Observation
From Social Events to Social Networks

• Modeling social events
  – 2 people => 2 nodes
  – unidirectional or bidirectional link

• Social network: union of many social events
Example:

*Alice in Wonderland*
Example:

Alice in Wonderland
Example:

*Alice in Wonderland*
Example:

Alice in Wonderland
Other Applications

• Diplomatic cables
• Social network of the Taliban Government of Afghanistan (as described
Social Events and Semantics

• The definition of “social event” is both:
  – broad (many different types of events) and
  – specific (requirements on cognitive state of two people)

• Precise semantics important:
  – What is the situation?
    • Buying an apple from someone  Interaction
    • Seeing someone across a room  Observation
    • Exceeding someone in height  ---
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and Semantics

• Social events are both:
  – broad (many different types of event) and
  – specific (requirements on cognitive state of two people)

• Specific semantics important:
  – What is the situation?
    • Buying an apple from someone
    • Seeing someone across a room
    • Exceeding someone in height
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and Semantics

• Social events are both:
  – broad (many different types of event) and
  – specific (requirements on cognitive state of two people)

• Specific semantics important:
  – What is the situation?
    • Buying an apple from someone
    • Seeing someone across a room
    • Exceeding someone in height
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and Semantics

• Social events are both:
  – broad (many different types of event) and
  – specific (requirements on cognitive state of two people)

• Specific semantics important:
  – What is the situation?
    • Buying an apple from someone  Interaction
    • Seeing someone across a room  Observation
    • Exceeding someone in height  ---
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and Semantics

• Social events are both:
  – broad (many different types of event) and
  – specific (requirements on cognitive state of two people)

• Specific semantics important:
  – What is the situation?
    • Buying an apple from someone
    • Seeing someone across a room
    • Exceeding someone in height
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and Semantics

• Social events are both:
  – broad (many different types of event) and
  – specific (requirements on cognitive state of two people)

• Specific semantics important:
  – What is the situation?
    • Buying an apple from someone  
    • Seeing someone across a room  
    • Exceeding someone in height
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and Semantics

• Social events are both:
  – broad (many different types of event) and
  – specific (requirements on cognitive state of two people)

• Specific semantics important:
  – What is the situation?
    • Buying an apple from someone  Interaction
    • Seeing someone across a room  Observation
    • Exceeding someone in height ---
  – What are the roles of the two people?
    • Sanjeev talked to Miriam about Sam and about José
Social Events and FrameNet

• Hypothesis: a complete correct FrameNet analysis would be helpful for detecting and classifying social events
• We use SEMAFOR as a black box semantic parser
Problem 1 with Using FrameNet: Compositionality

- FrameNet (and thus SEMAFOR) does not create a single semantic representation for a whole sentence.
- We construct single tree from spans and analyses contained in spans.
- This is not complex unless there are errors.
Coleman said he bought drugs from the defendants.
Problem 2 with Using FrameNet: Coverage

- FrameNet does not have complete lexical coverage
  - Nor does SEMAFOR
- Need to keep unanalyzed parts of sentence in syntactic representation (= deep dependency)
Coleman said he bought adulterated oxy from the defendants.
Problem 3 with Using FrameNet: Analysis Accuracy

- Semantic parsing is hard
- SEMAFOR makes mistakes (though getting better...)
- Need to have machine learning approach which can learn from errorful representations
- Solution: use tree kernels (and graph kernels) on semantic trees (and also on syntactic trees)
Some Results

Data: ACE, annotated for Social Events

<table>
<thead>
<tr>
<th>Model</th>
<th>Detection</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Bag of Words</td>
<td>.34</td>
<td>.39</td>
</tr>
<tr>
<td>Semantics+Rules</td>
<td>.51</td>
<td>.10</td>
</tr>
<tr>
<td>Syntax+ML</td>
<td>.46</td>
<td>.75</td>
</tr>
<tr>
<td>Syntax+Semantics+ML</td>
<td>.48</td>
<td>.74</td>
</tr>
</tbody>
</table>
Case Study 2: Text-to-Scene Generation

• Daniel Bauer, Bob Coyne, Julia Hirschberg, Richard Sproat
• WordsEye system: Coyne & Sproat 2001
Motivation: Grounded Semantics

• Grounded semantics applicable to many interesting NLP applications:
  – discourse modeling, virtual worlds, embedded systems, robotics, text-to-scene generation,…

• Need lexical semantic theory and resource to bridge between language and grounded semantics
  – Starting point: FrameNet

• Issue: what type of semantics?
Current WordsEye System

Input text: The very large silver ball is on the table. The ground is shiny. The table is under the small willow tree. The lion is one foot in front of the table. The lion is facing the ball. It is cloudy.
Text-to-Scene Generation

• Generate a graphical scene from a textual description that depicts the content of the description

• Types of descriptions:
  – Low-level (primitive spatial relations):
    The man is on the floor. He is kneeling.  
    He is holding the sponge.  
    The bucket is near the man.
  – High-level:
    The man is washing the floor
Levels of Scene Description

• High-Level:
  – Functional view: Who does what to whom
  – **Wash**(washer:x1, theme:x2)
  – Descriptions involves action/event verbs, complex entities...

• Low-level:
  – Realization view: How is it done?
    (graphical: what does it look like)
  – **On**(figure:x1, ground:x2), **Grasp**(grasper:x1, theme:x3),
    **Reach**(reacher: x1, ground:x2), **Kneel**(kneeler:x1)
  – Just spatial relations

• One high-level description → many low-level descriptions
• In graphics generation, low-level description ground high-level descriptions
Translating from High-Level Descriptions to Low-level Graphical Representations

• Requires three sources of knowledge:
  – Lexical Knowledge
    • Textual description to high-level semantic representation
  – Graphical Knowledge
    • Translate high-level semantics into low-level graphical relations
  – Factual Knowledge
    • Guide translation, rule out impossible/unlikely graphical representations.

• Use a common frame-based representation to bridge between language, functional and graphical meaning
  – “VigNet”

• Starting point: Frame Semantics[Fillmore, 1982]
Lexical Knowledge: FrameNet

- Bridge language and high-level semantic representation
- Can build on FrameNet
  - High-level semantics / functional view.
  - Mapping from syntax / lexicon to frame semantics by providing example annotations for each frame
  - Frame-to-Frame relations

\[
\begin{align*}
\text{[Mary]}_{\text{buyer}} & \quad \text{bought} & \quad \text{Commerce} \quad \text{buy} & \quad \text{[an apple]}_{\text{goods}} & \quad [\text{for $1]}_{\text{money}} \\
\text{Subj} & & \text{Obj} & & \text{PP(for)}
\end{align*}
\]
Problems with FrameNet: Compositionality

\[
\text{[Mary]}_{\text{buyer}} \text{ bought } \text{Commerce buy } \text{[an apple]}_{\text{goods}} \text{ [for $1]}_{\text{money}}
\]

- FrameNet annotations are ‘shallow’ (no semantic objects as arguments, just text spans)
  - Does not represent semantics of whole sentence in one structure
- Does not represent co-reference
- Solution: notion of “instance” of a frame
  - Allows representing semantics of whole sentence
  - Allows co-reference
Instantiating Frames: Types and Instances

- Frames describe concept types.
- When lexical items evoke a frame in a description, the frame is **instantiated**.
- All frames carry a ‘self’ frame element, which is bound to the instance of the frame.
- When instantiating a frame, bind all the frame elements to instances (which may be defined by another frame).

```
Commerce buy(self: i6, buyer: i4, seller: i1, goods: i2, money: i3)
Person(self:i4)
Person(self:i1)
Apple(self:i2)
Money(self:i3 amount:.... )

‘ Mary bought an apple for $1.’
```
Graphical Knowledge

- Need knowledge about arrangement of 3D models to depict a situation/event
- Low-level semantics, realization view
- Non-compositionality of verb meaning:
  - Correct visualization of verb depends on verb and its arguments

‘The man washed the floor’

‘The man washed the apple’
Many Wash Options

- Wash Small Fruit
  - Wash w/ sponge
- Wash Ground
  - Wash w/ mop
- Wash Hair
- Wash Vehicle
- Wash dog
- Wash window
Note: Lexicalization Differences Across Languages

• Verbs for ‘wash’ don’t simply mean ‘remove dirt’; English/French/Egyptian Arabic:
  – John **washed** the apple/laver/gasal
  – John **washed** the floor/laver, nettoyer/masaH
  – John **brushed** his teeth/laver/gasal
    • **brush** encodes the instrument!

• Support verb-noun constructions often not compositional (Persian)
Graphical Knowledge: Vignettes

- Vignettes extend frames by:
  1. Adding new specialized frames (extend frame hierarchy) based on selection restrictions for frame elements
  2. Optionally introducing new frame elements that participate in the visualization
  3. Decomposing into sub-frames:
     - link to specific 3D model types (frames describing entities)
     - describe graphical structure of a scene (frames describing events/situations)

<table>
<thead>
<tr>
<th>Commerce_counter( buyer, goods, money, seller)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ISA) Commerce_buy at_counter(partcpt1:buyer, partcpt2:seller, counter:c)</td>
</tr>
<tr>
<td>on(figure:goods, ground:c)</td>
</tr>
<tr>
<td>on(figure:money, ground:c)</td>
</tr>
</tbody>
</table>
Graphical Knowledge: Vignette Decomposition

- (temporal) subframe relation in FrameNet:

- New frame-to-frame relation `subframe_parallel`.
Factual (World) Knowledge

• Some ontological information already encoded in frame-to-frame relations (inheritance)
• In addition frame definitions for entity types need:
  - non-graphical properties of objects / attributes
  - information about parts
  - world knowledge (‘apples grow on trees’, `apples are bought in stores or markets’)
The man washes the stage with a sponge.
The man washes the stage with a sponge.
The man washes the stage with a sponge.

- **Man**(self)
- **Human**(self:self)
- **Male**(self:self)

- **Stage**(self)
- **Floor**(self:self)
- **SpongeTool**(self)

Grounded semantics through vignettes
The man washes the stage with a sponge.
The man washes the stage with a sponge.
The man washes the stage with a sponge.
The man washes the stage with a sponge.
The man washes the stage with a sponge.
Status of the VigNet Resource

VigNet currently contains:

• a small set of primitive spatial relations (on, next-to (direction and distance), in, direction..)
• small set (about 30) ‘abstract‘ vignettes
  – holding/touching target or patient, using handheld
• instruments, using stationary machine, human poses...
• several hundred verbal vignettes inheriting from and parameterizing abstract vignettes (ongoing...)
• about 2000 nominal vignettes mapping to about 3000 3D models (with physical attributes, parts, affordances)
• about 80 location vignettes (all rooms, including fixtures/affordances)
Summary

• FrameNet used successfully in two applications
  – Social network extraction
  – Text-to-scene generation
• Good level of abstraction
• But...
Conclusion from Both Case Studies

- FrameNet comes out of a lexicographic tradition
- NLP is needs semantic representations based on FrameNet, not just lexical entries with examples
- Great interest in sophisticated compositional (lexical) semantic representations now (AMR)
- Need for FrameNet to define such a representation (this is not super complicated)
- NLP needs annotated FrameNet-based semantic representation
- FrameNet parsers should produce trees, not annotate spans
Thank You!
Frame decompositions are declarative.

Simultaneously define properties of frame element fillers and restrict fillers to instances of frames that define this property. Can create frame elements for properties.

Or use ‘self’ frame element to define properties of frames for entity types.

<table>
<thead>
<tr>
<th>commerce_counter( buyer, goods, money, seller)</th>
</tr>
</thead>
<tbody>
<tr>
<td>size(figure:goods, size:small)</td>
</tr>
<tr>
<td>animate(self:seller)</td>
</tr>
<tr>
<td>animate(self:buyer)</td>
</tr>
</tbody>
</table>

(ISA) commerce

at_counter(partcpt1:buyer, partcpt2:seller, counter:c)
on(figure:goods, ground:c)
on(figure:money, ground:c)

apple( )

(ISA) fruit

size(figure:self, size:small)
shape(figure:self, shape:round)
Examples of high/low level descriptions (via Mechanical Turk)

Low-level:
A man is using the telephone.
The man is wearing a yellow vest.
The man has blonde hair.
The man has white skin.
A white rodent is inside a cage.
The cage is on a table.
The phone is on the table.
The cage has a handle.
A safe is in the background of the room.

#High-level:
The man is a scientist working with white rodents.

#High-level:
The man is talking to another scientist.

#High-level:
The man feels guilt at imprisoning a white rodent.

• Acquire typical language (hi/low) for 100 comic book scenes
• Each scene described by 5 different Turkers
Goal

• Automatic conversion from text to 3D scene
  – Text $\rightarrow$ semantics (FrameNet & semantic parsing)
  – High-level semantics $\rightarrow$ low-level semantics
  – Low-level semantics $\rightarrow$ 3D scene (WordsEye)
Example: Decomposing Meaning of *Wash*

*The man washed the floor.*

- Agent HOLDS sponge
- Agent NEAR bucket
- Agent KNEEL on floor

*The man washed the apple.*

- Agent FACING sink
- Agent HOLD patient (not shown)
- Agent IN-FRONT-OF sink
Example: Grounding of “of”

CONTAINMENT: bowl of cats

PART-OF: head of the cow

SIZE-OF: height of horse is..

ARRANGEMENT-OF: stack of cats

SUBSTANCE-OF: horse of stone

REPRESENTING: Picture of the girl
Outline

• Motivation and Goal
• Basic Approach
  – Semantic nodes, semantic relations, and Vignettes
• Formalizing Vignettes for grounded semantics
  • FrameNet and its limitations
  • Adding Vignette Semantics to FrameNet
• Populating VigNet
• Conclusion
Representing and Grounding Meaning

- **Semantic nodes** denote entities and types:
  - Types (*House, teenager, polar bear, …*)
  - Elaborated types (*A tiny house with a metallic front door*)
  - Events (*John washing the apple*)
  - Individuals (*Barack Obama, Homer Simpson, my house*)

- **Semantic relations** are applied to semantic nodes to give them meaning.
  - HABITAT-OF(*polar-bear, Arctic*)
  - SIZE-OF(*house-1, tiny*)
  - WASH(*JOHN, APPLE*)
  - Need inventory of semantic relations (*FrameNet frames!*)

- **Meta-relations** translate between semantic relations.
  - Used to decompose/ground meaning with Vignettes (*FrameNet f2f relations*)
Meta-relations to invoke and ground Vignettes

- **Vignettes** are semantic relations representing different ways of grounding semantics.
  - Vignettes inherit (via meta-relations) from parent relation
    - WASH ➔ WASH-SMALL-FRUIT
    - Add new semantic roles as needed for grounding (e.g., sink)
    - Selectional restrictions constrain types of the semantic roles
      - WASH-SMALL-FRUIT: SizeOf(PATIENT, small)
    - Retain high-level meaning and semantic roles from parent
  - Vignettes decompose into grounded relations using meta-relations
    - WASH-SMALL-FRUIT(man, apple) ➔
      - IN-FRONT-OF(man, sink)
      - HOLDS(man, apple)
      - FACING(man, sink)
      - REACH-TOWARD(man, sink)
Example: Representing and grounding meaning

Input: *The man washed the blue tile floor*

Create semantic nodes and relations
Example: Representing and grounding meaning

Input: *The man washed the blue tile floor*

Create semantic nodes and relations

Invoke vignette
Example: Representing and grounding meaning

Input: The man washed the blue tile floor

Create semantic nodes and relations

Invoke vignette

Decompose vignette into grounded relations
**Example:** Representing and grounding meaning

**Input:** The man washed the blue tile floor

Create semantic nodes and relations

Invoke vignette

Decompose vignette into grounded relations

Apply constraints and render
Outline

• Motivation and Goal
• Basic Approach
  – Semantic nodes, semantic relations, and Vignettes
• Formalizing Vignettes for grounded semantics
  • FrameNet and its limitations
  • Adding Vignette Semantics to FrameNet
• Populating VigNet
• Conclusion
Frame Semantics [Fillmore 1982]

FrameNet [Baker et al. 1998, Ruppenhofer et al. 2010]

• Word meaning can only be understood by referring to conceptual structure evoked by it.

• Frames are cognitive schemas describing relations between state or event participants.

• Role of syntax and lexicon:
  • Valence patterns of a lexical item map syntactic arguments to frame elements.

• FrameNet: Lexicographic resource containing frames, their relations and linguistic realizations.

• Can we use FrameNet to ground word meaning via conceptual structure/frames?
Problem 1: FrameNet frames are too general

- *walk* and *swim* are grouped in the Self_Motion frame, but have different graphical realizations.
- *Wash an apple, ... the floor, ... the car, ... hands, ... hair, etc.* have different graphical realizations.

Problem 2: FrameNet frames have little internal structure

- Frames are only defined by a list of Frame Elements (possibly with semantic types), frame-to-frame relations and an informal plaintext definition.
- No grounding: Need framework to decompose frames into (graphical) primitives.
Problem 3: FrameNet analyses of sentences are “shallow”

- In annotations frame elements are bound to text spans, not semantic objects.

Problem 4: FrameNet contains only lexical knowledge.

- Only source of conceptual knowledge: (very limited and rigid) semantic types, frame-to-frame relations.
- No world knowledge whatsoever.
  - But can use frames to assert world knowledge (once problem 3 is solved).
    - typical_habitat(lifeForm: Sloth, habitat: Rain_Forest).
Outline

• Motivation and Goal

• Basic Approach
  – Semantic types, semantic relations, and Vignettes

• Formalizing Vignettes for grounded semantics
  • FrameNet and its limitations
  • Adding Vignette Semantics to FrameNet

• Populating VigNet

• Using VigNet and conclusion
Frame Decompositions and Selectional Restrictions

• Need to decompose frames into subframes.
  – Spatial decomposition of locations/actions/complex objects.
  – Express conceptual knowledge
    • “Apples are small, round, edible and sweet fruits”.
    • Use for complex selectional restrictions on frame elements.
      “something edible”, “something small and round”, “a sweet fruit”

• Decomposition mechanism:
  – New frame-to-frame relation Subframe_parallel.
  – New Self frame element for each frame (assert conceptual knowledge)
Frame Decompositions

- FrameNet’s subframe and precedes relation

- New frame-to-frame relation **Subframe_parallel**.

Grounded semantics through vignettes
Semantic Nodes

• Problems:
  – Frames are conceptual descriptions, frame elements are just ‘variables’.
  – In annotations frame elements are filled with text spans.
  – But scenes are composed of concrete objects and relations.

• Frames are instantiated to semantic nodes.
• Semantic nodes are attached to frame elements.
• ‘self’ frame element points to instance of its frame.
Semantic Nodes and Frames

• Semantic Nodes:
  – Frame instances, types of objects or relations
  – Two types:
    1. Discourse referents (existentially quantified).
    2. Semantic classes for knowledge representation (universally quantified).

• Frames:
  – ‘Constructors’ for semantic nodes/intensions/descriptions of concepts
Outline

• Motivation and Goal
• Basic Approach
  – Semantic nodes, semantic relations, and Vignettes
• Formalizing Vignettes for grounded semantics
  • FrameNet and its limitations
  • Adding Vignette Semantics to FrameNet
• Populating VigNet
• Conclusion
Populating VigNet

• Use and extend FrameNet itself
  – Add frames for all lexical items
  – Add new graphically-oriented frames for grounding
  – Add valence patterns from VerbNet, etc.
  – Improve linguistic coverage using corpora

• Import WordsEye dictionary and object info

• Acquire knowledge with Mechanical Turk

• Mine relational knowledge from corpora and existing resources

• Add relevant knowledge by hand
  – Define abstract vignettes with parameterized decompositions
Import WordsEye Dictionary

- 2,200 3D objects and 10,000 images represented by semantic nodes
  - Many specialized subtypes and compound objects (e.g. a goat head mounted on a plank of wood)
  - Semantic relations between these nodes (CONTAINING, IS-A, COLOR-OF, HANDLE-OF, PART-OF ...)
- 15,000 lexical items corresponding to these nodes (including related terms and attributes)

### 3D objects

- **3D Object**
  - Associated words: weight
  - bench, sportsware, bench
  - Size: 0.0 feet

- **3D Object**
  - Associated words: furniture, television, furnishing, appliance, display device, communication device
  - Size: 4.0 feet

- **3D Object**
  - Associated words: coffee table, table
  - Size: 3.5 feet

- **3D Object**
  - Associated words: kitchen table, patio table, table
  - Size: 5.0 feet

- **3D Object**
  - Associated words: table, furniture
  - Size: 3.0 feet

- **3D Object**
  - Associated words: bar stool, stool
  - Size: 3.0 feet

### 2D Images and textures

- **B&W drawings**
- **Texture Maps**
- **Artwork**
- **Photographs**
Acquire Knowledge with Mechanical Turk
(Masoud Rouhizadeh, Margit Bowler, Jack Crawford) (cite Masoud’s paper)

- Default locations and parts
  - Show picture of 3D objects. Turkers provide locations/parts
    CONTAINING(container=SCHOOLHOUSE, contents=LOCKER)
    CONTAINING(container=SCHOOLHOUSE, contents=DESK)
    CONTAINING(container=SCHOOLHOUSE, contents=BLACKBOARD)
    HABITAT-OF(habitat=MEADOW, inhabitant=BUSH)
    HABITAT-OF(habitat=MEADOW, inhabitant=BIRD)
    HABITAT-OF(habitat=MEADOW, inhabitant=WILDFLOWER)

- Location Vignettes
  - Show picture of different types of rooms. Turker does:
    1. Identifies main objects
    2. Specifies spatial relations for those objects
    3. Identifies subobject types using 3D library (e.g. kitchen table versus picnic table)

- Action Vignettes
  - Provide action sentence and location vignette picture/objects. Turker does:
    1. Identifies additional objects and participants
    2. Identifies subobject types using 3D library
    3. Specifies spatial relations, facial expressions, and poses for those objects
Inferring Relations from Corpora
Sproat (2001): Inferring the environment in a text-to-scene conversion system

- Acquire default locations, times, and seasons for actions
  - Target terms: rooms (e.g. bedroom), time of day (e.g. morning), season (e.g. winter)
  - Corpora of 415 million words of English text with POS tagging
  - Extract tuples for verb-object (eg wash face) and verb-prep-object (get into bed) utilizing POS tags.
  - Compute association between the tuples and each of the target terms.

<table>
<thead>
<tr>
<th>Locations</th>
<th>Time of day</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tuple</strong></td>
<td><strong>Target term</strong></td>
</tr>
<tr>
<td>Wash clothes</td>
<td>Laundry room</td>
</tr>
<tr>
<td>Wash hands</td>
<td>Bathroom</td>
</tr>
<tr>
<td>Drive car</td>
<td>Garage</td>
</tr>
<tr>
<td>Go to bathroom</td>
<td>Bathroom</td>
</tr>
<tr>
<td>Brush teeth</td>
<td>Bathroom</td>
</tr>
<tr>
<td>Run car</td>
<td>Garage</td>
</tr>
<tr>
<td>Wash dishes</td>
<td>Kitchen</td>
</tr>
<tr>
<td>Go to store</td>
<td>Laundry room</td>
</tr>
<tr>
<td>Go to bed</td>
<td>Bedroom</td>
</tr>
<tr>
<td>Take shower</td>
<td>Bathroom</td>
</tr>
<tr>
<td>See in kitchen</td>
<td>Kitchen</td>
</tr>
<tr>
<td>Sit on sofa</td>
<td>Bedroom</td>
</tr>
<tr>
<td>Sit on bed</td>
<td>Bedroom</td>
</tr>
<tr>
<td>Sit on toilet</td>
<td>Bathroom</td>
</tr>
<tr>
<td>Sit at table</td>
<td>Kitchen</td>
</tr>
<tr>
<td>Hold knife</td>
<td>Bedroom</td>
</tr>
<tr>
<td>Climb over wall</td>
<td>bedroom</td>
</tr>
<tr>
<td>Sit on floor</td>
<td>hallway</td>
</tr>
</tbody>
</table>
Outline

• Motivation and Goal
• Basic Approach
  – Semantic nodes, semantic relations, and vignettes
• Formalizing Vignettes for grounded semantics
  • FrameNet and its limitations
  • Adding Vignette Semantics to FrameNet
• Populating VigNet
• Conclusion
Note: Verb Homophony and Polysemy

• FE disambiguates homophonous verb senses:
  – *The pilot banked to the right before the crash*
  – *The pilot banked at Chase before the crash*

• FE disambiguates polysemous verb senses:
  – *Patricia glared over at me*
  – *The sun glared down on us*
Conclusion – what’s new

• Vignette Semantics – a grounded lexical semantic theory based on FrameNet
  – Semantic nodes representing types and individuals
  – FrameNet frames provide inventory of semantic relations that can be applied to semantic nodes to express meaning
  – Vignettes are semantic relations that bridge between high-level and low-level semantic relations (via F2F meta-relations)
    • Up: Selectional restrictions leveraging semantic nodes and relations
    • Down: Subframe_parallel frame decomposition

• A lexical-semantic resource VigNet under development
  – Various techniques for populating Vignet

• Applying VigNet in text-to-scene conversion
This material is based upon work supported by the National Science Foundation under Grant No.IIS-0904361
Example: Representing and grounding meaning

Input: The man washed the blue tile floor

Create semantic nodes and relations

Vignette decomposition
Example: Representing and grounding meaning

Input: *The man washed the blue tile floor*

Create semantic nodes and relations

Vignette decomposition

Apply constraints and render
Leverage knowledge and abstract vignettes to increase coverage

• *John picked the apple*
  – Use “pick-from-tree” vignette based on selectional restriction of apple being from a tree.
  – Use knowledge to determine a) type of tree and b) location:
    – OriginOf(entity:*apple*, source:*apple-tree*)
    – Habitat(entity:*apple-tree*, location:*field*)
  – Decompose to:
    – InFrontOf(figure:*John*, ground: *apple-tree*)
    – ReachUp(agent:*John*, target: *apple*)
    – Attached(figure:*apple*, ground:*apple-tree*, part: *branch*)
    – LocatedIn(figure:*apple-tree*, ground:*field*)
WordsEye: Create Scenes from Low-Level Descriptions
Online system at http://bit.ly/wordseye

Input text: Santa Claus is on the white mountain range. He is blue. It is cloudy. A large yellow illuminator is in front of him. The alien is in front of him. The mountain range is shiny.
Grounding meaning through Vignettes

**Input:** The truck chased the man down the long road

Create semantic nodes and relations

Vignette decomposition
Grounding meaning through Vignettes (spend time)

Input: The truck chased the man down the long road

Create semantic nodes and relations

Vignette decomposition

Apply constraints and render (contextual objects added)
polar bear: a white colored bear that lives in the Arctic

Semantic node for “polar bear”

Semantic relation giving meaning to semantic node for polar bear
The old gray polar bear ate the fish

Semantic nodes for a particular polar bear and fish
The old gray polar bear ate the fish

Use meta-relations to do grounding
Representing Word Meaning

- **polar bear**: a bear with white fur that lives in arctic

- **teenager**: human between 13 and 19 years old
Using Semantic Nodes

• Semantic nodes can be:
  – Temporary or permanently stored in a knowledge base
  – Linked to lexicon, discourse, or ‘anonymous’

• Asserting world knowledge example:
  – \texttt{sn\_15123}: \texttt{Distance}(self: \texttt{sn\_15123}, ground:\texttt{earth}, figure:\texttt{moon},
    distance: \texttt{sn\_1314})
  – \texttt{sn\_1314}: \texttt{Range}(self: \texttt{sn\_1314}, min: 400,000km, max: 500,000km)
Conceptual Issues

• What goes in frames? Conceptual vs world knowledge issues.

• Deducing vignette structure from noisy field data? Or designing structure similar to FrameNet?

• Composing action vignettes with location vignettes.
What’s new about VigNet? Related work.

• Knowledge representation, Ontologies
  – Resources: ConceptNet/OpenMind, LabelMe, Cyc
  – Theory: FOL, Description Logics

• Grounded semantics
  – Event Logics, Conceptual Dependency, PAR/actionary

• Lexical semantic resources
  – FrameNet, Verbnet, WordNet, Ontonotes

• Text-to-scene
## Using VigNet: Vignettes corresponding to different senses of “Of” using selectional restrictions

<table>
<thead>
<tr>
<th>Text (A of B)</th>
<th>Conditions</th>
<th>Semantic Relation</th>
<th>Vignette Decomposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowl of cherries</td>
<td>A=container, B=plurality-or-mass</td>
<td>CONTAINER-OF (bowl, cherries)</td>
<td>A contains B</td>
</tr>
<tr>
<td>Slab of concrete</td>
<td>A=entity, B=substance</td>
<td>MADE-OF (slab, concrete)</td>
<td>A HAS-TEXTURE B</td>
</tr>
<tr>
<td>Picture of girl</td>
<td>A=representing, B=entity</td>
<td>REPRESENTS (picture, girl)</td>
<td>A HAS-TEXTURE ?C ?C REPRESENTS B</td>
</tr>
<tr>
<td>Arm of the chair</td>
<td>A=part-of(B), B=entity</td>
<td>PART-OF (chair, arm)</td>
<td></td>
</tr>
<tr>
<td>Height of the tree</td>
<td>A=size-property, B=Phys-entity</td>
<td>DIMENSION-OF (height, tree)</td>
<td></td>
</tr>
<tr>
<td>Stack of plates</td>
<td>A=arrangement, B=plurality</td>
<td>GROUPING-OF (stack, plates)</td>
<td>B ALIGNED-IN-DIRECTION A</td>
</tr>
</tbody>
</table>
Example: Sentence meaning diagram

- **John walked from the road to the old house with the red front door**
Example: create dependency structure

Input: The truck chased the man down the long road

Parse tree

Dependency relations

Using VigNet and future work
FrameNet for text-to-scene?

- Frames intuitively provides correct level of abstraction. However:
  1. Text to scene (and other applications) need more fine-grained sense distinctions. FrameNet frames are not graphically motivated.
  2. FrameNet frames do not have internal structure / no grounding.
  3. Shallow semantics: No obvious way of instantiating frames to formally represent the meaning of a text passage.
  4. FrameNet only records lexical and shallow conceptual knowledge. How can we assert world-knowledge?
Selectional Restrictions contd.

Apple(self)

Fruit(self)
Shape_of(figure:self, shape:spherical)
Size_of(figure:self, size:small)

Wash_small_fruit (washer, theme, sink)

theme = x
x = Fruit
Size_of(figure:x, size:small)
sink = Sink
Washer = Sentient

Front-Of(figure: washer, figure: sink)
Facing(figure:washer, figure:sink)
Grasp(grasper:washer, theme:theme)
Reach(reacher:washer, target:sink)

• Abbreviation: Frame(self:x) \(\rightarrow\) \(x=\text{Frame}\)
• Allow use of variables.
Frame-to-Frame Relations contd.

– Network of directed relations between frames.
  • Causative_of
  • Inchoative_of
  • Perspective_on

![Diagram of FrameNet and its limitations]