Deep Lexical Semantics, Case, Constructions, and FrameNet

Jerry R. Hobbs
USC/ISI
Marina del Rey, CA
Outline

1. Deep Lexical Semantics
2. Case
3. Constructions
4. FrameNet
The Big Picture

Speaker's Plan: Reasoning about goals and beliefs

Reasoning with World Knowledge
(Local pragmatics: coreference, predicate-strengthening, etc.)

Discourse Coherence

Observable to be explained

Logical Form

Syntax

Segment(w,e)

Syn(w1,e1,--)

CoRel(e1,e2,e)

Syn(w2,e2,--)

goal(i,c)

Cog'(c,u,e)

utter(i,u,w)
Utterance is an intentional act, intended to cause the hearer u to think about the conventional meaning e of string of words w.
The Big Picture

Speaker's Plan: Reasoning about goals and beliefs

Pragmatics: Explain why speaker i wants to convey information e
The Big Picture

Explain adjacency of discourse segments as conveying coherence relations.
Syntax is the explanation of adjacency as predicate-argument relations.
The best explanation of the occurrence of individual words is that they are intended to convey their conventional meanings.
The Big Picture

Syn(w₁,e₁,-,-)  Syn(w₂,e₂,-,-)  CoRel(e₁,e₂,e)

Segment(w,e)

The best explanation of the logical form usually solves problems of coreference, predicate-strengthening, etc.

Speaker's Plan: Reasoning about goals and beliefs

goal(i,c)  Cog'(c,u,e)

Shared variables so mutual influence

Reasoning with World Knowledge

Logical Form

Syntax

Discourse Coherence
The Big Picture

Observables to be explained

Segment \((w, e)\)

Observable to be explained

 uttered \((i, u, w)\)

Speaker's Plan:
Reasoning about goals and beliefs

Logical Form

Discourse Coherence

Shared variables so mutual influence

Syntax

Reasoning with World Knowledge
(Local pragmatics: coreference, predicate-strengthening, etc.)

CoRel \((e_1, e_2, e)\)

goal \((i, c)\)

Cog' \((c, u, e)\)
We need a common representation scheme for all these levels:

Ontologically promiscuous first-order logic

Reify states and events (eventualities) for scope-free logical forms, representational adequacy, etc.
Speaker's Plan:
Reasoning about goals and beliefs

Reasoning with World Knowledge
(Local pragmatics: coreference, predicate-strengthening, etc.)

Discourse Coherence

Observable to be explained

Logical Form

Syntax

Segment \( (w, e) \)

\( \text{Syn}(w_1, e_1, -,-) \)

\( \text{Syn}(w_2, e_2, -,-) \)

\( \text{CoRel}(e_1, e_2, e) \)

\( \text{utter}(i, u, w) \)

\( \text{goal}(i, c) \)

\( \text{Cog}'(c, u, e) \)
“Levels” of Processing

Syntax and Compositional Semantics:

\[ \text{Syn}(w_1,x,N,-,-,-,-) \land \text{Syn}(w_2,e,V,x,N,-,-,-) \]
\[ \rightarrow \text{Syn}(w_1 \, w_2,e,V,-,-,-,-) \]

\[ \text{Syn} \] is specialization of \text{mean}

\[ \text{Syn}(w_2,e,V,x,N,-,-,-) \] says string or VP \( w_2 \) describes situation \( e \) provided an NP subject describing \( x \) can be found in the right place

The rule says if \( w_1 \) is an NP describing \( x \) and \( w_2 \) is a VP describing \( e \) (if it had a subject \( x \)), then the concatenation \( w_1 \, w_2 \) describes \( e \) (and doesn’t need a subject)

HPSG converted into FOL
“Levels” of Processing

Lexical Axioms:

\[ \text{kill'(e,x,y) & living-thing(y) } \rightarrow \text{Syn(“kill”,e,V,x,N,y,N)} \]

- pred-arg structure (incl word sense)
- selectional constraint
- spelling or phonology
- category
- subcategorization
Lexical Decomposition:

\[
\text{kill}(x, y) \iff \text{cause}(x, \text{become}(\text{not}(\text{alive}(y))))
\]

\[
\text{kill}'(e_1, x, y) \iff \text{cause}'(e_1, x, e_2) \& \text{changeTo}'(e_2, e_3) \\
\quad \& \text{not}'(e_3, e_4) \& \text{alive}'(e_4, y)
\]

Need this to understand:

My roommate killed all my plants.  
He didn’t water them once while I was gone.
“Levels” of Processing

Lexical Decomposition:

cause’(e,x,become(not(alive(y)))) <---> kill’(e,x,y)

Core Theories:

water --> nourish; enable(nourish,alive)
“Levels” of Processing

Syntax and Compositional Semantics:

$$\text{Syn}(w_1, x, N, -,-,-,-) \land \text{Syn}(w_2, e, V, x, N, -,-,-) \implies \text{Syn}(w_1 \ w_2, e, V, -,-,-,-)$$

Lexical Axioms:

$$\text{kill}'(e, x, y) \land \text{living-thing}(y) \implies \text{Syn}(\text{“kill”}, e, V, x, N, y, N)$$

Lexical Decomposition:

$$\text{cause}'(e, x, \text{become}(\text{not}(\text{alive}(y)))) \iff \text{kill}'(e, x, y)$$

Core Theories:

$$\text{water} \implies \text{nourish}; \quad \text{enable}(\text{nourish}, \text{alive})$$
A Sentence Interpreted

Syntax:
- Syn(“My roommate killed my plant.”, e, -, -)
- Syn(“My roommate”, x, -, -)
- Syn(“killed my plant.”, e, x, -)
- Syn(“killed”, e, x, y)
- Syn(“my plant.”, y, -, -)
- Syn(“roommate(x,i)
- Syn(“killed”, e, x, y)
- Syn(“my plant.”, y, -, -)
- Syn(“killed”, e, x, y)
- Syn(“my plant.”, y, -, -)

Lexical Axioms:
- roommate(x,i)
- Syn(“killed”, e, x, y)
- Syn(“my plant.”, y, -, -)
- kill’(e, x, y)
- plant(y)

Lexical Decomposition:
- cause(x, become(not(alive(y))))
- cause(not(nourish(x,y)), not(alive(y)))

Core Theories:
- cause(not(water(x,y)), not(nourish(x,y)))

He didn’t water them.
Outline

1. Deep Lexical Semantics
2. Case
3. Constructions
4. FrameNet
The Case for Case

The relations between predicates and arguments can be classified into a small number of categories.

Fillmore’s original list:

- Agentive
- Instrumental
- Dative
- Factitive
- Locative
- Objective
- Comitative
- Benefactive

This proposal had a huge appeal among computational linguists; many lists developed, e.g., Source, Goal, (Inanimate) Cause, Time, etc.
Case and Lexical Decomposition

Chris moved the flower pots from the front yard to the back yard with a wheelbarrow.

\[
\text{cause}'(e_1,c,e_2) \land p'(e_2,w) \land \text{cause}'(e_3,e_2,e_4) \\
\land \text{change}'(e_4,e_5,e_6) \land \text{at}'(e_5,p,f) \land \text{at}'(e_6,p,b) 
\]

c causes an event involving w, which causes a change from p being at f to p being at b
Case and Lexical Decomposition

Chris moved the flower pots from the front yard to the back yard with a wheelbarrow.

\[
\text{cause}'(e_1,c,e_2) \& p'(e_2,w) \& \text{cause}'(e_3,e_2,e_4) \\
\& \text{change}'(e_4,e_5,e_6) \& \text{at}'(e_5,p,f) \& \text{at}'(e_6,p,b)
\]

Agent: the entity initiating a causal chain

Instrument: an entity mediating a causal chain

Object: (Patient, Theme) entity undergoing change of state or location

Source: beginning of the change

Goal: end of the change
Variations

Agent vs. Instrument (vs. Cause):

The tornado destroyed the barn.

Dative vs. Object:

+animate vs. -animate

Comitative:

cause’(e1, {c,d}, e2) & ....
Problem

These standard cases or semantic roles seem appropriate exactly insofar as the verb decomposes into a pattern resembling that of “move”.

“X lets Y Verb”: What is X? Agent?

not’(e₁,e₂) & cause’(e₂, x, e₃) & not’(e₃, e₄)

x is the entity that doesn’t initiate a causal chain

“X outnumbers Y”: What are X and Y? (Patient and Locative?)

==> In FrameNet case labels are idiosyncratic and only mnemonics

Me: Forget the case labels; do the decomposition
Outline

1. Deep Lexical Semantics
2. Case
3. Constructions
4. FrameNet
Constructions

A linguistic pattern whose meaning or function is not strictly predictable from the rules of compositional semantics and from the lexical semantics of its parts.

The majority:

A linguistic pattern whose conventionalized meaning or function is among the possible interpretations generated from the rules of compositional semantics and from the lexical semantics of its parts (its motivation), but would not necessarily be chosen as the correct interpretation without the convention.
“Let’s”

“Let us go.”: You and I should go together. [stilted]
       You should release us. [victims to kidnapper]

“Let’s go.”: You and I should go together.
“Let’s”

Interpret “Let’s go” as a grammatical sentence

Syn(“Let’s go”, e, V, u, a, -, -)

utter(e₀, i, u, “Let’s go”)

“You” is the addressee in utterance e₀

you(u, e₀)
“Let’s”

Contraction expanded

utter’(e₀,i,u,”Let’s go”)
“Let’s”

VP deconcatenated into Verb and Small Clause

utter’(e₀,i,u,“Let’s go”)

Syn(“Let’s go”,e,V,u,a,-,-)

Syn(“Let us go”,e,V,u,a,-,-)

Syn(“Let”,e,V,u,a,y,N,e₂,V.Tnsless.OC)

Syn(“us”,y,N.Acc,-,-,-,-)

Syn(“go”,e₂,V.Tnsless,-,-)

you(u,e₀)
“Let’s”

utter’(e₀,i,u,“Let’s go”)

Syn(“Let’s go”,e,V,u,a,,-,-)

Syn(“Let us go”,e,V,u,a,,-,-)

Syn(“Let”,e,V,u,a,y,N,e₂,V,Tnsless.OC)

Syn(“us”,y,N.Acc,,-,-,-)

Syn(“go”,e₂,V.Tnsless,-,-)

let’(e,u,y,e₂)

we(y,s,e₀)

plural(y,s)

go’(e₂,y,z₁,z₂)

“let” means let

“go” means go

“us” means we, a set s with type element y (and with i as a member)

you(u,e₀)
“Let’s”

utter’(e₀,i,u,“Let’s go”) 

Syn(“Let’s go”,e,V,u,a,-,-) 

Syn(“Let us go”,e,V,u,a,-,-) 

Syn(“Let”,e,V,u,a,y,N,e₂,V,Tnsless.OC) 

Syn(“us”,y,N.Acc,-,-,-,-) 

Syn(“go”,e₂,V,Tnsless,-,-) 

let’(e,u,y,e₂) 

we(y,s,e₀) 

plural(y,s) 

go’(e₂,y,z₁,z₂) 

Inclusive “we” is one possible interpretation of “we”
“Let’s”

utter'(e_0,i,u,"Let’s go")

Syn("Let’s go",e,V,u,a,-,-)

Syn("Let us go",e,V,u,a,-,-)

Syn("Let",e,V,u,a,y,N,e_2,V.Tnsless.OC)

Syn("us",y,N.Acc,-,-,-,)

Syn("go",e_2,V.Tnsless,-,-)

we(y,s,e_0)

plural(y,s)

go'(e_2,y,z_1,z_2)

you(u,e_0)

member(u,s)

Lexical decomposition of “let”

let'(e,u,y,e_2)

not(e_4) & cause'(e_4,u,e_5) & not'(e_5,e_2)
"Let’s"

Your not going causes the set “us” not to go

cause(e₆,e₅) & not'(e₆,e₇) & go'(e₇,u,z₁,z₂)

member(u,s)

utter'(e₀,i,u,"Let’s go")

let'(e,u,y,e₂)

plural(y,s)
go'(e₂,y,z₁,z₂)

t(_FILL,_) & cause'(e₄,u,e₅) & not'(e₅,e₂)

not(e₄)

you(u,e₀)

Syn("Let’s go",e,V,u,a,-,-)

Syn("Let us go",e,V,u,a,-,-)

Syn("Let",e,V,u,a,y,N,e₂,V.Tnsless.OC)

Syn("us",y,N.Acc,-,-,-)

Syn("go",e₂,V.Tnsless,-,-)
“Let’s”
This is the motivation for the convention, but it’s only one interpretation among many possible
"Let’s"

The convention:
You and we go as one set

- utter'(e_0, i, u, "Let’s go")
- Syn("Let’s go", e, V, u, a, -,-)
- Syn("Let us go", e, V, u, a, -,-)
- Syn("Let", e, V, u, a, y, N, e2, V, Tnsless, OC)
- Syn("us", y, N, Acc, -,-,-,-)
- Syn("go", e2, V, Tnsless, -,-)
- we(y, s, e_0)
- plural(y, s)
- go'(e_2, y, z_1, z_2)
- let'(e, u, y, e_2)
- you(u, e_0)
- member(u, s)
- cause(e_6, e_5) & not'(e_6, e_7) & go'(e_7, u, z_1, z_2)
“Let’s”

Let’s go

But we can unpack it

No, let’s stay here.

I’ve been ready to go.

No, you go without me.
“Let Alone”

“let” means not-cause-not
“let” subcategorizes for small clauses
“alone” means not part of some larger structure

Let that dog alone. It’s vicious:
Don’t cause the dog to be not alone by interacting with it.

Let “general” alone. He didn’t make colonel:
Don’t cause “general” to be not alone by entering it into the conversational record

Invert NP and Pred in small clause:
Let alone the very idea of him being general. He didn’t make colonel.

Reinterpret “let alone” as a conjunction, conveying order on a scale:
He didn’t make colonel, let alone general.
Don’t cause “general” to be not alone by entering it into the conversational record, because something lower on a scale is already false
“Let Alone”

“let alone” \( w \)

“let” \( w \) “alone”

“let”

let

not cause not

not part of composite entity

conversational record
“Let Alone”

“Let what alone?”

“Let alone” w

“let” w “alone”

“let”

let

not cause not

“You mentioned it first.”

“Forget I said it.”

w “alone”

w

“alone”

alone

not part of composite entity

conversational record
Cause Motion

He hit the ball out of the park.
They laughed him off the stage.
She blinked the snow from her eyelashes.

Interpret adjacency as “cause”, since predicate-argument relation is unavailable

Motion implicit in “from”

May reinterpret verb as subcategorizing for small clause with cause motion interpretation

Syn(\(w_1, e_1, v, \ldots\)) & Syn(\(w_2, e_2, p.\text{dir}, \ldots\)) & cause’(e, e_1, e_2)
\(\rightarrow\) syn(\(w_1w_2, e, v, \ldots\))
Outline

1. Deep Lexical Semantics
2. Case
3. Constructions
4. FrameNet
Where Does the Knowledge Come From?

Syntax:

Syn(“My roommate killed my plant.”, e, -,-)  
Syn(“My roommate”, x, -,-)  
Syn(“killed my plant.”, e, x, -)  
Syn(“killed”, e, x, y)  
Syn(“my plant.”, y, -,-)

Lexical Axioms:

He didn’t water them.

Lexical Decomposition:

cause(x, become(not(alive(y))))

Core Theories:

cause(not(nourish(x,y)), not(alive(y))))

cause(not(water(x,y)), not(nourish(x,y))))

He didn’t water them.
FrameNet into Axioms

(Ovchinnikova)

Giving Frame:

\[
\text{Giving}(e_1, x_1, x_2, x_3) & \text{Donor}(x_1, e_1) & \text{Recipient}(x_2, e_1) & \text{Theme}(x_3, e_1) \\
\rightarrow \text{give}'(e_1, x_1, x_3) \& \text{to}'(e_2, e_1, x_2)
\]

\[
\text{Giving}(e_1, x_1, x_2, x_3) & \text{Donor}(x_1, e_1) & \text{Recipient}(x_2, e_1) & \text{Theme}(x_3, e_1) \\
\rightarrow \text{hand}'(e_1, x_1, x_3) \& \text{to}'(e_2, e_1, x_2)
\]

Frame-Frame Relations:

\[
\text{Giving}(e_1, x_1, x_2, x_3) & \text{Donor}(x_1, e_1) & \text{Recipient}(x_2, e_1) & \text{Theme}(x_3, e_1) \\
\rightarrow \text{Getting}(e_2, x_1, x_2, x_3) \& \text{Source}(x_1, e_2) \& \text{Recipient}(x_2, e_2) \& \text{Theme}(x_3, e_2)
\]

Annotated corpus used to set weights

Also converted lexeme-synset relations, synset relations, derivational relations in WordNet into axioms
Size of Knowledge Base

Frame-Lexeme Axioms          49,100
Frame-Frame Axioms               5,300
Axioms from WordNet          383,000
Recognizing Textual Entailment

T: He became a boxing referee in 1964 and became most well-known for his decision against Mike Tyson, during the Holyfield fight, when Tyson bit Holyfield’s ear.

H: Mike Tyson bit Holyfield’s ear in 1964.

Cost(Int(KB=>H)) >> Cost(Int(KB+T=>H))?

Is the cost of an abductive proof of H much less with T than without T?
## Results on RTE-2

<table>
<thead>
<tr>
<th>Resource Configuration</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>No KB (lexical overlap)</td>
<td>57.3%</td>
</tr>
<tr>
<td>WordNet alone</td>
<td>59.6%</td>
</tr>
<tr>
<td>FrameNet alone</td>
<td>60.1%</td>
</tr>
<tr>
<td>WordNet + FrameNet</td>
<td>62.6%</td>
</tr>
</tbody>
</table>

No special tuning for RTE task.
Would have been 3rd in RTE-2; two leaders low 70s; most high 50s. Substantial improvements in inference engine since then.

Modest improvement with FrameNet
FrameNet a better resource for inference than WordNet
More knowledge sources are needed
Outline

1. Deep Lexical Semantics
2. Case
3. Constructions
4. FrameNet
Summary

Chuck Fillmore will be missed