Semantics: Meaning Representations and Computation

Speech and Language Processing
Selections from Chapters 17 and 18

Generic NLP System Architecture
Tasks in Computational Semantics

• Computational semantics aims to extract, interpret, and reason about the meaning of NL utterances, and includes:
  – Defining a meaning representation

  – Developing techniques for semantic analysis, to convert NL strings to meaning representations

  – Developing methods for reasoning about these representations and performing inference from them

Complexity of Computational Semantics

• Requires:
  – Knowledge of language: words, syntax, relationships b/t structure and meaning, composition procedures

  – Knowledge of the world: what are the objects that we refer to, how do they relate, what are their properties?

  – Reasoning: Given a representation and a world, what new conclusions – bits of meaning – can we infer?
More Semantic Considerations

• Which meaning representation to use?
• How to translate from syntax into the meaning representation?

• How to disambiguate word meaning? When?
• How to identify relations between words? What sort to identify?

Meaning or *Semantic* Representation Needed

• To represent questions from users
• To represent knowledge drawn from user input (e.g. location, preferences)
What Kinds of Semantic Representations Can We Use?

• Anything that allows us to
  – Answer wh-questions (What is the best restaurant in Oakland?)
  – Determine truth, e.g. for yes-no questions (Is Dave and Andy’s on Atwood?)
  – Draw inferences (If the Provost’s office is in the Cathedral and the Cathedral is the tallest building on campus, then the Provost’s office is in the tallest building on campus.)

What kinds of meaning do we want to capture?

• Categories/entities
  – Le Monde, Asian cuisine, vegetarian
• Events
  – taking a taxi, paying a check, eating a meal
• Time
  – Oct 30, next week, in 2 months, once a week
• Place
  – On the corner of 34th and Broadway
• Tense and Aspect
  – George knows how to run. George is running. George ran to the restaurant in 5 min.
• Beliefs, Desires and Intentions (BDI)
  – George wants/thinks/intends…. 
Meaning Representations

- All structures from set of symbols
  - Representational vocabulary
- Symbol structures correspond to:
  - Objects
  - Properties of objects
  - Relations among objects
- Can be viewed as:
  - Representation of meaning of linguistic input
  - Representation of state of world

Representing Meaning

\[\exists e, y \ \text{Having}(e) \land \text{Haver}(e, \text{Speaker}) \land \text{HadThing}(e, y) \land \text{Car}(y)\]

- First-order Logic
- Semantic Network
- Frame-Based

\[\text{Poss-BY} \quad \text{Haver: Speaker} \quad \text{HadThing: Car}\]
Representational Requirements

- Verifiability
- Unambiguous representations
- Canonical Form
- Inference and Variables
- Expressiveness
  - Should be able to express meaning of any NL sent

Verifiability

- Can a system compare
  - Description of state given by representation to
  - State of some world modeled by a knowledge base (kb)?
- Is the proposition encoded by the representation true?
- E.g.
  - Input: Does Maharani server vegetarian food?
  - Representation: Serves(Maharani,VegetarianFood)
  - KB: Set of assertions about restaurants
  - If representation matches in KB -> True
  - If not, False or Don’t Know
    - Is KB assumed complete or incomplete?
Unambiguous Representations

- Semantics is ambiguous:
  - *I wanna eat someplace close to UW*
    - Eat *at* someplace OR eat the restaurant

- (Final) Representation must be unambiguous
- Resolving the ambiguity?
  - Later

Canonical Form

- Input can have many meanings, and
- Many inputs can have same meaning
  - Flights from Seattle to Chicago
  - Are there any flights from Seattle to Chicago?
  - Do flights go from Seattle to Chicago?
  - Which flights are flown from Seattle to Chicago?
- Could all have different forms
  - Difficult to test in KB
- Single canonical form allows consistent verification
Inference

– Can vegetarians eat at Maharani?
– Does Maharani serve vegetarian food?
• Meanings are not identical, but
• Linked by facts in the world
• *Inference* allows system to draw valid conclusions from meaning rep. and KB
  – Serves(Maharani, VegetarianFood) =>
  – CanEat(Vegetarians, AtMaharani)

Variables

– *I want a restaurant that serves vegetarian food.*
• Can we match this in KB?
  – No restaurant specified, so no simple assertion match
• Solution:
  – Variables
    • Serves(x, VegetarianFood)
  – True if variable can be replaced by some object s.t. resulting proposition can match some assertion in KB
Meaning Structure of Language

• Human languages
  – Display basic predicate-argument structure

  – Employ variables

  – Employ quantifiers

  – Exhibit a (partially) compositional semantics

Predicate-Argument Structure

• Represent concepts and relationships
• Words behave like predicates:
  – Verbs, Adj, Adv:
    • Eat(John,VegetarianFood); Red(Ball)
• Some words behave like arguments:
  – Nouns: Eat(John,VegetarianFood); Red(Ball)
• Subcategorization frames indicate:
  – Number, Syntactic category, order of args
Semantic Roles

• Roles of entities in an event
  – E.g. John\textsubscript{AGENT} hit Bill\textsubscript{PATIENT}
• Semantic restrictions constrain entity types
  – The dog slept.
  – ?The rocks slept.

• Verb subcategorization links surface syntactic elements with semantic roles

Problems

• What exactly is a semantic role?
• What is the right/best set of roles?
• Are such roles universal?
• Can we automatically label syntactic constituents with their thematic roles?
First Order Predicate Calculus

- Not ideal as a meaning representation and doesn't do everything we want -- but a useful start
  - Supports the determination of truth
  - Supports compositionality of meaning
  - Supports question-answering (via variables)
  - Supports inference

NL Mapping to FOPC

- **Terms**: constants, functions, variables
  - **Constants**: objects in the world, e.g. Huey
  - **Functions**: concepts, e.g. sisterof(Huey)
  - **Variables**: x, e.g. sisterof(x)
- **Predicates**: symbols that refer to relations that hold among objects in some domain or properties that hold of some object in a domain
  - likes(Sue, pasta)
  - female(Sue) person(Sue)
• Logical connectives permit compositionality of meaning
  – pasta(x) → likes(Sue,x) “Sue likes pasta”
  – cat(Vera) ∧ odd(Vera) “Vera is an odd cat”
  – sleeping(Huey) ∨ eating(Huey) “Huey either is sleeping or eating or both”
• Sentences in FOPC can be assigned truth values
  – Atomic formulae are T or F based on their presence or absence in a DB (Closed World Assumption)
  – Composed meanings are inferred from DB and meaning of logical connectives

  – cat(Huey)
  – sibling(Huey, Vera)
  – cat(Huey) ∧ sibling(Huey, Vera) → cat(Vera)

• Limitations:
  – Do ‘and’ and ‘or’ in natural language really mean ‘∧’ and ‘∨’?
    Mary got married and had a baby. And then…
    Your money or your life!
  – Does ‘→’ mean ‘if’?
    If you go, I’ll meet you there. If it rains, we’ll get wet.
  – How do we represent other connectives?
    She was happy but ignorant.
• Quantifiers: \( \exists, \forall \)
  – Existential quantification: There is a unicorn in my garden. Some unicorn is in my garden.
  – Universal quantification: The unicorn is a mythical beast. Unicorns are mythical beasts.
  – How do we represent:
    • Many? A few? Several? A couple?

Lambda Expressions

• Lambda notation: (Church, 1940)
  – Allows abstraction over FOL formulas
    • Supports compositionality
  – Applied to logical terms to form exp.
    • Binds formal params to term
  – Essentially unnamed function w/params
    • Application substitutes terms for formal params
Examples

\[ x.P(x) \]
\[ x.P(x)(A) \]
\[ P(A) \]

\[ x. \ y.Near(x,y) \]
\[ x. \ y.Near(x,y)(\text{Bacaro}) \]
\[ y.Near(\text{Bacaro},y) \]
\[ y.Near(\text{Bacaro},y)(\text{Centro}) \]
\[ \text{Near}(\text{Bacaro},\text{Centro}) \]

Lambda Expressions

- Currying;
  - Converting multi-arguments preds to sequence of single argument preds
  - Why?
Lambda Expressions

• Currying;
  – Converting multi-arguments preds to sequence of single argument preds
  – Why?
    • Incrementally accumulates multiple arguments spread over different parts of parse tree

Temporal Representations

• How do we represent time and temporal relationships between events?
  – It seems only yesterday that Martha Stewart was in prison but now she has a popular TV show. There is no justice.

• Where do we get temporal information?
  – Verb tense
  – Temporal expressions
  – Sequence of presentation
Beliefs, Desires and Intentions

- Very hard to represent internal speaker states like believing, knowing, wanting, assuming, imagining
  - Not well modeled by a simple DB lookup approach so..
    - Truth in the world vs. truth in some possible world
    - George imagined that he could dance.
    - George believed that he could dance.
  - Augment FOPC with special modal operators that take logical formulae as arguments, e.g. believe, know
    - Believes(George, dance(George))
    - Knows(Bill, Believes(George, dance(George)))

Summary I

- Many difficult but interesting problems in full semantic representation
  - What do we need to represent?
  - How can we represent it?
- Current representations impoverished in many respects, e.g. FOPC
Summary II: Meaning Representation for Computational Semantics

• Requirements:
  – Verifiability, Unambiguous representation, Canonical Form, Inference, Variables, Expressiveness
• Solution:
  – First-Order Logic
    • Syntax
    • Semantics
    • Event Representation
• Next: Semantic Analysis
  – Deriving a meaning representation for an input

Syntax-driven Semantic Analysis

• Key: Principle of Compositionality
  – Meaning of sentence from meanings of parts
    • E.g. groupings and relations from syntax
• Question: Integration?
• Solution 1: Pipeline
  – Feed parse tree and sentence to semantic unit
  – Sub-Q: Ambiguity:
    • Approach: Keep all analyses, later stages will select
Simple Example

• AyCaramba serves meat.
  \[ \exists e \text{ Isa}(e, \text{Serving}) \land \text{Server}(e, \text{AyCaramba}) \land \text{Served}(e, \text{Meat}) \]

Rule-to-Rule

• Issue:
  – How do we know which pieces of the semantics link to what part of the analysis?
  – Need detailed information about sentence, parse tree
    • Infinitely many sentences & parse trees
    • Semantic mapping function per parse tree => intractable

• Solution:
  – Tie semantics to finite components of grammar
    • E.g. rules & lexicon
  – Augment grammar rules with semantic info
    • Aka “attachments”
      – Specify how RHS elements compose to LHS
Semantic Attachments

• Basic structure:
  – A-> a1….an {f(aj.sem,…ak.sem)}
  – A.sem

• Language for semantic attachments
  – Lambda calculus
    • Extends First Order Predicate Calculus (FOPC) with function application

Basic example

• Input: Maharani closed.
• Target output: Closed(Maharani)
Semantic Analysis Example

• Semantic attachments:
  – Each CFG production gets semantic attachment

• Maharani
  – ProperNoun -> Maharani {Maharani}
    • FOL constant to refer to object

  – NP -> ProperNoun {ProperNoun.sem}
    • No additional semantic info added

Semantic Attachment Example

• Phrase semantics is function of SA of children

• More complex functions are parameterized
  – E.g. Verb -> closed { λx.Closed(x) }
    • Unary predicate:
      – 1 arg = subject, not yet specified

  – VP -> Verb {Verb.sem}
    • No added information

  – S -> NP VP {VP.sem(NP.sem)}
    • Application = λx.Closed(x)(Maharani) = Closed(Maharani)
Semantic Attachment

• General pattern:
  – Grammar rules mostly lambda reductions
    • Functor and arguments
  – Most representation resides in lexicon

Refining Representation

• Add
  – Event-style model
  – Quantification
• Example II
  – Input: Every restaurant closed.
  – Target:
    $$\forall x \text{Restaurant}(x) \quad e \text{Closed}(e) \quad \text{ClosedThing}(e, x)$$
Refining Representation

- Idea: \( x\text{Restaurant}(x) \)
  - Good enough?
    - No: roughly ‘everything is a restaurant’
    - Saying something about all restaurants – nuclear scope
- Solution: Dummy predicate
  \( x\text{Restaurant}(x) \land Q(x) \)
  - Good enough?
    - No: no way to get \( Q(x) \) from elsewhere in sentence
- Solution: Lambda
  \( Q. \ x\text{Restaurant}(x) \land Q(x) \)

Updating Attachments

- Noun -> restaurant  \{ \lambda x.\text{Restaurant}(x) \}\}
- Nominal -> Noun  \{ \text{Noun.sem} \}
- Det -> Every  \{ P. Q. xP(x) \land Q(x) \}
- NP -> Det Nominal  \{ \text{Det.sem(Nom.sem)} \}
Full Representation

– Verb -> close $\{ x. eClosed(e) \text{ } ClosedThing}(e,x) \}$

– VP -> Verb $\{ \text{Verb}.sem \}$

– S -> NP VP $\{ \text{NP}.sem(\text{VP}.sem) \}$

Q. $x \text{Restaurant}(x) \quad Q(x)(\quad y. eClosed(e) \text{ } ClosedThing}(e,y))$
$x \text{Restaurant}(x) \quad y. eClosed(e) \text{ } ClosedThing}(e,y)(x)$
$x \text{Restaurant}(x) \quad eClosed(e) \text{ } ClosedThing}(e,x)$

Generalizing Attachments

– ProperNoun -> Maharani $\{ \text{Maharani} \}$

• Does this work in the new style?
– No, we turned the NP/VP application around

• New style: $\lambda x.x$(Maharani)
Strategy for Semantic Attachments

• General approach:
  – Create complex, lambda expressions with lexical items
    • Introduce quantifiers, predicates, terms
  – Percolate up semantics from child if non-branching
  – Apply semantics of one child to other through lambda
    • Combine elements, but don’t introduce new

Sidelight: Idioms

• Not purely compositional
  – E.g. kick the bucket = die
    – tip of the iceberg = beginning
• Handling:
  – Mix lexical items with constituents (word nps)
  – Allow non-compositional semantic attachments