Semantic Analysis.

inputs → Syntactic Analysis → Syntactic Structure → Symantic Analysis → Meaning Representation.

Transforming Natural language (NL) sentences into computer executable complete meaning representations (MRs) for Domain specific applications

Realistic semantic parsing currently entails domain dependence
- ⇒ Assigning each word a meaning.
- ⇒ Combining the meanings of words into sentences.

The knowledge Base (KB) have to formed in such a way that it satisfy the following properties

- Verifiability
  - the systems ability to compare representations to facts into memory.
  - Can statements be verified against a KB.

    " Does spice Island serve vegetarian food ?
    Serves (SpiceIland, Vegetarianfood).

- Unambiguousness
  - Give me the book ?
  - Which book ?

- Canonical Form.
  - Distinct inputs that mean the same thing shuld have the same meaning representation.
    - = Does Spice Island have vagetarian dishes?
    - = Do they have vagetarian ~~dishes at~~ food at Spice Island?
    - = Are vagetarian dishes served at Spice Island?
    - Does Spice Island serve vagetarian fare,

Make representations more compact
Use single representation for all senses in a ~~synset~~ synset.

Inference and Variable:

Draw a valid conclusions based on the meaning representations of inputs and its background knowledge.

Example:

I would like to find a resturant where I can get vagetarian food.

Answering this types of query requires the use of variables like

$\boxed{\text{Serves}(X, \text{Vagetarianfood})}$

Logical Representation of Sentence Meaning:

- First-order Logic (FOL) is a flexible, well-understood, and computationally tractable approach for knowledge representation.

CFG specification of the syntax of FOL representation.

Formula → AtomicFormula | Formula Connective Formula
| Quantifier Variable, ..., Formula
| ¬ Formula | (Formula)

AtomicFormula → Predicate (Term, ...)

Term → Function (Term, ...)
| Constant | Variable

Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Quantifier → $\forall$ | $\exists$

Constant → A | Vagetarian food | ...

Variable → x | y | ......

Predicate → Serves | Near | ......

Function → LocationOf | ....

Examples FOL

$\exists P(P(a) \wedge P(b))$ 2nd order

Everyone Likes chocolate.

$\forall x \{ \text{Person}(x) \Rightarrow \text{Likes}(x, \text{chocolate}))$

Someone likes chocolate

$\exists x \{ \text{Person}(x) \wedge \text{Likes}(x, \text{chocolate})$

Everyone likes chocolate unless they are allergic to it.

$\forall x (\text{Person}(x) \wedge \neg \text{allergic}(x, \text{chocolate})) \Rightarrow \text{Likes}(x, \text{chocolate}) \}$

A resturant near ICSI, serves Mexican Food.

$\exists x (\text{Resturant}(x) \wedge \text{serves}(x, \text{MexicanFood}) \wedge \text{Near}(\text{Locationof}(x), \text{Locationof}(\text{ICSI}))$

All vegetarian resturant serve vagetarian food.

$\forall x \, \text{VagetarianResturant}(x) \Rightarrow \text{serves}(x, \text{Vagetarian Food})$.

Lambda Notation

Provides a way to abstract from fully specified FOL formula in a way that will be ~~use-full~~ useful for Semantic Analysis.

Syntax of FOL to include expression of the following form: $\boxed{\lambda x \, P(x)}$

λ-reduction – Consists of simple textual replacement of λ variables.

$\boxed{\lambda x \, P(x)(A)}$

→ $\boxed{P(A)}$ After ~~θ~~ λ reduction

$\lambda x \, \lambda y \, \text{Near}(x, y) \, (\text{Kolkata})$

$\lambda y \, \text{Near}(\text{Kolkata}, y)$

$\lambda y \, \text{Near}(\text{Kolkata}, y) \, (\text{Howrah})$

$\text{Near}(\text{Kolkata}, \text{Howrah})$

Howrah is Near Kolkata

Syntax Driven Semantic Analysis

3

Augment the CFG grammar rules with semantic Attachments. These instructions specify how to compute the meaning representation ~~of~~ ~~of~~ of input.

Abstractly the Augmented rules have the following structure:-

$A \rightarrow \alpha_1 \cdots \alpha_n \{ f(\alpha_j.sem, \ldots \alpha_k.sem) \}$

Example

$S \rightarrow NP \; VP \quad \{ VP.sem(NP.sem) \}$

$NP \rightarrow ProperNoun \quad \{ ProperNoun.sem \}$

$ProperNoun \rightarrow \{ \cancel{SpiceIsland} \} \{ SpiceIland \}$

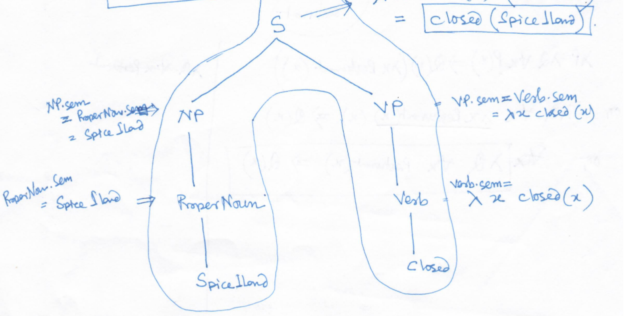
$VP \rightarrow Verb \quad \{ Verb.sem \}$

$Verb \rightarrow closed \quad \{ \lambda x. closed(x) \}$

$\boxed{\text{Spice Iland closed}}$

$\lambda x \; closed(x) \; (Spice Iland)$

$= \boxed{closed(Spice Iland)}$



~~Appear~~ the following Prod rule

~~NP~~

$NP \rightarrow Det \; Nominal \quad \{ Det.sem(Nominal.Sem) \}$

$Det \rightarrow every \quad \{ \lambda P. \lambda Q. \forall x \, P(x) \Rightarrow Q(x) \}$

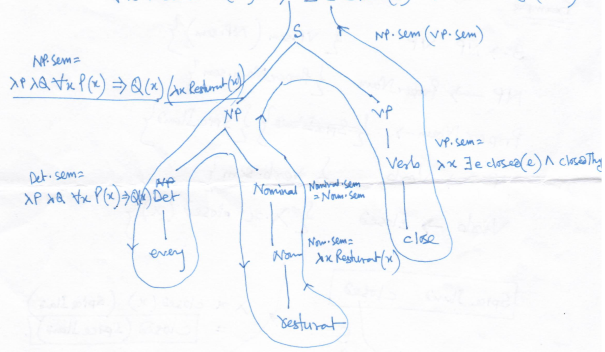
$Nominal \rightarrow Noun \quad \{ Noun.sem \}$

$Noun \rightarrow resturant \quad \{ \lambda x \; Resturant(x) \}$

$Verb \rightarrow close \; \{ \lambda x \; \exists e \; closed(e) \wedge closedThing(e, x) \}$

$\boxed{\text{Every resturant closed}}$

$\forall x \; Resturant(x) \Rightarrow \exists e \; closed(e) \wedge closedThing(e, x)$



$\lambda P \; \lambda Q \; \forall x \; P(x) \Rightarrow Q(x) \; (\lambda x \; Resturant(x))$

or, $\lambda Q \; \forall x \; \lambda x \; Resturant(x) \; (x) \Rightarrow Q(x)$

or, $\cancel{\forall x} \; \boxed{\lambda Q \; \forall x \; Resturant(x) \Rightarrow Q(x)}$

~~λQ ∀x Resturant~~

4

$\lambda Q \; \forall x \; Resturant(x) \Rightarrow Q(x) \; (\lambda x \; \exists e \; closed(e) \wedge closedThing(e, x$

$\forall x \; Resturant(x) \Rightarrow \lambda x \; \exists e \; closed(e) \wedge closedThing(e, x) \; (x)$

$\boxed{\forall x \; Resturant(x) \Rightarrow \exists e \; Closed(e) \wedge closedThing(e, x)}$.