Report on Syntactics, Semantics and Pragmatics

Given a sentence, how do we derive its conceptual meaning ?

Abhishek Nalla

20161115

3rd Year B.Tech CSE (Honors)

Introduction

Natural Language Understanding:

Understanding involves the following tasks –

1) Mapping the given input in natural language into useful representations.
2) Analyzing different aspects of the language.

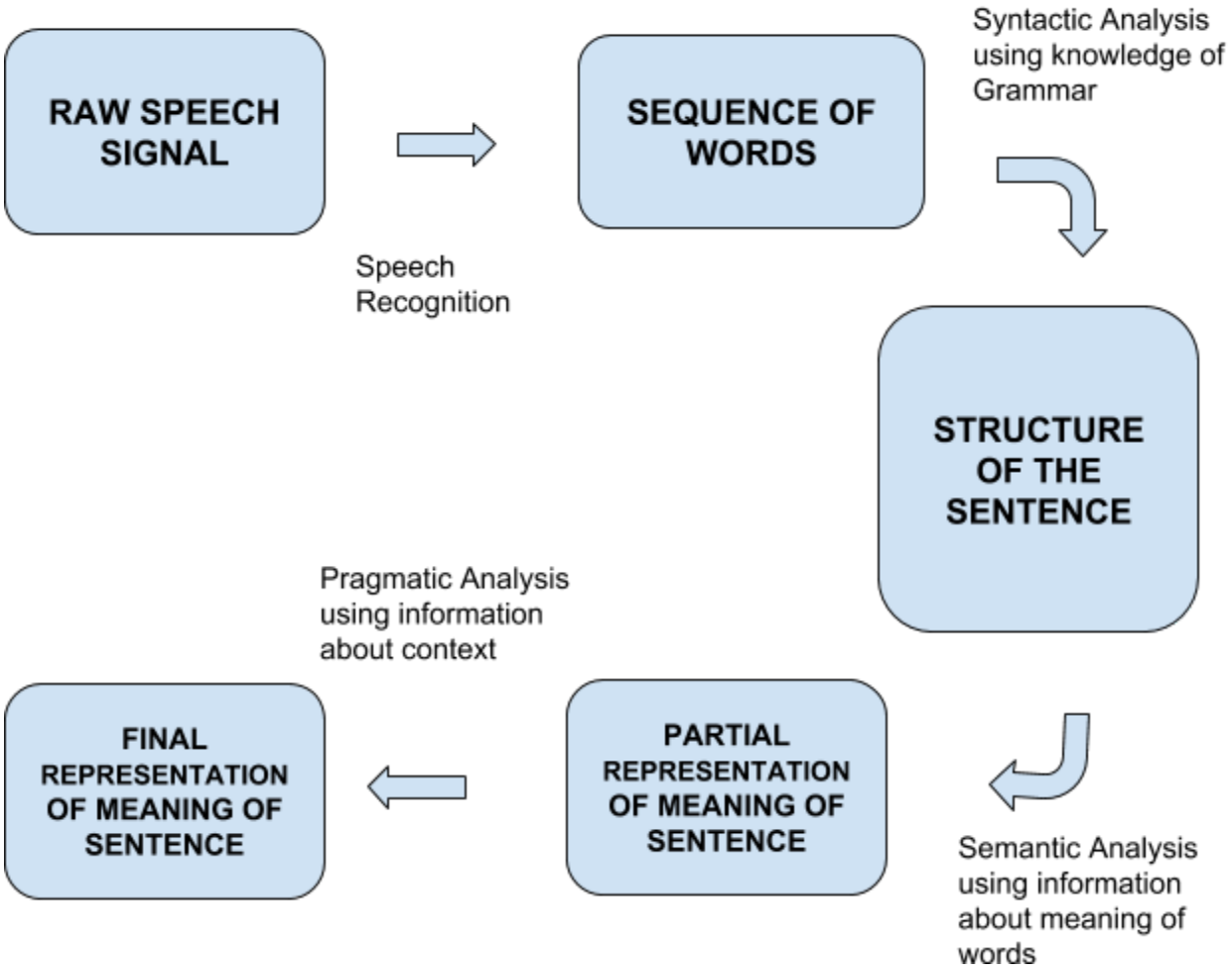
Natural Language Generation:

It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

1) **Text planning** – It includes retrieving the relevant content from knowledge base.
2) **Sentence planning** – It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
3) **Text Realization** – It is mapping sentence plan into sentence structure.

The NLU part is harder than the NLG part.

NLU Architecture



Difficulties in NLU

NL has an extremely rich form and structure.

It is very ambiguous. There can be different levels of ambiguity –

1) **Lexical ambiguity** – It is at very primitive level such as word-level.

   Ex. treating the word “board” as noun or verb?.

2) **Syntax Level ambiguity** – A sentence can be parsed in different ways.

   Ex. “He lifted the beetle with red cap.” – Did he use cap to lift the beetle or he lifted a beetle that had red cap?

3) **Referential ambiguity** – Referring to something using pronouns.

   Ex. Rima went to Gauri. She said, “I am tired.” – Exactly who is tired?

4) One input can mean different meanings.

5) Many inputs can mean the same thing.

Terminology

Syntax - It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.

Semantics - It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.

Pragmatics - It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.

Steps in Understanding a Sentence

Syntactic Analysis (Parsing)

It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.

Semantic Analysis

It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as “hot ice-cream”.

Pragmatic Analysis

During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Implementation Aspects of Syntactic Analysis

Solution : Context-Free Grammar

It is the grammar that consists rules with a single symbol on the left-hand side of the rewrite rules. Let us create grammar to parse a sentence – “The bird pecks the grains”.

Articles (DET) – a | an | the

Nouns – bird | birds | grain | grains

Noun Phrase (NP) – Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

Verbs – pecks | pecking | pecked

Verb Phrase (VP) – NP V | V NP

Adjectives (ADJ) – beautiful | small | chirping

The parse tree breaks down the sentence into structured parts so that the computer can easily understand and process it. In order for the parsing algorithm to construct this parse tree, a set of rewrite rules, which describe what tree structures are legal, need to be constructed.

These rules say that a certain symbol may be expanded in the tree by a sequence of other symbols. According to first order logic rule, if there are two strings Noun Phrase (NP) and Verb Phrase (VP), then the string combined by NP followed by VP is a sentence.

The rewrite rules for the sentence are as follows –

1) S → NP VP

2) NP → DET N | DET ADJ N

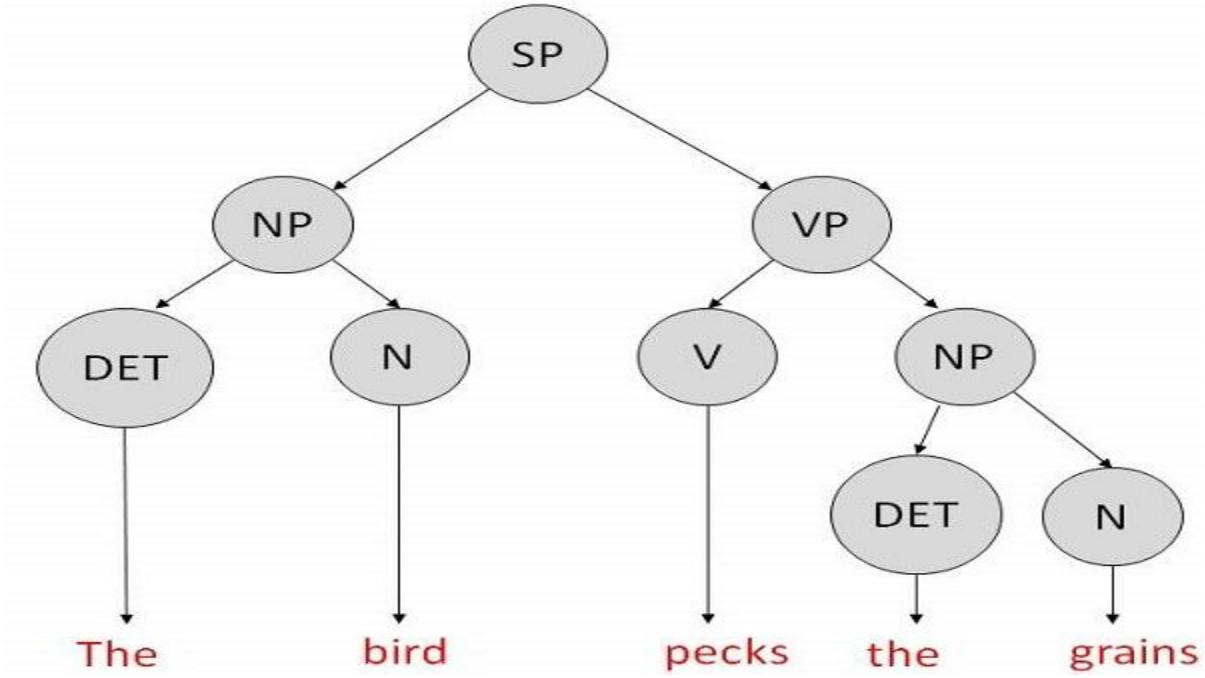
3) VP → V NP

DET → a | the

ADJ → beautiful | perching

N → bird | birds | grain | grains

V → peck | pecks | pecking



Merit – The simplest style of grammar, therefore widely used one.

Demerits :

1) They are not highly precise. For example, “The grains peck the bird”, is a syntactically correct according to parser, but even if it makes no sense, parser takes it as a correct sentence.

2) To bring out high precision, multiple sets of grammar need to be prepared. It may require a completely different sets of rules for parsing singular and plural variations, passive sentences, etc., which can lead to creation of huge set of rules that are unmanageable.

Implementation Aspects of Semantic Analysis

We have learnt how a parser constructs parse trees in the syntax analysis phase. The plain parse-tree constructed in that phase is generally of no use for a compiler, as it does not carry any information of how to evaluate the tree. The productions of context-free grammar, which makes the rules of the language, do not accommodate how to interpret them.

For example,

E → E + T

The above CFG production has no semantic rule associated with it, and it cannot help in making any sense of the production.

Semantic Understanding

Semantics of a language provide meaning to its constructs, like tokens and syntax structure. Semantics help interpret symbols, their types, and their relations with each other. Semantic analysis judges whether the syntax structure constructed in the source program derives any meaning or not.

CFG + semantic rules = Syntax Directed Definitions

For example:

int a = "string";

should not issue an error in lexical and syntax analysis phase, as it is

lexically and structurally correct, but it should generate a semantic error as the type of the assignment differs. These rules are set by the grammar of the language and evaluated in semantic analysis. The following tasks should be performed in semantic analysis:

1) Scope resolution
2) Type Checking
3) Array-bound checking

Solution: Attribute Grammar

Attribute grammar is a special form of context-free grammar where some additional information (attributes) are appended to one or more of its non-terminals in order to provide context-sensitive information. Each attribute has well-defined domain of values, such as integer, float, character, string, and expressions.

Attribute grammar is a medium to provide semantics to the context-free grammar and it can help specify the syntax and semantics of a programming language. Attribute grammar (when viewed as a parse-tree) can pass values or information among the nodes of a tree.

Example:

$$E \rightarrow E + T \{ E.value = E.value + T.value \}$$

The right part of the CFG contains the semantic rules that specify how the grammar should be interpreted. Here, the values of non-terminals E and T are added together and the result is copied to the non-terminal E.

Semantic attributes may be assigned to their values from their domain at the time of parsing and evaluated at the time of assignment or conditions. Based on the way the attributes get their values, they can be broadly divided into two categories : synthesized attributes and inherited attributes.

Synthesized attributes

These attributes get values from the attribute values of their child nodes. To illustrate, assume the following production:

$$S \rightarrow ABC$$

If S is taking values from its child nodes (A,B,C), then it is said to be a synthesized attribute, as the values of ABC are synthesized to S.

As in our previous example (E → E + T), the parent node E gets its value from its child node. Synthesized attributes never take values from their parent nodes or any sibling nodes.

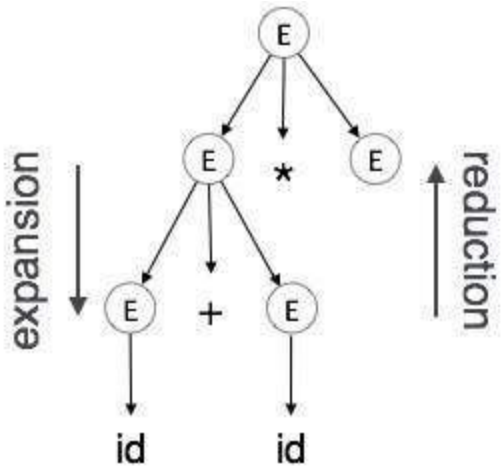
Inherited attributes

In contrast to synthesized attributes, inherited attributes can take values from parent and/or siblings. As in the following production,

$$S \rightarrow ABC$$

A can get values from S, B and C. B can take values from S, A, and C. Likewise, C can take values from S, A, and B.

Expansion : When a non-terminal is expanded to terminals as per a grammatical rule



Reduction : When a terminal is reduced to its corresponding non-terminal according to grammar rules. Syntax trees are parsed top-down and left to right. Whenever reduction occurs, we apply its corresponding semantic rules (actions).

Semantic analysis uses Syntax Directed Translations to perform the above tasks.

Semantic analyzer receives AST (Abstract Syntax Tree) from its previous stage (syntax analysis).

Semantic analyzer attaches attribute information with AST, which are called Attributed AST.

Attributes are two tuple value, <attribute name, attribute value>

For example:

int value = 5;
<type, “integer”>
<presentvalue, “5”>

For every production, we attach a semantic rule.

Implementation Aspects of Pragmatic Analysis

Pragmatic Understanding

Pragmatics is a subfield of linguistics which studies the ways in which context contributes to meaning. Pragmatics encompasses speech act theory, conversational implicature, talk in interaction and other approaches to language behavior in philosophy, sociology, linguistics and anthropology. Unlike semantics, which examines meaning that is conventional or "coded" in a given language, pragmatics studies how the transmission of meaning depends not only on structural and linguistic knowledge (e.g., grammar, lexicon, etc.) of the speaker and listener, but also on the context of the utterance, any preexisting knowledge about those involved, and other factors.

The sentence *"You have a green light"* is ambiguous. Without knowing the context, the identity of the speaker or the speaker's intent, it is difficult to infer the meaning with certainty. For example, it could mean:

- *the space that belongs to you has green ambient lighting;*
- *you are driving through a green traffic signal;*
- *you no longer have to wait to continue driving;*
- *you are permitted to proceed in a non-driving context;*
- *your body is cast in a greenish glow; or*
- *you possess a light bulb that is tinted green.*

Pragmatic interpretation of sentences

Pragmatics is often defined as the theory of the way we use language. Theories of pragmatics then link the language and its user while semantics links the syntactical entities of language with their meanings. One may think that a semantics for a language is a sufficient basis for the capacity to use that language. Through having knowledge of an appropriate semantics a competent would user know the meanings of words and sentences and this should be enough to use the language properly. The reality appears to be much more complicated. Very often, especially where common-sense language is concerned, sentences are uttered in a way which seems to have no relation to their meaning.

Solution: Referential Indexing

In pragmatics, there are two different types of meaning to consider: **semantico-referential meaning** and **indexical meaning.** Semantico-referential meaning refers to the aspect of meaning, which describes events in the world that are independent of the circumstance they are uttered in.

An example would be propositions such as:

"Santa Claus eats cookies."

In this case, the proposition is describing that Santa Claus eats cookies. The meaning of this proposition does not rely on whether or not Santa Claus is eating cookies at the time of its utterance. Santa

Claus could be eating cookies at any time and the meaning of the proposition would remain the same. The meaning is simply describing something that is the case in the world. In contrast, the proposition, "Santa Claus is eating a cookie right now," describes events that are happening at the time the proposition is uttered.

Indexical meaning, on the other hand, is dependent on the context of the utterance and has rules of use. By rules of use, it is meant that indexicals can tell you when they are used, but not what they actually mean.

Example: *"I"*

Whom *"I"* refers to depends on the context and the person uttering it.

As mentioned, these meanings are brought about through the relationship between the signified and the signifier. One way to define the relationship is by placing signs in two categories: **referential indexical signs,** also called "shifters," and **pure indexical signs.**

Referential indexical signs are signs where the meaning shifts depending on the context hence the nickname "shifters." 'I' would be considered a referential indexical sign. The referential aspect of its meaning would be '1st person singular' while the indexical aspect would be the person who is speaking (refer above for definitions of semantico-referential and indexical meaning). Another example would be:

"This"

Referential: singular count

Indexical: Close by

Links to Papers/Material Used:

https://www.nltk.org/book/ch08.html

http://mt-archive.info/MT-1968-Simmons.pdf

http://www.essay.uk.com/essays/linguistics/pragmatics/

http://www.dmi.unict.it/farinella/SMM/Lectures/Seminar3_20_12_2017.pdf