Natural Language Processing : AI Course Lecture 41, notes, slides

www.myreaders.info/ , RC Chakraborty, e-mail rcchak@gmail.com , June 01, 2010

www.myreaders.info/html/artificial_intelligence.html

www.myreaders.info

Natural Language Processing

Artificial Intelligence

Natural Language Processing, topics : Introduction, definition, formal language, linguistic and language processing, terms related to linguistic analysis, grammatical structure of utterances - sentence, constituents, phrases, classifications and structural rules; Syntactic Processing - context free grammar (CFG), terminal, non-terminal and start symbols, parser, Semantics and Pragmatics.

Natural Language Processing

Artificial Intelligence

Topics

(Lecture 41 , 1 hours)

Natural Language Processing

What is NLP ?

- NLP is Natural Language Processing.

  Natural languages are those spoken by people.

- NLP encompasses anything a computer needs to understand natural language (typed or spoken) and also generate the natural language.

- Natural Language Processing (NLP) is a subfield of Artificial intelligence and linguistic, devoted to make computers "understand" statements written in human languages.

1. Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence and linguistic, devoted to make computers "understand" statements written in human languages.

1.1 Natural Language

A natural language (or ordinary language) is a language that is spoken, written by humans for general-purpose communication.
Example : Hindi, English, French, and Chinese, etc.

A language is a system, a set of symbols and a set of rules (or grammar).

- The Symbols are combined to convey new information.
- The Rules govern the manipulation of symbols.

Natural Language Processing (NLP)

NLP encompasses anything a computer needs to understand natural language (typed or spoken) and also generate the natural language.

- **Natural Language Understanding (NLU) :**

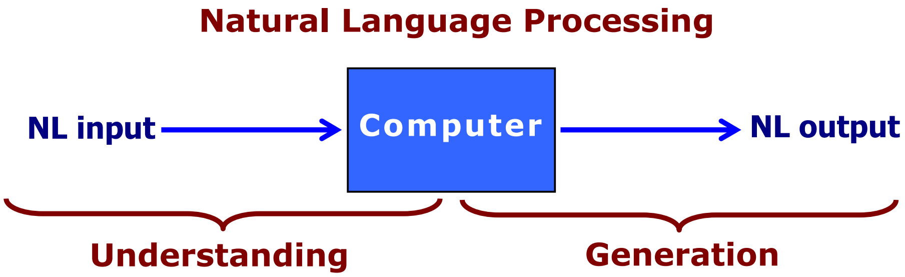
  The NLU task is understanding and reasoning while the input is a natural language.

  Here we ignore the issues of natural language generation.

- **Natural Language Generation (NLG) :**

  NLG is a subfield of natural language processing NLP.

  NLG is also referred to text generation.



1.2 Formal Language

Before defining formal language Language, we need to define symbols, alphabets, strings and words.

Symbol is a character, an abstract entity that has no meaning by itself.

e.g., Letters, digits and special characters

Alphabet is finite set of symbols;

an alphabet is often denoted by **Σ** (sigma)

e.g., **B = {0, 1}** says **B** is an alphabet of two symbols, **0** and **1**.

C = {a, b, c} says **C** is an alphabet of three symbols, **a, b** and **c**.

String or a word is a finite sequence of symbols from an alphabet.

e.g., **01110** and **111** are strings from the alphabet **B** above.

aaabccc and **b** are strings from the alphabet **C** above.

Language is a set of strings from an alphabet .

Formal language (or simply language) is a set **L** of strings over some finite alphabet Σ .

Formal language is described using formal grammars.

1.3 Linguistic and Language Processing

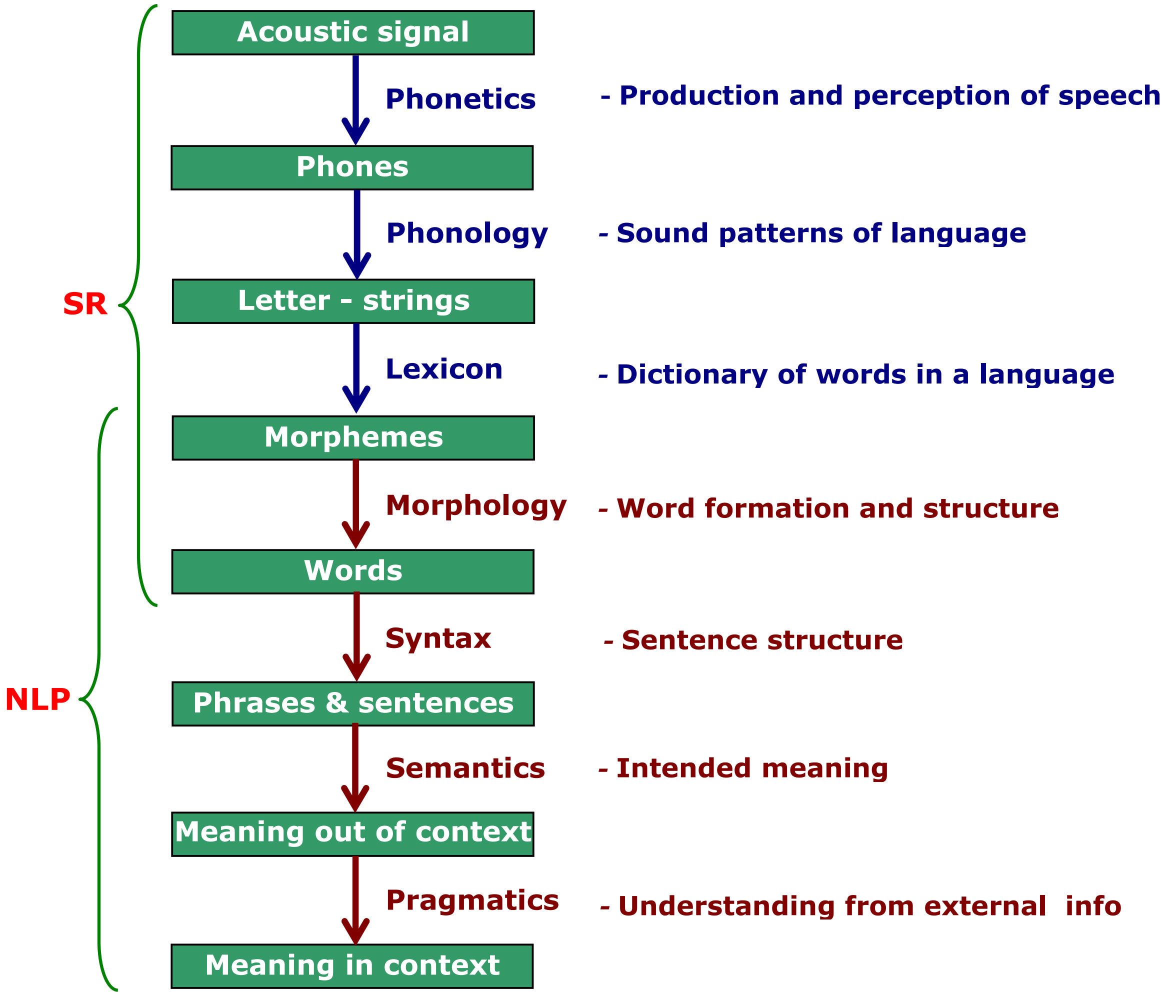
Linguistics is the science of language. Its study includes :

- sounds (phonology),
- word formation (morphology),
- sentence structure (syntax),
- meaning (semantics), and understanding (pragmatics) etc.

The levels of linguistic analysis are shown below.

- higher level corresponds to Speech Recognition (SR)
- lower levels corresponds to Natural Language Processing (NLP).

Levels Of Linguistic Analysis



- **Steps of Natural Language Processing (NLP)**

Natural Language Processing is done at 5 levels, as shown in the previous slide. These levels are briefly stated below.

- **Morphological and Lexical Analysis :**

  The lexicon of a language is its vocabulary, that include its words and expressions. Morphology is the identification, analysis and description of structure of words. The words are generally accepted as being the smallest units of syntax. The syntax refers to the rules and principles that govern the sentence structure of any individual language.

  Lexical analysis: The aim is to divide the text into paragraphs, sentences and words. the lexical analysis can not be performed in isolation from morphological and syntactic analysis

- **Syntactic Analysis :**

  Here the analysis is of words in a sentence to know the grammatical structure of the sentence. The words are transformed into structures that show how the words relate to each others. Some word sequences may be rejected if they violate the rules of the language for how words may be combined.

  Example : An English syntactic analyzer would reject the sentence say :

  " Boy the go the to store ".

- **Semantic Analysis :**

  It derives an absolute (dictionary definition) meaning from context; it determines the possible meanings of a sentence in a context.

  The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between the syntactic structures and objects in the task domain. The structures for which no such mapping is possible are rejected.

  Example : the sentence **"Colorless green ideas . . . "** would be rejected as semantically anomalous because colorless and green make no sense.

- **Discourse Integration :**

  The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it.

  Example : the word **" it "** in the sentence, **"you wanted it"** depends on the prior discourse context.

- **Pragmatic analysis :**

  It derives knowledge from external commonsense information; it means understanding the purposeful use of language in situations, particularly those aspects of language which require world knowledge; The idea is, what was said is reinterpreted to determine what was actually meant. Example : the sentence

  "Do you know what time it is ?"

  should be interpreted as a request.

1.4 Defining Terms related to Linguistic Analysis

The following terms are explained in next few slides.

Phones, Phonetics, Phonology, Strings, Lexicon, Words, Determiner, Morphology, Morphemes, Syntax, Semantics, Pragmatics, Phrase, and Sentence.

- **Terms**
  - **Phones**

    The Phones are acoustic patterns that are significant and distinguishable in some human language.

    Example : In English, the **L** - sounds at the beginning and end of the word **"loyal"**, are termed "light L" and "dark L" by linguists.

  - **Phonetics**

    Tells how acoustic signals are classified into phones.

  - **Phonology**

    Tells how phones are grouped together to form phonemes in particular human languages.

- **Strings**

  An alphabet is a finite set of symbols.

  Example : English alphabets

  { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z }

  A String is a sequence of symbols taken from an alphabet.

- **Lexicon**

  Lexicon is collection of information about words of a language.

  The information is about the lexical categories to which words belong.

  Example : **"pig"** is usually a noun **(N)**, but also occurs as a verb**(V)** and an adjective**(ADJ)**.

  Lexicon structure : as collection of lexical entries.

  Example : **("pig" N, V, ADJ)**

- **Words**

  Word is a unit of language that carries meaning.

  Example : words like bear, car, house are very different from
  words like run, sleep, think, and are different from
  words like in, under, about.

  These and other categories of words have names : nouns, verbs, prepositions, and so on.

  Words build phrases, which in turn build sentences.

- **Determiner**

  Determiners occur before nouns and indicate the kind of reference which the noun has.

  Example below shows determiners marked by "bold letters"

  the boy   **a** bus   **our** car
  these children   **both** hospitals

- **Morphology**

  Morphology is the analysis of words into morphemes, and conversely the synthesis of words from morphemes.

- **Morphemes**

  A smallest meaningful unit in the grammar of a language.

  A smallest linguistic unit that has semantic meaning.

  A unit of language immediately below the ‘word level’.

  A smallest part of a word that can carry a discrete meaning.

  Example : the word **"unbreakable"** has 3 morphemes:

1	"**un-**"	a bound morpheme;
2	"**-break-**"	a free morpheme; and
3	"**-able**"	a bound morpheme;

  Also **"un-"** is also a prefix; **"-able"** is a suffix; Both are affixes.

  Morphemes are of many types, stated in the next slide.

Types of Morphemes

‡ **Free Morphemes**

can appear stand alone, or "free" .

Example : **"town", "dog"** or with other lexemes **"town hall" , "dog house"**.

‡ **Bound Morphemes**

appear only together with other morphemes to form a lexeme.

Example : **"un-"** ; in general it tend to be prefix and suffix.

‡ **Inflectional Morphemes**

modify a word's tense, number, aspect, etc.

Example : **dog** morpheme with plural marker morpheme **s** becomes **dogs.**

‡ **Derivational Morphemes**

can be added to a word to derive another word.

Example : addition of **"-ness"** to **"happy"** gives "**happiness."**

‡ **Root Morpheme**

It is the primary lexical unit of a word; roots can be either free or bound morphemes; sometimes "root" is used to describe word minus its inflectional endings, but with its lexical endings.

Example : word **chatters** has the inflectional root or lemma **chatter**, but the lexical root **chat.**

Inflectional roots are often called stems, and a root in the stricter sense may be thought of as a mono-morphemic stem.

‡ **Null Morpheme**

It is an "invisible" affix, also called zero morpheme represented as either the figure zero **(0)**, the empty set symbol **Ø**, or its variant **Ø**. Adding a null morpheme is called null affixation, null derivation or zero derivation; null morpheme that contrasts singular morpheme with the plural morpheme.

e.g., **cat = cat + -0 = ROOT("cat") + SINGULAR**

cats = cat + -s = ROOT("cat") + PLURAL

- **Syntax**

  Syntax is the structure of language. It is the grammatical arrangement of words in a sentence to show its relationship to one another in a sentence; Syntax is finite set of rules that specifies a language;

  Syntax rules govern proper sentence structure;

  Syntax is represented by Parse Tree, a way to show the structure of a language fragment, or by a list.

- **Semantics**

  Semantic is Meaning of words / phrases/ sentences/ whole texts.

  Normally semantic is restricted to "meaning out of context" - that is, meaning as it can be determined without taking context into account.

- **Pragmatics**

  Pragmatics tell how language is used; that is 'meaning in context'.

  Example: if someone says **"the door is open"** then it is necessary to know which door **"the door"** refers to;

  Need to know what the intention of the speaker :

  could be a pure statement of fact,

  could be an explanation of how the cat got in, or

  could be a request to the person addressed to close the door.

1.5 Grammatical Structure of Utterances

Here sentence, constituent, phrase, classification and structural rule are explained.

- **Sentence**

  Sentence is a string of words satisfying grammatical rules of a language;

  Sentences are classified as simple, compound, and complex.

  Sentence is often abbreviated to "***S***".

  Sentence (S) : "The dog bites the cat".

- **Constituents**

  Assume that a phrase is a construction of some kind.

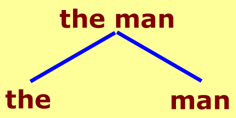
  Here construction means a syntactic arrangement that consists of parts, usually two, called "constituents".

  Examples : The phrase **the man** is a construction consists of two constituents **the** and **man**. A few more examples are shown below.

Phrase : the man

Constituents : the and man

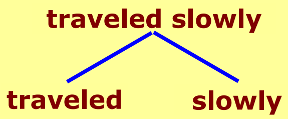
Construction :



Phrase : traveled slowly

Constituents : traveled and slowly.

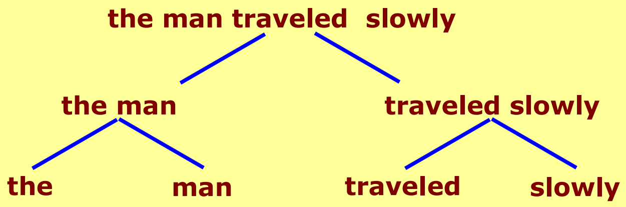
Construction :



Phrase : the man traveled slowly

Constituents four : the , man , traveled , slowly

Construction :



- **Phrase**

  A Phrase is a group of words (minimum is two) that functions as a single unit in the syntax of a sentence.

  e.g., 1: **"the house at the end of the street "** is a phrase, acts like noun.

  e.g., 2: **"end of the street "** is a phrase, acts like adjective;

  How phrases are formed is governed by phrase structure rules.

  Most phrases have a head or central word, which defines the type of phrase. Head is often the first word of the phrase. Some phrases, can be headless.

  e.g.,3: **"the rich"** is a noun phrase composed of a determiner and an adjective, but no noun.

  Phrases may be classified by the type of head they take.

  [Continued in next slide]

[Continued from previous slide]

Classification of Phrases : names (abbreviation)

The most accepted classifications for phrases are stated below.

- **Sentence (S) :** often abbreviated to "***S***".
- **Noun phrase (NP) :** noun or pronoun as head, or optionally accompanied by a set of modifiers; The possible modifiers include: determiners: articles (the, a) or adjectives (the red ball) etc ; example : "**the** black cat", "**a** cat on the mat".
- **Verb phrase (VP)** : verb as head, example : "**eat** cheese", "**jump** up and down".
- **Adjectival phrase (AP) :** adjective as head, example : "**full** of toys"
- **Adverbial phrase (AdvP) :** adverb as head, example : "**very carefully**"
- **Prepositional phrase (PP) :** preposition as head, example : "**in** love", "**over** the rainbow".
- **Determiner phrase (DP) :** determiner as head example : "**a** little dog", "**the** little dogs". In English, determiners are usually placed before the noun as a noun modifier that includes : articles (the, a), demonstratives (this, that), numerals (two, five, etc.), possessives (my, their, etc.), and quantifiers (some, many, etc.).

- **Phrase Structure Rules**

Phrase-structure rules are a way to describe language syntax. Rules determine what goes into phrase and how its constituents are ordered. They are used to break a sentence down to its constituent parts namely phrasal categories and lexical categories.

- Phrasal category include : noun phrase, verb phrase, prepositional phrase;
- Lexical category include : noun, verb, adjective, adverb, others.

Phrase structure rules are usually of the form A → **B C** ,

Meaning "constituent **A** is separated into two sub-constituents **B** and **C** " or simply " **A** consists of **B** followed by **C** " .

Examples :

‡ **S** → **NP VP** Reads : **S** consists of an **NP** followed by a **VP** ; means a sentence consists of a noun phrase followed by a verb phrase.

‡ **NP** → **Det N1** Reads : **NP** consists of an **Det** followed by a **N1** ; means a noun phrase consists of a determiner followed by a noun.

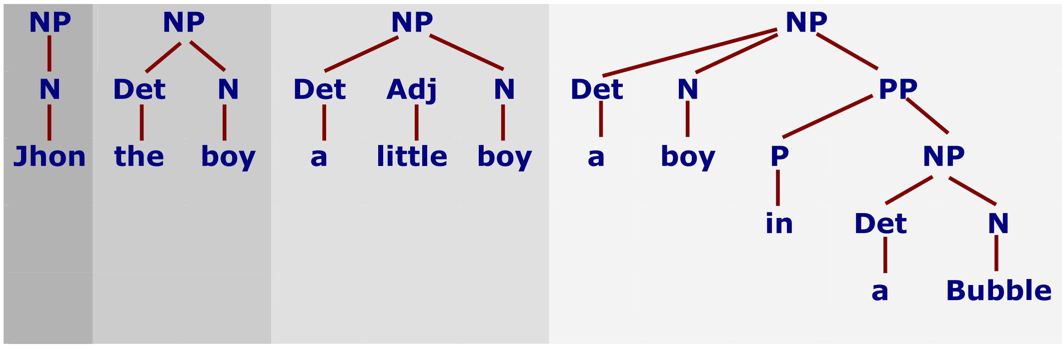
Phrase Structure Rules and Trees for Noun Phrase (NP)

Noun Phrase (NP)	
John	N
the boy	Det N
A little boy	Det Adj N
A boy in a bubble	Det N PP

Phrase Structure rules for NPs

NP → (Det) (Adj) N (PP)

Phrase Structure trees for NPs



2. Syntactic Processing

Syntactic Processing converts a flat input sentence into a hierarchical structure that corresponds to the units of meaning in the sentence.

The Syntactic processing has two main components :

- one is called grammar, and
- other is called parser.

‡ **Grammar** :

It is a declarative representation of syntactic facts about the language.

It is the specification of the legal structures of a language.

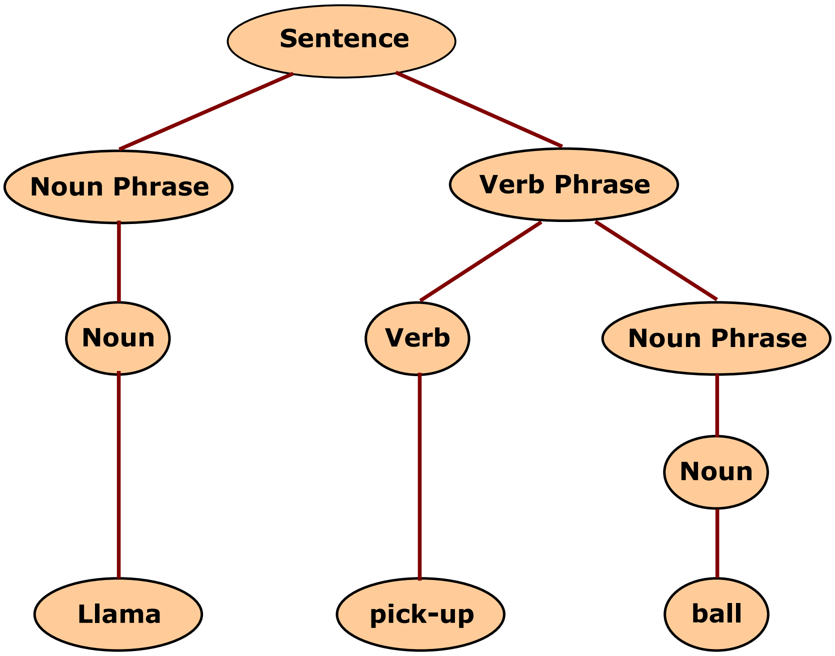
It has three basic components : terminal symbols, non-terminal symbols, and rules (productions) .

‡ **Parser** :

It is a procedure that compares the grammar against input sentences to produce a parsed structures called **parse tree.**

Example 1 : Sentence **"Llama pickup ball"**.

Parse Tree Structure (PS)



2.1 Context Free Grammar (CFG)

In formal language theory, a context free grammar is a grammar where every production rules is of the form: $A \rightarrow \alpha$ where A is a single symbol called **non-terminal** , and α is a **string** that is a sequence of symbols of terminals and/or non-terminals (possibly empty).

Note : The difference with an arbitrary grammars is that the left hand side of a production rule is always a single nonterminal symbol rather than a string of terminal and/or nonterminal symbols.

- **Terminal , Non-Terminal and Start Symbols**

The terminal and non-terminal symbols are those symbols that are used to construct production rules in a formal grammar.

‡ **Terminal Symbol**

Any symbol used in the grammar which does not appear on the left-hand-side of some rule (ie. has no definition) is called a terminal symbol. Terminal symbols cannot be broken down into smaller units without losing their literal meaning.

‡ **Non-Terminal Symbol**

Symbols that are defined by rules are called non-terminal symbol. Each production rule defines the non-terminal symbol. Like the above rule states that "whenever we see an A, we can replace it with α ".

‡ A non-terminal may have more than one definition, in that case we use symbol "|" as the union operator;

Example 1: $A \rightarrow \alpha \mid \beta$ states that "whenever we see A, we can replace it with α or with β ''.

Similarly, if a rule is **NP** → **Det N** | **Prop** then the vertical slash on the right side is a convention used to represent that the **NP** can be replaced either by **Det N** or by **Prop**. Thus, this is really two rules.

Example 2: **S** → **NP VP** states that the symbol **S** is replaced by the symbols **NP** and **VP**.

‡ One special non-terminal is called **Start symbol**, usually written **S**. The production rules for this symbol are usually written first in a grammar.

- **How Grammar works ?**

Grammar starts with the start symbol, then successively applies the production rules (replacing the L.H.S. with the R.H.S.) until reaches to a word which contains no non-terminals. This is known as a derivation.

‡ Anything which can be derived from the start symbol by applying the production rules is called a sentential form.

‡ Any grammar may have an infinite number of sentences;
The set of all such sentences is the language defined by that grammar.

‡ Example of grammar :

S → X c   X → Y X   Y → a | b

The above grammar shows that it can derive all words which start arbitrarily and have many '**a**'s or '**b**'s and finish with a '**c**'. This language is defined by the regular expression **(a | b) * c.** The " * " indicates that the character immediately to its left may be repeated any number of times, including zero. Thus **ab*c** would match **"ac", "abc", "abbc", "abbbc", "abbbbbbbbc"**, and any string that starts with an "**a**", is followed by a sequence of "**b**"'s, and ends with a "**c**".

‡ **Regular Expression**

Every regular expression can be converted to a grammar, but not every grammar can be converted back to a regular expression;
Any grammar which can be converted back to a regular expression is called a regular grammar; the language it defines is a regular language.

Regular Expression	→	**Grammar**
Regular Expression	←	**Regular Grammar**

‡ **Regular Grammars**

A regular grammar is a grammar where all of the production rules are of one of the following forms:

A → **a** B   or   A → **a**

where A and B represent any single non-terminal, and
a represents any single terminal, or the empty string.

2.2 Parsar

A parser is a program, that accepts as input a sequence of words in a natural language and breaks them up into parts (nouns, verbs, and their attributes), to be managed by other programming.

- Parsing can be defined as the act of analyzing the grammaticality an utterance according to some specific grammar.
- Parsing is the process to check, that a particular sequence of words in a sentence correspond to a language defined by its grammar.
- Parsing means show how we can get from the start symbol of the grammar to the sequence of words using the production rules.
- The output of a parser is *a Parse tree.*

Parse Tree is a way of representing the output of a parser.

- Each phrasal constituent found during parsing becomes a branch node of the parse tree;
- the words of the sentence become the leaves of the parse tree;
- there can be more than one parse tree for a single sentence;

- **Parsing**

To parse a sentence, it is necessary to find a way in which the sentence could have been generated from the start symbol. There two ways to do : One, Top-Down Parsing and the other, Bottom–UP Parsing.

- **Top-Down Parsing**

  Begin with the start symbol and apply the grammar rules forward until the symbols at the terminals of the tree corresponds to the components of the sentence being parsed.

- **Bottom–UP Parsing**

  Begin with the sentence to be parsed and apply the grammar rules backward until a single tree whose terminals are the wards of the sentence and whose top node is the start symbol has been produced.

Note : The choice between these two approaches is similar to the choice between forward and backward reasoning in other problem solving tasks. The most important consideration is the branching factors. Some times these two approaches are combined in to a single method called bottom–up parsing with top-down filtering.

• Modeling a Sentence using Phase Structure

Every sentence consists of an internal structure which could be modeled with the phrase structure.

Algorithm : Steps

‡ Apply rules on an proposition

‡ The base proposition would be :

S (the root, ie the sentence).

‡ The first production rule would be :

(NP = noun phrase, VP = verb phrase)

S -> (NP, VP)

‡ Apply rules for the 'branches'

NP -> noun VP -> verb, NP

‡ The verb and noun have terminal nodes which could be any word in the lexicon for the appropriate category.

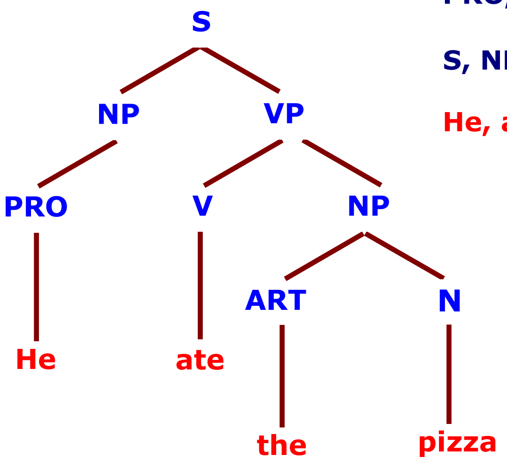
‡ The end is a tree with the words as terminal nodes, which is referred as the sentence.

Example : Parse tree

- sentence "He ate the pizza",
- apply the grammar with rules

  S -> NP VP, NP -> PRO, NP -> ART N, VP -> V NP,
- the lexicon structure is

  ("ate" V) ("he" PRO) ("pizza" N) ("the" ART)
- The parse tree is



PRO, V, ART, N - lexical non-terminals

S, NP, VP - phrasal non-terminal

He, ate, the, pizza - words or terminal

3. Semantics and Pragmatics

The semantics and pragmatics, are the two stages of analysis concerned with getting at the meaning of a sentence.

- In the first stage (semantics) a partial representation of the meaning is obtained based on the possible syntactic structure(s) of the sentence and the meanings of the words in that sentence.
- In the second stage (pragmatic), the meaning is elaborated based on : the contextual and the world knowledge.

For the difference between these stages, consider the sentence:

"He asked for the boss".

From knowledge of the meaning of the words and the structure of the sentence we can work out that :

- Someone (who is male) asked for someone who is a boss.
- We can't say who these people are and why the first guy wanted the second.
- If we know something about the context (including the last few sentences spoken/written) we may be able to work these things out.
- Maybe the last sentence was **"Fred had just been sacked.''**
- From our general knowledge that bosses generally sack people : if people want to speak to people who sack them it is generally to complain about it.
- We could then really start to get at the meaning of the sentence : **"Fred wants to complain to his boss about getting sacked".**

4. References : Textbooks

1. *"Artificial Intelligence", by Elaine Rich and Kevin Knight, (2006), McGraw Hill companies Inc., Chapter 15, page 377-426.*

2. *"Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, (2002), Prentice Hall, Chapter 23, page 834-861.*

3. *"Artificial Intelligence: Structures and Strategies for Complex Problem Solving", by George F. Luger, (2002), Addison-Wesley, Chapter 15, page 619-632.*

4. *"Artificial Intelligence: Theory and Practice", by Thomas Dean, (1994), Addison-Wesley, Chapter 10, Page 489-538.*

5. *Related documents from open source, mainly internet. An exhaustive list is being prepared for inclusion at a later date.*