Parul® University

ARTIFICIAL INTELLIGENCE II

Prof. Mehulkumar Dalwadi
IT & Computer Science

Parul® University

DIGITAL LEARNING CONTENT

Unit-4

Natural Language Processing

Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking

Parul® University

What is NLP?

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans through natural language. The ultimate goal of NLP is to enable computers to understand, interpret, and generate human language in a valuable way.

Importance of NLP: NLP is crucial in many applications such as machine translation, sentiment analysis, speech recognition, and chatbots, making it a fundamental technology in the AI landscape.

Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, and Spell Checking.

Parul® University

NLP Techniques

1. Text Processing and Preprocessing In NLP
2. Syntax and Parsing In NLP
3. Semantic Analysis
4. Information Extraction
5. Text Classification in NLP
6. Language Generation
7. Speech Processing
8. Question Answering
9. Dialogue Systems
10. Sentiment and Emotion Analysis in NLP

Applications of Natural Language Processing (NLP):

Spam Filters: One of the most irritating things about email is spam. Gmail uses natural language processing (NLP) to discern which emails are legitimate and which are spam. These spam filters look at the text in all the emails you receive and try to figure out what it means to see if it's spam or not.

Algorithmic Trading: Algorithmic trading is used for predicting stock market conditions. Using NLP, this technology examines news headlines about companies and stocks and attempts to comprehend their meaning in order to determine if you should buy, sell, or hold certain stocks.

Questions Answering: NLP can be seen in action by using Google Search or Siri Services. A major use of NLP is to make search engines understand the meaning of what we are asking and generate natural language in return to give us the answers.

Summarizing Information: On the internet, there is a lot of information, and a lot of it comes in the form of long documents or articles. NLP is used to decipher the meaning of the data and then provides shorter summaries of the data so that humans can comprehend it more quickly.

NLP Pipeline

In comparison to general machine learning pipelines, In NLP we need to perform some extra processing steps. The region is very simple that machines don't understand the text. Here our biggest problem is How to make the text understandable for machines. Some of the most common problems we face while performing NLP tasks are mentioned below.

Data Acquisition
Text Cleaning
Text Preprocessing
Feature Engineering
Model Building
Evaluation
Deployment

Ambiguity and Uncertainty in Language

Lexical Ambiguity

The ambiguity of a single word is called lexical ambiguity. For example, treating the word silver as a noun, an adjective, or a verb.

Syntactic Ambiguity

This kind of ambiguity occurs when a sentence is parsed in different ways. For example, the sentence “The man saw the girl with the telescope”. It is ambiguous whether the man saw the girl carrying a telescope or he saw her through his telescope.

Semantic Ambiguity

This kind of ambiguity occurs when the meaning of the words themselves can be misinterpreted. In other words, semantic ambiguity happens when a sentence contains an ambiguous word or phrase. For example, the sentence “The car hit the pole while it was moving” is having semantic ambiguity because the interpretations can be “The car, while moving, hit the pole” and “The car hit the pole while the pole was moving”.

Parul® University

Ambiguity and Uncertainty in Language

Anaphoric Ambiguity

This kind of ambiguity arises due to the use of anaphora entities in discourse. For example, the horse ran up the hill. It was very steep. It soon got tired. Here, the anaphoric reference of “it” in two situations cause ambiguity.

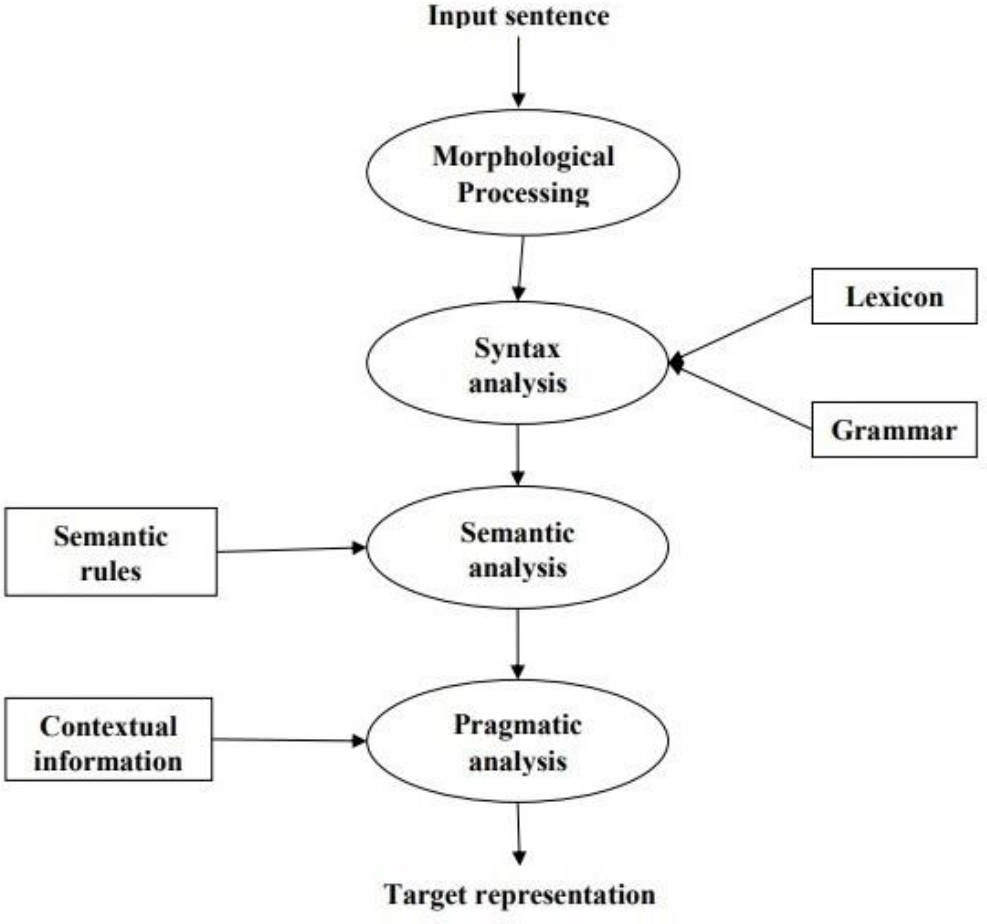
Pragmatic ambiguity

Such kind of ambiguity refers to the situation where the context of a phrase gives it multiple interpretations. In simple words, we can say that pragmatic ambiguity arises when the statement is not specific. For example, the sentence “I like you too” can have multiple interpretations like I like you (just like you like me), I like you (just like someone else dose).

Parul® University

DIGITAL LEARNING CONTENT

NLP Phases



NLP Phases

Morphological Processing

It is the first phase of NLP. The purpose of this phase is to break chunks of language input into sets of tokens corresponding to paragraphs, sentences and words. For example, a word like “uneasy” can be broken into two sub-word tokens as “un-easy”.

Syntax Analysis

It is the second phase of NLP. The purpose of this phase is two folds: to check that a sentence is well formed or not and to break it up into a structure that shows the syntactic relationships between the different words. For example, the sentence like “The school goes to the boy” would be rejected by syntax analyzer or parser.

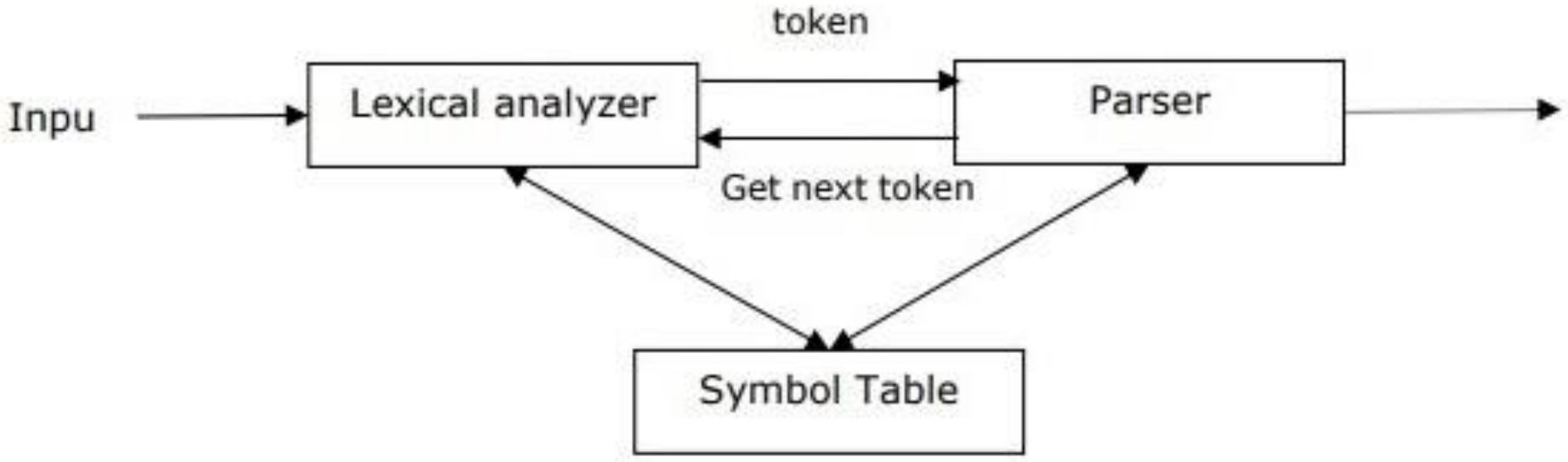
Parul® University

Natural Language Processing - Syntactic Analysis

Syntactic analysis or parsing may be defined as the process of analyzing the strings of symbols in natural language conforming to the rules of formal grammar. The origin of the word ‘parsing’ is from Latin word ‘pars’ which means ‘part’.

Concept of Parser

It is used to implement the task of parsing. It may be defined as the software component designed for taking input data (text) and giving structural representation of the input after checking for correct syntax as per formal grammar. It also builds a data structure generally in the form of parse tree or abstract syntax tree or other hierarchical structure.


token
Inpu
Lexical analyzer
Parser
Get next token
Symbol Table

Types of Parsing

Derivation divides parsing into the followings two types −

Top-down Parsing

Bottom-up Parsing

Top-down Parsing
In this kind of parsing, the parser starts constructing the parse tree from the start symbol and then tries to transform the start symbol to the input. The most common form of topdown parsing uses recursive procedure to process the input. The main disadvantage of recursive descent parsing is backtracking.

Bottom-up Parsing
In this kind of parsing, the parser starts with the input symbol and tries to construct the parser tree up to the start symbol.

Concept of Derivation

In order to get the input string, we need a sequence of production rules. Derivation is a set of production rules. During parsing, we need to decide the non-terminal, which is to be replaced along with deciding the production rule with the help of which the non-terminal will be replaced.

Types of Derivation

In this section, we will learn about the two types of derivations, which can be used to decide which non-terminal to be replaced with production rule −

Left-most Derivation

In the left-most derivation, the sentential form of an input is scanned and replaced from the left to the right. The sentential form in this case is called the left-sentential form.

Right-most Derivation

In the left-most derivation, the sentential form of an input is scanned and replaced from right to left. The sentential form in this case is called the right-sentential form.

Concept of Grammar

Grammar is very essential and important to describe the syntactic structure of well-formed programs. In the literary sense, they denote syntactical rules for conversation in natural languages. Linguistics have attempted to define grammars since the inception of natural languages like English, Hindi, etc.

Mathematically, a grammar G can be formally written as a 4-tuple (N, T, S, P) where −

N or VN = set of non-terminal symbols, i.e., variables.

T or ∑ = set of terminal symbols.

S = Start symbol where S ∈ N

P denotes the Production rules for Terminals as well as Non-terminals. It has the form α → β, where α and β are strings on VN ∪ ∑ and least one symbol of α belongs to VN

Phrase Structure or Constituency Grammar

Phrase structure grammar, introduced by Noam Chomsky, is based on the constituency relation. That is why it is also called constituency grammar. It is opposite to dependency grammar.

Example
Before giving an example of constituency grammar, we need to know the fundamental points about constituency grammar and constituency relation.

All the related frameworks view the sentence structure in terms of constituency relation.

The constituency relation is derived from the subject-predicate division of Latin as well as Greek grammar.

The basic clause structure is understood in terms of noun phrase NP and verb phrase VP.

For example, let's consider a simple PSG rule:

S -> NP VP

Rule 1: states that a sentence (S) can be divided into a noun phrase (NP) followed by a verb phrase (VP).

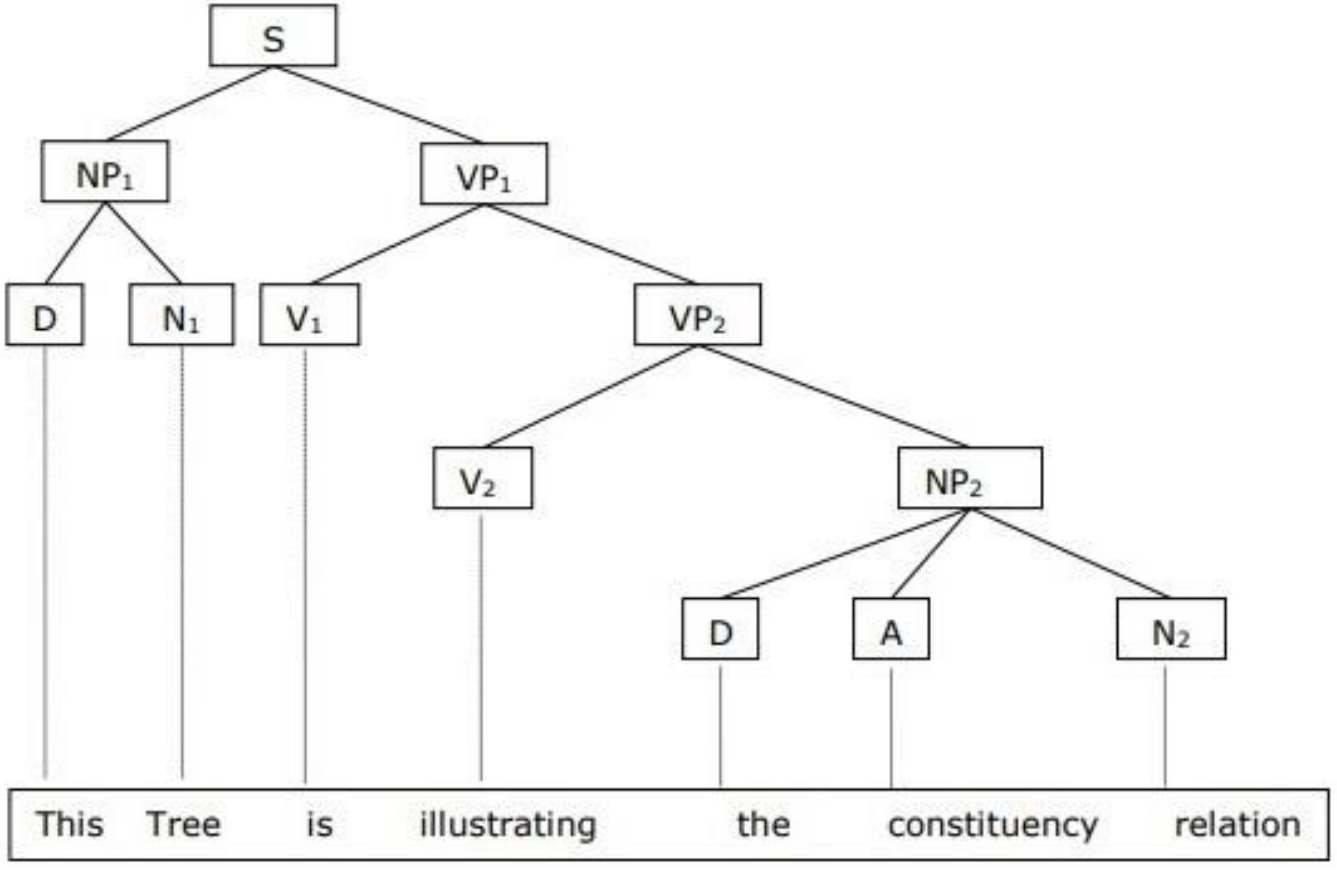
The constituents NP and VP can be further expanded using other grammar rules.

For instance:

- NP -> Det N
- VP -> V NP

Rule 2 : indicates that an NP can consist of a determiner (Det) followed by a noun (N),

Rule 3 : states that a VP can be formed by a verb (V) followed by an NP.


S
NP1
VP1
D
N1
V1
VP2
V2
NP2
D
A
N2
This
Tree
is
illustrating
the
constituency
relation

Dependency Grammar

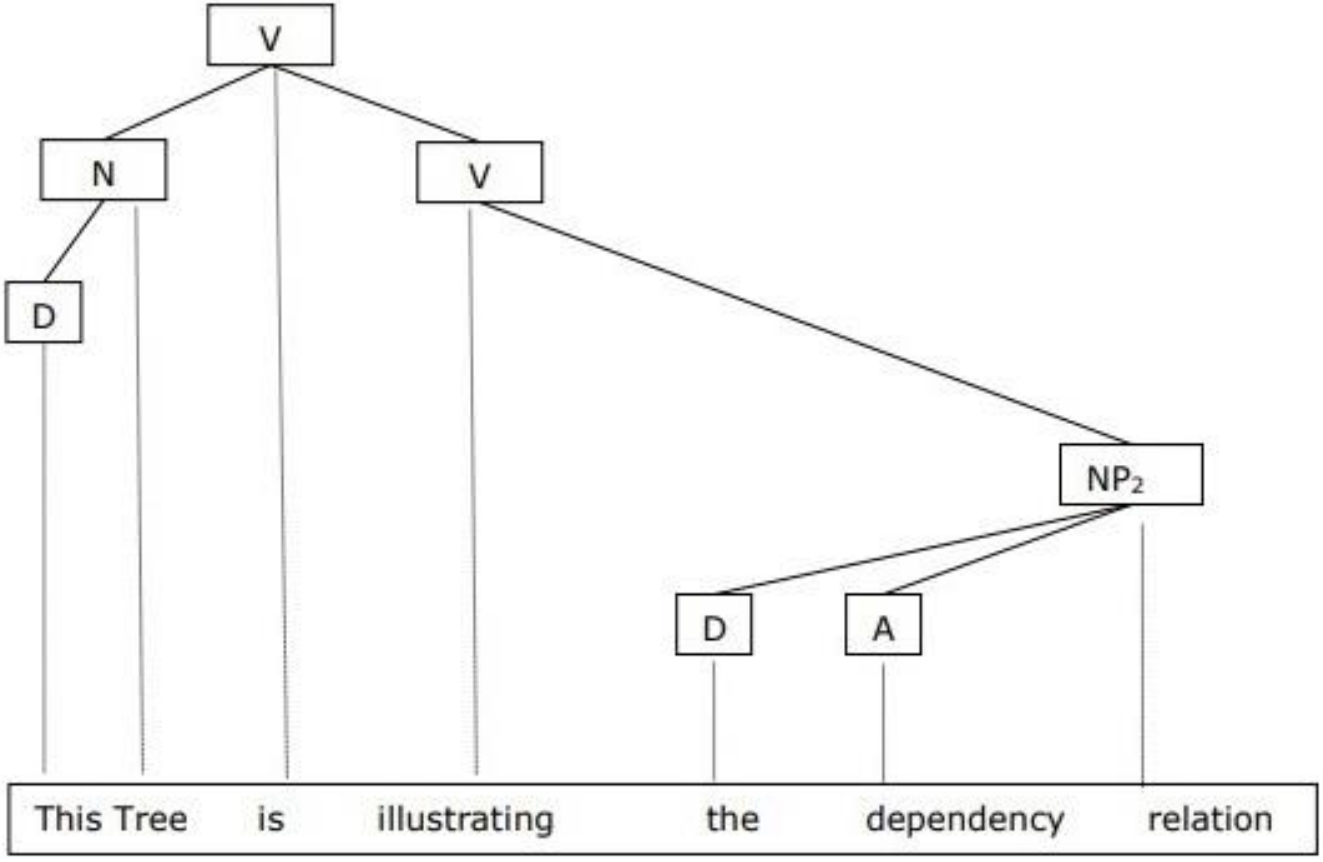
It is opposite to the constituency grammar and based on dependency relation. It was introduced by Lucien Tesniere. Dependency grammar (DG) is opposite to the constituency grammar because it lacks phrasal nodes.

Example
Before giving an example of Dependency grammar, we need to know the fundamental points about Dependency grammar and Dependency relation.

In DG, the linguistic units, i.e., words are connected to each other by directed links.

The verb becomes the center of the clause structure.

Every other syntactic units are connected to the verb in terms of directed link. These syntactic units are called dependencies.


V
N
V
D
NP2
D
A
This Tree
is
illustrating
the
dependency
relation

Parul® University

Natural Language Processing - Semantic Analysis

Studying meaning of individual word
It is the first part of the semantic analysis in which the study of the meaning of individual words is performed. This part is called lexical semantics.

Studying the combination of individual words
In the second part, the individual words will be combined to provide meaning in sentences.

The most important task of semantic analysis is to get the proper meaning of the sentence. For example, analyze the sentence “Ram is great.” In this sentence, the speaker is talking either about Lord Ram or about a person whose name is Ram. That is why the job, to get the proper meaning of the sentence, of semantic analyzer is important.

Elements of Semantic Analysis

Hyponymy

It may be defined as the relationship between a generic term and instances of that generic term. Here the generic term is called hypernym and its instances are called hyponyms. For example, the word color is hypernym and the color blue, yellow etc. are hyponyms.

Homonymy

It may be defined as the words having same spelling or same form but having different and unrelated meaning. For example, the word “Bat” is a homonymy word because bat can be an implement to hit a ball or bat is a nocturnal flying mammal also.

Polysemy

Polysemy is a Greek word, which means “many signs”. It is a word or phrase with different but related sense. In other words, we can say that polysemy has the same spelling but different and related meaning.

Building Blocks of Semantic System

In word representation or representation of the meaning of the words, the following building blocks play an important role −

Entities − It represents the individual such as a particular person, location etc. For example, Haryana. India, Ram all are entities.

Concepts − It represents the general category of the individuals such as a person, city, etc.

Relations − It represents the relationship between entities and concept. For example, Ram is a person.

Predicates − It represents the verb structures. For example, semantic roles and case grammar are the examples of predicates.

Need of Meaning Representations

A question that arises here is why do we need meaning representation? Followings are the reasons for the same −

Linking of linguistic elements to non-linguistic elements
The very first reason is that with the help of meaning representation the linking of linguistic elements to the non-linguistic elements can be done.

Representing variety at lexical level
With the help of meaning representation, unambiguous, canonical forms can be represented at the lexical level.

Can be used for reasoning
Meaning representation can be used to reason for verifying what is true in the world as well as to infer the knowledge from the semantic representation.

Disclosure Integration:

While processing a language there can arise one major ambiguity known as referential ambiguity. Referential ambiguity is the ambiguity that can arise when a reference to a word cannot be determined. For example,

Ram won the race.
Mohan ate half of a pizza.
He liked it.

In the above example, “He” can be Ram or Mohan. This creates an ambiguity. The word “He” shows dependency on both sentences. This is known as disclosure integration. It means when an individual sentence relies upon the sentence that comes before it. Like in the above example the third sentence relies upon the sentence before it. Hence the goal of this model is to remove referential ambiguity.

Parul® University

Natural Language Discourse Processing

An important question regarding discourse is what kind of structure the discourse must have. The answer to this question depends upon the segmentation we applied on discourse. Discourse segmentations may be defined as determining the types of structures for large discourse. It is quite difficult to implement discourse segmentation, but it is very important for information retrieval, text summarization and information extraction kind of applications.

Algorithms for Discourse Segmentation

Unsupervised Discourse Segmentation

The class of unsupervised discourse segmentation is often represented as linear segmentation. We can understand the task of linear segmentation with the help of an example. In the example, there is a task of segmenting the text into multi-paragraph units; the units represent the passage of the original text. These algorithms are dependent on cohesion that may be defined as the use of certain linguistic devices to tie the textual units together. On the other hand, lexicon cohesion is the cohesion that is indicated by the relationship between two or more words in two units like the use of synonyms.

Algorithms for Discourse Segmentation

Supervised Discourse Segmentation

The earlier method does not have any hand-labeled segment boundaries. On the other hand, supervised discourse segmentation needs to have boundary-labeled training data. It is very easy to acquire the same. In supervised discourse segmentation, discourse marker or cue words play an important role. Discourse marker or cue word is a word or phrase that functions to signal discourse structure. These discourse markers are domain-specific.

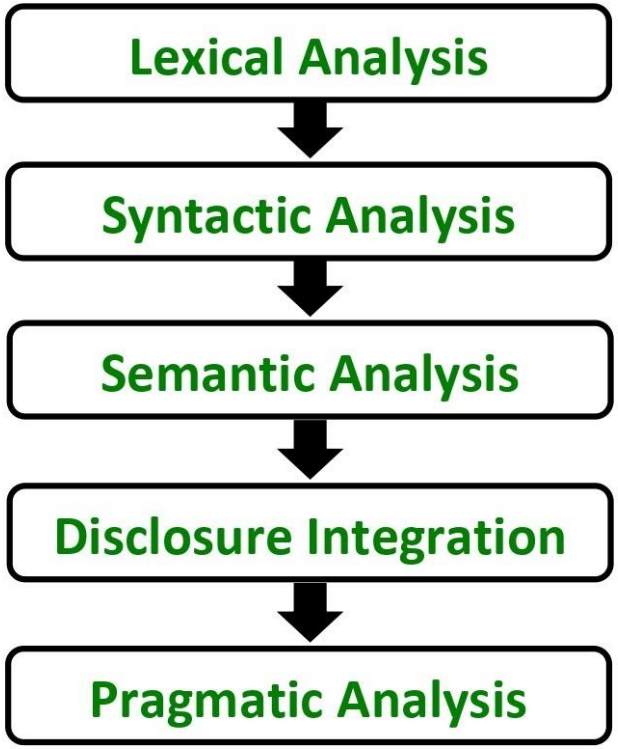
Pragmatic Analysis

The pragmatic analysis means handling the situation in a much more practical or realistic manner than using a theoretical approach. As we know that a sentence can have different meanings in various situations. For example, The average is 18.
The average is 18. (average may be of sequence)
The average is 18. (average may be of a vehicle)
The average is 18. (average may be of a mathematical term)

We can see that for the same input there can be different perceptions. To interpret the meaning of the sentence we need to understand the situation. To tackle such problems we use pragmatic analysis. The pragmatic analysis tends to make the understanding of the language much more clear and easy to interpret.

Implementation:

The five phases discussed above for Language processing are required to follow an order. Each phase takes its input from the previous phase's output and sends it along to the next phase for processing. While this process input can get rejected half-way if it does not follow the rules defining it for the next phase.



How to Deal With Spelling Errors in NLP?

Dealing with spelling errors in Natural Language Processing (NLP) tasks is crucial for improving the accuracy of text-processing applications. Here are several techniques commonly used to handle spelling errors:

Spell Checking:

Dictionary-Based Approaches: Utilize a dictionary or lexicon to check if each word in the text is spelled correctly. If a word is not found in the dictionary, it is considered a potential spelling error.

Edit Distance Algorithms: Algorithms such as Levenshtein distance or Damerau-Levenshtein distance measure the minimum number of edits (insertions, deletions, substitutions, or transpositions) required to transform one word into another. Words with small edit distances to known words in the dictionary can be suggested as corrections.

Phonetic Matching:

Soundex and Metaphone: Phonetic algorithms map words to phonetic representations based on their pronunciation. Words with similar phonetic representations are likely to be spelled similarly, even if spelled differently. This technique helps in identifying spelling errors where words sound alike but are spelled differently.

Language Models:
Statistical Language Models: Use statistical models trained on large text corpora to estimate the probability of a word sequence. Language models can help in identifying likely corrections for misspelled words based on the context of surrounding words.
Neural Language Models: Modern neural language models like Transformer-based models (e.g., BERT, GPT) are effective at predicting and correcting spelling errors by considering the context of the entire sentence. Fine-tuning these models on spelling correction tasks can yield highly accurate results.
Rule-Based Approaches:
Pattern Matching: Apply regular expressions or pattern-matching rules to detect common types of spelling errors, such as repeated characters, missing characters, or transposed letters.
Language-Specific Rules: Develop language-specific spelling correction rules based on common misspellings, phonetic patterns, or morphological rules.

Ensemble Methods:

Combining Multiple Approaches: Combine the outputs of different spelling correction methods, such as spell checking, phonetic matching, and language models, using ensemble techniques to improve accuracy and robustness.

User Feedback:

Interactive Correction: Allow users to provide feedback on suggested corrections and incorporate this feedback to improve the spelling correction system over time. This can be achieved through interactive interfaces or feedback mechanisms in applications.

Domain-Specific Customization:

Custom Dictionaries: Create domain-specific dictionaries or lexicons containing relevant terms and vocabulary to improve the accuracy of spelling correction in specific domains or industries.

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in