Natural Language Processing

Tokenization, Stemming and Lemmatization

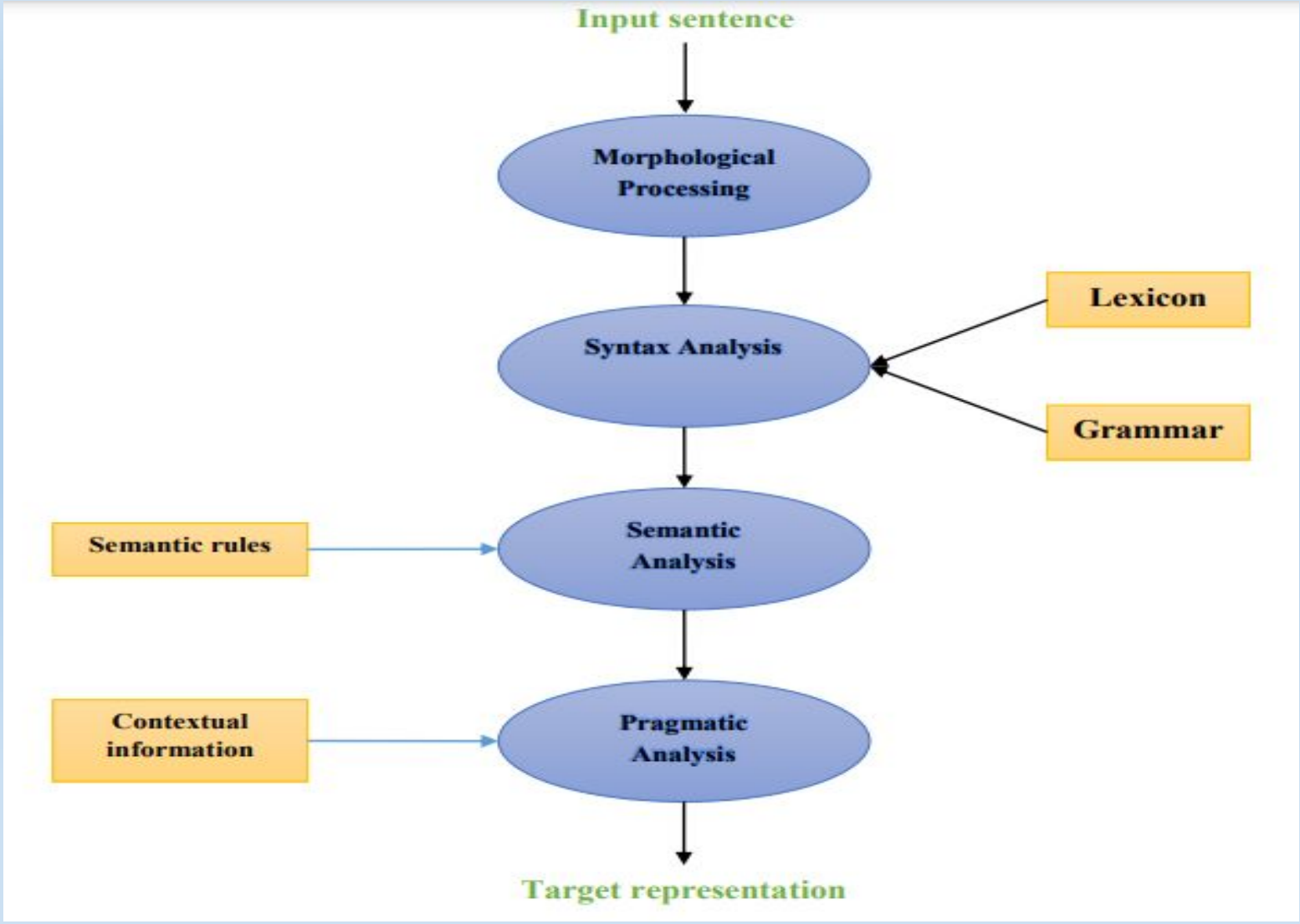
NLTK

CMR UNIVERSITY

- Natural language processing (NLP) is a field that focuses on making natural human language usable by computer programs.
- **NLTK**, or Natural Language Toolkit, is a Python package that is used for NLP.

- A lot of the data analyzing is unstructured data and contains human-readable text. Before analyzing the data programmatically, need to preprocess it.

Components of NLP



- **Morphological Processing**
  - Morphological processing is the first component of NLP. It includes breaking of chunks of language input into sets of tokens corresponding to paragraphs, sentences and words
- **Syntax Analysis**
  - Syntax Analysis is the second component, is one of the most important components of NLP. The purposes of this component are as follows:
  - To check that a sentence is well formed or not.
  - To break it up into a structure that shows the syntactic relationships between the different words.
  - E.g. The sentences like “The school goes to the student” would be rejected by syntax analyzer.
- **Semantic analysis**
  - Semantic Analysis is the third component of NLP which is used to check the meaningfulness of the text. It includes drawing exact meaning, or we can say dictionary meaning from the text.
  - E.g. The sentences like “It’s a hot ice-cream.” would be discarded by semantic analyzer

- Pragmatic analysis
  - Pragmatic analysis is the fourth component of NLP.
  - It includes fitting the actual objects or events that exist in each context with object references obtained by previous component i.e. semantic analysis.
  - E.g. The sentences like “Put the fruits in the basket on the table” can have two semantic interpretations hence the pragmatic analyzer will choose between these two possibilities.

NLP using Python

CMR UNIVERSITY

- Natural language toolkit (NLTK) is the most popular library for natural language processing (NLP) which was written in Python and has a big community behind it.
- Install the NLTK packages by running the following code:
  - import nltk
  - nltk.download()

Tokenization

CMR UNIVERSITY

- It may be defined as the process of breaking up a piece of text into smaller parts, such as sentences and words. These smaller parts are called tokens.
- For example,
  - a word is a token in a sentence, and a sentence is a token in a paragraph.
  - NLP is used to build applications such as sentiment analysis, QA systems, language translation, smart chatbots, voice systems, etc.,
  - Hence, in order to build them, it becomes vital to understand the pattern in the text.
  - The tokens, mentioned above, are very useful in finding and understanding these patterns.
  - We can consider tokenization as the base step for other recipes such as stemming and lemmatization.

Tokenize words

A sentence or data can be split into words using the method **word_tokenize()**:

```
from nltk.tokenize import sent_tokenize, word_tokenize

data = "All work and no play makes jack a dull boy, all work and no play"
print(word_tokenize(data))
```

This will output:

```
['All', 'work', 'and', 'no', 'play', 'makes', 'jack', 'dull', 'boy', ',', 'all', 'work', 'and', 'no'
```

All of them are words except the comma. Special characters are treated as separate tokens.

Tokenizing sentences

The same principle can be applied to sentences. Simply change the to **sent_tokenize()**

We have added two sentences to the variable data:

```
from nltk.tokenize import sent_tokenize, word_tokenize

data = "All work and no play makes jack dull boy. All work and no play makes jack a dull boy."
print(sent_tokenize(data))
```

Outputs:

```
['All work and no play makes jack dull boy.', 'All work and no play makes jack a dull boy.']
```

Stemming

CMR UNIVERSITY

- **Stemming** is a method of normalization of words in Natural Language Processing.
- It is a technique in which a set of words in a sentence are converted into a sequence to shorten its lookup.
- In this method, the words having the same meaning but have some variations according to the context or sentence are normalized.
- In another word, there is one root word, but there are many variations of the same words.
- For example, the root word is “eat” and it’s variations are “eats, eating, eaten and like so”. In the same way, with the help of Stemming in Python, we can find the root word of any variations.

```
1  from nltk.stem import PorterStemmer
2  e_words= ["wait", "waiting", "waited", "waits"]
3  ps =PorterStemmer()
4  for w in e_words:
5      rootWord=ps.stem(w)
6      print(rootWord)
7
```

- There is a stem module in NLTk which is imported. If ifyou import the complete module, then the program becomes heavy as it contains thousands of lines of codes. So from the entire stem module, we only imported “PorterStemmer.”
- We prepared a dummy list of variation data of the same word.
- An object is created which belongs to class nltk.stem.porter.PorterStemmer.
- Further, we passed it to PorterStemmer one by one using “for” loop. Finally, we got output root word of each word mentioned in the list.

Lemmatization

- **Lemmatization** in NLTK is the algorithmic process of finding the lemma of a word depending on its meaning and context.
- Lemmatization usually refers to the morphological analysis of words, which aims to remove inflectional endings.
- It helps in returning the base or dictionary form of a word known as the lemma.

Why is Lemmatization better than Stemming?

CMR UNIVERSITY

- Stemming algorithm works by cutting the suffix from the word. In a broader sense cuts either the beginning or end of the word.
- On the contrary, Lemmatization is a more powerful operation, and it takes into consideration morphological analysis of the words.
- It returns the lemma which is the base form of all its inflectional forms. In-depth linguistic knowledge is required to create dictionaries and look for the proper form of the word.
- Stemming is a general operation while lemmatization is an intelligent operation where the proper form will be looked in the dictionary. Hence, lemmatization helps in forming better machine learning features.

Stemming Code

```
import nltk
from nltk.stem.porter import PorterStemmer
porter_stemmer  = PorterStemmer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
        print("Stemming for {} is {}".format(w,porter_stemmer.stem(w)))
```

```
Stemming for studies is studi
Stemming for studying is studi
Stemming for cries is cri
Stemming for cry is cri
```

Lemmatization Code

CMR UNIVERSITY

```
import nltk
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
        print("Lemma for {} is {}".format(w, wordnet_lemmatizer.lemmatize(w)))
```

```
Lemma for studies is study
Lemma for studying is studying
Lemma for cries is cry
Lemma for cry is cry
```

https://www.youtube.com/watch?v=p1ccbR2P_xA