Layers of NLP architecture

Pragmatic
Semantics
Syntax
Morphology
Phonology
Phonetics - sound

Imp:

Q) write a prog. to tokenize a given statement

token = word

corpus = collection of tokens/words like paragraph

Imp:

Q) WAP to remove the stop words

Imp:

Q) WAP to carry out stemming or lemmitization for given tokens.

1. Tokenization (seprator can be space or comma or etc.)
2. Removal of stop words (is, are, the, of, this...)
3. Stemming / Lemmitization (root word like watching → watch, worked → work, removes suffix)
4. POS tagging
5. Chunking
6. TF-IDF (Term Freq. - Inverse document freq.)

There are 4 imp. types of stemmers

1. Porter stemmer
2. Lancaster stemmer
3. SnowBall stemmer
4. Regular Expression stemmer

1. Porter Stemmer (1980s)

It was basically used to find the root of the word by removing suffix of a given word

2. Lancaster Stemmer

It is more aggressive than porter stemmer since it works also with the prefix of the words to find the root of the word

3. Snowball stemmer

It was basically an update of porter stemmer which permitted stemming or lemmitization of foreign words apart of english lang.

4. Regular expression stemmer

It is customized based stemming modu that permits to generate root of the given tokens based on regular expression.

Imp:
Q) WAP to carry out parts of speech tagging

Important POS tagger

1. Rule based Parts of Speech tagging

These POS tagger is based on handcrafted rules to assign parts of speech to the provided tokens

2. Probabilistic based tagging → Viterbi algo (Markov model)

It is based on Viterbi algo which is designed on Hidden Markov Model (HMM)

This tagger basically tries to find the parts of speech for every given token based on the recent collection of tokens than the set of tokens used a long time before.

It is based on freq of a typical word that helps to decide the inclination of the paragraph

3. Transformation tagger

It is based on training data of the corpus which helps to upgrade the context of the words and one of the popular example is •Average Perceptron Tagger.

4. Deep Learning

When the corpus is provided to the RNN. It helps to understand the concept and collect only the required set of tokens in the database and unnecessary tokens are removed.

Q) WAP to carry out chunking of the words based on the parts of speech tagging.

Chunking is basically carried out to group or cluster a set of tokens once parts of speech tagging has been provided to ~~the~~ each token. The chunkers try to collect meaningful tokens that help to implement meaning to the given sentence and the command can be understood by the machine.

Based on parts of speech, the chunking is carried out in following categories:

Noun phrases

The tokens are clusted depending on a noun phrase in the given token & its associated phrases that can result into meaningful conversations.

Similarly other categories are

Verb phrases Helps to identify given verb in given sentence & possible task associated with it.

Preposition chunking

Similar to above ones.

Named Entity chunking

Helps to identify names of given person, place or object specifically associated with the nouns. This chunking helps to understand the inclination of the complete paragraph.

Clause Chunking

The clause chunking deals with particular set of tokens that are assumed to be unchangable as they might depict the policies given by an organization or legal rules & regulations.

Parsing

Parsing helps to decide the syntax to the given NLP machine. It makes use of the different syntax rules provided by the language rules. Most of the parsing approaches are statistical, probablistic or machine learning based. It makes use of different syntactic categories which generally appear in the abbreviation format.

Parsing Compiler → Syntactic structure → syntactic categories

Statistical Probablistic OR Machine Learning Based

np → noun phrase
vp → verb phrase
det → determiner
s → sentence
n → noun
tv → transitive verb
iv → intransitive verb
prep → preposition
adj → adjective
NN → proper noun

4 types of Parsing :-

Rule based Parsing

Syntax rules are predefined for a particular language.

They are generally not preferred since a complete database defining all the possible rules have tobe set as an input which causes high retrival complexities.

Probablistic based Parsing

Where parsing technique utilise ML algos. to learn the patterns from the annoted corpora.

Morphological parsing

It deals with 3rd layer of NLP architecture & carries out the following function

1. Formation of words from alphabets
2. Finding the origin of word ie. stemming or lemmatization
3. Gramatical structure for a given sentence
4. Use of prefixes or suffix with proper spelling & syntax rules
5. Parts of speech identification to identify role of every word in a given sentence.

Components of Morphological parsing

1. Stems
2. Prefixes / Suffix
3. Word order

Stems

Stems is a type of morpheme.

It is the core meaning of any given vocabulary. In other words it is root of word which is exagerated over time using a suffix or infix.

Stems has got affixes which add an additional meaning to a given root of the word.

Affixes can be divided into 4 diff. types :

1. suffix : It follows the stem
   Eg. cat → cats
2. Prefix : Preceding the stem
   grace → disgrace
3. Infix : addition of extra word in between roots
   eg: cupful → cupsful
4. Circumfixes : They precede & follow the stem
   eg. understand → misunderstanding

Word Order

There are 3 different components of word order :-

1. Lexicon : It is basically a list of affixes along with the parts of speech
   eg. describing whether the stem is a noun or a word
2. Morphotactics : which tries to map the morphone order
3. Orthographic rules : which helps to desc. how the stem can add additional meaning to it.
   eg. city can not be written as citys but cities

Describe passing & its types.

Types of derivations

Explain phrase structure grammer

Describe context free Grammer

WAP to find the similar words using word to vector model from 2 paragraphs

Stemming & its diff. categories of stemmers

Stemming is an NLP technique basically used to remove affixes & find the root words.

Lematization involves understanding of ctx. and morphological order of words. Once this is achieved, it is able to find the base form of the word.

2. Stemming is a rule based algo whereas lematization utilizes dynamic approach
3. Stemming produces stems which may not be the actual word whereas lematization produces an output of valid words that can be found in dictionary.
4. Stemming produces approx. accurate result where lematization is more precise & has better accuracy.

Stemming has reduced time complexity & is more scalable whereas in lematization time complexity increases & scalability feature is reduced.

Different stemmers & diff. NLP packages are being used for stemming

Regular expression stemmers permits custom rules & regulations to find the root word to given word. It generally categorize the design to end the regular expression pattern & customise as per convenience & is considered to have dynamic approach.

Lematizers (5 types)

1. Rule based lematizers

Early ones to be produced based on linguistic patterns that helps remove ambiguity from language.

2. Dictionary based lematizers

Uses dictionary mapping & patters and strictly follows alphabetical order of dictionary.

considered to be slow since it requires to lookover through all words of dictionary for lematization.

3. Hybrid lematizers

Makes use of rule based techniques along with the dictionary based mapping.

4. ML based lematizers

When ML algos are utilized to find the probable root word, they are considered to be an intelligent approach than the previous ones.

5. Transformer based lematizers

In present scenario, large amount of data are analysed & mapped with req. vocabulary using transformer which are basically a high end deep learning algorithms.

TF-IDF comprises of 2 parts:
Term frequency
Inverse document freq.

TFIDF is a statistical methodology in NLP that helps to decide the importance of given words in the collection of documents.

TF
It tries to calculate how many times a given word appears in a document
It can be calculated by using formula

$$TF = \frac{\text{no. of times a term appears}}{\text{total no. of terms in a corpus}}$$

IDF helps to understand the importance of the term in the collection of given corpus. It can be calculated as

$$\frac{\text{total no. of corpus}}{\text{no. of corpus containing term T}}$$

Eg. In a library out of 1500 books, assume 400 books has a term 'NLP' in it.

$$\therefore IDF = \log\left(\frac{1500}{400}\right)$$

There are basically 4 diff. types of TFIDF

1. Normalised TF-IDF
2. Probabilistic TF-IDF
3. Absolute TF-IDF
4. Best matching 25 TF-IDF

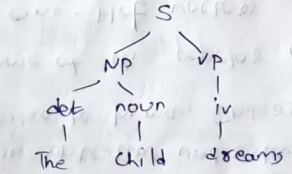
Normalised TF-IDF

values are required to be normalised before processing the documents.

PARSING

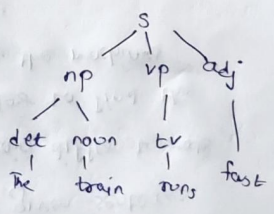
s → The child dreams

s → np vp
np → det noun
vp → iv



s → The train runs fast
noun verb adjective

s → np vp adj
np → det noun
vp → transitive verb
adj → adj



In order to automate the given production rules for a lang. chomskey hierarchy based parsing is carried out which helps to replace certain terminals or non-terminals in the pre-defined prod. rules to generate new set of prod. rules that can help to understand new commands given by the humans making sure that parsing loop does not enter into infinite loop.

Q) Describe 2 diff. techniques in NLP that helps to generate words from given sentences.

In NLP the basic task is to obtain different words and their relevant freq. in order to find the basic contextual understanding by the machines similar to human understanding.

Some of the popular ML techniques are

1. One-Hot vectors
2. Bag of words

These algos work with numeric values by converting all the words in the form of numeric values.

One Hot Vector

It makes a matrix comprising of 0 & 1 where each word is defined as 1 and the remaining words are depicted as 0.

	Rome	Paris	Washington
Rome	1	0	0
Paris	0	1	0
Washington	0	0	1

One Hot vector are ~~considerably~~ expensive and they do not consider the similarity of words.

In case corpus comprises of 1000 words it may result into a very large matrix thereby increasing time & space complexity.

Bag of Words

Bag of words is another approach where we take a document & find out frequencies of each word in the listed corpus.

These frequencies are utilized by ML algos. for various applications and to simplify the task of finding the inclination of the given corpus.

Bag of words is a collection of all the words that are present in the document along with their freq.

India is a country. India has many states.

BoW = { 'India': 2, 'is': 1, 'a': 1, 'country': 1, 'has': 1, 'many': 1, 'states': 1 }

Problems with bag of words

Bag of words is too simple algorithm which uses the contextual understanding & also does not follow the order of a particular sequence which creates at times confusion ~~at the time~~ for the analysis purpose.