Kushal Dubey
21 October 2024 20:33

NLP

Natural language processing.
↓
Computational linguistic language.

Computer science + Artificial intelligence + Human language.

Language which machines can easily understand by humans is known as NLP.

Applications of NLP :-

1) Speech Recognition → Google assistant & Siri
2) Sentimental analysis → we can analyze Facebook & twitter comments & tweets that which are good or bad known as sentimental analysis
3) Machine translators → Google translator.
4) Chatbots etc.

NLP Diagram :-

I/p → Automated speech recognition → NLU → NLG → o/p

Eg:- google translate.

steps of NLP :-

Tokenization → tokens.
↓
Steeming → Stems (stemming): Know, Knows, Kn-ing → Know
↓
Lemmitization → lemmitization: Date, Dated, Dating → Date; Die, Died, Dying → Die (meaning); Date, Dated, Dating → Date (meaning)
↓
Pos tag → Nouns, verbs, Pronouns
↓
Name recognition → name, location, company
↓
Chunkization → chunks: pronoun, noun, [illegible].

Imp

* Levels of NLP :-

1) Morphological Analysis :- It is a type of analysis used to make the tokens in the sentence.

For Eg:- truthfulness → truth, ful, ness

Generally study of words / root of words.

2) Syntax analysis :- Sentence is syntaxically correct or not.

Dog ate my Homework → X Incorrect.

Eg:- Anurag ate the rat.

Anurag → noun, ate → verb, the → Determinal, rat → Noun.

NP VP NP
↓
Syntaxically correct.

3) Semantic analysis :- word makes some sense or not.

Eg:- Plant ! Industry / organism → syntaxically correct → semantically not correct.

4) Pragmatic Analysis :- If sentence is syntaxically & semantically correct but it should have more than one meaning then it is known as Pragmatic analysis.

Eg:- Dipesh loves his Girlfriend & Manan does too.

Means → Manan loves his girlfriend
or
Manan loves Dipesh girlfriend

Two meaning

5) Discourse analysis :- It is analysis which is been understandable when the sentence affects before.

* Ambiguity :- It means if a sentence has more than one meaning in itself is called ambiguity.

Eg:- Anurag eat a bat → animal
I dont have my baseball bat → thing
} 2 meanings → ambiguity

Types of Ambiguity :-
1) Lexical ambiguity
2) Syntax ambiguity
3) Semantic ambiguity
4) Pragmatic ambiguity

* Lexical :- If a word have more than one meaning known as Lexical ambiguity.

Eg:- I can (→ verb) play cricket.
Give me that can → thing
} 2 meanings in word.

[illegible] If a sentence have more than one

* Lexical. ~~σ~~ known as Lexical ambiguity.

Eg.- I can (verb) play cricket.
Give me that can → thing } 2 meanings in word

* Syntatic Ambiguity:- If a sentence have more than one meaning known as Syntatic Ambiguity

Eg.- Abid saw man with binnocular.
Abid saw the men carrying binnoculars. } Sentence have 2 meaning
Abid saw the man through the binnoculars.

* Semantic:- 2 other meanings in sentence.

* Pragmatic:- 2 meaning in sentence.

* Morphological parsing:- Breaking words into stems or chunks or tokens.

Rules → stem → root work
Rules → afix → prefix, suffix, infix

Eg:- Mangoes.
Mango | es. → stem

afix → landed → suffix
→ arguard → prefix.
→ passersby → Infix.

* Rules depend upon morphology analysis.
1) Lexicon → {stems / afix}
2) Morphotactics → meaningful words — Eg.- useableness → use, able, ness; ~~able useness~~ ×
3) orthographic rules

lady + s = ladys ×

lady + s = ladies ✓

Generally we have to focus of stem & afix & also the meaningful sentence or word.

* N gram model:-

*. Language model.

$$P(x,y) = P(x)\cdot P(y/x)$$
$$P(x,y,z) = P(x)\cdot P(y/x)\cdot P(z/x,y)$$

Eg.- $P(x,y) = P(x)\cdot P(y/x)$
= Google it
$P(\text{google}, \text{it}) = P(\text{google})\cdot P(\text{it}/\text{google})$.

Eg.- She is dead
$P(x,y,z) = P(x)\cdot P(y/x)\cdot P(z/x,y)$
$\Rightarrow P(\text{she})\cdot P(\text{is}/\text{she})\cdot P(\text{dead}/\text{she is})$

* Markov assumption:- use generally for more words in a sentence.

Eg:- I wish I was a unicorn.
Always start the last word.
$P(\text{unicorn}/\text{I wish I was a}) \approx P(\text{unicorn}/a)$
or
$P(\text{unicorn}/\text{was a})$
Means it can include the last word in sentence and may include last second as well as last third word in it known as Markov assumption.

Q
Examples of N gram numericals:-
1) <S> I am a human </s>
2) <S> I am not a stone </S>
3) <S> I I live in Mumbai </S>
Check the probability of <S> I I am not </s> using bigram

$P(\text{I I am not}) = P(I/\langle s\rangle)\cdot P(I/I)\cdot P(\text{am}/I)\cdot P(\text{not}/\text{am})\cdot P(\langle/s\rangle/\text{not})$

$$\Rightarrow P\left(\frac{\langle S\rangle / I}{\langle S\rangle}\right)\cdot P\left(\frac{I/I}{I}\right)\cdot P\left(\frac{I/\text{am}}{P(I)}\right)\cdot \frac{P(\text{am}/\text{not})}{P(\text{am})}\cdot P\left(\frac{\text{not}/\langle/s\rangle}{\text{not}}\right)$$

$$\Rightarrow \frac{3}{3}\cdot\left(\frac{1}{4}\right)\cdot\frac{1}{2}\frac{1}{2}\cdot 0 = 0 \text{ Ans}$$

$P(I \langle S\rangle)$
$= P\left(\frac{\langle S\rangle . I}{\langle S\rangle}\right)\times P(\text{am}/I)$
$P\left(\frac{I/\text{am}}{I}\right)$
$P(\text{not}/\text{am})$
$P\left(\frac{\text{am not}}{\text{am}}\right)$
$P\left(\frac{\langle S\rangle/\text{not}}{P(\text{not} \langle S\rangle)}\right)$ not

Q
$P(\langle s\rangle \text{ I want english food } \langle/s\rangle)$
$P(I/\langle s\rangle)\times P(\text{want}/I)\cdot P(\text{english}/\text{want})\cdot P(\text{food}/\text{english})\cdot P(\langle/s\rangle \mid \text{food})$

$$P\left(\frac{\langle s\rangle / I}{\langle s\rangle}\right)\cdot P\left(\frac{I/\text{want}}{I}\right)\cdot P\left(\frac{\text{want}/\text{english}}{\text{want}}\right)\cdot P\left(\frac{\text{english}/\text{food}}{\text{english}}\right)\cdot P\left(\frac{\text{food}/\langle/s\rangle}{\text{food}}\right)$$

$$\frac{1}{1}\times\frac{1}{1}\times\frac{1}{1}\times\frac{1}{1}\times\frac{1}{1} = 1$$

Q
<S> I live in Delhi </s>
<S> Delhi is the capital of india </s>
<S> The capital of Raj is Jaipur </s>
<S> MVJ is in Jaipur </s>
Calculate <S> I live in Jaipur </s>

$P(\text{I live in Jaipur}) = P(I/\langle s\rangle)\cdot P(\text{live}/I)\cdot P(\text{in}/\text{live})\cdot P(\text{Jaipur}/\text{in})\cdot P(\langle/s\rangle/\text{Jaipur})$.

<s> MUJ is in Jaipur </s>

Calculate <s> I live in Jaipur </s>

$P(\text{I live in Jaipur}) = P(\text{I}/\langle s\rangle) \cdot P(\text{live}/\text{I}) \cdot P(\text{in}/\text{live}) \cdot P(\text{Jaipur}/\text{in}) \cdot P(\langle/s\rangle/\text{Jaipur})$.

$+\; P\left(\frac{\langle s\rangle/\text{I}}{\langle s\rangle}\right) \cdot P\left(\frac{\text{I live}}{\text{I}}\right) P\left(\frac{\text{live in}}{\text{live}}\right) \cdot P\left(\frac{\text{in Jaipur}}{\text{in}}\right)$

$P\left(\frac{\text{Jaipur}\langle/s\rangle}{\text{Jaipur}}\right)$

$\frac{1}{4} \times \frac{1}{1} \times \frac{1}{1} \times \frac{1}{2} \times \frac{2}{2}$

$\frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$ $\quad 8\overline{)1.0}\; 0.128 = 0.12$

* Consider the following data.

<s> I am Jack </s>

<s> Jack I am </s>

<s> Jack I like </s>

<s> Jack I do like </s>

<s> do I like Jack </s>

Assume that we use bigram based language model on data.

What is most probable next word predicted by model?

1) <s> Jack ___ I (2) <s> Jack I do ___ I like (3) Jack I am Jack ___ I.

(4) <s> do I like ___ </s>

Solution:- $P(\text{I}\,|\,\langle s\rangle) = P\left(\frac{\langle s\rangle\ \text{I}}{\langle s\rangle}\right) = \frac{1}{5}$

$P(\text{Jack}\,|\,\langle s\rangle) = P\left(\frac{\langle s\rangle\ \text{Jack}}{\langle s\rangle}\right) = 3/5$

$P(\text{am}/\text{I}) = P\left(\frac{\text{I am}}{\text{I}}\right) = \frac{2}{5}$

$P(\text{like}/\text{I}) = P\left(\frac{\text{I like}}{\text{I}}\right) = 2/5$

$P(\langle s\rangle/\text{Jack}) = P\left(\frac{\text{Jack}\ \langle s\rangle}{\text{Jack}}\right) = \frac{2}{5}$

$P(\langle s\rangle\ \text{am}) = P\left(\frac{\text{am}\ \langle s\rangle}{\text{am}}\right) = 1/2$

$P(\text{Jack}/\text{like}) = P\left(\frac{\text{like Jack}}{\text{like}}\right) = 1/3$

$P(\text{I}/\text{do}) = P\left(\frac{\text{do I}}{\text{do}}\right) = 1/2$

$P(\text{do}/\text{I}) = P\left(\frac{\text{I do}}{\text{I}}\right) = \frac{1}{5}$

$P(\text{do}/\langle s\rangle) = P\left(\frac{\langle s\rangle\ \text{do}}{\langle s\rangle}\right)$

$\frac{1}{5}$

* $P(\langle s\rangle\,|\,\text{like}) = P\left(\frac{\text{like}\ \langle s\rangle}{\text{like}}\right)$

$\frac{2}{3}$

$P(\text{like}/\text{do}) = P\left(\frac{\text{do like}}{\text{do}}\right) = \frac{1}{2}$

$P(\text{I}/\text{Jack}) = P\left(\frac{\text{Jack}/\text{I}}{\text{Jack}}\right) = 3/5$

Test sentence

Q The arabian knights are the fairy tailes taker of the east).

Soln:- $P(\text{The}/\langle s\rangle) = P\left(\frac{\langle s\rangle\ \text{The}}{\langle s\rangle}\right) = \frac{2}{3}$

$P(\text{arabian}\,|\,\text{The}) = P\left(\frac{\text{The arabian}}{\text{The}}\right) = \frac{1}{2} = 0.5$

$P(\text{knights}\,|\,\text{arabian}) = P\left(\frac{\text{arabian knights}}{\text{arabian}}\right) = \frac{2}{2} = 1$

$P(\text{are}\,/\,\text{knights}) = P\left(\frac{\text{knights are}}{\text{knights}}\right) = 1/2 = 0.5$

$P(\text{the}\,/\,\text{are}) = P\left(\frac{\text{are}/\text{the}}{\text{are}}\right) = \frac{1}{2} = 0.5$

$P(\text{fairy}\,|\,\text{the}) = P\left(\frac{\text{the fairy}}{\text{the}}\right) = \frac{1}{3} = 0.33$

$P(\text{tales}\,|\,\text{fairy}) = P\left(\frac{\text{fairy tailes}}{\text{fairy}}\right) = \frac{1}{1} = 1$

$P(\text{fairy}|\text{the}) = \frac{}{\text{the}} = \frac{}{3} = 0.33$

$P(\text{tales}|\text{fairy}) = \frac{P(\text{fairy tales})}{\text{fairy}} = \frac{1}{1} = 1$

$P(\text{of}|\text{tales}) = \frac{P(\text{tales of})}{\text{tales}} = \frac{1}{1} = 1$

$P(\text{the}|\text{of}) = \frac{P(\text{of the})}{\text{of}} = \frac{2}{2} = 1$

$P(\text{east}|\text{the}) = \frac{P(\text{the east})}{\text{the}} = \frac{1}{3} = 0.33$

$\Rightarrow \frac{2}{3} \times \frac{1}{2} \times 1 \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{3} \times 1 \times 1 \times 1 \times \frac{1}{3}$

$= 0.0061728395$

(*) Parsing:- How to create the parse tree using CFG rules.

→ Q construct a parse tree for the following sentence using CFG rules.

→ The man read this book.

rules:- S → NP VP , S → AUX NP VP , Nom → Noun

VP → verb NP , S → VP , NP → DET Nom , Nom → Noun Nom

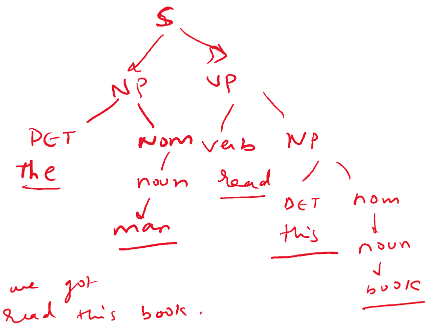
DET → that | this | a | the.

Noun → book | flight | meal | man

verb → book | include | read

AUX → does.

Ans:-



Hence we got
The man read this book.

(*) Top Down & Bottom up parsing:-

Grammar :- S → VP

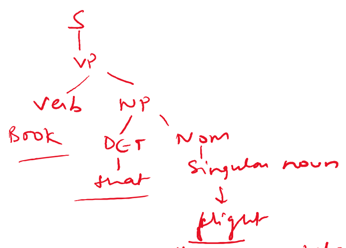
VP → Verb NP

NP → DET Nom , NP → DET Nom , DET → that,

Nom → Singular Noun , Verb → Book , Singular noun → flight.

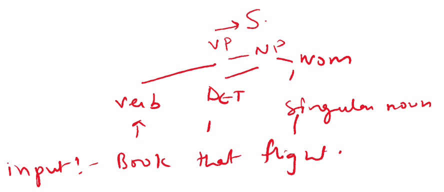
→ input → Book that flight.

Top Down parsing approach:-



→ Hence "Book that flight" is completed.

* Bottom up parsing:-



LHS → terminal (only one)
RHS → nonterminal (more than one)

CKY only work on the CNF
CNF → Chomskey Normal form.

CKY only work on the CNF → Chomsky Normal form.

CNF rules :-
- → one single terminal ✓
- → two non terminal ✓
- → combination of terminal & non terminal ✗
- → Single non terminal ✗

If we want to convert the CNF if one terminal & nonterminal is given then we have to take the dummy model.

Eg:- NP → the Nominal
(the: terminal ✗; Nominal: NT)

we can introduce dummy non terminal
Det → the
NP → Det Nominal

Eg:- NP → Det Noun PP ✗
Nominal → Noun PP
NP → Det Nominal ✓ → CNF

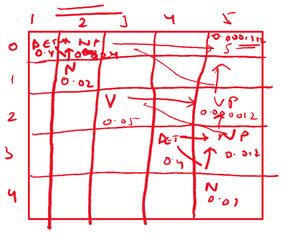
Q How to convert CFG to CNF using rules?

S → NP VP [0.80]
NP → DET N [0.1]
VP → V NP [0.20]
(syntactic words grammar)

V → includes [0.05]
DET → the [0.4]
DET → a [0.4]
N → meal [0.01]
N → flight [0.02]
(Lexical words)

0 The 1 flight 2 includes 3 a 4 meal 5

5×5



Using CKY

Probabilistic CKY . NP 0.3 → DET 0.4, N 0.02 = 0.3 × 0.4 × 0.02 = 0.024

VP → 0.20 → V 0.05, NP 0.012 ⇒ 0.000012

NP 0.3 → DET 0.4, N 0.01 ⇒ 0.012

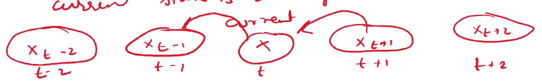
S 0.80 → NP 0.024, VP 0.000012

S ⇒ 0.0000192

* Markov Rule :-

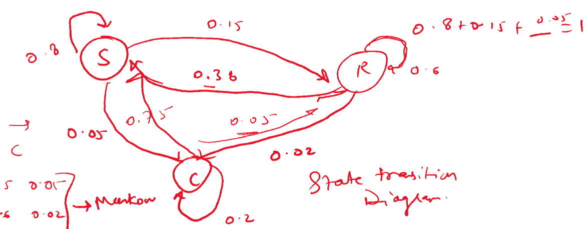
It says if a future state will only depend upon the current state not the past state event how current state is occuring.

Eg:-



$P(X_t / X_{t-1})$ → conditional probability.

$P(X_{t+1} / X_t)$ → X_{t+1} → X_t



0.8 + 0.15 + 0.05 = 1

State transition Diagram.

Future →

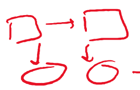
$$
\begin{array}{c|ccc}
 & S & R & C \\
S & 0.8 & 0.15 & 0.05 \\
R & 0.38 & 0.6 & 0.02 \\
C & 0.75 & 0.05 & 0.2
\end{array}
$$
→ Markov

current

Transition probability matrix

HMM → Hidden markov model → hidden states have. we have observable things also.

Eg:- umb, coat, short — S



→ hidden state
→ observable state.

Eg:- N (hidden), T (observable)
$2^3 = 8$

Postagging

P(C, C, U, SU, RA, CL)
= P(C/SU) × P(C/RA) × P(U/CL)
× P(SU) × P(RA/SU) × P(CL/RA)

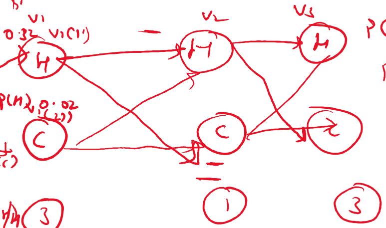
$P(C, C, U, SU, RA, CL)$

$= P(C/SU) \times P(C/RA) \times P(U/CL)$

$\times P(SU) \times P(RA/SU) \times P(CL/RA)$

{markov depend upon before state}

Eg:-



$P(3, H) = P(3/H) \cdot P(H)$

$P(3, C) \to P(3/C) \cdot P(C)$

$P(1, H) = P(1/H) \cdot P(H/H)$ 3

$P(1, C) = P(1/C) \cdot P(C/H)$

$P(3, H) = P(3/H) \cdot P(H/H)$

$P(3, C) = P(3/C) \cdot P(C/H)$

$P(3, C) = P(3/C) \cdot P(C/C)$

$P(3, H) = P(3/H) \cdot P(H/C)$

$P(1, C) = P(1/C) \cdot P(C/C)$

$P(1, H) = P(1/H) \cdot P(H/C)$

And if we will multiply all terms we will get highest which is the combination.