NLP

→ an automated system capable of processing and analyzing text data

machine translation

→ conversion of one language to other

Zipf's law

A relationship between the frequency of a word and it position in its list

$$f \propto \frac{1}{r}$$

(or)

$$f_r = k$$

The number of means m of a word obeys the law

$$m \propto \sqrt{f}$$

Heap's law:-

Let $|V|$ size of vocabulary & N be no. of tokens

$$|V| = kN^{\beta}$$

Typically

$k = 10 - 100$

$\beta = 0.4 - 0.6$ (roughly sq root)

Text pre-processing :-

Tokenization is process of segmenting a string of characters into words.

sentence Tokenization → splitting into sentences.

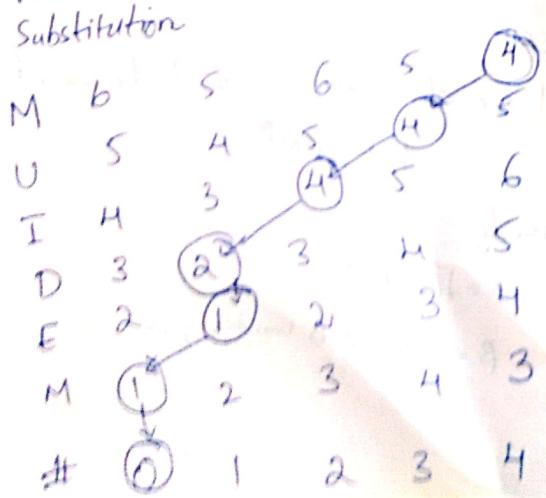
Normalization
- Case folding
- stemming
- Lemmatization

Morphology :-
↳ internal structure of words

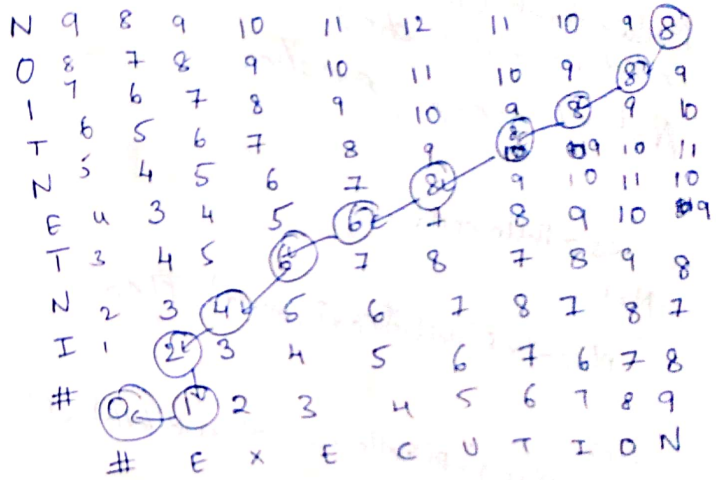
Porter's algorithm

PyEnchant → package (spelling correction)

- Insertion
- Deletion
- Substitution



MEDIUM
*ENUMds



~~INTENTION~~
~~EXECUTION~~

INTE*NTION
*EXECUTION

O(mn) → Edit distance

O(m+n) → backtracking

Levenshtein Edit distance

	Correct	Error
Correct	TP	TN
Not correct	FP	FN

True Positive
False +ve
True -ve
False -ve

Probabilistic Language models:-
Machine translation
Context based spell check
Natural language generation.

<s> Helloworld </s>

N=1

P(<s>) x P(Hello) x P(world) x P(</s>)

N=2

P(<s>) x P(Hello/<s>) x P(world/Hello) x P(</s>|...)

N=3

P(<s>) x P(Hello/<s>) x P(world/<s>Hello) x P(</s>|...)

bigrams
the dog smelled like a skunk

the dog
dog smelled
smelled like
like a
a skunk

the dog smelled
dog smelled like
smelled like a
like a skunk

11542/90000, 7035/88000, 6268/7855, 5071/65000, 8791/7855, 2461/5000

P(I|<s>) = 3/5

P(Sam|<s>) = 1/3

P(</s>|Sam) = 0

P(am|I) = 2/4

P(do|I) = 2/4

P(Sam|like) = 2/3

P(like|I, do) = 0

P(Sam|not like) = 2/3

P(Sam|do not like) = 2/3

allegation	3/7	(3+3)/(7+21) = 6/28	x 7/4	1.5
reports	2/7	5/28	5/28 x 7	1.25
claim	1/7	4/28	4/28 x 7	= 1
request	1/7	4/28	4/28 x 7	1
attack	0/7	3/28	3/28 x 7	0.75
man	0/7	3/28	3/28 x 7	0.75
outcome	0/7	3/28	3/28 x 7	0.75

"data sparsity

Backoff & Interpolation :-

$$c(w_i/w_{i-2},w_{i-1}) = \lambda_1\left(w_i/w_{i-2},w_{i-1}\right) + \lambda_2\left(w_i/w_{i-1}\right) + \lambda_3$$

$$\lambda = \lambda_1 + \lambda_2 + \lambda_3$$

$$c^* = \frac{(c+1)\,N_{c+1}}{N_c}$$

3 allegation
2 reports
1 claims
1 request
0 attack
0 man
0 outcome

Given that in the trainig corpora the ~~bigrams~~ denied the allegal

Estimate the modified count c^* and the modified probability of the unseen trigram. Also find P_w for existing data. Use good turing estimate for calculation

Answer: $N_0 = 3$ (unseen trigrams) $N_1 = 2$ $N_2 = 1$ $N_3 = 1$ (total no. of unseen trigram in left side)

$$c_0^* = \frac{1\cdot 2}{3} = \frac{2}{3}$$

$$c_1^* = \frac{2\cdot 1}{2} = 1$$

$$c_2^* = \frac{3\cdot 1}{1} = 3$$

$$P_0^* = \frac{c^*}{N} = \frac{2}{3}\times\frac{1}{7} = \frac{2}{21}$$

$$P_1^* = 1\times\frac{1}{7} = \frac{1}{7}$$

$$P_2^* = 3\times\frac{1}{7} = \frac{3}{7}$$

P_3^*

clock goes tick
Time on clock
Time on clock is twelve

clock – 3 time – 2
goes – 1 on – 2
tick – 1 ~~clock – 2~~
is – 1
twelve – 1

The unseen unigrams in the trainig corpora are value, welcome, wait usig good turig estimate. Find the modified probability values of P^*(value) P^*(time), P^*(tick)

$N_0 = 3$ $N_1 = 4$ $N_2 = 2$ $N_3 = 1$

$$c_0^* = \frac{1\cdot 4}{3} = \frac{4}{3}$$

$$c_1^* = \frac{2\cdot 2}{4} = \frac{4}{4} = 1$$

$$c_2^* = \frac{3\cdot 1}{2} = 1.5$$

c_3^* = —

P_0^*

$\frac{4}{3}\times\frac{1}{14} = \frac{11}{112}$ $\frac{4}{42}$

$1\times\frac{1}{14} = \frac{3}{32}$ $\frac{1}{11}$

$\frac{3}{2}\times\frac{1}{14} = \frac{1}{16}$ $\frac{3}{22}$

P^*(value) = $\frac{4}{33}$

P^*(time) = $\frac{3}{22}$

P(tick) = $\frac{1}{11}$

If size of vocabulary V is given as input then no. of unseen grams should be calculated as (unigrams)
V-(bigrams intialized)

V^2 - (bigrams) (bigrams)

V^3 - (trigrams) (trigrams)

Witten Bell

$$P^* = \frac{C^*}{N} = \frac{1}{Z}\left(\frac{T}{N+T}\right)$$

$$C^* = \frac{N}{Z}\left(\frac{T}{N+T}\right) \quad C=0$$

$$C^* = C1\left(\frac{N}{N+T}\right) \quad C\neq 0$$

POS Tagging:-

CRF - Conditional Random Field

HMM - Hidden Markov Model.

denied the allgations = 3
denied the claims = 1
reports = 2
request = 1

T=4

denied the attack
denied the attack

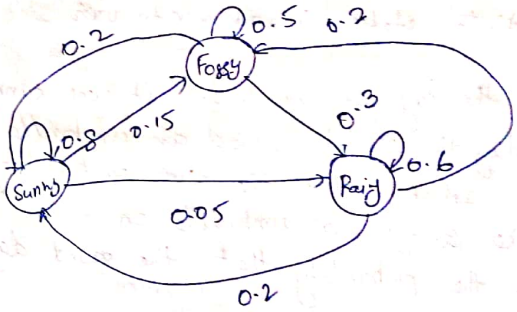
C	Nc	C^*
0	3	$\frac{7}{3}\times\frac{4}{11} = \frac{28}{33}$
1	2	$\frac{2}{1}\times\frac{7}{11} = \frac{14}{11}$
2	1	$\frac{2}{1}\times\frac{7}{11} = \frac{14}{11}$
3	1	$\frac{3}{1}\times\frac{7}{11} = \frac{21}{11}$

POS Tagging:-

Classification — sequence labelling

Assume transitions in weather are indicated by a series of states which indicate what is the likelyhood tomorrow's weather given today's.

Today \	Sunny	Rainy	Foggy
Sunny	0.8	0.05	0.15
Rain	0.2	0.6	0.2
Foggy	0.2	0.3	0.5



Given that it is sunny today what is the probability that tommorrow is sunny and day after is rainy.

$$P(w_2=\text{sunny}, w_3=\text{rainy} \mid w_1=\text{sunny}) = P(w_2=\text{"sunny"} \mid w_1=\text{"sunny"}) \times P(w_3=\text{rainy} \mid w_2=\text{"sunny"})$$

0.0410

$$= 0.08 \times 0.05 = 0.04$$

$P(w_2 = \text{Rain} / \text{foggy}) \times P(\text{rainy} / \text{rainy})$

$= 0.3 \times 0.6$

$= 0.18$

$P(w_3 = \text{rainy} / w_1 = \text{foggy})$

$0.2 \times 0.5 + 0.3 \times 0.6 + 0.5 \times 0.3$

$= 0.34$

Hidden Markov Model is a state where one of the intermediate stated is unknown to user

Suppose the day you were locked it was sunny the next day the caretaker carried an umbrella into the room assumes that the prior probability of the caretaker carrying an umbrella on any day is. What is the probability that the second day was

	Probability of carrying umbrella
Sunny	0.1
Rain	0.8
Foggy	0.3

Markow

$$P(w_1, w_2 \ldots / (u_1 u_2 \ldots u_n)_2 \; P(u_1, u_2 \ldots u_n / w_1, w_2 \ldots w_n) \times P(w_1, w_2 \ldots w_n)$$

$$P(u_1, u_2 \ldots u_n)$$

$$P(u_1, u_2 \ldots u_n / w_1, w_2, \ldots w_n) = \pi P(u_i / w_i)$$

$$P(w_2 = \text{Rain}, w_1 = \text{Sunny} / u_2 = \text{True})$$

$$= \frac{P(w_1 = \text{Sunny} / u_2 = \text{True})}{P(w_1 = \text{Sunny})}$$

$$= P(u_2 = \text{True} / w_2 = \text{Rain}) * P(w_2 = \text{Rain} / w_1 = \text{Sunny})$$

$$P(u_2 = \text{True}) \times P(w_1 = \text{Sunny})$$

$$= \frac{0.8 \times 0.05}{0.5} = 0.08$$

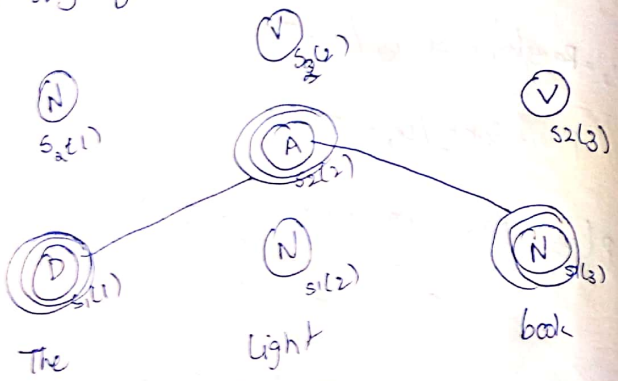
	Det	Noun	Adj	Verb
Det	0	0.5	0.3	0.0001
Noun	0	0.2	0.002	0.3
Adj	0	0.2	0	0.001
Verb	0	0.3	0	0.1

Tag transition probability $P(t_i / t_{i-1})$

	D	N	A	V
The	0.3	0.1	0	0
light	0	0.03	0.2	0.06
book	0	0.03	0	0.001

Word Prob (w_i / t_i)

Given the phrase the light book. use HMM with Witterbi algorithm to find most suitable tag sequence for the given phrase. The transition and the word emission probability and emission probabilities learnt from the training are given in tables. Assume that the probability of any tag starting the sentence is same



Level 1

$S_1(1) = 0.25 \times 0.3 = 0.075$

$S_2(1) = 0.25 \times 0.1 = 0.025$

$S_{12} = \begin{cases} S_{11} \times P(N|D) \times P(\text{light}|N) \\ = 0.075 \times 0.5 \times 0.003 = 0.1125 \times 10^{-3} \\ S_{21} \times P(N|N) \times P(\text{light}|N) \\ = 0.025 \times 0.2 \times 0.003 = 1.5 \times 10^{-5} \end{cases}$

$S_{22} = \begin{cases} S_{11} \times P(A|D) \times P(\text{light}|A) \\ = 0.075 \times 0.3 \times 0.2 = 4.5 \times 10^{-3} \\ S_{21} \times P(A|N) \times P(\text{light}|A) \\ = 0.025 \times 0.002 \times 0.2 = 1 \times 10^{-5} \end{cases}$

$S_{23} \begin{cases} S_{11} \times P(V|D) \times P(\text{light}|V) \\ = 0.075 \times 0.0001 \times 0.06 = 4.5 \times 10^{-8} \\ S_{21} \times P(V|N) \times P(\text{light}|V) \\ 0.025 \times 0.3 \times 0.06 = 4.5 \times 10^{-4} \end{cases}$

$S_{31} = \{ S_{12} \times P(N|N) \times P(\text{book}|N)$

$= 0.1125 \times 10^{-3} \times 0.2 \times 0.03 = 6.75 \times 10^{-7}$

$S_{22} \times P(N|A) \times P(\text{book}|N)$

$= 4.5 \times 10^{-3} \times 0.2 \times 0.03 = 2.7 \times 10^{-5}$

$S_{23} \times P(N|V) \times P(\text{book}|N)$

$= 4.5 \times 10^{-4} \times 0.3 \times 0.03 = 4.05 \times 10^{-6}$

$S_{32} = S_{12} \times P(V|N) \times P(\text{book}|V)$

$= 0.1125 \times 10^{-3} \times 0.3 \times 0.001 = 3.375 \times 10^{-8}$

$S_{22} \times P(V|A) \times P(\text{book}|V)$

$4.5 \times 10^{-3} \times 0.001 \times 0.001 = 4.5 \times 10^{-9}$

$S_{23} \times P(V|V) \times P(\text{book}|V)$

$= 4.5 \times 10^{-4} \times 0.1 \times 0.001 = 4.5 \times 10^{-8}$

stemming

morpheme

affixes → suffix | prefix | infix | circumfix
- suffix ↓ add at end to change meaning
- prefix ↓ at beginning
- infix ↓ in between
- circumfix ↓ add both sides

concative morphology (possible to break & get root)

non-concative morphology (not possible to break & get root)

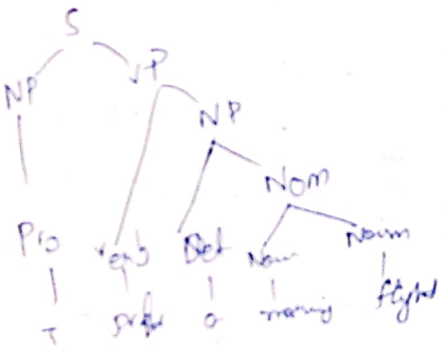
Σ :- better

uses: context based spellcheck, information extraction

inflectional morphology

derivational morphology

Context Free Language for English



Bracket Notation of parse tree

[S [NP [Pro I]] [Vp [Verb Prefer] [NP [Det a] [Nom [Noun morning] [Noun flight]]]]]

Chomsky Normal form

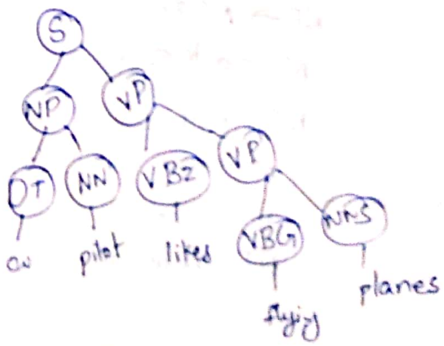
Either exactly two non-terminals on RHS

Or 1 terminal symbol on RHS

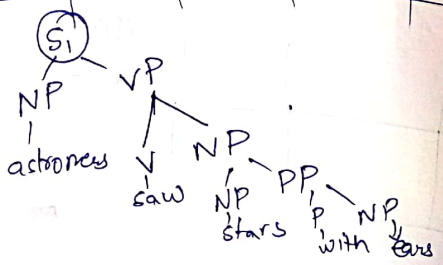
CKY Algorithm

0 a 1	pilot 2	likes 3	flying 4	planes 5
DT 01	NP 02	– 03	– 04	S, S 05
	NN 12	– 13	– 14	– 15
		VBZ 23	– 24	VP_1, VP_2 25
			VBG, JJ 34	VP, NP 35
				NNS 45

S → NP VP
VP → VBG NNS
VP → VBZ NP
VP → VBZ NP
NP → DT NN
NP → JJ NNS
DT → a
NN → pilot
VBZ → likes
VBG → flying
JJ → flying
NNS → planes



astronomers	saw	stars	with	ears
NP	–	S	–	S_1, S_2
	V, NP	VP	–	VP, VP
		N.P	–	NP
			P	PP
				NP



S → NP VP
VP → V NP
VP → VP PP
PP → P NP
P → with
V → saw

NP → NP PP
NP → astronomy
NP → ears
NP → saw
NP → stars
NP → telescope

Homonomy < Homophony
Homograph

Polysemy

Synonymy

Antonyms

Hyponymy

Meronomy