27/Aug

Project Gutenberg

1830's → slavery → citizen → spike

Bhashini project → under IND Govt.

→ corpus → english & corpura - Latin.

30/Aug:- Sept 28 → midterm presentation

Midterm Quiz → Reasoning Questions.

No coding

Conceptual Technological Questions

(ML part)

(Open Laptop/Notes)

(Tut before midterm)

(duration → 1+ hr)

(handwritten)

Papers:- (4, 5, 6, 7) in document

(set of slides) - AI snake oil

(fraud)

claim itself is fraud

dataset based on fradulent

NLTK - chapter 3:- urllib, utf-8

decoder

for html → parser ⇒ Beautiful soup.

* Reading local files → 'w' (to write) → f.write(" ")

(override)

→ 'a' → append

f.open

f.close

* NLP Pipeline.

→ fxn:- wordpunct_tokenize } takes care of punctuations also.

* Strings:-

* regex:- import re

↳ nltk.corpus.words.words } list of words

(ending → ing$)
(begin → ^win)

[ ] → one of set match
a+ } → ≥1 of a
a* } → ≥0 of a → Kleene closure.

Normalizing text → everything lower case

Stemming → finding smallest possible word

Lemmatization → process of stemming.

regex to remove various suffixes of a word.

Segmentation ✓ Tokenisation / word segmentation.
(Ex:- chinese → no spaces)

Ch4:-

Generator Expr → X

→ Questions of style:- → procedural → lot of functions
→ declarative → lot of variables & iterations.

→ variable ~~scope~~ & parameter type.
(X) → checking not allowed.

Program dev:

↳ Error management → generally no coding discipline like C++ / Java
↓
gdb

Algorithmic design:-
↳(less resources / time)
↓ (grandious) ↓ faster → small
(mergesort vs bubblesort)

Ch-5:- → 1st task of ML in lang:- Pos Tagger
classifier learning task.

→ (position, context) of 1 word ⟺ (") other words

words = nltk.word_tokenize("sentence")
nltk.pos_tag(words).

~~[illegible]~~ context

Tagged corpora:-

Pos Tagger → (Default Tagger → "Noun")
(Regex Tagger → ("ing" → gerunds .. etc)
(lookup Tagger → ↑'s model size } → (45% → 94%) accuracy.

↑'s ?

| (a followed by b))

bigram → a followed by b

unigram tagger ≅ lookup tagger.

→ ("Using NLTK make a Pos Tagger") → ✓