08 January 2024 19:13

NLP -> Natural language Processing

Sub-sections

NLU
Natural language understanding

NLG
Natural language generation
- New text generate

Basic terms of any language:

① Phonemes -> Smallest unit of any language
characters, Speech, sound

② Morphemes & lexemes
↓ words
↓ Run, Running
Swim, swimming

③ Syntax -> phrases, Sentences

④ Context -> meaning

-> Combination of syntaxes

NLP applications:

1) Sentiment analysis -> Text classification

Tweets -> +ve
-ve
Neutral

Movie Reviews -> +ve
-ve
Neutral

Text	label
Tweets, Movie Reviews	+ve -ve Neutral

Product / services -> % customer satisfied

Data collection
- Survey
- google forms
- audio files

2) Document classification

Text	label
Document	Document name

Document content	Document name
	- Aadhar - PAN - DL - VID

③ Text summarization

Text summarization

extractive T.S. | Abstractive T.S

extracting only important lines/ sentences from the data

- creating new summary (NLG)

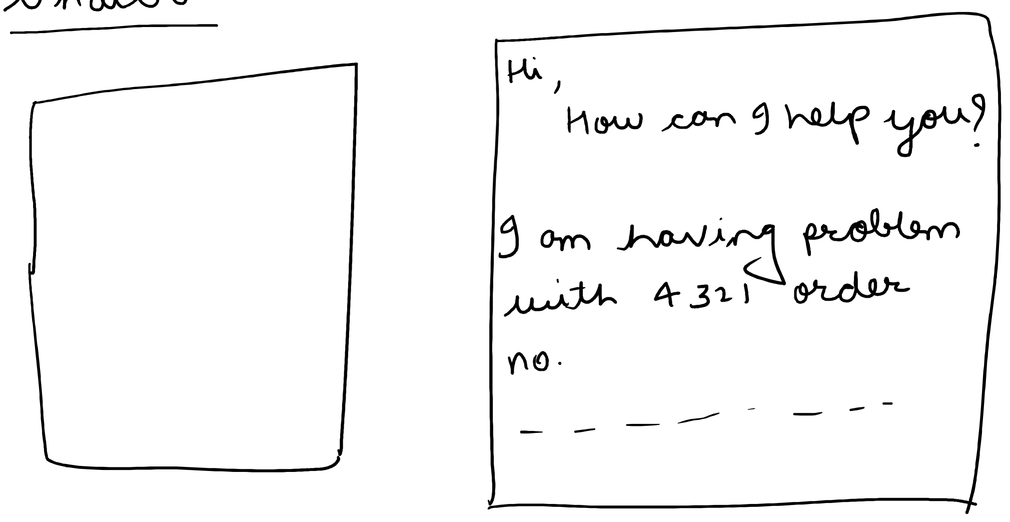
④ Topic modelling / identification

100 documents
- → 20 → Sports → ground, player, match, etc.
- → 30 → geography
- → 50 → ethics

- Hidden pattern between text

100 documents
- → 20 → 0 →
- → 30 → 1 →
- → 50 → 2 →

⑤ Chatbot

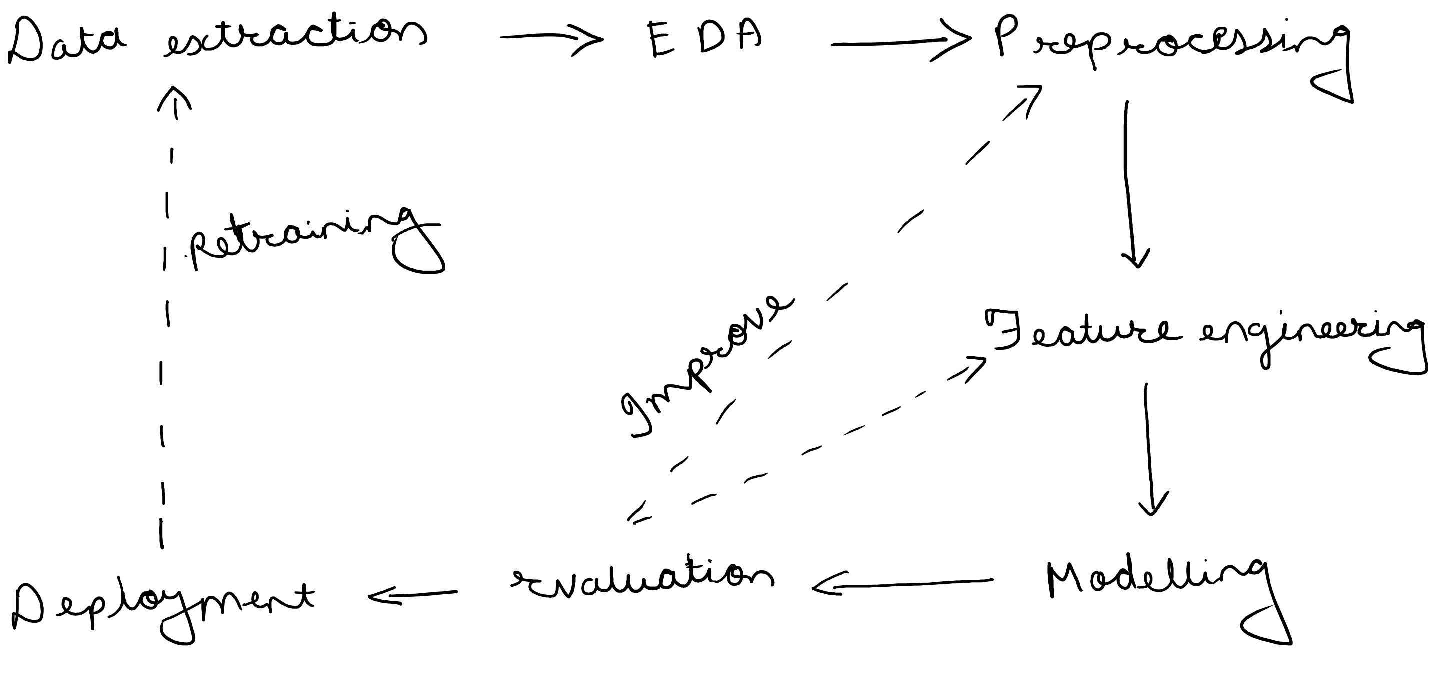


Hi,
How can I help you?

I am having problem with 4321 order no.

- automatic response generate

NLP Pipeline



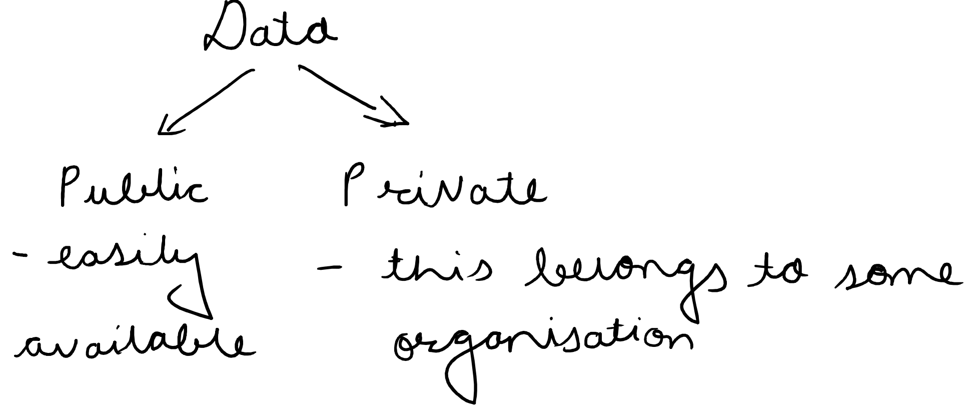
① Data extraction -> Data formats -> json
txt
csv
images -> OCR
↓
Pytesseract

↓

Pytesseract

Amazon textract

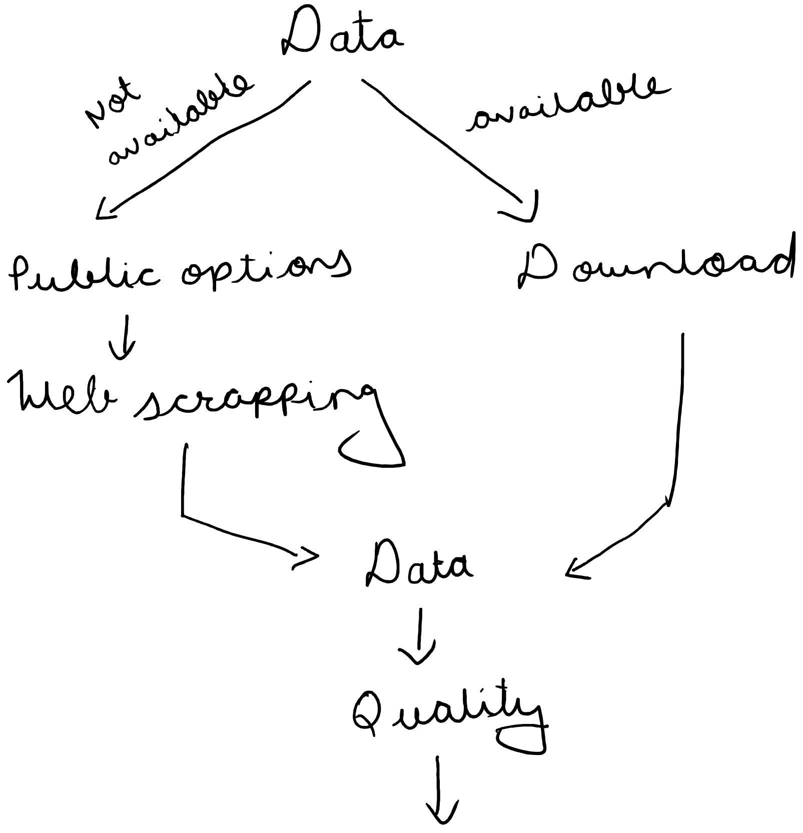
Google vision



Data

Public
- easily available

Private
- this belongs to some organisation



Data

Not available

available

Public options

Download

Web scrapping

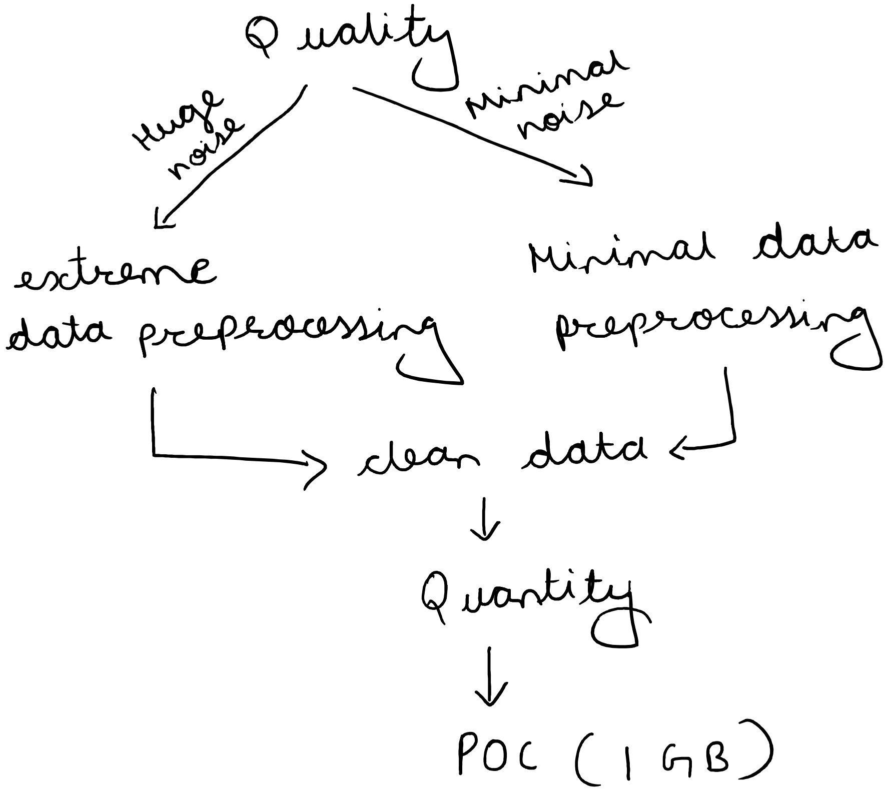
Data

Quality

What is noise ?

" We >>>> loved $@ /\ # the product ______

" ________ " will definitely ---- recommend."



Problems regarding data:

① Quantity

② Quality

③ exact data is not available for our use case.

④ Do not have continuous flow of data.

Data cycles → Monthly → 1✓ 2✗ 3✓ 4✗

Quarterly +

Yearly

EDA : NLP → ① Ngram
② Word cloud
③ Keyphrase extraction

Ngram → ① Unigram ② Bigram ③ Trigram
④ Quadragram

" Rajesh is hardworking guy."

Unigram = [Rajesh, is, hardworking, guy]

Bigram = [Rajesh is, is hardworking,
hardworking guy]

Trigram = [Rajesh is hardworking,
is hardworking guy]

Why Ngrams ?

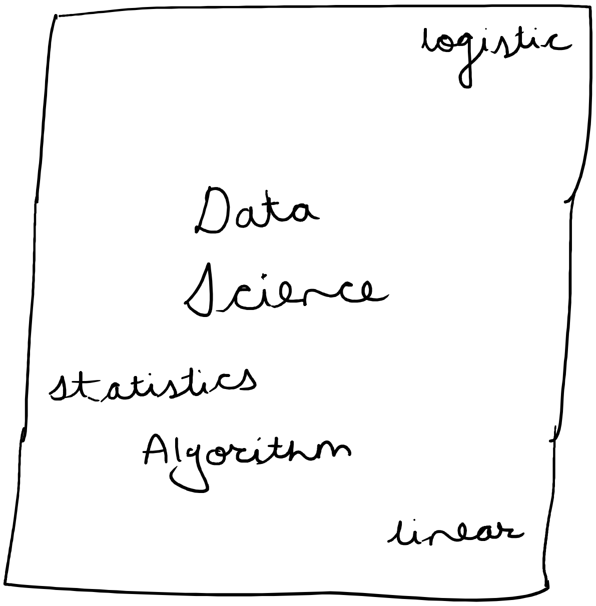
① To get insights from the data
Words understanding

Positive reviews → +ve words

Negative reviews → -ve words

② To get domain specific stopwords

② **Word cloud**



Word frequency ↑
Word font ↑

Word frequency ↓
Word font ↓

③ Keyphrase / Keyword extraction

To extract → Important Keyphrase / Keyword from text

→ RAKE, YAKE

We are learning NLP.

⇓

Preprocessing : →

① Tokenization ⟶ ① Sentence tokenization

① Tokenization ⟶ ① Sentence tokenization
② Word tokenization

"We are learning NLP.
NLP is a huge domain."

Sent = [We are learing NLP., NLP is a huge domain.]

Tokens = [We, are, learning, NLP, ., NLP, is, a, huge, domain, .]

Sentence tokenization ─┬→ Syntax
├→ Punctuation → . , ? , !
└→ Conjunction → and, but

② Normalization → GREAT ⟶ Num1
⇓ great ⟶ Num 2

Single case ⟶ lower case
upper case

③ Remove punctuation / Symbols

from string import punctuation

. , ; [''] ? $ @ , - - - - -

④ Remove stopwords

↙ ↘

language specific   Domain specific

"Rajesh is suffering from cancer. Right now Doctor Pravin is treating him. We have given him xyz tablet."

Health domain → Doctor, tablet, capsule, treatment, etc.

⑤ Lemmatization & stemming ⇒

Lemmatization	stemming
↓	↓
lemma	Stem → Greek words
↓	↓
Meaningful Root word	Root word
↓	↓
word net dictionary	aggressive prunner
↓	↓
Running	Running → Run
↓	↓
Run	Swimming → Swim
	↓

X

Run ↓

Believe → Beli ×

(6) Contraction mapping → expanding text

didn't → did not
doesn't → does not
haven't → have not

I didn't like the movie → like movie

I liked the movie → like movie

I did not like the movie → not like movie

I liked the movie → like movie

Stopword_list = [I, me, haven't, didn't,
no, nor, not, - - -]
× × ×

Stopword list . remove ("no")
("nor")
("not")

(7) Handling accented characters → unidecode library

a → $\hat{a}$, $\bar{a}$, $\grave{a}$, - - - -

b → $\hat{b}$, $\bar{b}$, $\grave{b}$

a → a, a, ȧ, - - - -

b → b̂, b̄, ḃ, - - - - -

âble → ble

a ble

⑧ Autocorrection → correct spellings of words

→ autocorrect library

→ text blob library

Feature engineering

text → Numerical format / vectors

text → Word embedding

Word embedding

- Frequency Based → word frequency
  - ① Count vectorizer ✓
  - ② T F I D F ✓
- Prediction based → Algorithm
  - ① Word 2 Vec ✓
  - ② Fast text
  - ③ Doc 2 Vec

Modelling → Data → Numerical format

⇓

Build model

logistic regression , SVM, Rondom forest,
Adaboost, Naive Bayes, Decision tree class,
RNN, LSTM

<u>Evaluation</u> : Accuracy
Precision
Recall

Low accuracy —> Preprocessing
↓
Feature engineering
↓
Frequency based
↓
prediction based

Good accuracy —> Deployment