* Topic Modelling

$K \rightarrow$ # of Topics
$N \rightarrow$ # of words
$\theta_m \rightarrow$ Distribution over topics
$Z \rightarrow$ index $[1, \dots, K]$
$w \rightarrow$ index of words $[1 \dots |V|]$

} Paper: David Blei, Jordan & Ng 2003

* Word embedding

Dictionary $\rightarrow V$
# of words $\rightarrow |V|$

] → Glove Stanford
→ word2vec Google } two famous word embeddings

word2vec
→ CBOW (continuous bag of word) (Dealing row (w) in co-occurre table)
→ skipgram (dealing column in co-occurence table)

* Tagging

→ POS
→ NER (People, locations, organizations & misc)

CHUNKING (entity detection)
↓
NER (entity classification)

* SEQ to SEQ

Text

- Corpus/Collection
- Document
- word (like pixels/superpixels)
- Vocabulary

Image

- Training Set → S
- Total images in S → N
- For each image, x :
  → let "T" be its total superpixels

So, $x = [\underbrace{x^1, x^2, \dots, x^T}_{T \text{ superpixels}}]$

And now:

$x^1 \in \mathbb{R}^n$ ("n" features for each superpixel)

$x \in \mathbb{R}^D$

NLP Applications

- → Document Classification (Supervised)
- → Topic Modelling (Unsupervised)
- → word embedding
- → Tagging
- → Seq2Seq

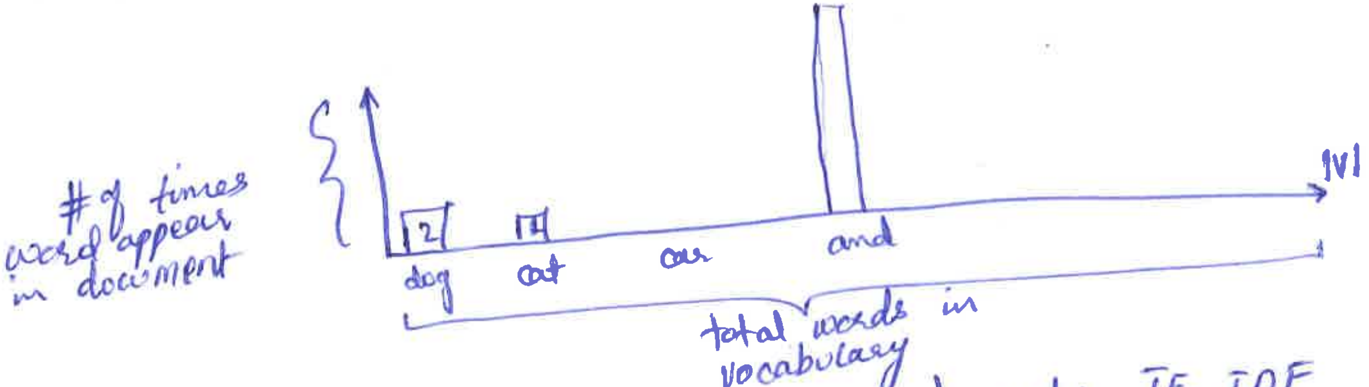
Document Classification

Assign a document its label

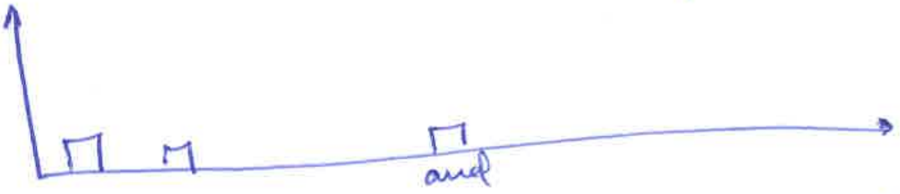
$$x \in R^D \longrightarrow y$$

where x is document

Now, how to represent document as a vector? (As histogram of words)



↓ Apply TF-IDF (term-frequency - inverse document frequency)



TF-IDF: statistical measure that tells how important a word is in the document.

Topic Modelling (Unsupervised Problem)

Example

Find 20 main topics in

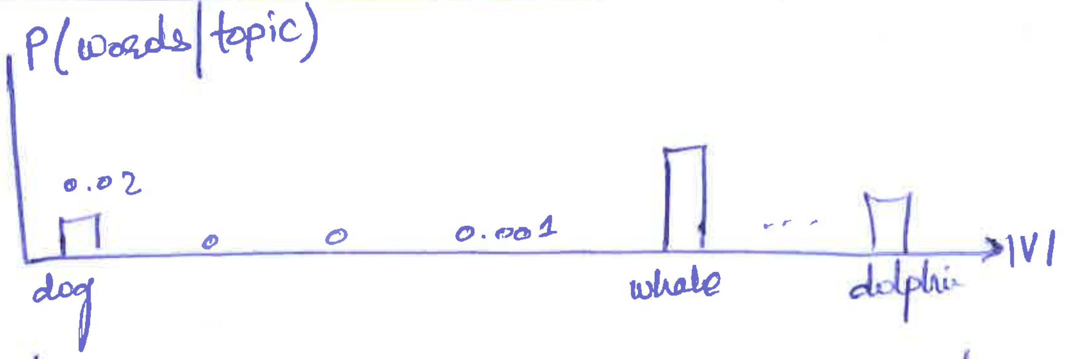
a $\underbrace{\text{corpus}}_{\text{collection OR Training Set}}$ of $\underbrace{\text{"N" documents}}_{\text{like "N" images}}$

Topic

A topic is a distribution over the words in Vocabulary "V"

$|V| \sim 50{,}000 - 200{,}000$

For 20 topics, we will have 20 such distributions



This vector can be taken for a problem of topic labelling

PAPer/Code:

1) To understand topic modelling, read Latent Dirichlet Allocation (David Blei 2003)

2) Use scikit learn (for lda)

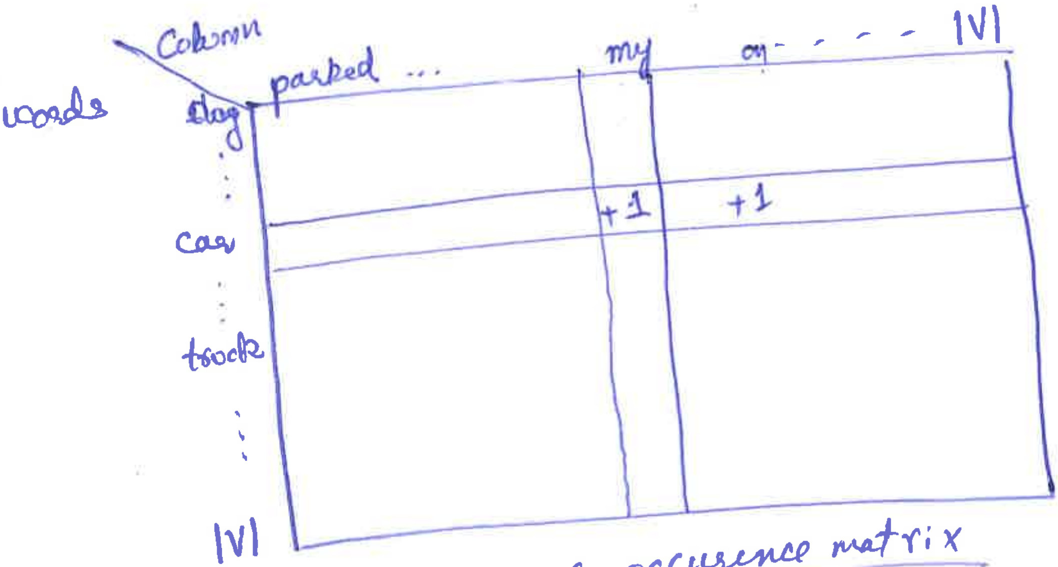
Word Embeddings

- Two famous word embeddings
  - Glove (Stanford)
  - Word2Vec (Google)

Example

Today, I parked my $\boxed{\text{car}}$ on street.

$\underbrace{\text{my car on}}_{\text{window = 3 words}}$



co-occurence matrix

We can find P(W, C), P(W|C) & P(C|W) in co-occurences matrix

* In word2vec → CBOW (continuous bag of words)
"Here we deal with rows in co-occurence matrix"

→ Skipgram
("Here we deal with column in co-occurence matrix"

TAGGING

Tag a word in a sentence

POS tagging : { Noun, adj, verb... }

→ Old way of tagging : look up tables

→ New way :

1) word embedding (~300-500D)

3) trigram → 900D [concatenation of 3-words] of wordembeddings

4) train Classifier

$\underset{X}{900D} \rightarrow y$

NAMED - Entity Recognition (NER)

{ People
location
organisation
MUC → events
→ number
→ time expression

Example

B-loc I-loc
LAS Vegas

B-LOC I org I-org I-org
Common wealth bank of Australia

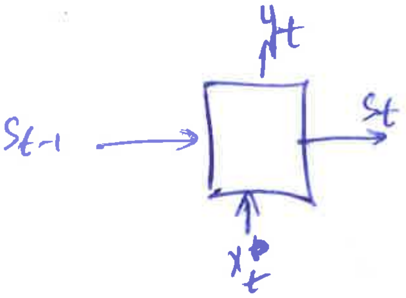
CHUNKING

Entity (detection) or segmentation

.... entity .... entity ....

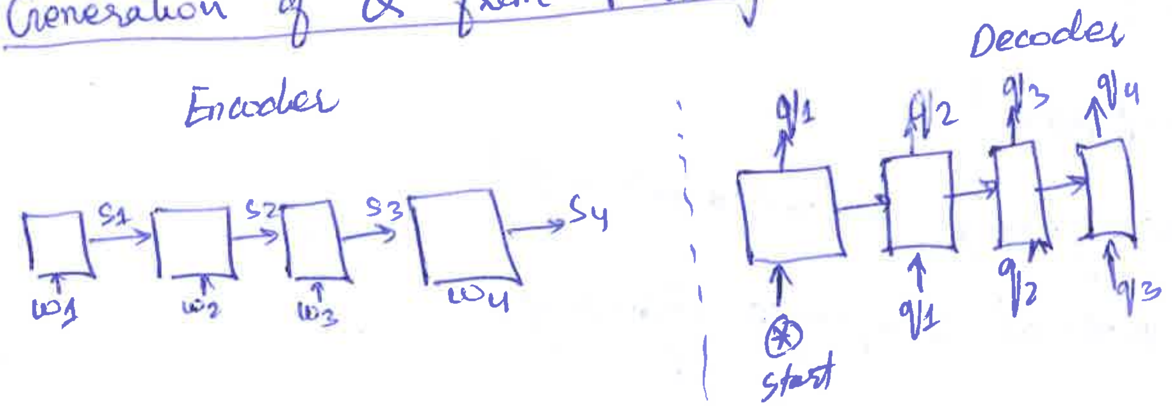
SEQ 2 SEQ → Generates text
(Natural language generator)

P: input
Q: output



Three famous RNNs { Jordan, Elman, LSTM

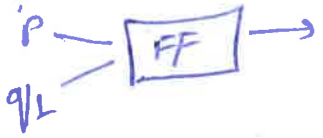
Generation of Q from P using Encoder-Decoder:



In attention Networks

$s_1 \rightarrow s_1 \times a_1$
$s_2 \rightarrow s_2 \times a_2$
$s_3 \rightarrow s_3 \times a_3$

a_1, a_2, a_3 are attention weights. They can be learnt as follows:



Paper

1) seq2seq → Sutskever 2015
2) seq2seq Attention → Badhanau

Other tasks

1) Co-reference Resolution:
Shaukat did some ... He told

2) Relation Extraction:
Father-son, mother-son ...

3) Taxonomy extraction
corpus → graph

4) Machine translation
French → English → English improved

5) Summarization

6) Matching (BiMPM)

7) RTE (recognising textual entailment)

8) NLI (Natural language inference)