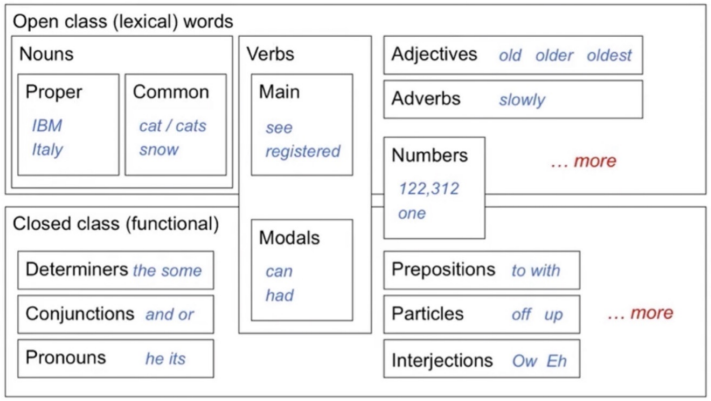
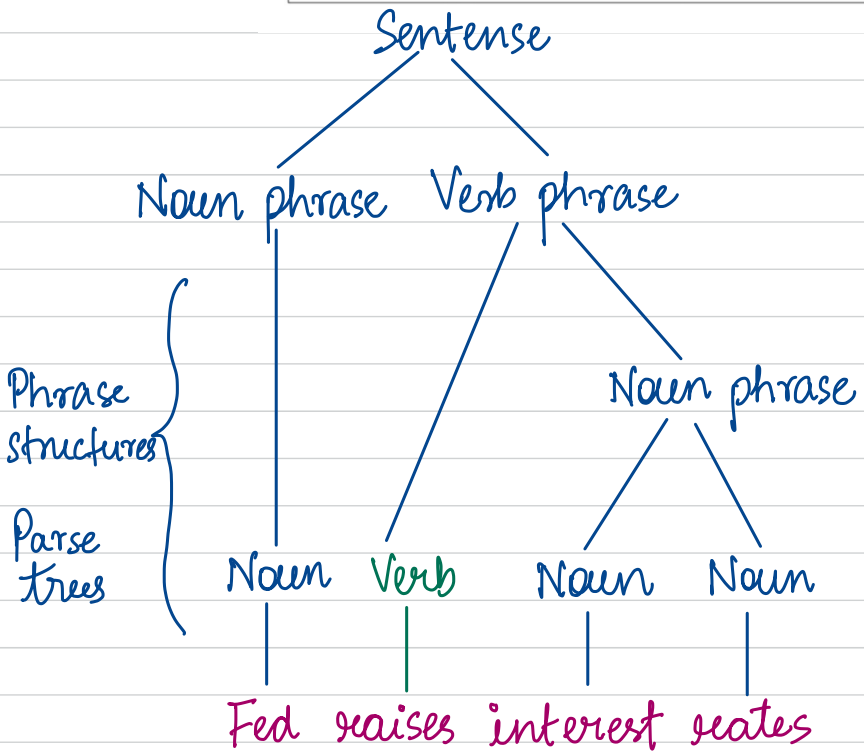
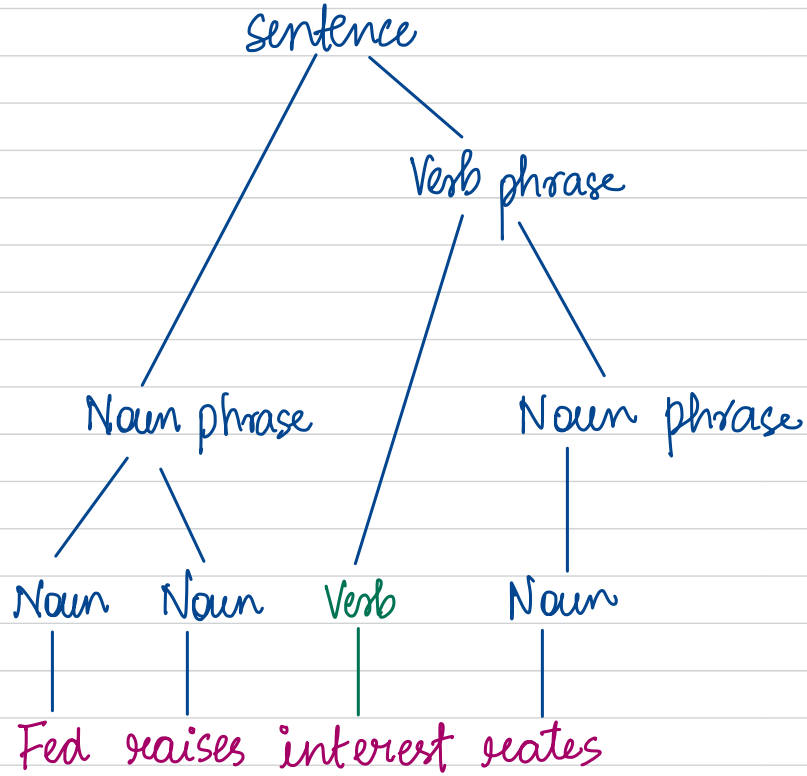
Parts of Speech Tagging

Goal: Tag each token with a part of speech.

Computational linguistic.







Named Entity Recognition

The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply. London

Person
Date
Location
Organi-
zation

How do we represent the meaning of words?

Recall one hot encoding

cat	[0 0 0 0 1]
dog	[1 0 0 0 0]
chair	[0 0 1 0 0]

$v = (v_1, v_2, \ldots, v_m)$

$w = (w_1, w_2, \ldots, w_m)$

$v \cdot w = \sum_{i=1}^{m} v_i w_i$

Similarity between two words is captured using dot product.

dot product is high

dot product is 0.

cat · dog = 0

cat · chair = 0

dog · chair = 0

cat · cat = 1

Problems with One-hot encoding for word-vectors

- long vectors
- does not capture similarity bet^{n} words.

eg. Dell notebook battery size.
Dell laptop battery capacity

Distributional Similarity

you shall know a word by the company it keeps. - J.R. Firth.

eg. melancholic

He has been in a melancholic mood since his girlfriend left him.

↓

sad, depressed, lonely

We build a dense vector for each word type so that it is good at predicting other words appearing in its neighbourhood.

Word2Vec (2013)

Two algorithms

(i) Skip gram

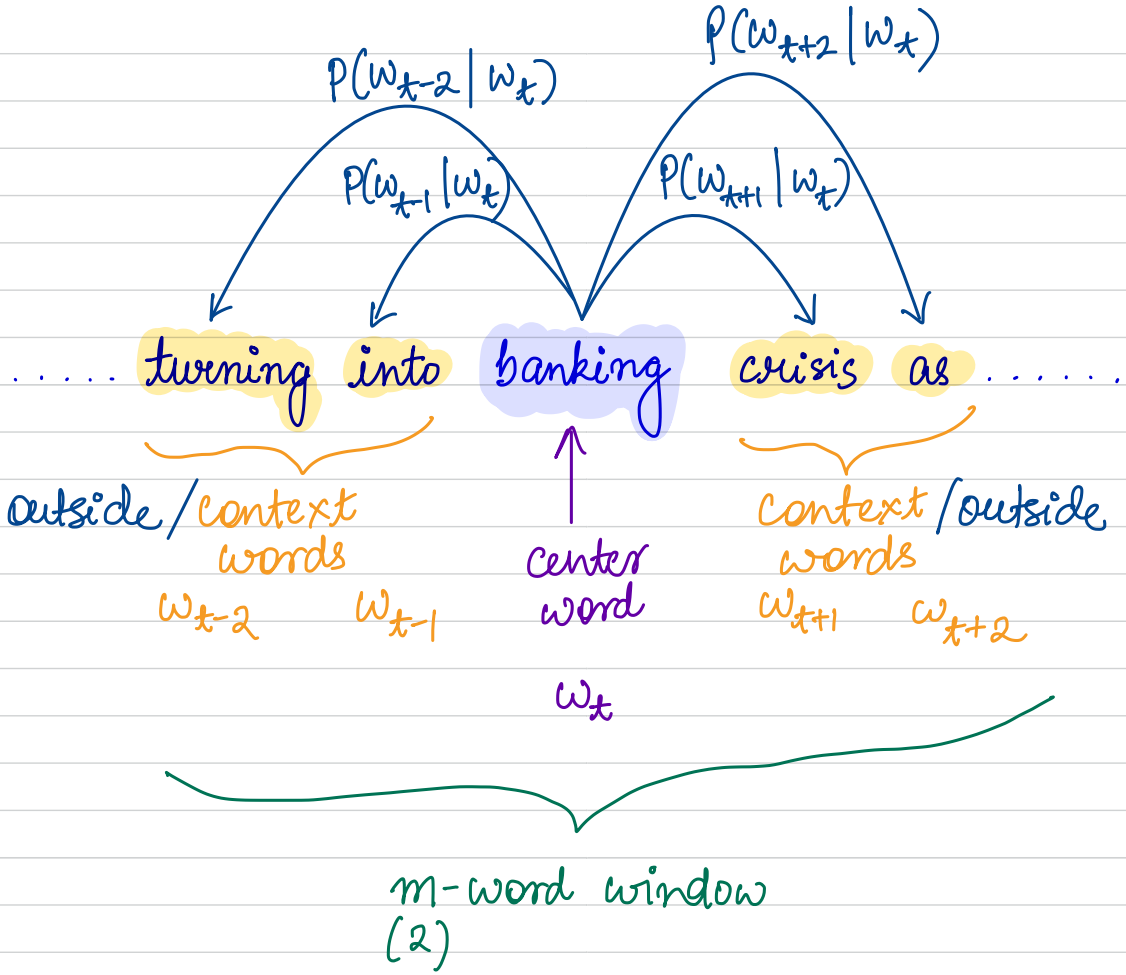
(ii) Continuous BOW

Two training methods

(i) Hierarchical Softmax

(ii) Negative Sampling

} Softmax



for each word $t = 1, 2, 3, \ldots$

Predict surrounding words in a window of radius m.

Objective: Maximize the probability of any context word given the center word.

eg. given w_t we want to maximize

$$P(w_{t-2} \mid w_t)\, P(w_{t-1} \mid w_t)\, P(w_{t+1} \mid w_t)\, P(w_{t+2} \mid w_t)$$

for window size = m,

$$\max \prod_{\substack{-m \le j \le m \\ j \neq 0}} P(w_{t+j} \mid w_t) \quad \text{for } w_t$$

We need to perform the above maximization for all words.

$$\max \mathcal{L}(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \neq 0}} P(w_{t+j} \mid w_t; \theta)$$

T → # words

θ → parameters of the model.

$\mathcal{L}(\theta)$ → likelihood function

negative log likelihood

$$\min J(\theta) = -\sum_{t=1}^{T} \sum_{-m \le j \le m,\ j \neq 0} \log\left(P(w_{t+j} \mid w_t; \theta)\right)$$

To parameterize P we use softmax.

$$P(o \mid c) = \frac{\exp(u_o^T u_c)}{\sum_{i=1}^{T} \exp(u_i^T u_c)}$$

o: outside word, c: center word

if u_o and u_c appear together, $u_o^T u_c$ is high and therefore prob. is also high!

vector representation for outside word : u_o.

" " " center " : u_c.

Softmax turns no.s into probabilities.

$$\theta = \begin{bmatrix} u_{abacus} \\ \vdots \\ u_{zebra} \end{bmatrix}$$

use GD

$$\theta_{i+1} \leftarrow \theta_i - \alpha \nabla J(\theta)\Big|_{\theta_i}$$

Continuous BOW

Predict the center word from sum of surrounding words.

Count Based Methods

Create Co-Occurance matrix

Example Dataset :

(1) I like deep learning .

(2) I like NLP .

(3) I enjoy flying .

P =

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Problem: Very high dimensional

How can we reduce dimensionality?

- PCA (Square matrices) or SVD (rectangle matrices) (singular value decomposition)

throw away dimensions corresponding to small singular values.

Glove : Global Vectors for Word Representation (2013)

$$\min J(\Theta) = \frac{1}{2} \sum_{i,j=1}^{T} f(P_{ij}) \left(u_i^T u_j - \log P_{ij} \right)^2$$

$(i,j)^{th}$ entry of the co-occurence matrix

$f(P_{ij})$: weight
co-occurences which are frequent with low values.

$u_i^T u_j$: high / low

$\log P_{ij}$: high / low

9:18 AM