

<800x2479 sparse matrix of type '<class 'numpy.int64'>'
with 52937 stored elements in Compressed Sparse Row format>

800 × 2479

| | T_1 | F_2 | T_3 | F_4 | T_5 |
|----|-------|-------|-------|-------|-------|
| D1 | 1 | 0 | 0 | 1 | 0 |
| D2 | 0 | 0 | 0 | 1 | 0 |
| D3 | 0 | 0 | 0 | 1 | 0 |
| D4 | 0 | 0 | 1 | 1 | 0 |

$$\begin{matrix} (0,0) \rightarrow 1 & \dots & (3,4) \rightarrow 5 \\ (0,2) = 1 & \vdots \end{matrix}$$

<40000x25748 sparse matrix of type '<class 'numpy.int64'>'
with 3230445 stored elements in Compressed Sparse Row format>

$$F1 \text{ score} = \frac{2 \times \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{precision} = \frac{TP}{TP+FP}$$

$$\text{recall} = \frac{TP}{TP+FN}$$

| Actual | Predicted | Prediction Type | |
|----------|-----------|-----------------|-----------|
| | | negative | positive |
| positive | negative | (FN) | |
| positive | positive | (TP) | |
| negative | positive | FP | |
| | | | → Type II |
| | | | → Type I. |

Confusion matrix

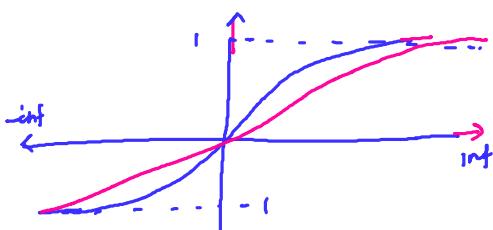
- positive ⇒ has cond ✓
- negative = does not cond. ✓

Sentiment Analysis - VADER

Valence Aware Dictionary & sEntiment Reasoner
 "I hate tea" → "negative" & 100%

→ ~~X~~ hate tea
 Neg Neu
 -2.7 0
 $\Rightarrow -2.7 \Rightarrow \text{Total score}$

→ ~~X~~ hate tea the taste in bad
 -2.7 0 0 0 0 -2.5
 $\Rightarrow -5.2 \Rightarrow \text{Total score}$



Compound Score = $\frac{\text{Total score}}{\sqrt{\text{Total score}^2 + \alpha}}$

$\downarrow 15$

Total score = -2.7

Compound Score = $\frac{-2.7}{\sqrt{(-2.7)^2 + 15}} =$

~~X~~ hate tea and ~~X~~ love coffee
 (-2.7) 0 0 3.2 0

Total score = 0.5

~~X~~ hate tea

| | | | |
|---------------|---------|---------------|--------------------------------------|
| Neg | Neu | \Rightarrow | Total score $\Rightarrow 4.7$ |
| -2.7 | 0 | | |
| abs[-2.7] + 1 | (0) + 1 | | $y.$ of pos score $= 0/4.7 = 0\%$ |
| 3.7 | 1 | | $y.$ of neg score $= 3.7/4.7 = 0.78$ |
| | | | $y.$ of neu score $= 1/4.7 = 0.22$ |

| | <u>heat</u> | <u>tea</u> | <u>and</u> | | <u>love</u> | <u>coffee</u> |
|----------------|-------------|------------|------------|--|-------------|---------------|
| Original Score | -2.7 | 0 | 0 | | 3.2 | 0 |
| Avg Score | 2.7 + 1 | 0 + 1 | 0 + 1 | | 3.2 + 1 | 0 + 1 |
| Total Score | 3.7 | 1 | 1 | | 4.2 | 1 |

$$\% \text{ of pos score} = \frac{4.2}{10.9} = 0.385$$

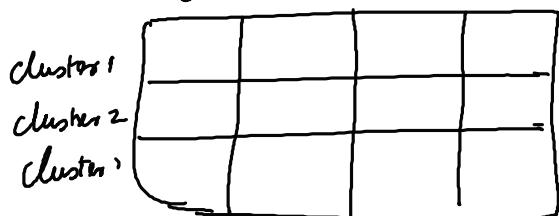
$$\% \text{ of neg score} = \frac{3.7}{10.9} = 0.339$$

$$\% \text{ of neu score} = \frac{3}{10.9} = 0.275$$

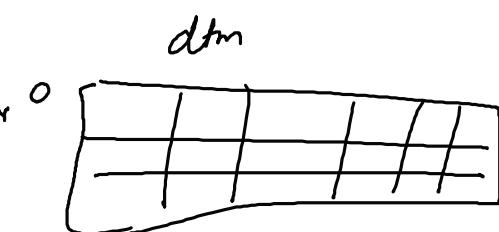
Document clustering

↓ Unsupervised

Clustering algorithms
(K-Means)
Average distance



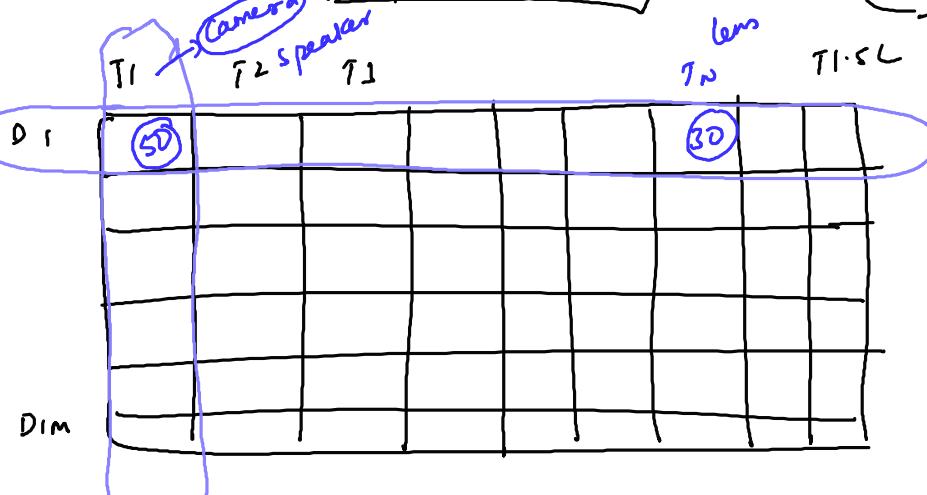
Topic modelling
(LSA, LDA)



df-dtm

cluster 2
camera

T₂ speaker



1M x 1SL

Document vector dimension: No. of columns in DTM \approx Vocab size
Word vector dimension: No. of rows in DTM \approx No. of document

Word2Vec

Shallow neural network - Relationship b/w words

The credit card is not having enough benefit
Credit card penalty is very high

I missed my credit card bill this month. Please help.

Doc1
Doc2
:
Doc1000

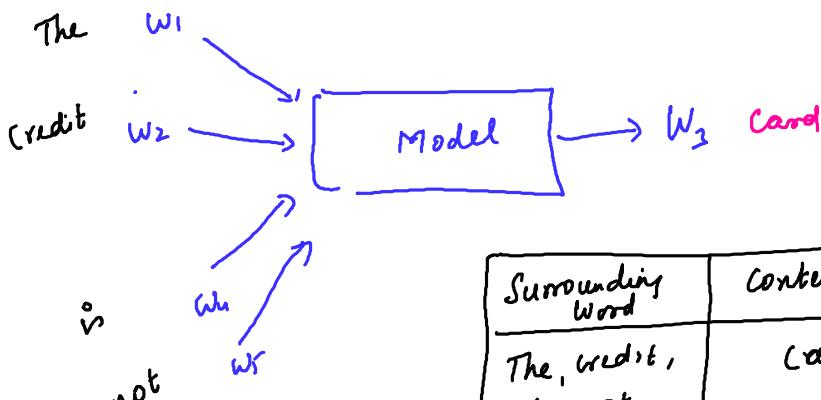
Bigram

| Input Word | Content Word |
|------------|--------------|
| The | credit |
| credit | card |
| card | is |
| is | not |
| not | having |



| 2 input word | Content word |
|--------------|--------------|
| The, credit | card |
| credit, card | is |
| card, is | not |
| is, not | having |
| not, having | penalty |

The credit card is not having enough benefit



| Surrounding word | Content word |
|------------------------------|--------------|
| The, credit, is, not | card |
| credit, card, not, having | is |

Word2Vec
Sparse, high dimension
less content vector representation

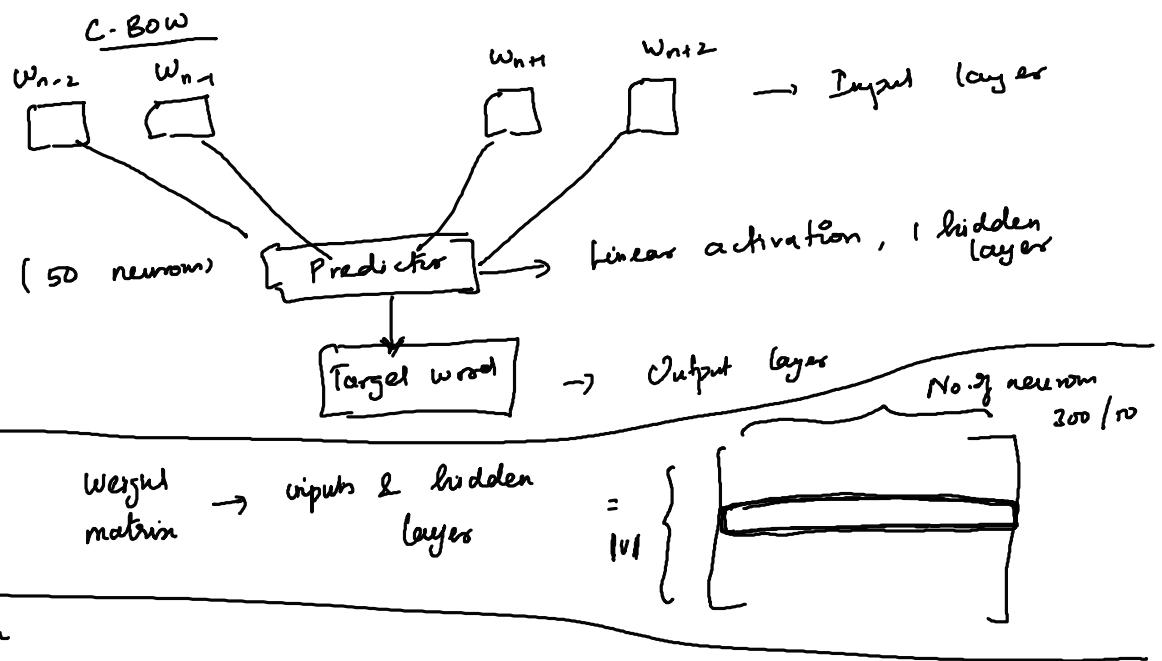
Word2Vec dense, low dimension,
content preserved vector
representation of words

DTM

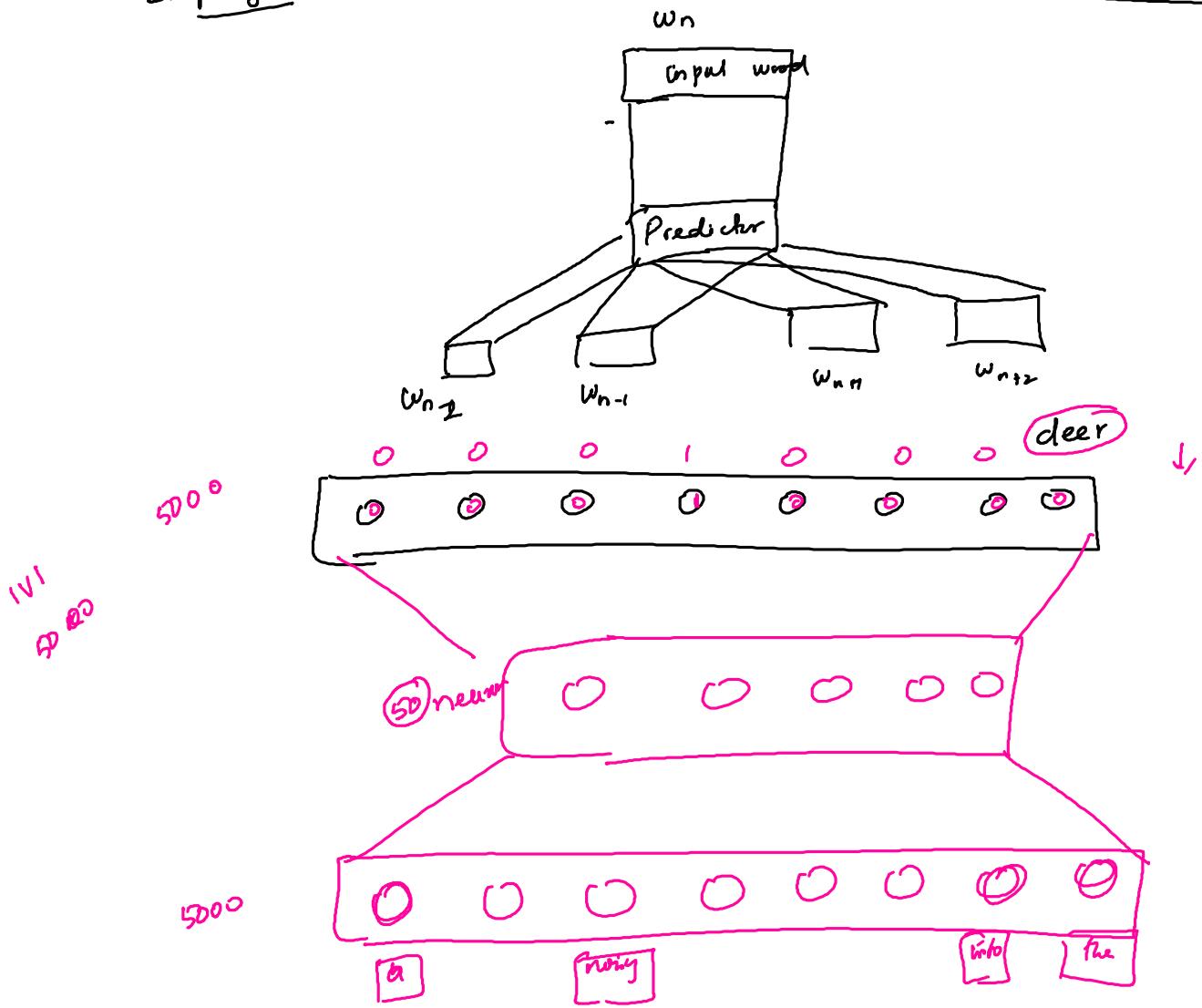
- Sparse
- High dimension
- Less content is used
- Frequency based
- Varies corpus to corpus

Word Embeddings (word2vec)

- Dense
- Customized dimension - Low dimension
- More content is used
- Based on shallow neural network
- Trained
- Pre-trained word embeddings
or
Customized word embedding is also possible



Skip-gram



$$w_1 \Rightarrow [v_{1,1} \dots v_{1,d_1} \dots v_{1,d_2} \dots v_{1,d_3} \dots \dots v_{1,d_D}]$$

vocab = 1000 words

$$= \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \vdots \\ w_{1000} \end{bmatrix}$$

$1000 \times 5D$

[[- - - - -],
[- - - - - -],
[- - - -]]