		musk	that	price	market	investor	iphone	itunes	gigafactory
Apple {	article1	0	0.005	0.01	0.05	0.05	0.7	0.8	0
	article2	0	0.002	0.008	0.01	0.002	0.7	0.8	0.
	article3	0.77	0.05	0.01	0.05	0.05	0	0	0
Tesla {	article4	0.75	0	0	0	0	0	0	0.87

TF-IDF Representations (Vectorizers)

Limitations

1. As n increased, dimensionality sparsity increases
2. Doesn't capture the relationship between words
3. Doesn't address out of vocabulary (OOV) problem

TF-IDF & Bag of words (BOW) limitations

Vector size can be huge (Too many compute resources, memory)
sparse → most values are 0
Do not capture the meaning of words. (Except n-grams)

Word embeddings

⇒ Similar words have similar vectors
⇒ Dimensions are low

good $\begin{bmatrix} 3. \\ 4. \\ \vdots \end{bmatrix}$ great $\begin{bmatrix} 3.1 \\ 4.3 \\ \vdots \end{bmatrix}$

Dense representation
↳ not too many zeros

Word embedding techniques

CBOW / Skip gram	Based on Transformer Architecture	Based on LSTM
Word2vec	BERT	ELMO
Glove	GPT	
fastText		

Word2Vec

King - man + woman = Queen

Glove
- glove-wiki-gigaword-100
- glove-twitter-100

Word2Vec
- word2vec google-news-300
- word2vec nuscorpora-300

BERT
- BioBert
- FinBert

BERT
- ALBERT
- RoBERTa

Word embedding

good → $\begin{bmatrix} 3.1 \\ 4.4 \\ \vdots \end{bmatrix}$

Sentence embedding

I am good → $\begin{bmatrix} 3 \\ 2 \\ \vdots \end{bmatrix}$

Doc embedding

Age ... → $\begin{bmatrix} 0.5 \\ 2.2 \\ 5.4 \\ \vdots \end{bmatrix}$

Gensim → Another Word Embedding Library.

Sentence embedding using Gensim Library

(Reduces Overfitting)

not

(can handle out of vocabulary problems)

Word2Vec

King - man + Woman = Queen

$\begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} - \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} + \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} = \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix}$

King

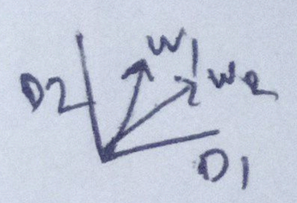
authority = 1

has tail = 0

rich = 1

gender = -1

[1, 0, 1, -1]



	battle	horse	king	man	queen	women
authority	0	0.01	1	0.2	1	0.2
event	1	0	0	0	0	0
rich	0	0.1	0	0	0	0
has tail	0	1	-1	-1	1	1
gender	0	1	+1	0.3	1	0.2

Embeddings are not hand crafted, They are learnt during Neural Network training

Huge Dataset → Matrix with unique words → Split into 2 matrices (Dimension)

going = (x_{1going}, x_{2going}, ... $x_{i going}$)

Steps

Take a fake problem

Solve using neural network

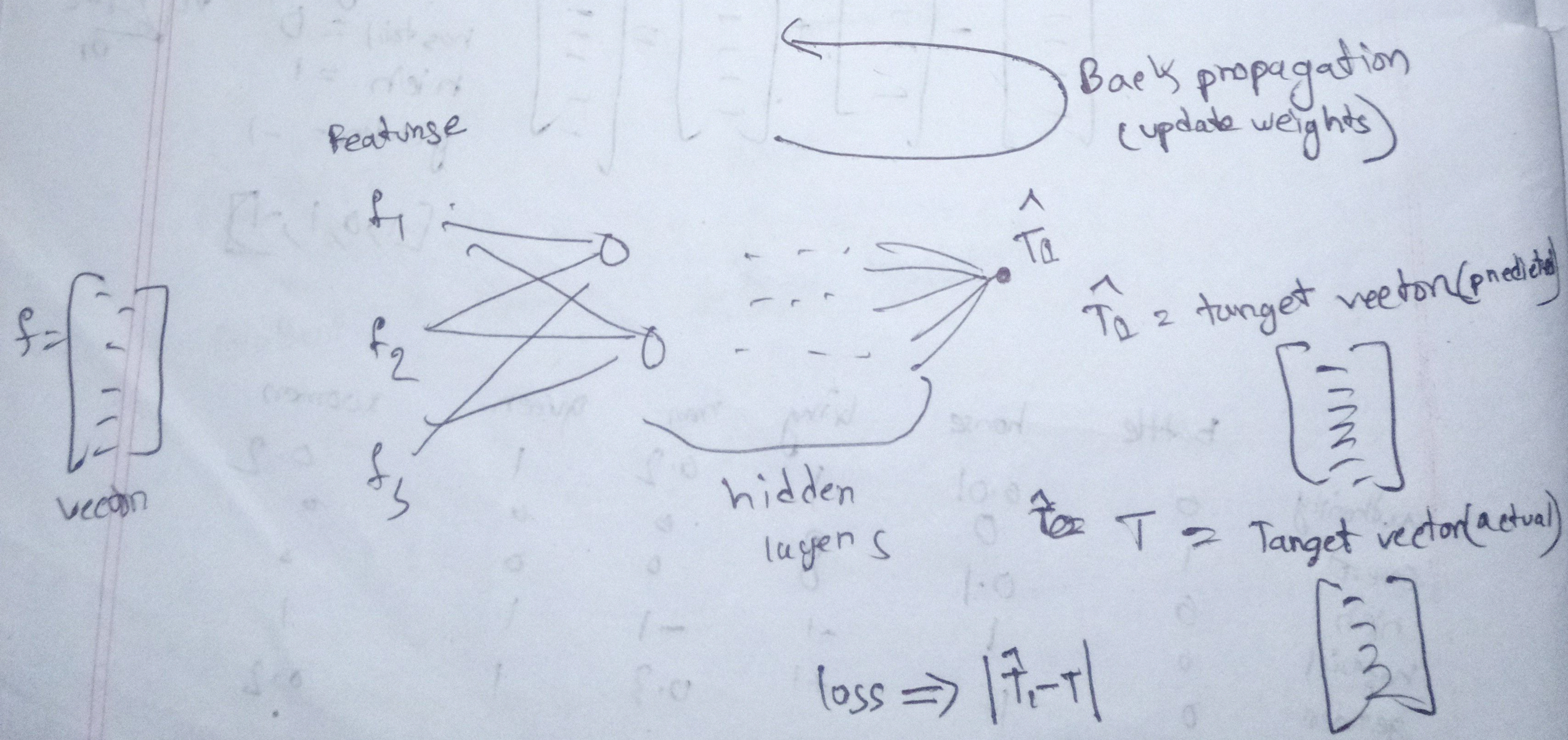
Get Word embedding as a side effect

Fake problem: fill in a missing word in a sentence

Example:

A King ~~onde~~ lived

Features/vectors — Target variable ⇒ Self-~~sup~~ supervised



CBOW → Continuous Bag of Words
Given context words predict target word

$\underbrace{\text{King}}_{\text{Target}}$ $\underbrace{\text{ordered his}}_{\text{context}}$

Skip Gram → Given the target predict context words (opposite of CBOW)

FastText → can handle OOV problem | lightweight, fast, requires less resources, good for solid baseline.
↳ Similar to Word2Vec

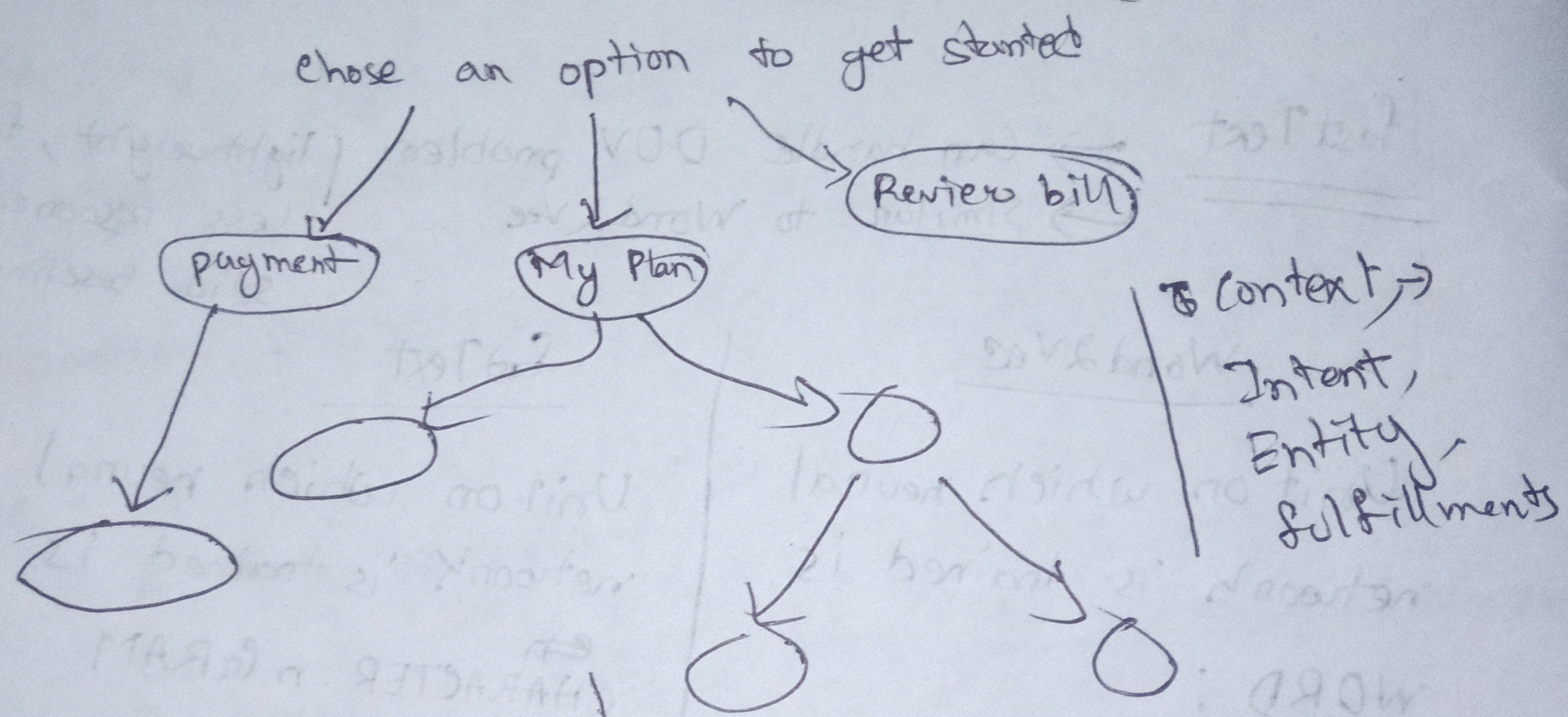
Word2Vec	FastText
Unit on which neural network is trained is WORD : ▭ ▭ capable	Unit on which neural network is trained is CHARACTER n GRAM capable n=3 cap apa pab able ble
can not handle Out of vocabulary problem	can handle Out of vocabulary problem.

FastText
↳ It is the first choice when you want to train custom word embeddings for your domain.

fastText is also a Library

Introduction to Chatbots → One of the Biggest Application (NLP)

Flow based chatbots - Using Rules (selecting options)



Process:

Dialogflow ↔ python ↔ mysql ~~Diag~~
fastAPI

Open Ended chatbots

↳ Chat gpt ....

Fixed rule-based chatbots
↓
Pure programming,
No ML
[Entity, Intent, context and others]

NL/NLP chatbots
↓ Using a chatbot framework
Dialogflow, RASA, Amazon Lex,
↕
Python, fastapi
↕
mysql

↓ Custom Implementation -
OpenAI API, GPT3/4, LLMs, BERT, Hugging face Bloom, Foundational models

Dialogflow
⇕
Python fastAPI
⇕
mysql

Benefits of Using Chatbot in Any Business
Scalability, 24 by 7 availability, Cost saving, Better customer service

Latent Semantic Analysis:

Relationship between documents and set of terms. Then they produce set of concepts [generating concepts]

Documents
Article 1 (A1)
Article 2 (A2)
Article 3 (A3)
Article 4 (A4)
Article 5 (A5)
Article 6 (A6)

Music (A1)	(A3) (A5) Food
News (A4) (A6) (A5)	Tech (A1) (A2)

probabilities	
A1-85%	A3-100% A5-73%
A4-100% A6-100% A5-27%	A2-100% A1-15%

BOW

Words / Document	going	to	today	i	am	it	is	rain	not	outside
1	1	1	1	0	0	1	1	1	0	0
2	1	0	1	1	1	0	0	0	1	1
3	1	1	0	1	1	0	0	0	0	0

$m \times n$ matrix

m = Documents / rows

N = words / columns

SVD (Singular value Decomposition)

$$A_{[m \times n]} = U_{m \times r} * S_{[r \times r]} * (V_{n \times r})^T$$

A: Input Data Matrix
* $m \times n$ matrix (m = Documents, n = words / features)

U: Left Singular Matrix
* $m \times r$ (m = Documents, r = concepts)

S: Rank Matrix (Diagonal Matrix) [N]
* $r \times r$ (r = rank of A)

V: Right ~~Sing~~ Singular Matrix
* $n \times r$ (n = words / features, r = concepts)

Applications

Article Bucketing Websites
Finding relations between words / articles.
* Page Indexing in Search Engines.

~~wordnet~~

Word Net → word synonyms and antonyms. / Negation words

Wordnet → using Antonyms

(not happy ⇒ unhappy
not_happy)

Building a chatbot step by step

1. Scope of work (SOW)
2. Chatpot platform selection
3. Diaglogflow setup, (Define intent, context, fulfillment)
4. Build Backend
5. Website intregation.