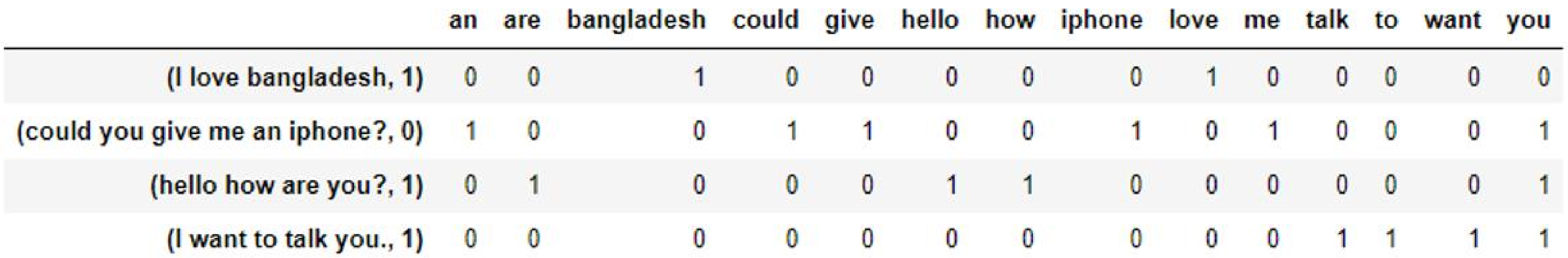
Different Techniques to Represent Words as Vectors:
Vectorizer

Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which used to find word predictions, word similarities/semantics. The process of converting words into numbers are called Vectorization.



	an	are	bangladesh	could	give	hello	how	iphone	love	me	talk	to	want	you
(I love bangladesh, 1)	0	0	1	0	0	0	0	0	1	0	0	0	0	0
(could you give me an iphone?, 0)	1	0	0	1	1	0	0	1	0	1	0	0	0	1
(hello how are you?, 1)	0	1	0	0	0	1	1	0	0	0	0	0	0	1
(I want to talk you., 1)	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Different Techniques to Represent Words as Vectors:
Vectorizer

Feature transformation: transformation of data to improve the accuracy of the algorithm.

Feature selection: removing unnecessary features.

Feature extraction: transformation of raw data into features suitable for modeling.

Different Techniques to Represent Words as Vectors:
Vectorizer

- Bag of Words: Count Vectorizer
- TF-IDF Vectorizer
- Word2Vec

Different Techniques to Represent Words as Vectors: Vectorizer

Count Vectorizer

Count Vectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.

Class 12

AI Quest

Different Techniques to Represent Words as Vectors:
Vectorizer

Sentences:

Count Vectorizer

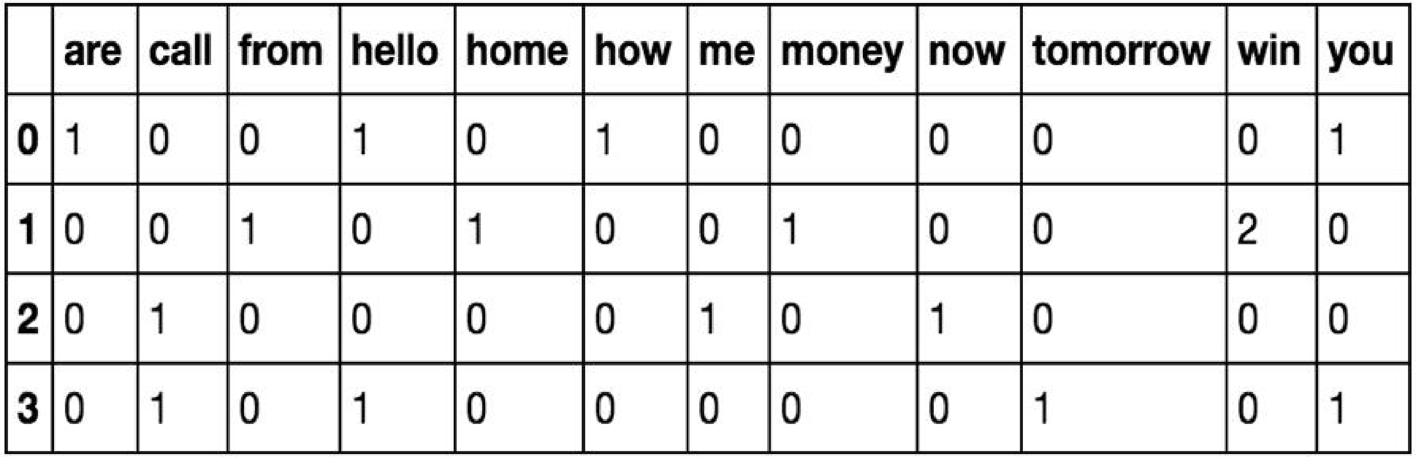
['Hello, how are you!',
'Win money, win from home.',
'Call me now',
'Hello, Call you tomorrow?']

Hello, How are you?

Win money, win from home

Call me now

Hello, Call you tomorrow?



	are	call	from	hello	home	how	me	money	now	tomorrow	win	you
0	1	0	0	1	0	1	0	0	0	0	0	1
1	0	0	1	0	1	0	0	1	0	0	2	0
2	0	1	0	0	0	0	1	0	1	0	0	0
3	0	1	0	1	0	0	0	0	0	1	0	1

Class 12

AI Quest

Different Techniques to Represent Words as Vectors: Vectorizer

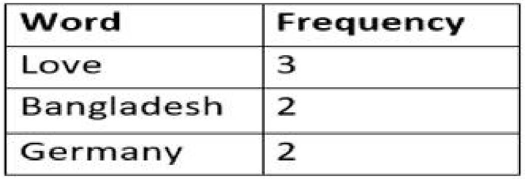
TF-IDF Vectorizer

Dataset:

Sentence1 = Love Bangladesh

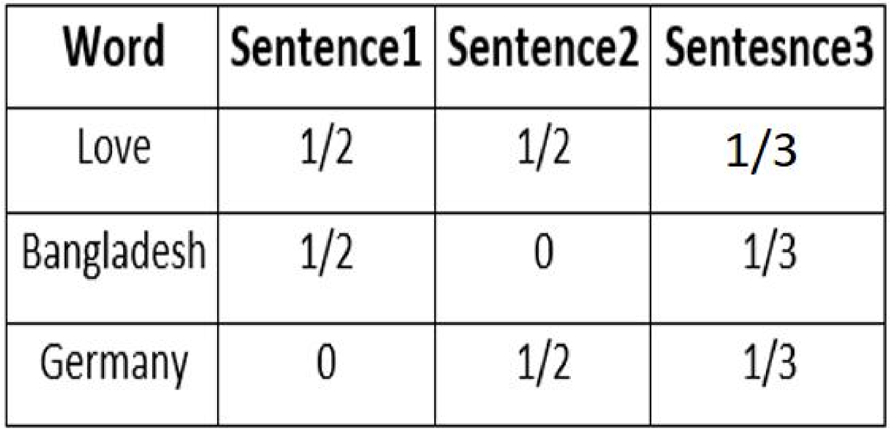
Sentence2 = Love Germany

Sentence3 = Love Bangladesh Germany



Word	Frequency
Love	3
Bangladesh	2
Germany	2

$$\text{Term Frequency (TF)} = \frac{\text{Number of repetition of word in sentence}}{\text{Total word in sentence}}$$



Word	Sentence1	Sentence2	Sentesnce3
Love	1/2	1/2	1/3
Bangladesh	1/2	0	1/3
Germany	0	1/2	1/3

Different Techniques to Represent Words as Vectors:
Vectorizer

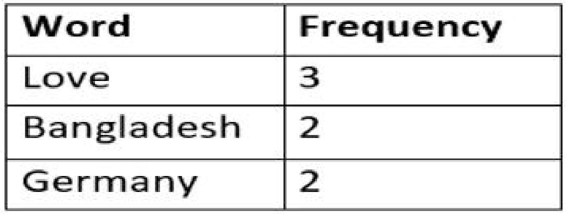
TF-IDF Vectorizer

Dataset:

Sentence1 = Love Bangladesh

Sentence2 = Love Germany

Sentence3 = Love Bangladesh Germany



Word	Frequency
Love	3
Bangladesh	2
Germany	2

$$\text{IDF} = \log \frac{\text{Total number of sentence}}{\text{No of word contain the sentence}}$$

Word	IDF
Love	Log 3/3 = 0
Bangladesh	Log 3/2 =
Germany	Log 3/2 =

Different Techniques to Represent Words as Vectors: Vectorizer

TF-IDF Vectorizer

TF

Word	Sentence1	Sentence2	Sentesnce3
Love	1/2	1/2	1/2
Bangladesh	1/2	0	1/3
Germany	0	1/2	1/3

IDF

Word	IDF
Love	Log 3/3 = 0
Bangladesh	Log 3/2 =
Germany	Log 3/2 =

TF * IDF =

Sentences	Love	Bangladesh	Germany
Sentence1	0	1/2 * log 3/2	0
Sentence2	0	0	1/2 * log 3/2
Sentence3	0	1/3 * log 3/2	1/3* log 3/2

TF-idf vs Count Vectorizer

TF-IDF is better than Count Vectorizers because it not only focuses on the frequency of words present in the corpus but also provides the importance of the words. The term "df" is called document frequency which means in how many documents the word "subfield" is present within corpus.

Can TF IDF Be Negative?

- No. The lowest value is 0. Both term frequency and inverse document frequency are positive numbers.

Different Techniques to Represent Words as Vectors:
Vectorizer

Word2Vec Vectorizer

Word2vec is a Neural Network that processes text by “vectorizing” words. Its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus.

Class 12

AI Quest

Different Techniques to Represent Words as Vectors: Vectorizer

Word2Vec Vectorizer

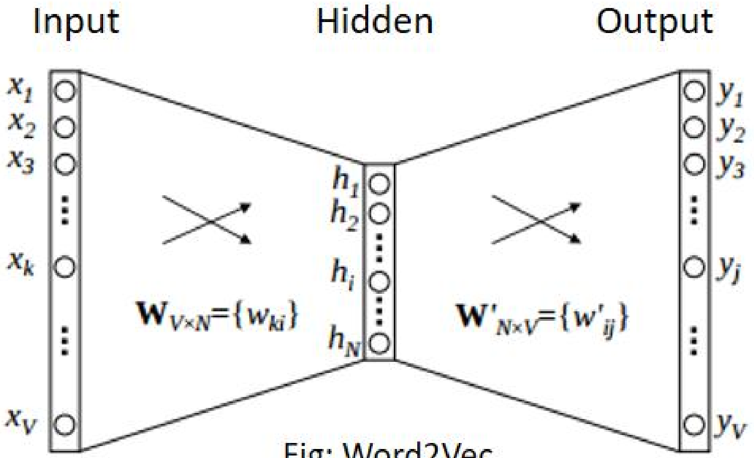
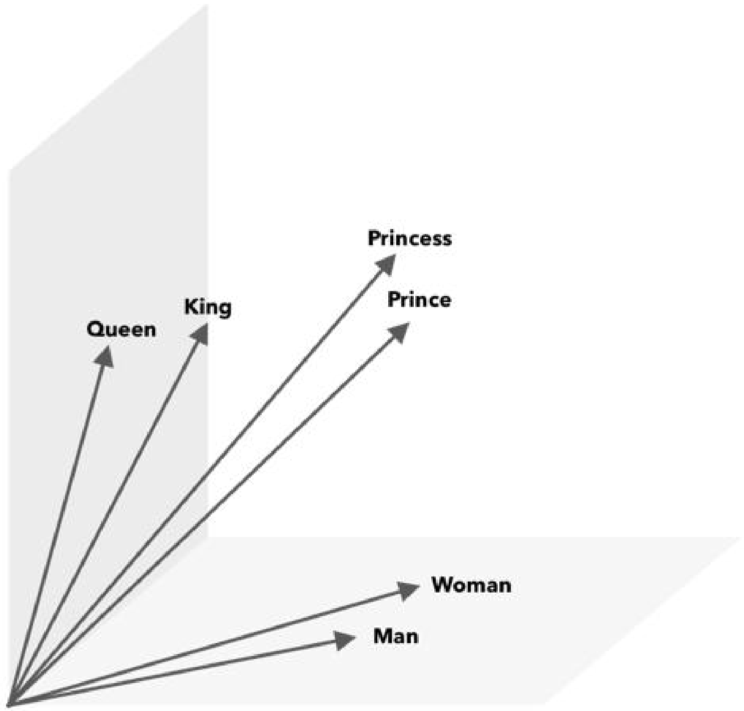


Fig: Word2Vec

Different Techniques to Represent Words as Vectors: Vectorizer

Word2Vec Vectorizer



- King – Man + Women = Queen
- Prince + mom = Queen
- vec("king") - vec("man") + vec("woman") =~ vec("queen")