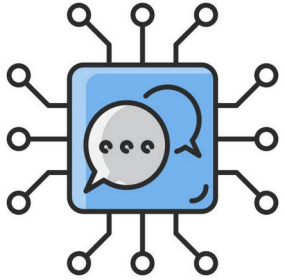
# Word2Vec Model

Tushar B. Kute,
http://tusharkute.com



# What is Word Embedding?

- Word Embedding is a word representation type that allows machine learning algorithms to understand words with similar meanings.
- It is a language modeling and feature learning technique to map words into vectors of real numbers using neural networks, probabilistic models, or dimension reduction on the word co-occurrence matrix.
- Some word embedding models are Word2vec (Google), Glove (Stanford), and fastest (Facebook).

# What is Word Embedding?

- Word Embedding is also called as distributed semantic model or distributed represented or semantic vector space or vector space model.
- As you read these names, you come across the word semantic which means categorizing similar words together.
- For example fruits like apple, mango, banana should be placed close whereas books will be far away from these words.
- In a broader sense, word embedding will create the vector of fruits which will be placed far away from vector representation of books.

# Where Word Embedding is used?

- Compute similar words: Word embedding is used to suggest similar words to the word being subjected to the prediction model. Along with that it also suggests dissimilar words, as well as most common words.
- Create a group of related words: It is used for semantic grouping which will group things of similar characteristic together and dissimilar far away.
- Feature for text classification: Text is mapped into arrays of vectors which is fed to the model for training as well as prediction. Text-based classifier models cannot be trained on the string, so this will convert the text into machine trainable form. Further its features of building semantic help in text-based classification.

# Where Word Embedding is used?

- Document clustering: is another application where Word Embedding Word2vec is widely used
- Natural language processing: There are many applications where word embedding is useful and wins over feature extraction phases such as parts of speech tagging, sentimental analysis, and syntactic analysis.
  - Now we have got some knowledge of word embedding. Some light is also thrown on different models to implement word embedding.

# Word2Vec

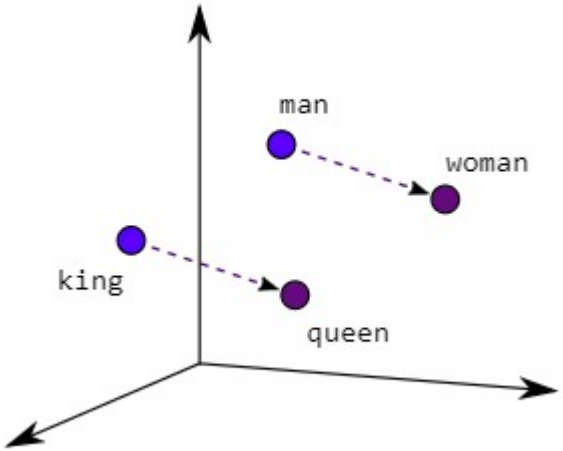
- Word2vec is a technique/model to produce word embedding for better word representation.
- It is a natural language processing method that captures a large number of precise syntactic and semantic word relationships.
- It is a shallow two-layered neural network that can detect synonymous words and suggest additional words for partial sentences once it is trained.

# Word2Vec

- Word2vec is a two-layer network where there is input one hidden layer and output.
- Word2vec was developed by a group of researcher headed by Tomas Mikolov at Google.
- Word2vec is better and more efficient that latent semantic analysis model.

# Word2Vec

- As seen in the image below where word embeddings are plotted, similar meaning words are closer in space, indicating their semantic similarity.



# Why Word2Vec?

- Word2vec represents words in vector space representation.
- Words are represented in the form of vectors and placement is done in such a way that similar meaning words appear together and dissimilar words are located far away.
- This is also termed as a semantic relationship. Neural networks do not understand text instead they understand only numbers.
- Word Embedding provides a way to convert text to a numeric vector.

# Why Word2Vec?

- Word2vec reconstructs the linguistic context of words. Before going further let us understand, what is linguistic context?
- In general life scenario when we speak or write to communicate, other people try to figure out what is objective of the sentence.
- For example, “What is the temperature of India”, here the context is the user wants to know “temperature of India” which is context. In short, the main objective of a sentence is context.
- Word or sentence surrounding spoken or written language (disclosure) helps in determining the meaning of context. Word2vec learns vector representation of words through the contexts.

# What Word2Vec does?

- Before Word Embedding
  - It is important to know which approach is used before word embedding and what are its demerits and then we will move to the topic of how demerits are overcome by Word embedding using Word2vec approach.

# Bag of Words

- It ignores the order of the word, for example, this is bad = bad is this.
- It ignores the context of words. Suppose If I write the sentence “He loved books. Education is best found in books”.
- It would create two vectors one for “He loved books” and other for “Education is best found in books.” It would treat both of them orthogonal which makes them independent, but in reality, they are related to each other

# How it works?

- Word2vec learns word by predicting its surrounding context. For example, let us take the word “He loves Football.”
- We want to calculate the Word2vec for the word: loves.
- Suppose
  - loves = Vin. P(Vout / Vin) is calculated

    where,

    Vin is the input word.

    P is the probability of likelihood.

    Vout is the output word.

# How it works?

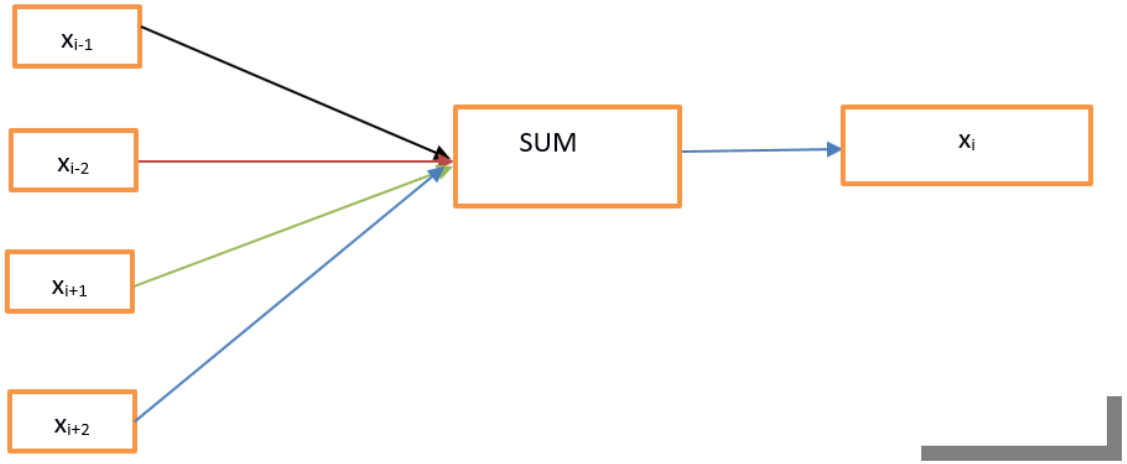
- Word loves moves over each word in the corpus. Syntactic as well as the Semantic relationship between words is encoded. This helps in finding similar and analogies words.
- All random features of the word loves is calculated. These features are changed or update concerning neighbor or context words with the help of a Back Propagation method.
- Another way of learning is that if the context of two words are similar or two words have similar features, then such words are related.

# Word2vec Architecture

- There are two architectures used by Word2vec:
  - Continuous Bag of words (CBOW)
  - Skip gram

# CBOW

- In CBOW, the current word is predicted using the window of surrounding context windows. For example, if wi-1,wi-2,wi+1,wi+2are given words or context, this model will provide wi



# CBOW

- Continuous Bag of Words (CBOW) is a neural network architecture used in Natural Language Processing (NLP) for learning word embeddings.
- These embeddings are vector representations of words that capture their semantic meaning and relationships with other words.

# CBOW

- The Problem:
  - Traditional methods like one-hot encoding represent words as sparse vectors with only one active element. This doesn't capture any semantic information or relationships between words. CBOW aims to overcome this limitation by learning dense vectors that encode meaning.
- The Approach:
  - CBOW predicts the surrounding words (context) given a target word. This means it takes a single word as input (e.g., "happy") and tries to predict words around it (e.g., "joyful," "excited"). By learning to predict context, the model implicitly learns the semantic meaning of the target word and its relationships with other words.

# CBOW

- The Architecture:
  - An input layer that receives the one-hot encoded vector of the target word.
  - A hidden layer that learns the word representation based on the input.
  - An output layer that predicts the probability distributions of the surrounding words.

# Skip-Gram

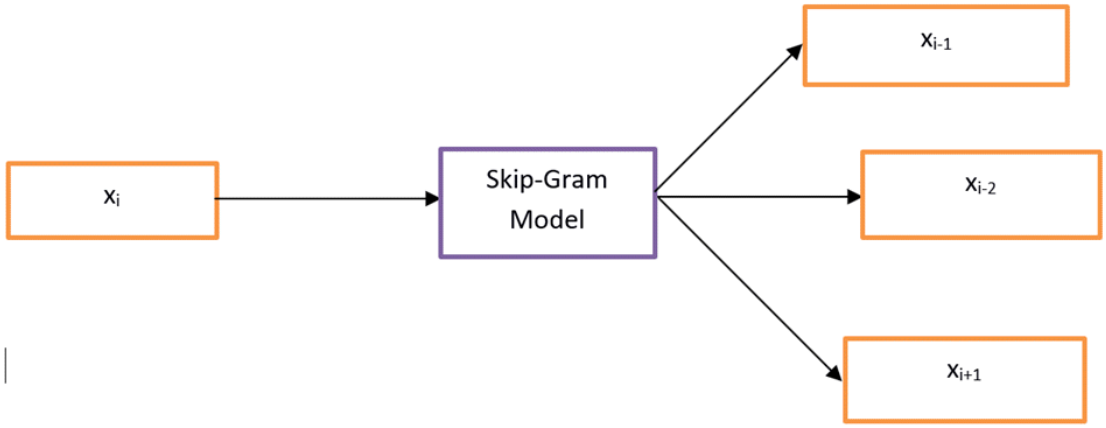
- The skip-gram model is a popular method for learning word representations or word embeddings in Natural Language Processing (NLP).
- It aims to capture the semantic meaning and relationships between words by mapping them to vectors in a low-dimensional space.

# Skip-Gram

- Imagine you have a large corpus of text like books, articles, or online reviews.
- You want to understand the relationships between words and how they connect within sentences.
- However, directly representing words as one-hot vectors (a vector of zeros with a single 1 for the specific word) is inefficient and doesn't capture semantic meaning.

# Skip-Gram

- Skip-Gram performs opposite of CBOW which implies that it predicts the given sequence or context from the word. You can reverse the example to understand it. If wi is given, this will predict the context or wi-1,wi-2,wi+1,wi+2.



# Skip-Gram: Steps

- Target Word and Context: It focuses on a single word (the target word) at a time and tries to predict the surrounding words (the context words) within a specific window size (e.g., 2 words before and after).
- Neural Network: The model utilizes a neural network with an input layer representing the target word and an output layer for each possible context word.

# Skip-Gram: Steps

- Training the Network: During training, the network compares the predicted probabilities of each context word with the actual words present in the surrounding window. The weights of the neural network are adjusted to minimize the prediction error.
- Word Embeddings: As the network learns, it develops internal representations of words (embeddings) stored in the hidden layer. These vectors capture the semantic meaning and relationships between words based on their co-occurrence and prediction patterns.

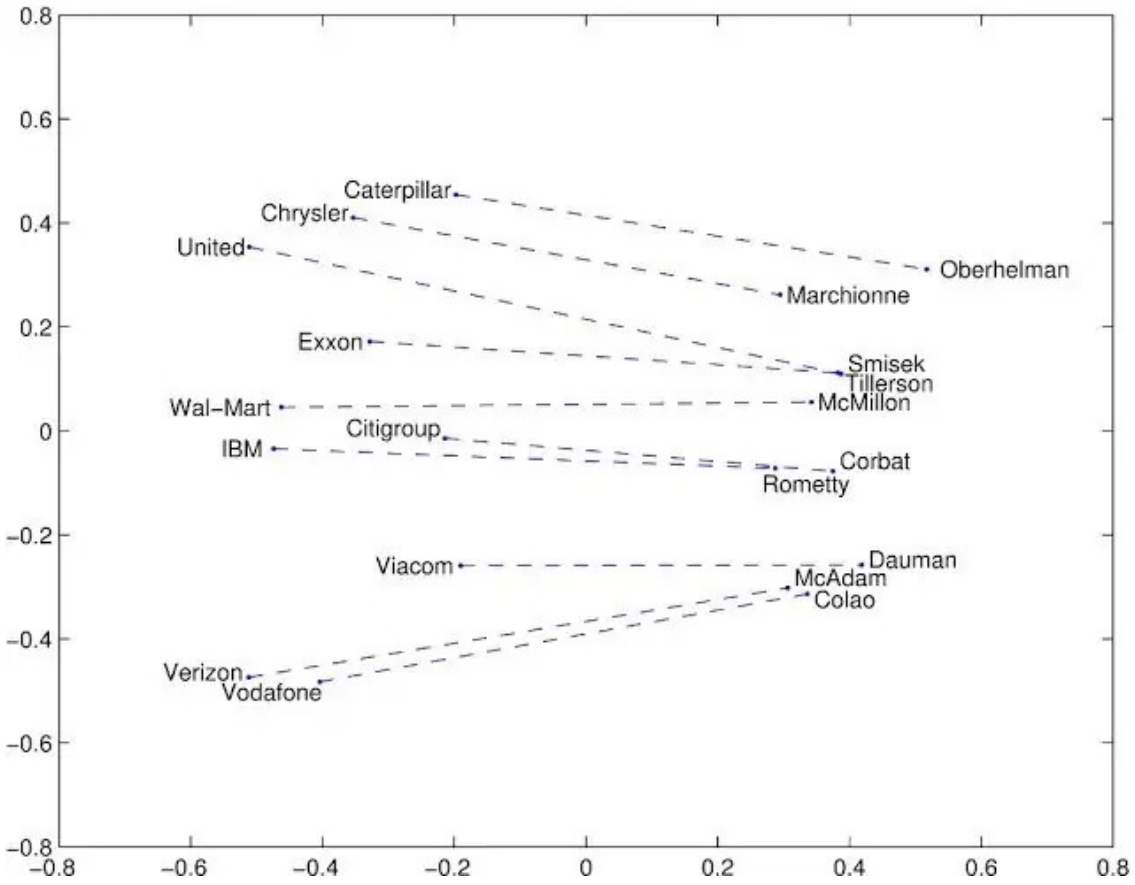
# Which one to choose?

- CBOW is several times faster than skip gram and provides a better frequency for frequent words whereas skip gram needs a small amount of training data and represents even rare words or phrases.

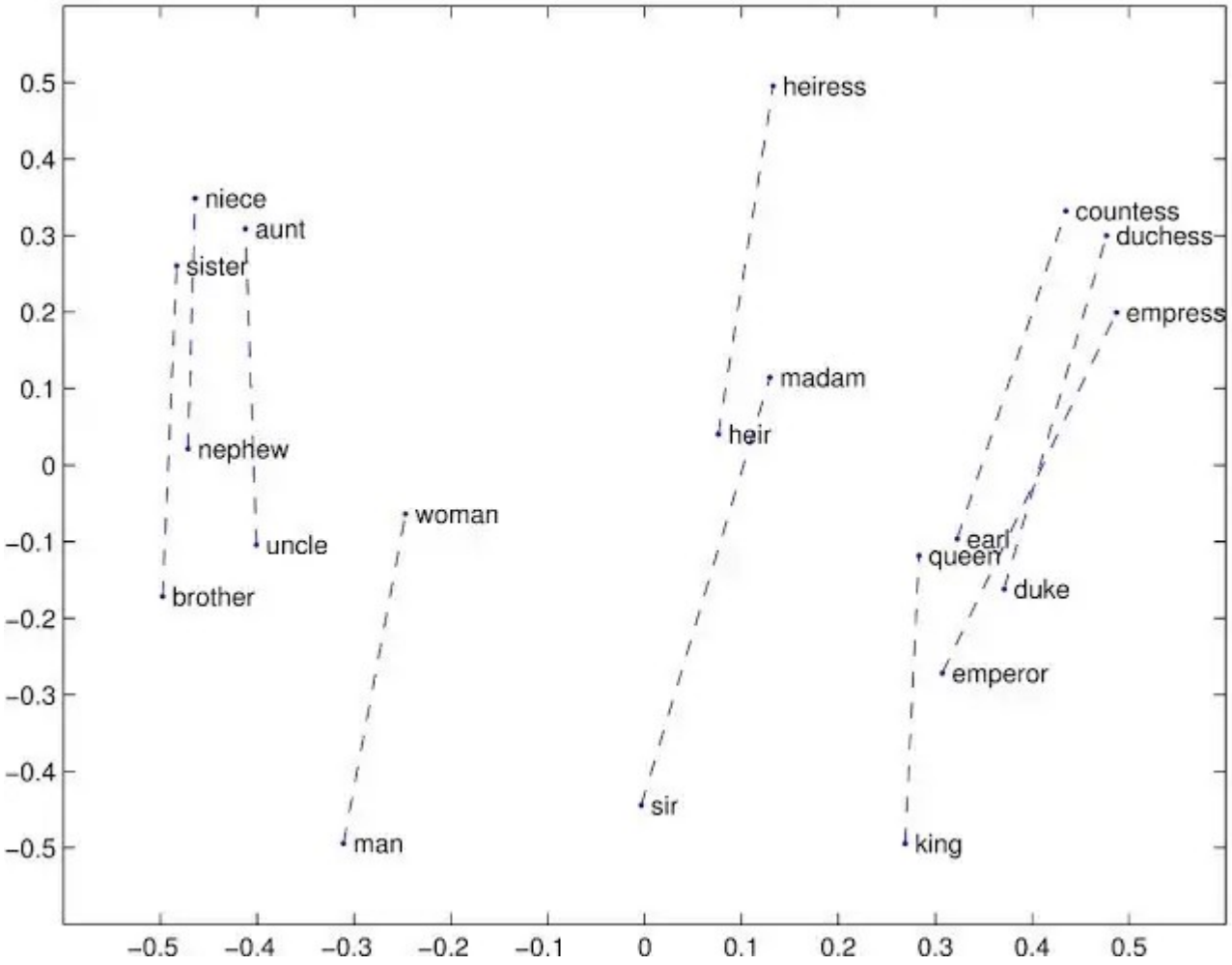
# GloVe

- GloVe stands for global vectors for word representation.
- It is an unsupervised learning algorithm developed by Stanford for generating word embeddings by aggregating global word-word co-occurrence matrix from a corpus.
- The resulting embeddings show interesting linear substructures of the word in vector space.

# GloVe


0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
Caterpillar
Chrysler
United
Oberhelman
Marchionne
Exxon
Smisek
Tillerson
Wal−Mart
McMillon
Citigroup
IBM
Corbat
Rometty
Viacom
Dauman
McAdam
Colao
Verizon
Vodafone

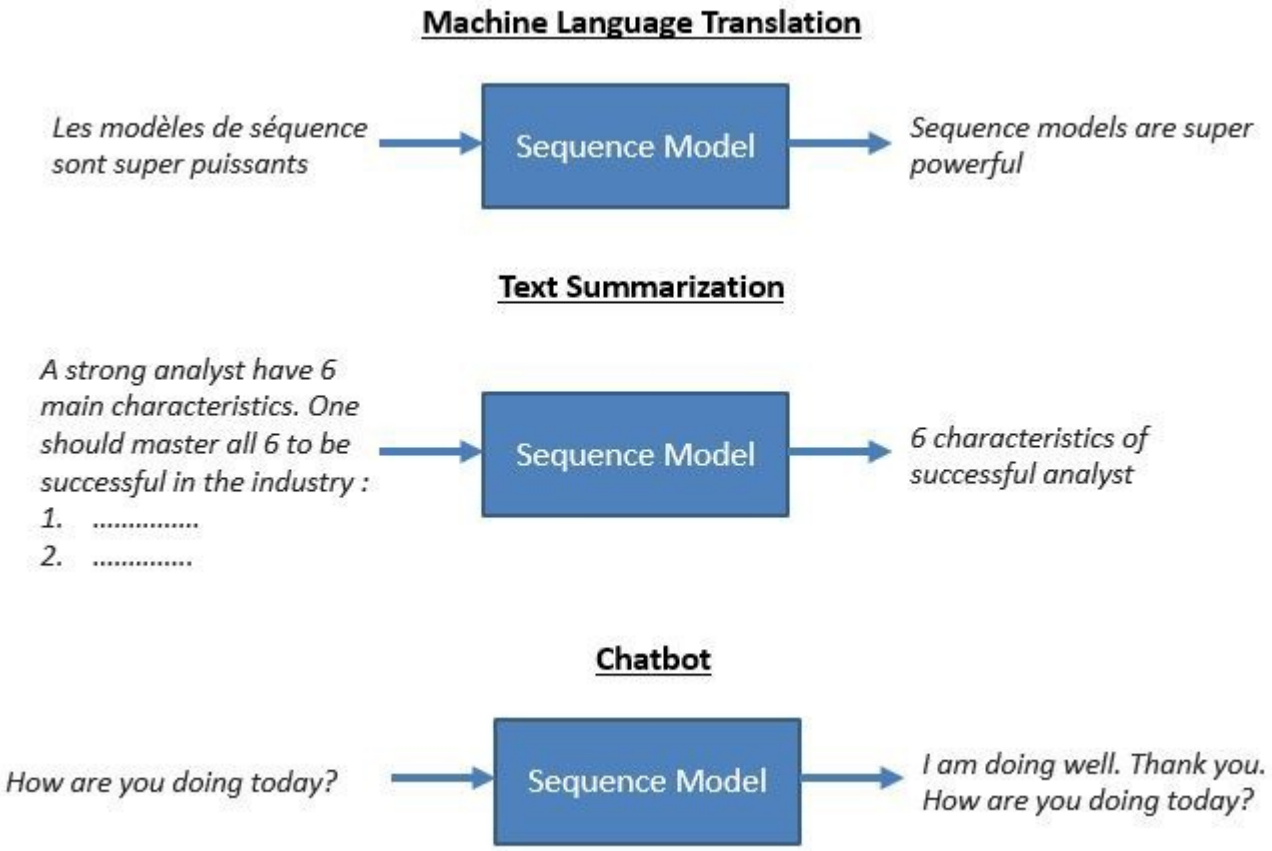
# GloVe


heiress
niece
aunt
sister
countess
duchess
empress
madam
nephew
heir
woman
uncle
earl
queen
duke
brother
emperor
sir
man
king
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.5
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5

# Seq2Seq

- Sequence to Sequence (often abbreviated to seq2seq) models is a special class of Recurrent Neural Network architectures that we typically use (but not restricted) to solve complex Language problems like Machine Translation, Question Answering, creating Chatbots, Text Summarization, etc.

# Seq2Seq


Machine Language Translation
Les modèles de séquence sont super puissants
Sequence Model
Sequence models are super powerful
Text Summarization
A strong analyst have 6 main characteristics. One should master all 6 to be successful in the industry :
1. .....
2. .....
Sequence Model
6 characteristics of successful analyst
Chatbot
How are you doing today?
Sequence Model
I am doing well. Thank you. How are you doing today?

# Thank you

*This presentation is created using LibreOffice Impress 7.4.1.2, can be used freely as per GNU General Public License*

@mitu_skillologies

@mITuSkillologies

@mitu_group

@mitu-skillologies

@MITUSkillologies

kaggle

@mituskillologies

**Web Resources**
https://mitu.co.in
http://tusharkute.com

@mituskillologies

**contact@mitu.co.in**

**tushar@tusharkute.com**