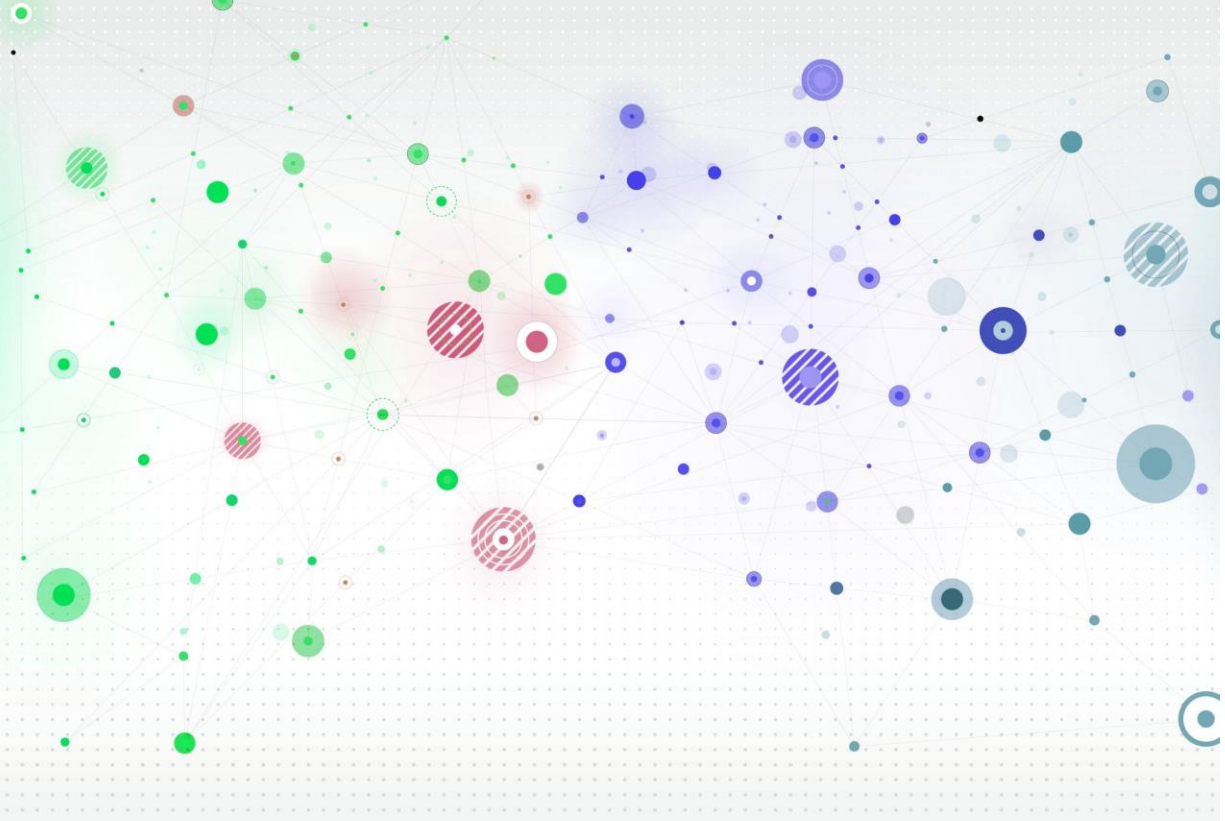
WORD REPRESENTATIONS



BAG OF WORDS
TF-IDF
WORD2VEC

Prashant Kodali
PhD Scholar @IIIT H
prashant.kodali@research.iiit.ac.in

TOPICS COVERED SO FAR

- **Lecture - 1**: Python basics includes list, tuple, dictionary, loops and functions
- **Lecture - 2**: Additional concepts in python, Introduction to Numpy, pandas, and matplotlib
- **Lecture - 3**: Regular expressions, stop words, lemmatization, stemming, Tokenization and Challenges in tokenization
- **Lecture - 4**: Spelling mistake detection and correction with minimum edit distance, Chunking and NER, POS tagging
- **Lecture - 5**: Language Modeling, Smoothing

OVERVIEW

- Why Representations?
- Bag of words
  - In practice
- TF-IDF
  - Motivation
  - In practice
- Word2vec
  - Motivation
  - In practice

WHY DO WE NEED REPRESENTATIONS?

- To feed text into any ML algo we need **features**.
- Strength/Richness of these features
  - Should be sufficient for your task
- Bag of words, TF-IDF, word2vec are some examples.
  - Basic but still powerful for some tasks

BAG OF WORDS

- Break down text into the constituent words and represent the counts of these words in a vector
- Ignores
  - Word order
  - Any grammar
- Useful for some IR tasks
  - Document classification
- Steps involved:
  - Build vocab: extract all unique words in a corpus
  - Measure which words are present, along with their counts
- Can be scaled to grams : to capture some context
- Issues
  - Sparsity
  - Ignores any sense of grammar
  - Frequently occurring words will dominate over the words which are seen fewer times in the corpus

TF - IDF

- intended to reflect how important a word is to a document in a collection or corpus.
- A way to weight the occurrence scores
- Consists of two terms
  - Term frequency

$$\frac{\#_d(w)}{\sum_{w' \in d} \#_d(w')}$$

$$\frac{\#_d(w)}{\sum_{w' \in d} \#_d(w')} \times \log \frac{|D|}{|\{d \in D : w \in d\}|}$$

  - Inverse Documents Frequency

$$\log \frac{|D|}{|\{d \in D : w \in d\}|}$$

EXAMPLE OF TF-IDF

Sentence A : The car is driven on the road.
Sentence B : The truck is driven on the highway.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

DISTRIBUTED SEMANTICS : WORD2VEC

Representing words by their context

- Distributional semantics: **A word's meaning is given by the words that frequently appear close-by**
  - *"You shall know a word by the company it keeps"* (J. R. Firth 1957: 11)
  - One of the most successful ideas of modern statistical NLP!
- When a word *w* appears in a text, its context is the set of words that appear nearby (within a fixed-size window).
- Use the many contexts of *w* to build up a representation of *w*

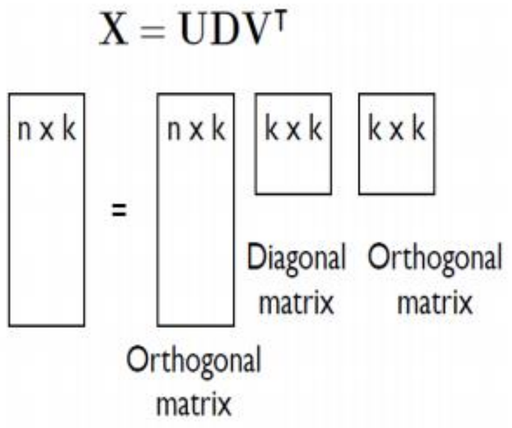
…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*

These context words will represent ***banking***

15

TO REDUCE DIMENSIONALITY

- Singular Value Decomposition on X to get a USV^T decomposition.
- use the rows of U as the word embeddings for all words in our dictionary



3. Word2vec: Overview

Word2vec (Mikolov et al. 2013) is a framework for learning word vectors

Idea:

- We have a large corpus of text
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word c and context ("outside") words o
- Use the similarity of the word vectors for c and o to calculate the probability of o given c (or vice versa)
- Keep adjusting the word vectors to maximize this probability

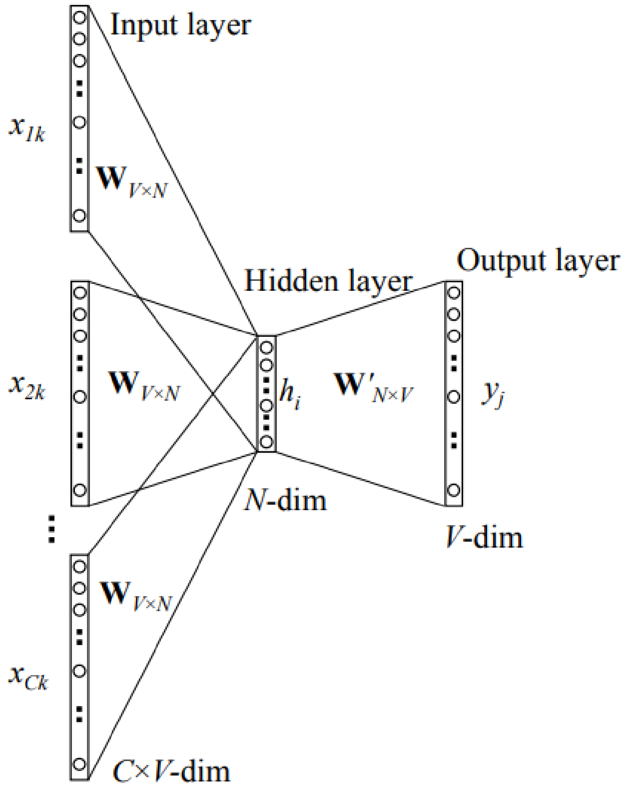


Figure 2: Continuous bag-of-word model

6

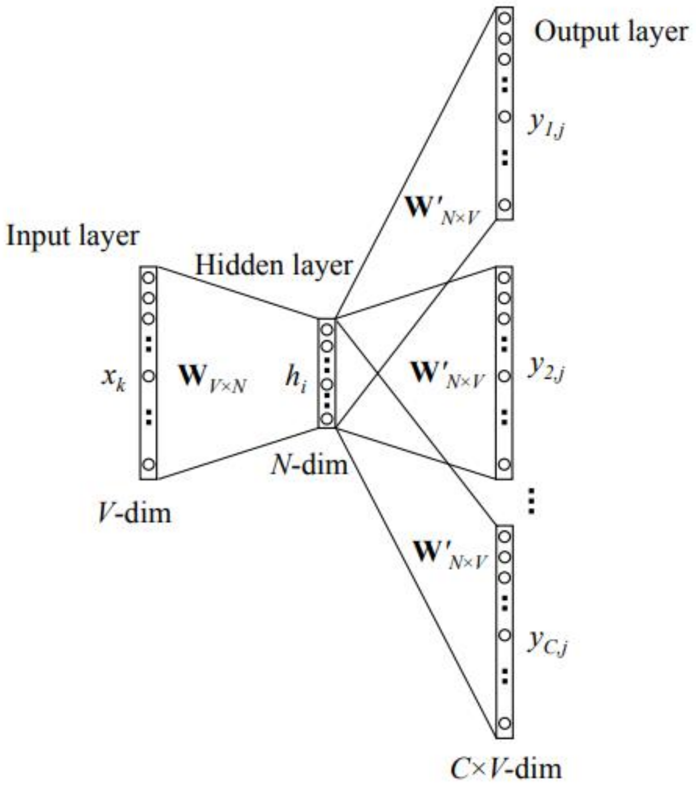


Figure 3: The skip-gram model.

LETS LOOK AT HOW TO IMPLEMENT THESE!

TOPIC FOR NEXT SESSION

- Neural Networks Basics:
  - Feedforward Neural Network,
  - Forward and Back propagation,
  - Activation Functions,
  - Gradient Descent