Statistical NLP

* BOW (sparse)
* TF-IDF (sparse)
* Token
* Stem
* Lemma
* Stop Words
* Word Cloud

Sequential NLP

* Word2vec (dense)
* Implementation word2vec
* Introduction RNN and its variation
* Vanishing and exploding gradient
* LSTM
* LSTM use-cases
* GRU
* Implementation: sentiment analysis LSTM + keras

Frequency-based embedding: Traditional word vector embedding

Prediction-based embedding: Deep neural word vector embedding

One of the key ideas in NLP is how we can efficiently convert words into numeric vectors which can then be “fed into” various machine learning models to perform predictions.

**word vector:**

Computers are unable to understand the concepts of words. In order to process natural language a mechanism for representing text is required.

The standard mechanism for text representation are **word vectors** where *words or phrases from a given language vocabulary are mapped to vectors of real numbers.*

Word embedding is a necessary step in performing efficient natural language processing in your machine learning models. **Word Embeddings are the texts converted into numbers** and there may be different numerical representations of the same text.

Why do we need Word Embeddings?:

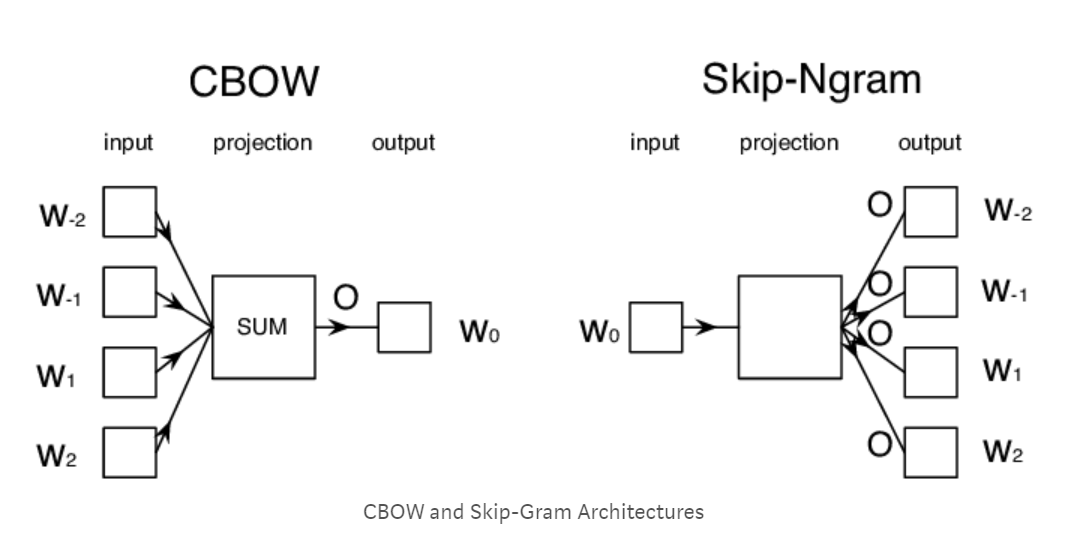
The current key embedding technique is called “Word2Vec”

WORD2VEC:

* It is the most powerful techniques in converting text to a vector.
* Its input is a text corpus and its output is a set of vectors.
* It turns text into a numerical form that deep nets can understand.
* This technique actually takes the semantic meaning of the words into consideration.
* Unlike, tf-IDF, bow: they do not take semantic meaning into consideration.
* Currently, w2v is almost state-of-the-art in NLP.

These methods were a prediction based on the sense that they provided probabilities to the words and proved to be state of the art for tasks like word analogies and word similarities.

* Word2vec is not a single algorithm but a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model.
* Both of these are shallow neural networks which map word(s) to the target variable which is also a word(s).
* Both of these techniques learn weights which act as word vector representations.

  + **CBOW:**
  + Predict the probability of a word given a context.
  + Input (context word) -- CBOW -- Output (target / focus word)
  + **SKIP-GRAM:**
  + Predict the context given a word.
  + Input (target / focus word) -- SKIP-GRAM -- Output (context word)
  + 
  + \* A context word may be a single word or a group of words.