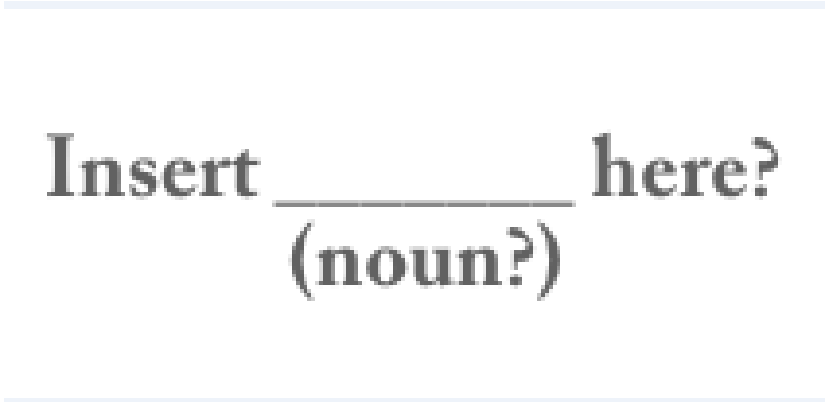
CS365 Course Project

Billion Word Imputation

Guide: Prof. Amitabha Mukherjee

Group 20:
Aayush Mudgal [12008]
Shruti Bhargava [13671]

Problem Statement

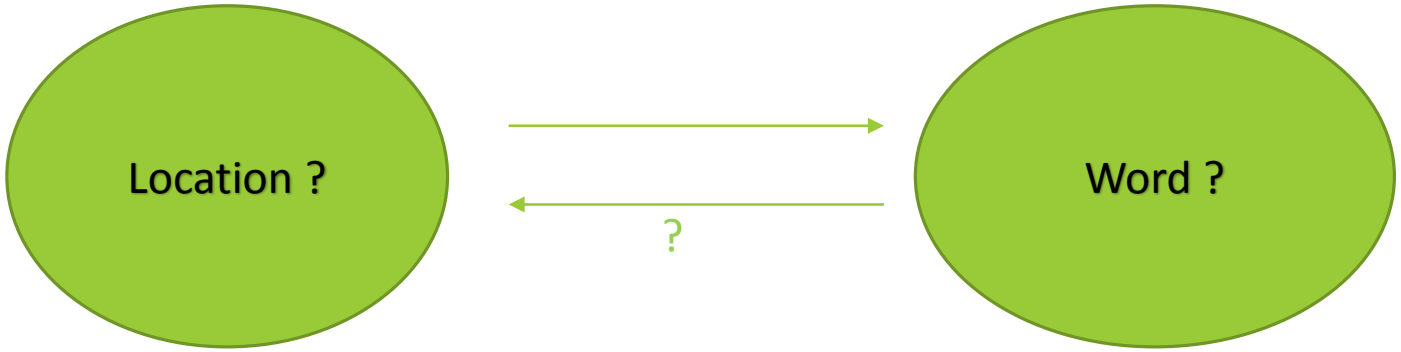


Problem Description : https://www.kaggle.com/c/billion-word-imputation

Examples :

1. “Michael described Sarah to a at the shelter .”
   - “Michael described Sarah to a _____? at the shelter.
2. “He added that people should not mess with mother nature , and let sharks be .”

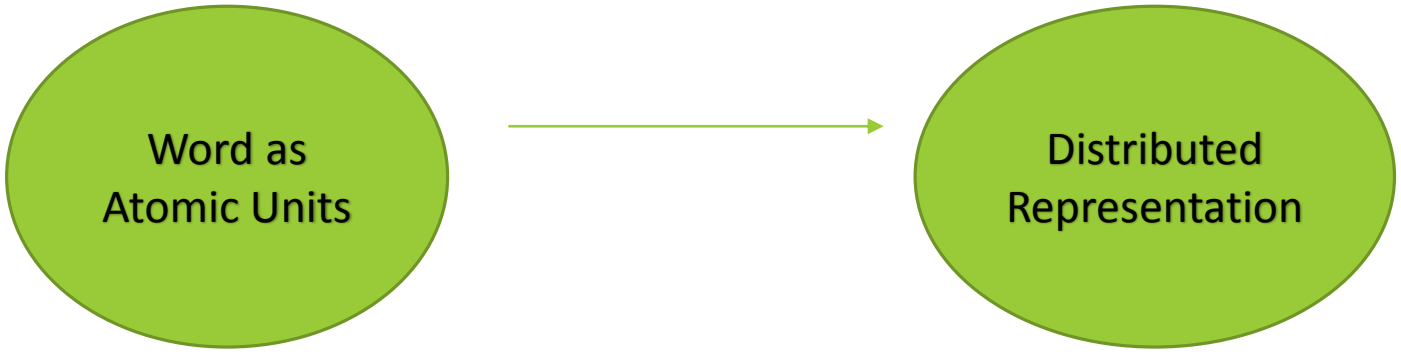
Basic Approach



1. Language modelling using Word2Vec
2. Strengthening using HMM / NLP Parser

Skip Gram VS N Gram

- Data is Sparse
- Example Sentence : “I hit the tennis ball”
- Word level trigrams: “I hit the”, “hit the tennis” and “the tennis ball”
- But skipping the word tennis, results in an equally important trigram



Word2vec by Mikolov et al.(2013)

Two architectures

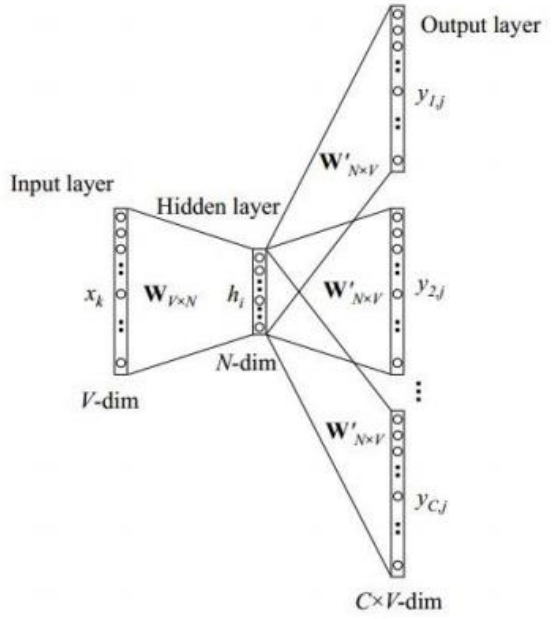
1. Continuous Bag-of-Word
   - Predict the word given the context

2. Skip Gram
   - Predict the context given the word
   - The training objective is to find word representations that are useful for predicting the surrounding words in a sentence or a document

Skip Gram Method

Given a sequence of training words w1, w2, w3, . . . , wT , the objective of the Skip-gram model is to maximize the average log probability :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$



c is the size of the training context (which can be a function of the center word wt)

Skip Gram Method

The basic Skip-gram formulation defines p(w_{t+j} |w_t) using the softmax function

$$p(w_O|w_I) = \frac{\exp\left(v'^{\top}_{w_O} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left(v'^{\top}_{w} v_{w_I}\right)}$$

where v_w and v'_w are the “input” and “output” vector representations of w
W is the number of words in the vocabulary.
IMPRACTICAL because the cost of computing ∇ log p(wO|wI) is proportional to W, which is often large (105–107 terms).

Sub-Sampling of Frequent Words

- The most frequent words like “in”, “the”, “a” can easily occur hundreds of millions of times (e.g., “in”, “the”, and “a”).
- Such words usually provide less information value than the rare words
- Example : Observation of France and Paris is much more beneficial
  - Than the frequent occurrence of “France” and “the”
- Vector representation of frequent words do not change significantly after training on several million examples

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

Skip-Gram Model : Limitation

- Word representations are limited by their inability to represent idiomatic phrases that are not compositions of the individual words.
- Example, “Boston Globe” is a newspaper, and not “Boston” + “Globe”

Therefore, using vectors to represent the **whole phrases** makes the Skip-gram model considerably more expressive.

Questions ?

Refrences

1. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in Neural Information Processing Systems*. 2013.

2. Mnih, Andriy, and Koray Kavukcuoglu. "Learning word embeddings efficiently with noise-contrastive estimation." *Advances in Neural Information Processing Systems*. 2013.

3. A Closer Look at Skip-gram Modelling David Guthrie, Ben Allison, W. Liu, Louise Guthrie, and Yorick Wilks. *Proceedings of the Fifth international Conference on Language Resources and Evaluation (LREC-2006), Genoa, Italy,* (*2006*)

4. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

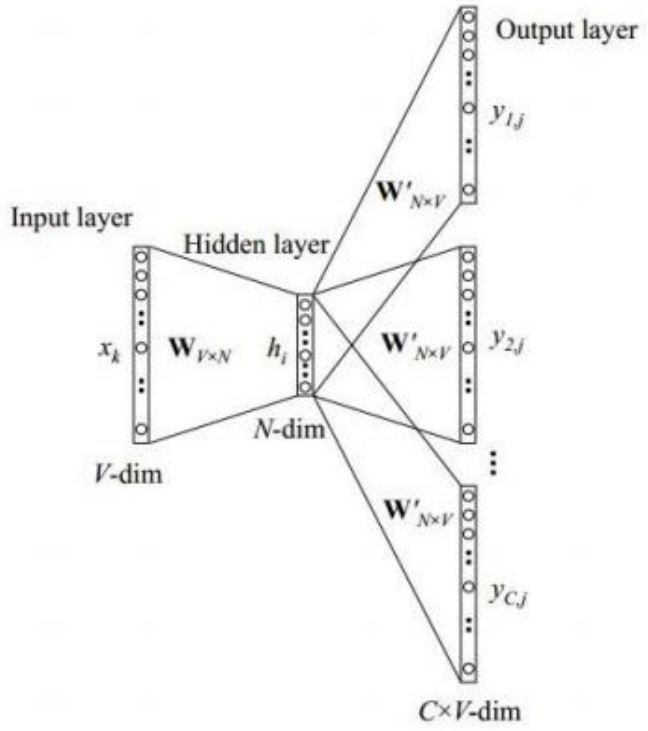
Challenge Description and Data : https://www.kaggle.com/c/billion-word-imputation

Hidden Markov Models

1. States : Parts of Speech
2. Combine Word2Vec with HMM

Skip-Gram Method

- Vocabulary size is V
- Hidden layer size is N
- Input Vector : One-hot encoded vector, i.e. only one node of $\{ X_{\{1\}}, X_{\{2\}}, \ldots\ldots, X_{\{v\}} \}$ is 1 and others 0
- Weights between the input layer and the output layer is represented by a VxN matrix W



Skip-Gram Method

- h=$x^T W = v_{Wi}$
- v_{Wi} is the vector representation of the input word w_i
- $u_j = {v'_{w_j}}^T . h$
- u_j is the score of each word in vocabulary and v'_{Wi} is the j-th column of matrix W'