SMAI Project Report

Team 19 - word2vec representation

Abstract

word2vec is a group of related models that are used to produce word embeddings. Word embeddings are vector representations of words. They are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the vector space. These distributed representations of words in a vector space help learning algorithms to achieve better performance in natural language processing (NLP) tasks by grouping similar words. They serve as the most commonly used form of input feature representation of natural language text to deep learning models for NLP tasks.

1 Introduction

The approach used to learn these vector representations were borrowed from the improved Skip-gram and CBOW algorithms. The first paper presents several improvements and extensions of the original Skip-gram model. They show that subsampling of frequent words during training results in a significant speedup (around 2x - 10x) and improves accuracy of the representations of less frequent words. In addition, they present a simplified variant of Noise Contrastive Estimation (NCE) for training the Skip-gram model that results in faster training and better vector representations for frequent words, compared to more complex hierarchical softmax that was used in the prior work.

The second, CBOW algorithm is similar to the Skip-gram approach except that it learns vectos by predicting the missing center word from the context instead of the other way around as in the Skip-gram approach. They call this architecture a bag-of-words model as the order of words in the history does not influence the vector space projection. Furthermore, they use context words from the left and right of the center word. This enabled the model to capture both past and future dependence of words on the missing word.

The main objective of the project was implement these word2vec algorithms to generate word embeddings over plain english text and domain specific data like medical text. The learnt vectors were used to perform a comparative analysis between the information encoded by vectors trained on plain english text and those trained on medical data. This helped in verifying that such vectors are able to retain domain knowledge and information.

2 Datasets

The medical data used for the purpose of this project was a mixture of medical research articles and news articles which were crawled as follows.

1. Medical research articles published on Nature with the "Medical Research" and "Medical Reviews". A total of 337 MB worth of research articles The sections scraped from those articles were (whichever were available): Abstract, Introduction, Methods, Results and Discussion
2. Entire text of medical news articles published on MedicalNewsToday from 2011 to April 2019. This amounted to 42 MB of textual data.

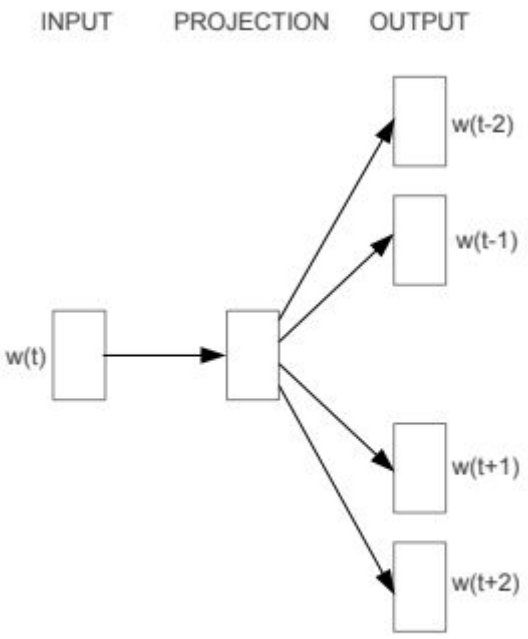
The scraped data was in its rawest form. It was rife with noise in the form of punctuation marks and special characters, single letter, numeric and alphanumeric tokens. Thus the cleaning of raw data was essential and it was performed via a script. The exact step-wise procedure followed was as follows:

1. All 's were removed
2. Sentence tokenization was performed
3. All non alphabetical characters were removed
4. All stop words were removed
5. All words of length less than and equal to 2 were removed

After cleaning the raw data was reduced to 268 MB. The cleaned medical data contained 37.5 million tokens in total and a frequent vocabulary of 46,800 words is considered.

We also use a small Wikipedia dump enwik9 as a substitute for plain english text. Of this data, a subsample of size 400 MB is taken which after cleaning reduces to 271 MB (comparable in size to the clean medical domain data). Thera are a total of 42.5 million tokens and a frequent vocabulary of 73,240 words is considered.

3 Skip-gram



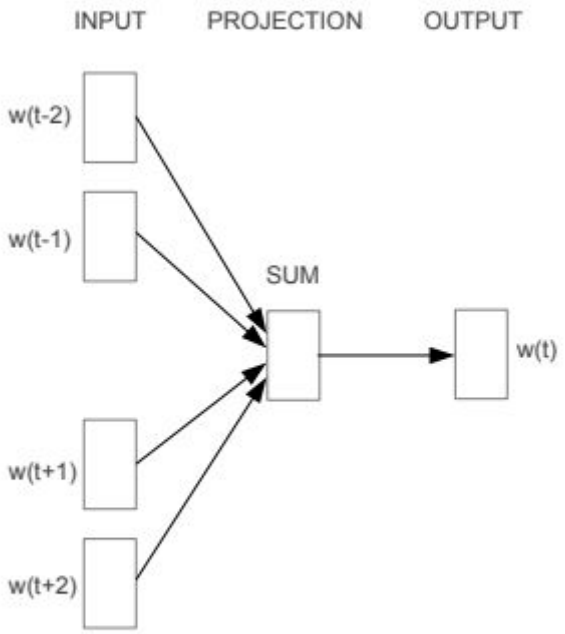
Skip-gram

Skip-gram model tries to maximize classification of a word based on another word in the same sentence. More precisely, it use each current word as an input to a log-linear classifier with continuous projection layer, and predict words within a certain range before and after the current word. Increasing the range improves quality of the resulting word vectors, but it also increases the computational complexity. Since the more distant words are usually less related to the current word than those close to it, so it gives less weight to the distant words by sampling less from those words in the training examples. The training complexity of this architecture is proportional to

$$Q = C \times (D + D \times \log_2(V))$$

where C is the maximum distance of the words. Thus, if we choose C = 5, for each training word we will select randomly a number R in range < 1; C >, and then use R words from history and R words from the future of the current word as correct labels. This will require us to do R × 2 word classifications, with the current word as input, and each of the R + R words as output. In the following experiments, we use C = 10.

4 Continuous bag-of-words



CBOW

CBOW Model is similar to the feedforward NNLM, where the non-linear hidden layer is removed and the projection layer is shared for all words (not just the projection matrix); thus, all words get projected into the same position (their vectors are averaged). This architecture is called a bag-of-words model as the order of words in the history does not influence the projection. Furthermore, we also use words from the future; a log-linear classifier with four future and four history words at the input is built, where the training criterion is to correctly classify the current (middle) word. Training complexity is then

$$Q = N \times D + D \times \log_2(V)$$

This model is denoted as CBOW as unlike standard bag-of-words model, it uses continuous distributed representation of the context. Note that the weight matrix between the input and the projection layer is shared for all word positions in the same way as in the NNLM.

5 Experimental Setup

Both the models were trained on cleaned versions of the raw text. The experimental parameters used to train both the models on the medical data and plain english text.

Parameter	Value	Comment
Minimum token frequency	15	There were 153,731 tokens with occurrence count less than 15. Such words which occur very rarely add noise to the model and cause an explosion in the training vocabulary.
Vector dimension	30	The vocabulary size wasn't exceedingly big and the total number of contexts were also fewer as compared to the original papers so a vector of size 30 seemed suitable
Number of epochs	5	Further increase in the training iterations didn't bring significant difference in the quality of vectors.
Initial learning rate	0.025	Over increasing iterations the learning rate was diminished.
Window size	5	This is the standard value for most word2vec models.

6 Evaluation criteria

1. Nearest neighbors

The cosine similarity between two word vectors provides an effective method for measuring the linguistic or semantic similarity of the

corresponding words. This helps in finding the nearest neighbours for any given word.

2. **Linear substructures**

   The word vectors also capture in a quantitative way the nuance necessary to distinguish words. They are designed such that the vector differences and additions capture as much as possible, the meaning specified by the juxtaposition of two or more words. Such vector operations also help solve different tasks like selecting out-of-the-list words, by computing average vector for a list of words and finding the most distant word vector.

7 Results and analysis

Examples of 3-nearest neighbours relationships

Word	Medical domain	Wikipedia
drug	drugs, antibiotic, artemisinin	drugs, marijuana, cocaine
brain	neurochemical, cns, cerebral	tumor, nervous, tissue
attack	attacks, palpitation, fluttering	assault, attacks, enemy
france	finland, netherlands, italy	italy, austria, germany
pathogen	organisms, pathogens, microorganisms	bacterium, parasite, tumors

Examples of 3-NN with linear substructure relationships

Words	Medical domain	Wikipedia
france - paris + rome	mexico, rome, finland	gaul, sparta, persia
bigger - big + high	higher, low, greater	higher, low, greater
brains - brain + kidney	kidneys, isam, liver	smoker, drinker, drank
woman - man + men	women, men, woman	women, men, mothers
artery - artery + neuron	neuronal, neuron, microglial	collectible, iconic, ramped

t-SNE visualisations