Project: Drug Prediction
Task: To Design A Sentence Completion System.

Abdullah Faiz Ur Rahman Khilji

National Institute of Technology Silchar

abdullahkhilji.nits@gmail.com
17-1-5-091@student.nits.ac.in

B.Tech in Computer Science and Engineering, Grad: 2021

Summer 2018

Abstract

The Task given to me was to Design a sentence completion system.
Given first 4 seed words of a sentence, it should predict the next word.
The training corpus used was from Sentiment140 (`http://help.sentiment140.com/for-students/`)

Contents

Chapter 1

Instruction to use the `task.py` program

1.1 Rename the corpus file

The name of the corpus file taken in the program is taken as `taskData.csv` , thus rename the file to `taskData.csv` on the 31st line of the code.
This should not be confused with `vocab.csv` that the program itself will generate on line 134.
The program contains a total of 300 lines.

Chapter 2

Methodology

2.1 Corpus

For completing the given task, as per stated in the abstract of the task the corpus was taken from Sentiment140 (`http://help.sentiment140.com/for-students/`). The data given there was a collection of tweets, and thus the data provided required detailed attention to pre processing.

2.2 Pre Processing

The pre processing stage required removing all the special characters or punctuation marks.
Even the capitalized letters were converted to small letters, and then Stemming was also done.
Thus the data know has only small character letters of the english alphabet alongwith spaces, This final data thus obtained will be referred to as refined data throughout this report.

Tweets less than 5 words were also removed to decrease complexity. Two different list were then formed.

Chapter 3

Preparing for Mapping of Words as Vectors

3.1 Grouping

Tweets having less than 5 words were also removed to decrease complexity.
Two different list were then formed one containing the first four words in an array (This is the input array, containing just the seed words)
And
The second one containing just the output array (The word that should be predicted after taking the seed words as input)

Chapter 4

Creating Word Vectors

4.1 Creating a Bag of Words

A bag of words is now formed from the refined corpus, containing only the words that are relevant.

4.2 Word2Vec

Word2vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text. Word2Vec comes together with the open source machine learning framework TensorFlow.
Thus, now after forming a collection of words (bag of words) the mapping of vector according to the relation of the words there mapping of equivalent pointing vector in 2 Dimension is thus made.
This is stored in a seperate list.

Chapter 5

Creating an input DataFrame i.e. converting the array of words to an array of vectors of each input word

5.1 The Process

Each word from the array of input Data is compared with the now formed bag of vector of words, and if it is found to be same the corresponding x1, y1 vector co-ordinates are placed in a new array corresponding to the position of the same element. The same procedure is repeated for the whole sliced array.
This process is repeated to both the input as well as output data's.

5.2 What if the word is not found in the bag of vector of words ?

To take care of this, all the elements of this array is automatically initially initialised to zero using numpy function zeros().

5.3 Wouldn't this cause a problem during training ?

To make a good model, this step is necessary.
We will have to thus remove the redundant cells and we remove all the cells corresponding to that cell.
In this way we can train a quite good model.

5.4 Train and test sets.

The data is divided into 20% and 80% ratio's for obvious reasons.

Chapter 6

Training the model and finding the predictions.

6.1 The Neural Networks

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.
Neural nets are widely used in pattern recognition because of their ability to generalise and to respond to unexpected inputs/patterns. During training, neurons are taught to recognise various specific patterns and whether to fire or not when that pattern is received. If a pattern is received during the execution stage that is not associated with an output, the neuron selects the output that corresponds to the pattern from the set of patterns that it has been taught of, that is least different from the input. This is called generalisation.

6.2 Feeding input to the Neural Network.

After all pre processing and getting a fine collection of bag of vector of words that are arranged by the position according to their relation with each other i.e. words that are related to each other get automatically clustered together.(To demonstrate this, I have thus also plotted all the pints corresponding to the vectors using MatPlotLib and you will find it in the output.)

Chapter 7

The Final Chapter

7.1 Output Vector

The neural network will now intelligently find a good relation between the input and output vectors, and will give a good prediction.

7.2 Comparing the test data with prediction

After training the data we should then compare it with the test data.
But since after so many refinements the data of size 1,00,000 reduces to mere 74,311 sentences having 431 bag's of vector words that can be trusted to come from spoken english, the % further reduces after splitting into test and train sets's.s Thus, for this task a sufficient size of data should be trained and this requires time.

The sample of 1,00,000 tweets gives an accuracy of 50-60% after some epochs.
Here it is important to note that the output word is found by finding the vector closest to the output vector in the bag of vector of words.

Bibliography

[1] Harrison Kinsley *pythonprogramming.net*. YouTube NLTK, Neural Network Videos

[2] Learn to create Machine Learning Algorithms in Python and R from two Data Science Experts. *Neural Network Models and Applications*. Kirill Eremenko, Hadelin de Ponteves, SuperDataScience Team