iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 7 Issue: IV Month of publication: April 2019

DOI: https://doi.org/10.22214/ijraset.2019.4474

Sentence Completion using NLP Techniques

Umang Rupareliya[1], Henali Shah[2], Pragnya Kulkarni[3], Nirmala Baloorkar (Shinde)[4]

[1, 2, 3]Student, [4]Assistant Professor, K.J.S.C.E., Department of Computer Engineering, Vidyavihar, Mumbai

Abstract: The paper aims to study, analyse and compare the results of sentence completion task based on SAT style questions. In this paper, we plan to apply and compare various approaches for automated sentence completion which include Latent Semantic Indexing and Recurrent Neural Networks. Results from the past research stated that the LSA model outperforms the conventional n-gram model models on the Microsoft Research Sentence Completion Challenge. Hence, we will analyse and compare the results of the LSA model with RNN model to decide which of them performs better considering the scale of the data.

The tasks involve training on a large corpus of unannotated text, to then try to predict the missing words in the test set which contains thousands of sentences where one word is missing and five alternatives for the missing word.

Methods using local information and global information for the task of sentence completion are used and we find that method using global information (Latent Semantic Analysis) proves to be better than the method using local information (Recurrent Neural Network). We compare our approach to Microsoft research sentence completion challenge by extending RNN to LSTM with RNN.

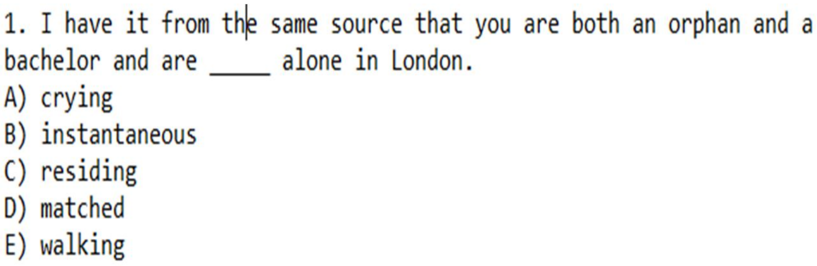
Keywords: Natural Language Processing, Recurrent Neural Networks, Latent Semantic Analysis, Microsoft Research.

I. INTRODUCTION

Sentence Completion plays a pivotal role in English language and communication. It triggers cognitive skills to interpret and evaluate the written sentences/words. Sentence Completion assists in efficient propagation of ideas and clear interaction with each other.

There are a variety of options to complete sentence which is semantically and syntactically correct but, in this paper, we have taken the most effective one by restricting to specific input options/value only. Till date, a few publications have focused on automatic and semi-automatic methods used to solve the questions of sentence completion. This paucity is mostly due to the attributable difficult nature of the given task, involving logical reasoning to both general and semantic knowledge.

The problem of Sentence completion is to measure grammatical and semantic correctness and to choose the most appropriate sentence/word to fill the blank. This tests the strength of algorithms to distinguish right from wrong based on a variety of sentence-level phenomena.



1. I have it from the same source that you are both an orphan and a bachelor and are _____ alone in London.
A) crying
B) instantaneous
C) residing
D) matched
E) walking

Figure 1.1 An example of sentence completion

For each sentence the task is to determine which of the five options for that sentence is the correct one. We aim to improve accuracy for most number of sentences completion problems. It should be accurate enough to provide answers better than N-gram models. We are going to use different NLP techniques to do the same.

Initially, we plan to predict the words in the sentences using RNN models. To improve on the RNN model, we propose to implement an alternative methodology, which is based on Latent Semantic Analysis, to address the problem of text completion. LSA/LSI improves the RNN based approach by considering also terms distant from the word to predict.

In this paper, we will investigate various methods to predict right answers to the given sentence completion questions. Also, we will be finding the accuracy of predicting appropriate word for given SAT style sentences. Finally, an overall solution which is easily implementable for fully working application purposes is developed.

II. RELATED WORK

The former work which is based on related lines to ours is derived from Microsoft Research Sentence Completion Challenge (G. Zweig, Christopher J. C. Burges, 2011) in which Microsoft has proposed a set of various English sentences. [7] Each sentence has been associated with the four impostor options, in which each word in the original sentence is replaced by an impostor word with similar occurrence statistics. The task for each sentence is to determine out of five options which is the correct option for the given sentence. This task is similar to the SAT language test. The question was generated using the following two steps. At first, the candidate sentence which contains an infrequent word was selected and the alternative word was determined automatically with the n-gram language model by sampling. The n-gram model used intermediate history as lexicon, which resulted in the words that are "suitable" locally, but there is no other reason to expect them to make it sense globally. In the second step, obvious incorrect choices are eliminated because they contained some grammatical errors. Data used was from the five of Conan Doyle's Sherlock Holmes novels: The Sign of the Four (1890), The Adventures of Sherlock Holmes (1892), The Hound of the Baskervilles (1892), The Memoirs of Sherlock Holmes (1894), and The Valley of Fear (1915). N-gram model was trained on 540 texts consisting mainly of 19th-Century Novels. This paper further explores a simple 4-gram model which achieved 34% correct on the test suite, smoothed N-gram model which improved by 5% absolute on the simple baseline to achieve 39% correct on the test set and then Latent Semantic Analysis Similarity results in best performance of 49% correct results.

Other work was carried out by Piotr Mirowski and Andreas Vlachos in 2015 in which they proposed a novel language dependency RNN model, which associated the syntactic dependencies into the RNN formulation. [4] They evaluated the performance on the Microsoft Sentence Research Sentence completion task which improved over RNN by 10 points in accuracy. The training was carried out on 522 19th century novels from Project Gutenberg. All the processing were performed using the Stanford CoreNLP toolkit. The test set contains 1040 sentences.

Note that their future work includes incorporating Long Short-Term Memory RNNs to handle longer syntactic dependencies whereas we incorporated Long Short-Term Memory (LSTM) in our RNN model for improving it's accuracy.

Research reported for Latent Semantic Analysis in the three articles- Foltz, Kintsch & Landauer (1998), Rehder, et al. (1998), and Wolfe, et al. (1998)—exploited a new theory of knowledge representation which determines the similarity of meaning of words and passages by analysing the large text corpora. [3] The paper claimed that LSA measures the similarity of meaning of words from text. Results that were evaluated are:(1) the meaning similarities match closely to those of humans, (2) LSA's rate of acquisition of such knowledge from text approximates that of humans, and (3) these accomplishments depend strongly on the dimensionality of the representation. LSA model was trained by running the Singular Vector Decomposition (SVD) which was analysed on a large corpus of representative English. In the experiment, an SVD was performed on text segments consisting of 500 characters or less taken from beginning portions of each of 30,473 articles in the encyclopaedia, a total of 4.5 million words of text. This resulted in a vector for each of 60 thousand words. Measured this way, LSA proved to be better than n-gram and RNN model.

III. PROPOSED METHODOLOGY

A. Recurrent Neural Network-

The system model that is to be implemented for sentence completion is explained below:

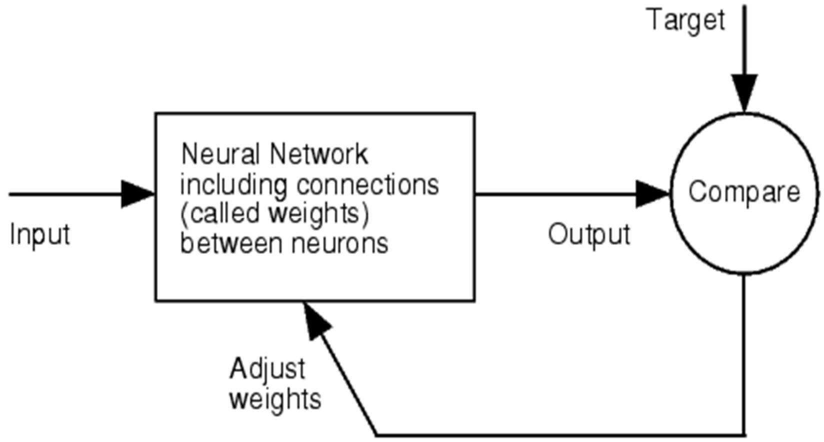


Figure 3.1: Basic Architecture for using RNN for sentence completion

For sequence classification problems, RNN is considered to be more effective because it is able to preserve important data from the precedent inputs and this data can be used to modify the current output. [5] Long Short-Term Memory is a RNN architecture that is used to address the problem of training over long sequences and preserving memory. LSTMs solve the gradient problem by introducing a few more gates that control access to the cell state. Recurrent Neural Network is used when your data is treated as a sequence, where the particular order of the data-points matter. For multi-layered neural networks when given a certain input, they tag the input as belonging to one of the many classes. They are trained using different algorithms. These networks do their jobs but their limitation lies in handling inputs which come in a sequence.

A lot of information is present in the context of the word which can only be determined by looking at the words near the given word. The entire sequence has to be studied to determine the output. Here Recurrent Neural Networks (RNNs) come to rescue. The output of every input becomes a part of next input for the next item of the sequence, as RNN traverses the input sequence.

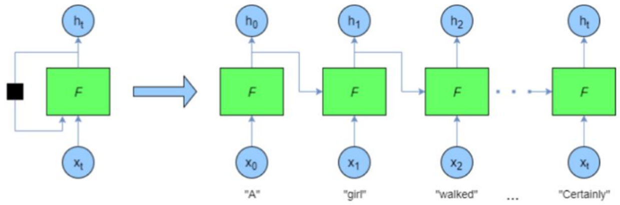


Figure 3.2 Basic structure of RNN for implementation

The network shows how one can supply a stream of continuous data to the recurrent neural network. For example, we, first, supply the word vector for "A" to the network *F* – the output of the nodes in *F* are fed into the "next" network and also acts as a stand-alone output (h0). The next network *F* at time *t=1* takes the next word vector for "girl" and the previous output h0 is stored into its hidden nodes, generating the next output (h1) and so on.

If we have sentence of m words, a language model allows us to predict the probability of observing the sentence (in a given dataset) as:

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \qquad \text{..........} \ (I)$$

In words, it uses conditional probability, the probability of a sentence is the product of probabilities of each word given the words that came before it. So, the probability of the sentence "She went to give all sweets" would be: the probability of "sweets" given "She went to give all", then multiplying with the probability of "all", and so on

B. Latent Semantic Analysis

Latent Semantic Analysis (LSA) which is applied to a large corpus of text is a method used to extract and represent the contextual meaning of words using statistical computations. [4] LSA is an automatic statistical technique used to extract and infer relations of contingent usage of words in passages. It is not any conventional Natural Language Processing (NLP) technique; it uses no knowledge bases, semantic networks, humanly constructed dictionaries, syntactic parsers, it takes input as a raw text parsed into uniquely character strings which separates into meaningful passages such as sentences. It takes into account the distributional hypothesis which states that words that are close in meaning will occur in similar pieces of text.

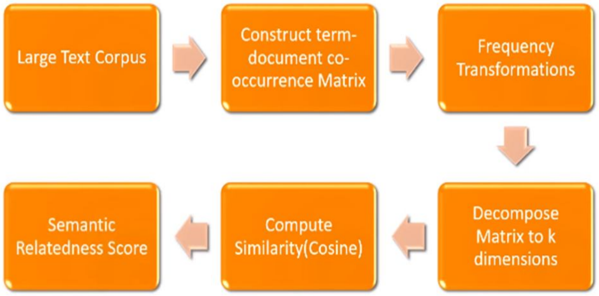

Large Text Corpus
Construct term-document co-occurrence Matrix
Frequency Transformations
Decompose Matrix to k dimensions
Compute Similarity(Cosine)
Semantic Relatedness Score

Figure 3.3: Steps in LSA

We start with large text Corpus including hundreds of documents in various areas of studies. We then represent this corpus as term document co-occurrence matrix. Then we perform certain frequency transformations on the matrix after which we decompose to k dimensions and compute similarities between terms and documents to get semantic relatedness score.

We use concepts of linear algebra for these two steps to be able to do computations on the corpus we use the vector space model or vsm which is an algebraic model for representing documents as vectors in the space where dictionary terms are used as dimensions. We express d documents in the space of T dictionary terms.

LSA represents the content as the matrix where each row corresponds to a different word while each column corresponds to a text passage. Each cell in the matrix contains the frequency of the word in its row which appeared in the passage is denoted by column. Next, the cell entries are subjected to a preliminary transformation, in which each cell frequency is weighted by a function that expresses the word's importance in the particular passage in addition to the degree to which the word type carries information in the domain of discourse.

LSA uses Singular Vector Decomposition (SVD) to generate the vectors of the particular text. The matrix X(term-document) is used for calculating two matrices:

$$Y = X_T X$$
$$Z = X X_T \qquad \text{................(II)}$$

Where: X: terms to the document matrix

Y: documents the document matrix

Z: terms to the term matrix

Thus, term document matrix, X, is divided into three matrices as follows

$$X = LSR_T \qquad \text{............(III)}$$

Where: L: Term for Concept weight matrix

RT: Concept to Document the weight matrix

S: Diagonal matrix which represents the concept weights

S is calculated by taking the square root of the eigenvalues of matrix Y.

IV. IMPLEMENTATION

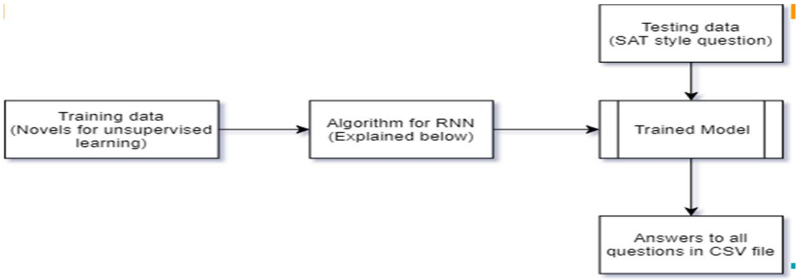


Figure 4.1: Proposed Methodology for Implementation

A. Dataset

The dataset used for Sentence Completion is the Microsoft Research Sentence Completion Challenge dataset. This dataset developed by Microsoft Research was publicly made available in 2012 (G. Zweig, C. Burges, 2012) in an effort to improve the research in the field of semantic and syntactic analysis. The dataset is based on five of Conan Doyle's Sherlock Holmes books that were a part of Project Gutenberg. We have used Holmes_Training_Data to train our model. It has 1040 sentence completion tasks based on the SAT question format. Each task has five options out of which only one is correct but all five fit likely well into the sentence. The training set contains 522 books from Project Gutenberg open corpus, each of them having adequate headers. We have a testing dataset which includes 1040 questions along with their five different options. Princeton review dataset with 11 practice test for SAT is used. Testing dataset is used to see how well the system is trained to be able to answer different unseen questions**.**

B. Preprocessing

To apply the different models to the Sentence Completion, pre-processing of dataset is required. Pre-processing in done in three steps, which involve removal of tab spaces, converting everything to lowercase and removal of punctuation and stop words.

C. Interface

A Desktop application is created for sentence completion for the user base - students, schools, colleges and examination centre. The interface is designed using C#. There are mainly two parts in the interface. The first one being "Upload the CSV file" where the user uploads the testing questions with five options each and the second one "Ask a question" where user can get an answer for the single question from the five available options. Both these tasks can be performed using RNN, LSA and both (RNN and LSA). Hence accuracy of these algorithms can be compared and evaluated.

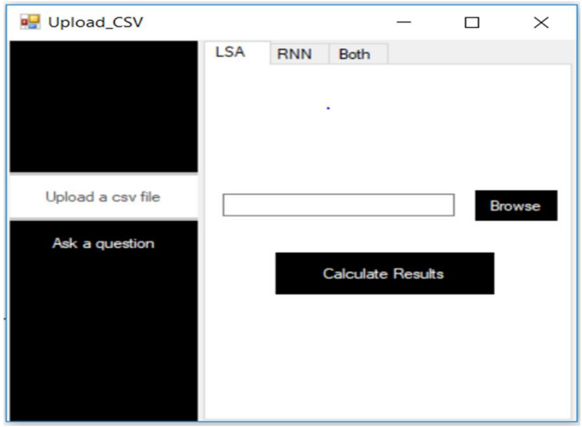


Figure 4.2: User Interface

All the models have been implemented in Python3 using Jupyter Notebooks and Colaboratory by Google Research. We've used Python libraries like NLTK, Gensim, SciPy, Numpy and Tensorflow.

For RNN, we first ran the code that contains all the functions required for training the files. Then, we actually implemented the RNN model. Training was carried out to extract the checkpoints. Then we carried out testing by providing input testing data to predict the right answers, which is then stored in the separate file.

For LSA, we first implemented a bag of words model to the corpus followed by the word vectors. Then, we computed the cosine similarity between the candidate answer and the question statement. Finally, we computed average cosine similarity for all the features and returned the option with the highest average cosine similarity.

V. EXPERIMENTAL RESULTS

As per the proposed system model, we first supplied 1024 test questions to the RNN model to predict one correct answer out of the five available options. Result shows that the accuracy of RNN model is 45.86%. Similarly, when the same questions were supplied to the LSA model, it was seen that LSA outperforms RNN with an improved accuracy of 51.08%.

VI. FUTURE WORK AND LIMITATIONS

Finding the dataset for the project was an important part as training and testing data should be similar in terms of vocabulary and the type of grammar and punctuations being used must resemble. Dataset we decided satisfies the above condition as the SAT papers use British English which is same as the training dataset. Training large dataset requires the use of GPU.

In future, we are planning to add a similarly sized dataset in accordance with Wikipedia, and also present the results found by asking human judges to perform the test. These human tests will be done in house which will allow us to provide additional statistics about the judges' backgrounds, for example, whether or not they are native-born English speakers, their level of education and so on.

VII. CONCLUSION

In this paper we have examined various methods that will predict the right answers for the given sentence-completion questions. These questions are appealing because they probe the ability to distinguish semantically coherent sentences from incoherent ones, and yet involve no more context than the single sentence. We have solved the problem by implementing the approaches of namely, Recurrent Neural Networks and Latent Semantic Analysis. The accuracy achieved by LSA was considerably more than RNN.

VIII. ACKNOWLEDGEMENT

We would like to thank all the members who helped investing their time on our project, and also we would like to express gratitude to the entire 'anonymous' expert for their insights on our study work. We would like to extend our heartfelt gratitude to our mentor Prof Nirmala Shinde for guiding us throughout our project work.

REFERENCES

[1] Geoffrey Zweig, John C. Platt Christopher Meek Christopher J.C. Burges,Ainur Yessenalina ,Qiang Liu. Computational Approaches to Sentence Completion, 2012.

[2] Aubrie M. Woods Carnegie Mellon University. Exploiting Linguistic Features for Sentence Completion,2016.

[3] Thomas K Landauer, Peter W. Foltz , Darrell Laham. An Introduction to Latent Semantic Analysis,1998.

[4] Joseph Gubbins, Andreas Vlachos.Dependency Language models for sentence completion,2013.

[5] Understanding RNN model. https://www.youtube.com/watch?v=Keqep_PKrY8

[6] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space,2013.

[7] Geoffrey Zweig and Christopher J.C. Burges. The Microsoft Research Sentence Completion Challenge, 2011.