

PROFESSIONAL TRAINING REPORT

At

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in
Information Technology

By

MUTHUMULA SREELATHA

(REG.NO 37120050)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119.

JANUARY 2021

DECLARATION

I **MUTHUMULA SREELATHA** (Reg. No. 37120050) here by declare that the Project Report entitled “RETROSPECTIVE READER FOR MACHINE READING COMPREHENSION” done by me under the guidance of **DR. JEBERSON RETNA RAJ**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE :

PLACE :

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management Of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, Dean, School of Computing ,

Dr. SUBHASHINI M.E., Ph.D., Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. JEBERSON RETNA RAJ., for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of project

Training Certificate



☎ 040-23710566, 040-29560566
✉ rk@vblptechsolutions.com
📍 SAI RANGA APARTMENT, #G-01,
B-17, SR NAGAR, MADHURA NAGAR,
HYDERABAD, TELANGANA, 500038.

CERTIFICATE OF COMPLETION

Date : 13th Dec, 2019

Dear Ms. Muthumula Sreelatha,

This is to certify that **Muthumula Sreelatha** of **Sathyabhama Institute of Science and Technology** has successfully completed her internship at **VBLP Tech Solutions Pvt. Ltd.** from **3rd Dec, 2019** to **3rd Jan, 2020**.

During the Internship, she worked on the project entitled **Object Detection Using Python**

She was found punctual, hardworking and interested to learn the technologies. During the internship she demonstrated good skills with self—motivative attitude towards learning.

Her association with the team was fruitful. We wish her all the best for future!

From


VBLP Tech Solutions Pvt. Ltd.
V Ramakrishna
Managing Director
Ph: 040-23710566/765995858



www.vblptechsolutions.com

Deep Learning



07/05/2020

Muthumula Sreelatha

has successfully completed the online, non-credit Specialization

Deep Learning

The Deep Learning Specialization is designed to prepare learners to participate in the development of cutting-edge AI technology, and to understand the capability, the challenges, and the consequences of the rise of deep learning. Through five interconnected courses, learners develop a profound knowledge of the hottest AI algorithms, mastering deep learning from its foundations (neural networks) to its industry applications (Computer Vision, Natural Language Processing, Speech Recognition, etc.).


Adjunct Professor
Andrew Ng
Computer Science

The online specialization earned in this certificate may draw on material from courses taught on campus, but the recorded courses are not equivalent to on-campus courses. Participation in this online specialization does not constitute an endorsement by the university. This certificate does not confer a University grade, course credit or degree, and it does not verify the identity of the signer.

Verify this certificate at:
coursera.org/verify/specialization/FQ18F7Y6EYE



5 Courses

How Google does Machine Learning

Launching into Machine Learning

Introduction to TensorFlow

Feature Engineering

Art and Science of Machine Learning

Google Cloud

May 26, 2020

Muthumula Sreelatha

has successfully completed the online, non-credit Specialization

Machine Learning with TensorFlow on Google Cloud Platform

This five-course online specialization teaches course participants how to write distributed machine learning models that scale in Tensorflow, scale out the training of those models, and offer high-performance predictions. Also featured is the conversion of raw data to features in a way that allows ML to learn important characteristics from the data and bring human insight to bear on the problem. It also teaches how to incorporate the right mix of parameters that yields accurate, generalized models and knowledge of the theory to solve specific types of ML problems. Course participants experimented with end-to-end ML, starting from building an ML-focused strategy and progressing into model training, optimization, and productionalization with hands-on labs using Google Cloud Platform.

The online specialization named in this certificate may draw on material from courses taught on-campus, but the included courses are not equivalent to on-campus courses. Participation in this online specialization does not constitute enrollment at this university. This certificate does not confer a University grade, course credit or degree, and it does not verify the identity of the learner.

Google Cloud Training

Verify this certificate at:
coursera.org/verify/specialization/5UZF9YGFNT

Roll No: NPTEL19C50755122346

To
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
CHENNAI

119/128



No. of credits recommended by NPTEL:3

Score	Type of Certificate
≥ 90	Elite+Gold
75-89	Elite+Silver
≥ 60	Elite
40-59	Successfully completed the course
< 40	No Certificate



Elite

NPTEL Online Certification

(Funded by the Ministry of HRD, Govt. of India)

This certificate is awarded to

M.SREELATHA

for successfully completing the course

Programming in Java

with a consolidated score of **75** %

Online Assignments	19.16/25	Proctored Exam	55.5/75
--------------------	----------	----------------	---------

Total number of candidates certified in this course: **8377**



Jan-Apr 2019
(12 week course)

A. Goswami
Prof. Adrijit Goswami
Dean, Continuing Education & NPTEL Coordinator
IIT Kharagpur



Indian Institute of Technology Kharagpur



Roll No: NPTEL19C50755122346

To validate and check scores: <http://nptel.ac.in/npoc>

Certificate of Completion

**This is to certify that *Sreelatha Reddy*
successfully completed 18 total hours of *The
Complete Python 3 Course: Beginner to Advanced!*
online course on June 12, 2020**

Joseph Delgadillo

Joseph Delgadillo, Instructor

Nick Germaine

Nick Germaine, Instructor

&



Certificate no: UC-4f05b2da-2858-4a33-a8c2-0f3457f2fa98
Certificate url: ude.my/UC-4f05b2da-2858-4a33-a8c2-0f3457f2fa98

#BeAble



Certificate of Completion
Congratulations, Sreelatha M

Advance Your Skills in Deep Learning and Neural Networks

Learning Path completed on May 28, 2020 • 16 hours 30 min

By continuing to learn, you have expanded your perspective, sharpened your skills, and made yourself even more in demand.

Top skills covered

Natural Language Processing (NLP), TensorFlow, Deep Learning, Neural Networks, Machine Learning, Artificial Intelligence (AI), JavaScript, Python (Programming Language), Amazon Web Services (AWS)

VP, Learning Content at LinkedIn

LinkedIn Learning
1000 W Maude Ave
Sunnyvale, CA 94085

Certificate id: AfZV6lcknWNYPc6sFjF40hmEJxEz



 LEARNING

Certificate of Completion
Congratulations, Sreelatha M

Become a Machine Learning Specialist

Learning Path completed on May 27, 2020 • 17 hours 18 min

By continuing to learn, you have expanded your perspective, sharpened your skills, and made yourself even more in demand.

Top skills covered

Data Science

VP, Learning Content at LinkedIn

LinkedIn Learning
1000 W Maude Ave
Sunnyvale, CA 94085

Certificate Id: Afu3PhKGX4YmDhx_86nhRHr3IP2



Credential id:



roBSsCLpxG

PROFESSIONAL — CERTIFICATE —

awarded to

Muthumula Sreelatha

has successfully completed course in

Machine Learning & Artificial Intelligence

offered by Amity Future Academy

July 13, 2020

DIRECTOR
AMITY FUTURE ACADEMY

Verify at learningweek.amityfutureacademy.com/mod/customcert/verify_certificate.php by entering credential id

ABSTRACT

Machine reading comprehension(MRC) is an AI challenge that requires machine to determine the correct answer to question based on given passage.

Stanford Question Answering Dataset(SQuAD) is a reading comprehension dataset. The dataset contains 24 layers, 16 attention heads & 340 millions parameters. BERT is designed to pre-train deep bidirectional representations from unlabelled text by jointly conditioning on both left & right context in all layers. BERT is basically a trained Transformer Encoder stack, with twelve in the Base version and twenty-four in the Large version, compared to 6 encoder layers in the original Transformer. BERT encoders have larger feedforward networks(768 and 1024 nodes in Base and Large respectively) and more attention heads(12 and 16 respectively). BERT was trained on Wikipedia and Books Corpus, a dataset containing +10,000 books of different genres. The pre-trained model can then be fine-tuned on small-data NLP tasks like question answering and semantic analysis resulting in substantial accuracy improvements compared to training on these datasets from scratch. The rapid growth of the development of Internet-based information systems increases the demand for natural language interface that are easy to set up and maintain the relevant answers to the questions.

TABLE OF CONTENTS

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGENO
	ABSTRACT	vi
	LIST OF FIGURES	Vii
1.	INTRODUCTION	1
	1.1 FEATURES OF PYTHON	1
	1.2FEATURES OF MACHINE LEARNIN	2
	1.3 DEEP LEARNING	3
	1.4 NATURAL LANGUAGE PROCESSING	4
	1.5 OUTLINE OF PROJECT	5
	1.6 LITERATURE REVIEW	6
2.	AIM AND SCOPE	8
	2.1 AIM	9
	2.2 INPUTS AND OUTPUTS	10
	2.2 SCOPE	11
3.	EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMS USED	12
	3.1 ALGORITHMS USED	

4.	REQUIREMENT SPECIFICATION	22
	4.1 SOFTWARE REQUIREMENTS	22
	4.2 HARDWARE REQUIREMENTS	23
5.	PROJECT DESCRIPTION	24
	5.1 FLOW CHART	24
	5.2 ALGORITHMS	28
6.	RESULTS AND DISCUSSION	30
	6.1 RESULTS	31
	6.1.1 SCREENSHOTS	32
7.	CONCLUSION AND FUTURE WORK	40
	7.1 CONCLUSION	40
	7.2 FUTURE WORK	41
	REFERENCES	42

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.3.1	ARCHITECTURE OF DEEP LEARNING	3
4.1	COSINE SIMILARITY DIAGRAM	13
4.1.2	WORD2VEC DIAGRAM	15
4.1.3	SKIP GRAM DIAGRAM	16
5.1	BASIC ATTENTION VISUALIZATION	18
5.1.2	TRANSFORMER ENCODER & DECODER	20
5.1.3	INPUT EMBEDDING DIAGRAM	21
5.1.4	BERT TRANSFORMER DIGRAM	22
6.1	F1 SCORE	23
6.1.2	BAR PLOT	24
6.1.3	OUTPUT SCREENSHOTS	25

CHAPTER 1

INTRODUCTION

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

FEATURES OF PYTHON

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamic typed language.

1.Easy to code: Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

2.Free and Open Source: Python language is freely available at official website and you can download it from the given download link below click on the Download python keyword. Download Python Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3.Object-OrientedLanguage: One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4.GUI Programming Support: Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wx Python or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature: Python is a Extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language: Python language is also a portable language, for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language: Python is also an Integrated language because we can easily integrated python with other language like c,c++ etc.

9. Interpreted Language: Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library: Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language: Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don't need to specify the type of variable

1.2 MACHINE LEARNING

Introduction to Machine Learning: Machine learning is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning beings with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

1.3 DEEP LEARNING

Introduction to Deep Learning: There are different ways to define deep learning but in general, without becoming too technical, deep learning is a class of machine learning algorithms that can be applied to the world of structure or unstructured data.

If we focus on unstructured information, we usually consider text mining as the typical task performed by deep learning solution. There are several algorithms in the Machine Learning world for performing text mining or, in other words, extracting knowledge from text, but they all share a similar core of statistics (distribution and frequency of keywords) and cooccurrences. The three most common applications of text mining solutions are automatic categorization, entity and concept extraction and natural language question answering.

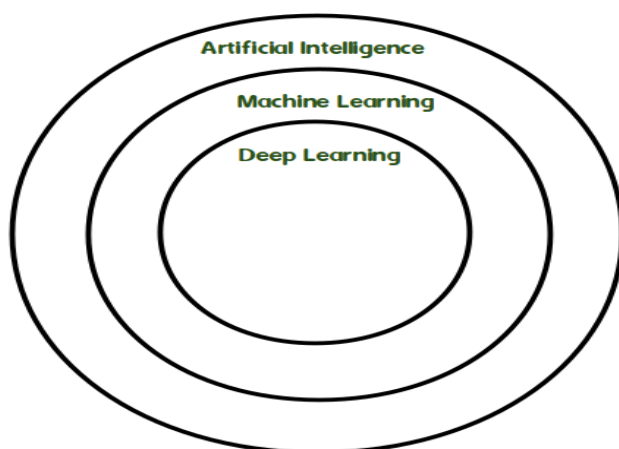


Fig 1.3.1 Architecture of Deep Learning

1.4 Natural Language Processing (NLP)

Introduction to NLP: NLP is the relationship between computers and human language. More specifically, natural language processing is the computer understanding, analysis, manipulation, and/or generation of natural language (according to dictionary.com).

Natural language refers to speech analysis in both audible speech, as well as text of a language. NLP systems capture meaning from an input of words (sentences, paragraphs, page, etc.) in the form a structured output (which varies greatly depending on the application). Natural language processing is a fundamental element of artificial intelligence.

Natural language processing, however, is more than just speech analysis. There are a variety of approaches for processing human language. This include:

Symbolic Approach: The symbolic approach to natural language processing is based on human-developed rules and lexicons. In other words, the basis behind this approach is in generally accepted rules of speech within a given language which are materialized and recorded by linguistic experts for computer systems to follow.

Statistical Approach: The statistical approach o natural language processing is based on observable and recurring examples of linguistic phenomena. Models based on statistics recognize recurring examples of linguistic phenomena. Models based on statistics recognize recurring themes through mathematical analysis of large text corpora. By identifying trends in large samples of text the computer system can develop its own linguistic rules that it will use to analyze future input and/or the generation of language output.

Connectionist Approach: The connectionist approach to natural language processing is a combination of the symbolic and statistical approaches. This approach starts with generally accepted rules of language and tailors them to specific application from input derived from statistical inference.

INTRODUCTION TO BERT TRANSFORMER

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task architecture.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), Multi NLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

The ultimate goal of natural language processing is for computers to achieve human-like comprehension of texts/language. When this is achieved, computer systems will be able to understand, draw inferences from, summarize, translate and generate accurate and natural human text and language.

1.5 OUTLINE OF PROJECT

The rapid growth of the development of Internet-based information systems increases the demand for natural language interface that are easy to set up and maintain.

The main goal of this project is to find the text for any question and answer by context provided. This is a closed dataset meaning that the answer to a question is always a part of the context and also a continuous span of context by training machine using predefined dataset of question and answers using Tensor flow, torch, huge face transformers, regular expression, identifiers, tokenizers, Word2vec, bag of words, glove embeddings, Bert transformer model and deep learning algorithms.

1.5 LITERATURE REVIEW

As the project determines is to find the text for any new question and context provided. This is a closed dataset meaning that the answer to a question is always a part of the context and also a continuous span of context by using SQUAT dataset. We use deep learning techniques and Tensor flow, torch, huge face transformers, regular expression, identifiers, tokenizers, Word2vec, bag of words, glove embeddings, Bert transformer model to train for future analysis of question answering for a passage/text/paragraph/articles.

STUDY OF THE SYSTEM

- 1.Numpy
- 2.Pandas
- 3.Matplotlib
- 4.Scikit –learn

1.Numpy: is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

2.Pandas: Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical

steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

3.Matplotlib: Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object -oriented interface or via a set of functions familiar to MATLAB users.

4.Scikit –learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

NumPy: Base n-dimensional array package

SciPy: Fundamental library for scientific computing

Matplotlib: Comprehensive 2D/3D plotting

IPython: Enhanced interactive console

Sympy: Symbolic mathematics

Pandas: Data structures and analysis

Extensions or modules for SciPy are conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn.

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The main aim of the project is to find the text for any question and answer by context provided.

SQuAD Dataset

Stanford Question Answering Dataset (SQuAD) is a new reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage. With 100,000+ question-answer pairs on 500+ articles, SQuAD is significantly larger than previous reading comprehension datasets.

There has been a rapid progress on the SQuAD dataset with some of the latest models achieving human level accuracy in the task of question answering!

Examples of context, question and answer on SQuAD

Context — Apollo ran from 1961 to 1972, and was supported by the two-man Gemini program which ran concurrently with it from 1962 to 1966. Gemini missions developed some of the space travel techniques that were necessary for the success of the Apollo missions. Apollo used Saturn family rockets as launch vehicles. Apollo/Saturn vehicles were also used for an Apollo Applications Program, which consisted of Skylab, a space station that supported three manned missions in 1973–74, and the Apollo–Soyuz Test Project, a joint Earth orbit mission with the Soviet Union in 1975.

Question — What space station supported three manned missions in 1973–1974?

Answer — Skylab

Key Features of SQuAD:

- i) It is a closed dataset meaning that the answer to a question is always a part of the context and also a continuous span of context
- ii) So the problem of finding an answer can be simplified as finding the start index and the end index of the context that corresponds to the answers
- iii) 75% of answers are less than equal to 4 words long

2.2 INPUT AND OUTPUTS

The following some are the project inputs and outputs

INPUTS:

Importing the all required packages like tensorflow, huggingface Pytorch, transformers, tokenizers, identifiers, word2vec, glove embeddings, bert transformer, numpy, pandas, matplotlib, scikit-learn, pprint, and required deep learning algorithms packages

This dataset contains 24 layers, 16 attention heads and 340 millions parameters.

OUTPUTS:

Preprocseing the importing data frame for imputing nulls with the related information. All are displaying cleaned outputs.

After applying deep learning algorithms tensorflow, transformers, torch it will give good results and visualization plots.

SCOPE:

The rapid growth of the development of Internet-based information systems increases the demand for natural language interface that are easy to set up and maintain the relevant answers to the questions.

One of the biggest challenging Natural Language Processing (NLP) is the shortage of training data. Because NLP is a diversified field with many distinct tasks, most task-specific datasets contain only a few thousand or a few hundred thousand human-labelled training examples. However, modern deep learning-based NLP models see benefits from much larger amounts of data, improving when trained on millions, or *billions*, of annotated training examples. To help close this gap in data, researchers have developed a variety of techniques for training general purpose language representation models using the enormous amount of unannotated text on the web (known as *pre-training*). The pre-trained model can then be *fine-tuned* on small-data NLP tasks like question answering and semantic analysis resulting in substantial accuracy improvements compared to training on these datasets from scratch.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMS USED

3.1 ALGORITHMS USED:

a) TensorFlow 2.0: TensorFlow 2.0 is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- a) Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- b) Computing the gradient of arbitrary differentiable expressions.
- c) Scaling computation to many devices (e.g. the Summit supercomputer at Oak Ridge National Lab, which spans 27,000 GPUs).
- d) Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

b) Torch: Torch is an open source machine learning library, a scientific computing framework, and a script language based on the Lua programming language. It provides a wide range of algorithms for deep learning, and uses the scripting language and an underlying C implementation.

c)Regular Expression: A *regular expression* is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. Regular expressions are widely used in UNIX world.

The Python module `re` provides full support for Perl-like regular expressions in Python. The `re` module raises the exception `re.error` if an error occurs while compiling or using a regular expression.

d) Cosine Similarity: Cosine similarity is a metric used to determine how similar the documents are irrespective of their size.

Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. In this context, the two vectors I am talking about are arrays containing the word counts of two documents.

Example

Let's suppose you have 3 documents based on a couple of star cricket players Sachin Tendulkar and Dhoni. Two of the documents (A) and (B) are from the wikipedia pages on the respective players and the third document (C) is a smaller snippet from Dhoni's wikipedia page.

The Three Documents and Similarity Metrics



Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

Doc Sachin: Wiki page on Sachin Tendulkar	Doc Dhoni: Wiki page on Dhoni	Doc Dhoni_Small: Subsection of wiki on Dhoni
Dhoni - 10	Dhoni - 400	Dhoni - 10
Cricket - 50	Cricket - 100	Cricket - 5
Sachin - 200	Sachin - 20	Sachin - 1

Document - Term Matrix (Word Counts)				Similarity Metrics			
Word Counts	"Dhoni"	"Cricket"	"Sachin"	Similarity or Distance Metrics	Total Common Words	Euclidean distance	Cosine Similarity
Doc Sachin	10	50	200	Doc Sachin & Doc Dhoni	$10 + 50 + 10 = 70$	432.4	0.15
Doc Dhoni	400	100	20	Doc Dhoni & Doc Dhoni_Small	$20 + 10 + 7 = 37$	204.0	0.23
Doc Dhoni_Small	10	5	1	Doc Sachin & Doc Dhoni_Small	$10 + 10 + 7 = 27$	401.85	0.77

Fig 3.1.1 Cosine similarity diagram

- e) **Word Embedding: Word Embeddings** is a language modelling technique used for mapping words to vectors of real numbers. It represents words or phrases in vector space with several dimensions. Word embeddings can be generated using various methods like neural networks, co-occurrence matrix, probabilistic models, etc.

Word2Vec consists of models for generating word embedding. These models are shallow two layer neural networks having one input layer, one hidden layer and one output layer. Word2Vec utilizes two architectures.

1)CBOW (Continuous Bag of Words): CBOW model predicts the current word given context words within specific window. The input layer contains the context words and the output layer contains the current word. The hidden layer contains the number of dimensions in which we want to represent current word present at the output layer.

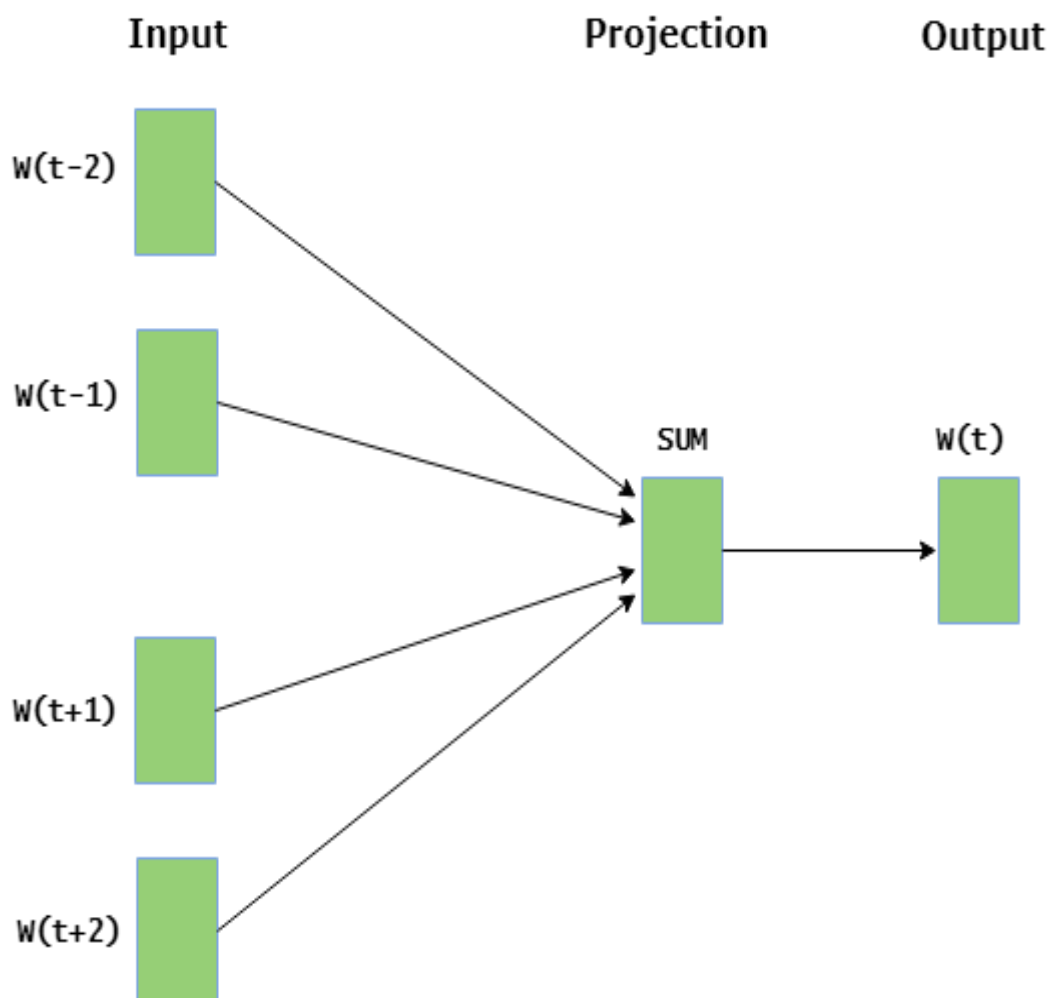


Fig 3.1.1.2 Word2vec diagram

2) SKIP GRAM Model: Skip gram predicts the surrounding context words within specific window given current word. The input layer contains the current word and the output layer contains the context words. The hidden layer contains the number of dimensions in which we want to represent current word present at the input layer.

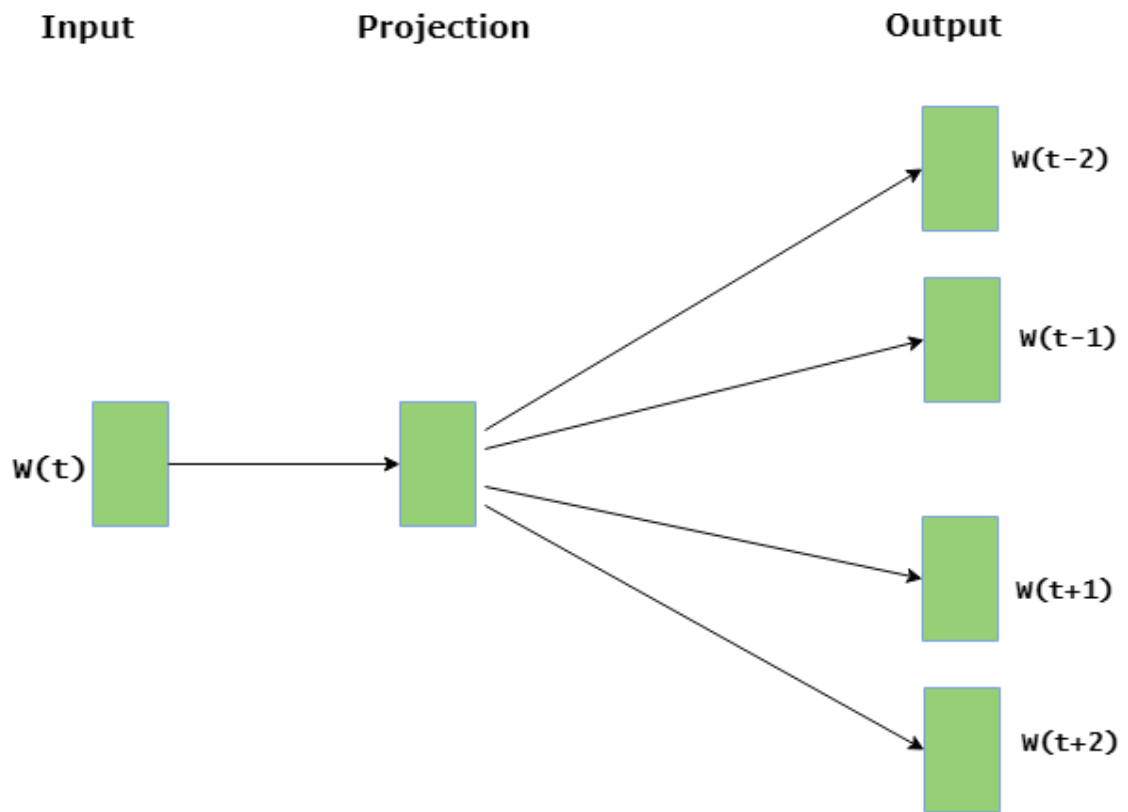


Fig 3.1.1.3 Skip gram diagram

3) GLOVE Embedding: Global vector for word representation is an unsupervised machine learning for obtaining vector representation words. Simply put, GloVe allows us to take a corpus of text, and intuitively transform each word in that corpus into a position in a high-dimensional space. This means that similar words will be placed together.

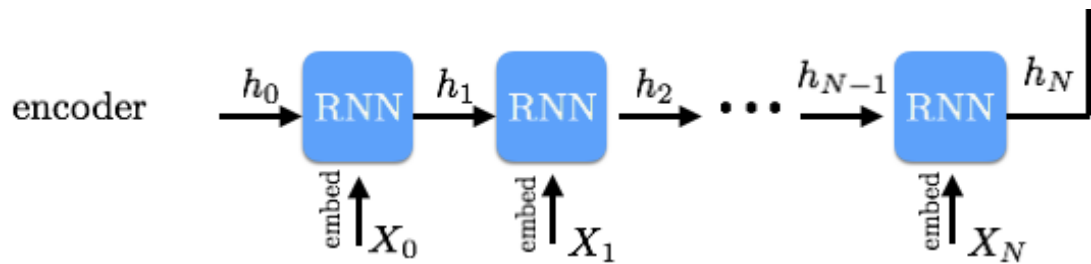
f) BERT Model:

Machine Comprehension Model- Key Components

i) Embedding Layer

The training dataset for the model consists of context and corresponding questions. Both of these can be broken into individual words and then these words converted into Word Embeddings using pretrained vector like GLoVe vectors.. Word Embeddings are much better at capturing the context around the words than using a one hot vector for every word. For this problem I used 100 dimension GloVe word embeddings and didn't tune them during the training process since we didn't have sufficient data.

ii) Encoder Layer

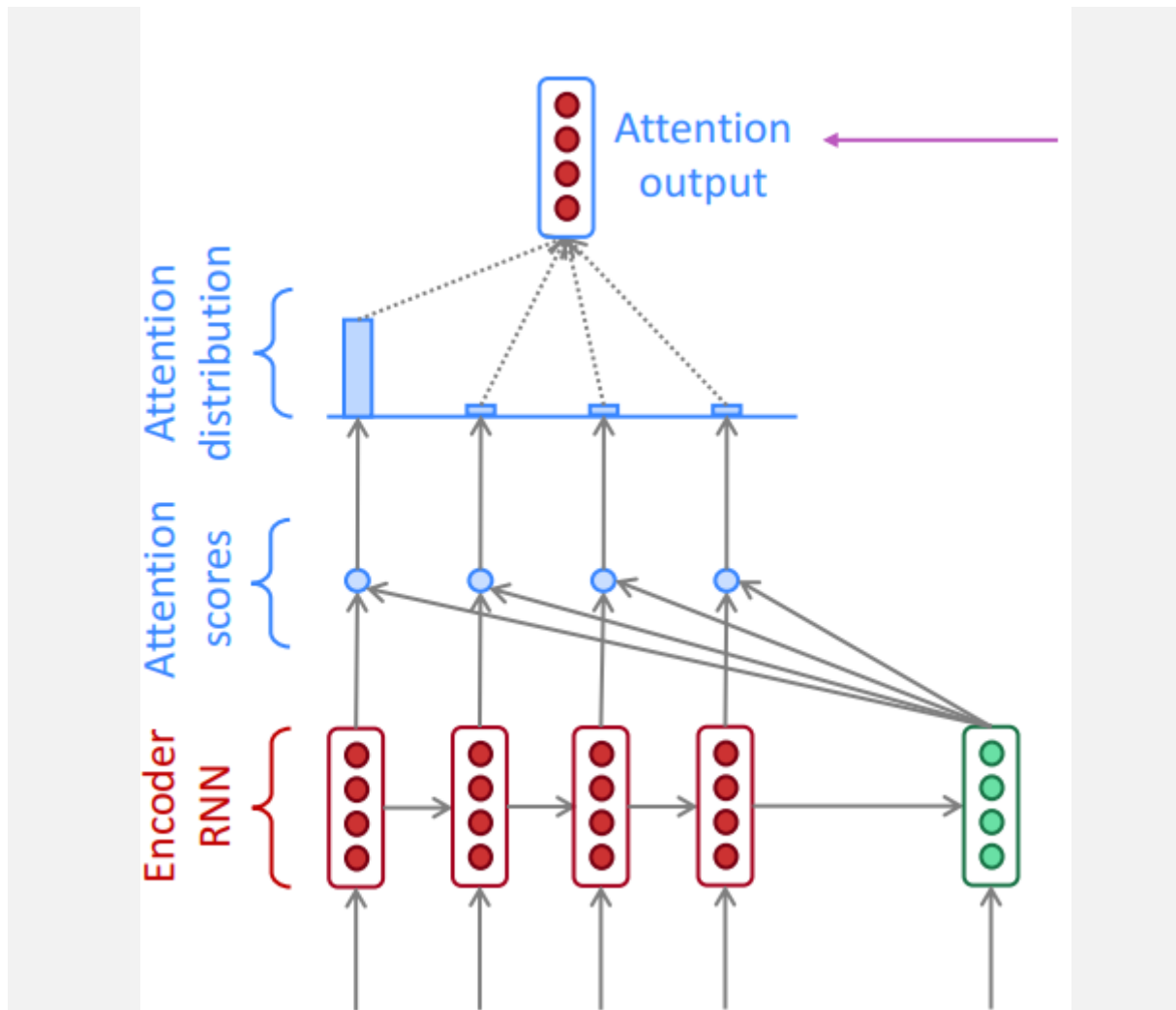


The next layer we add in the model is a RNN based Encoder layer. We would like each word in the context to be aware of words before it and after it. A bi-directional GRU/LSTM can help do that. The output of the RNN is a series of hidden vectors in the forward and backward direction and we concatenate them. Similarly we can use the same RNN Encoder to create question hidden vectors.

iii) Attention Layer

Up til now we have a hidden vector for context and a hidden vector for question. To figure out the answer we need to look at the two together. This is where attention comes in. It is the key component in the Question Answering system since it helps us decide, given the question which words in the context should I "attend" to. Lets start with the simplest possible attention model:

Dot product attention



Basic Attention Visualisation

The dot product attention would be that for each context vector c_i we multiply each question vector q_j to get vector e_i (attention scores in the figure above). Then we take a softmax over e_i to get α_i (attention distribution in the figure above). Softmax ensures that the sum of all e_i is 1. Finally we calculate a_i as the product of the attention

distribution α_i and the corresponding question vector (attention output in the figure above). Dot product attention is also described in the equations below

$$\mathbf{e}^i = [c_i^T q_1, \dots, c_i^T q_M] \in \mathbb{R}^M$$

$$\alpha^i = \text{softmax}(\mathbf{e}^i) \in \mathbb{R}^M$$

$$\mathbf{a}_i = \sum_{j=1}^M \alpha_j^i q_j \in \mathbb{R}^{2h}$$

More Complex Attention- BiDAF Attention

We can run the SQuAD model with the basic attention layer described above but the performance would not be good. More complex attention leads to much better performance.

The main idea is that attention should flow both ways — from the context to the question and from the question to the context.

We first compute the similarity matrix $S \in \mathbb{R}^{N \times M}$, which contains a similarity score S_{ij} for each pair (c_i, q_j) of context and question hidden states. $S_{ij} = w^T \text{sim}[c_i; q_j; c_i \circ q_j]$ $\in \mathbb{R}$ Here, $c_i \circ q_j$ is an elementwise product and $w_{\text{sim}} \in \mathbb{R}^{6h}$ is a weight vector.

$$S_{ij} = w_{\text{sim}}^T [c_i; q_j; c_i \circ q_j] \in \mathbb{R}$$

we perform Context-to-Question (C2Q) Attention. (This is similar to the dot product attention described above). We take the row-wise softmax of S to obtain attention distributions α_i , which we use to take weighted sums of the question hidden states q_j , yielding C2Q attention outputs $\mathbf{a}_i$.

$$\alpha^i = \text{softmax}(S_{i,:}) \in \mathbb{R}^M \quad \forall i \in \{1, \dots, N\}$$

$$a_i = \sum_{j=1}^M \alpha_j^i q_j \in \mathbb{R}^{2h} \quad \forall i \in \{1, \dots, N\}$$

we perform Question-to-Context(Q2C) Attention. For each context location $i \in \{1, \dots, N\}$, we take the max of the corresponding row of the similarity matrix, $m_i = \max_j S_{ij} \in \mathbb{R}$. Then we take the softmax over the resulting vector $m \in \mathbb{R}^N$ — this gives us an attention distribution $\beta \in \mathbb{R}^N$ over context locations. We then use β to take a weighted sum of the context hidden states c_i — this is the Q2C attention output c' .

$$m_i = \max_j S_{ij} \in \mathbb{R} \quad \forall i \in \{1, \dots, N\}$$

$$\beta = \text{softmax}(m) \in \mathbb{R}^N$$

$$c' = \sum_{i=1}^N \beta_i c_i \in \mathbb{R}^{2h}$$

Finally for each context position c_i we combine the output from C2Q attention and Q2C attention as described

$$b_i = [c_i; a_i; c_i \circ a_i; c_i \circ c'] \in \mathbb{R}^{8h} \quad \forall i \in \{1, \dots, N\}$$

where $\circ$ represents elementwise multiplication.

iv) Output Layer

The final layer of the model is a softmax output layer that helps us decide the start and the end index for the answer span. We combine the context hidden states and the attention vector from the previous layer to create blended reps. These blended reps become the input to a fully connected layer which uses softmax to create a p_{start} vector with probability for start index and a p_{end} vector with probability for end index. Since we know that most answers start and end index are max 15 words apart, we can look for start and end index that maximize $p_{\text{start}} * p_{\text{end}}$

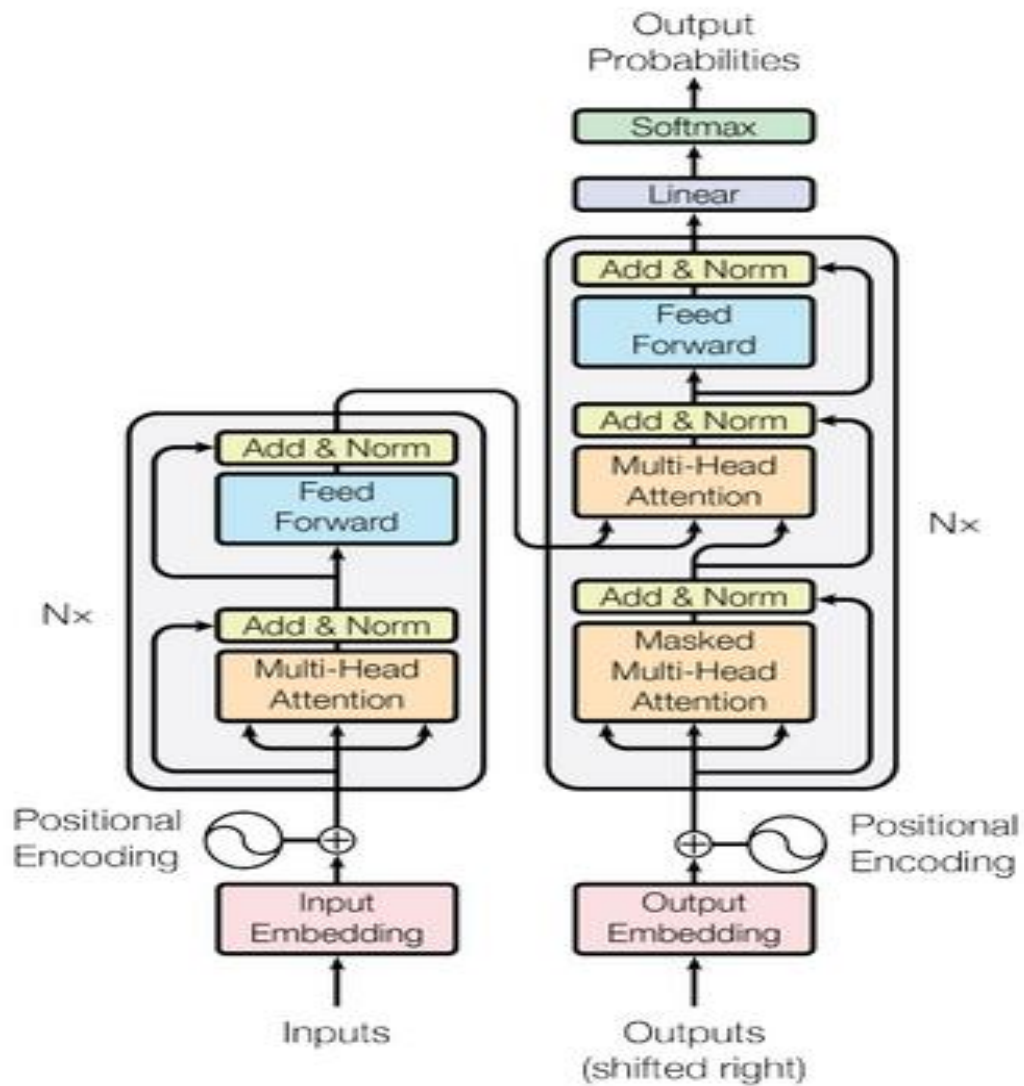


Fig 3.1.1.4 Transformer Encoder and Decoder diagram

g) **Input Representation:** BERT can handle single sentences or pairs of sentences with a one token sequence. The input representation for a given token is constructed by summing over three different embeddings, corresponding to the token, segment, and position

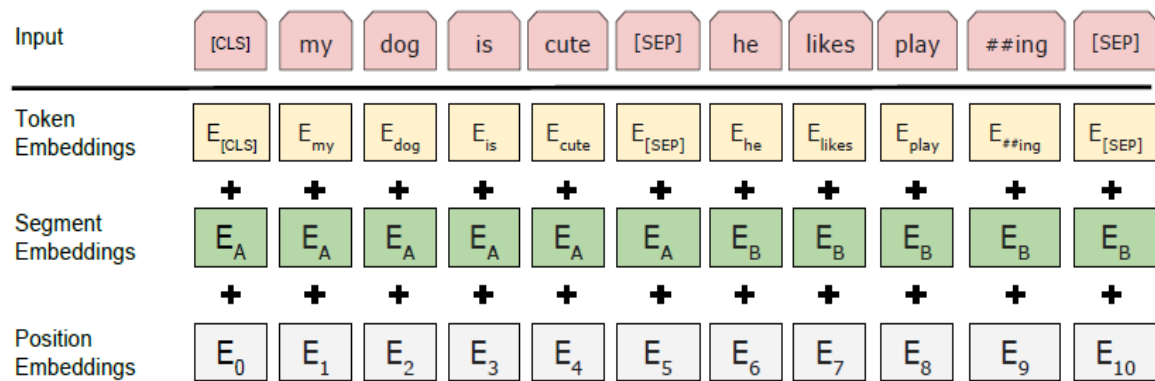
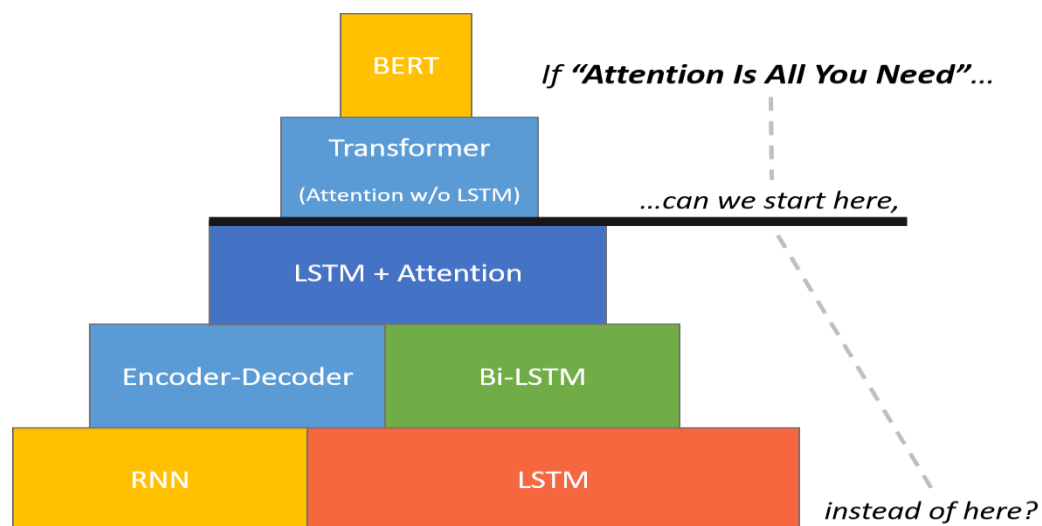


Fig 3.1.1.5 Input Embedding diagram

h) BERT Transformer: BERT is basically a trained Transformer Encoder stack, with twelve in the Base version, and twenty-four in the Large version, compared to 6 encoder layers in the original Transformer.

BERT encoders have larger feedforward networks (768 and 1024 nodes in Base and Large respectively) and more attention heads (12 and 16 respectively). BERT was trained on Wikipedia and Book Corpus, a dataset containing +10,000 books of different genres.



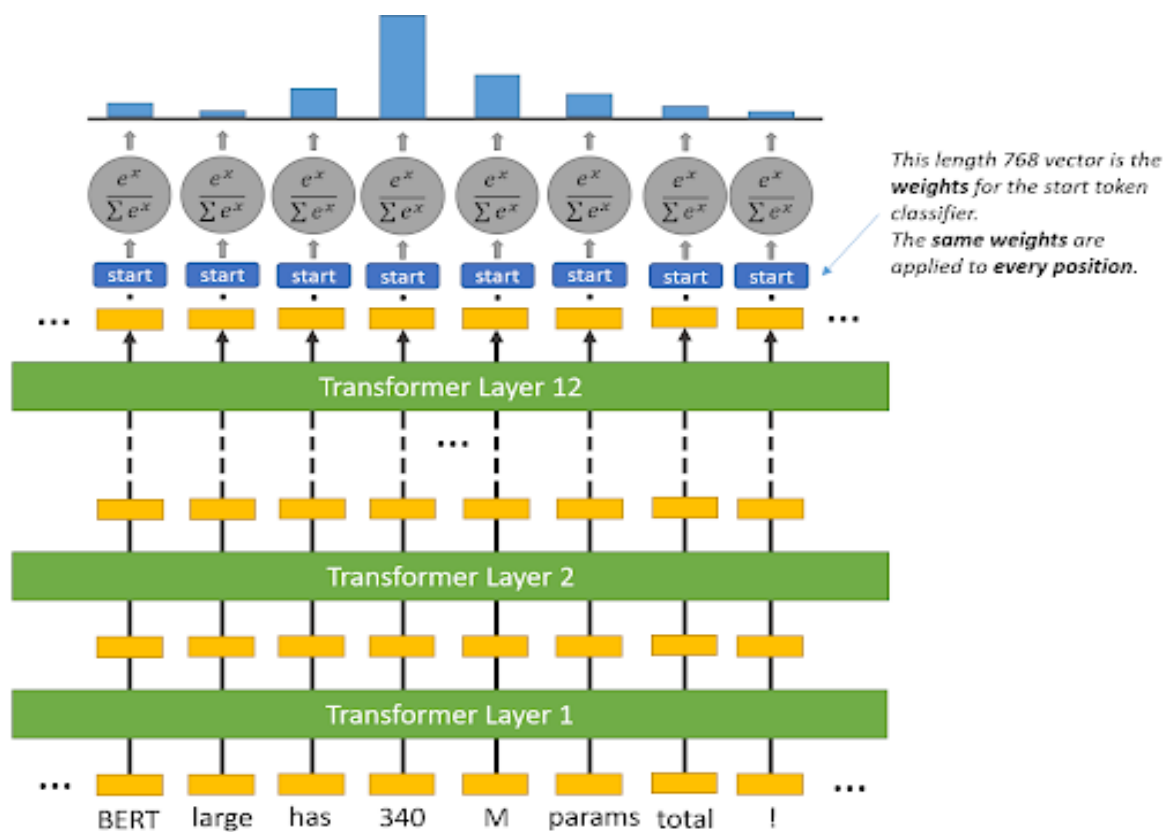


Fig 3.1.1.7 BERT Transformer diagram

F1 Score:

```

new_predictions[uuid] = final_answer
return new_predictions

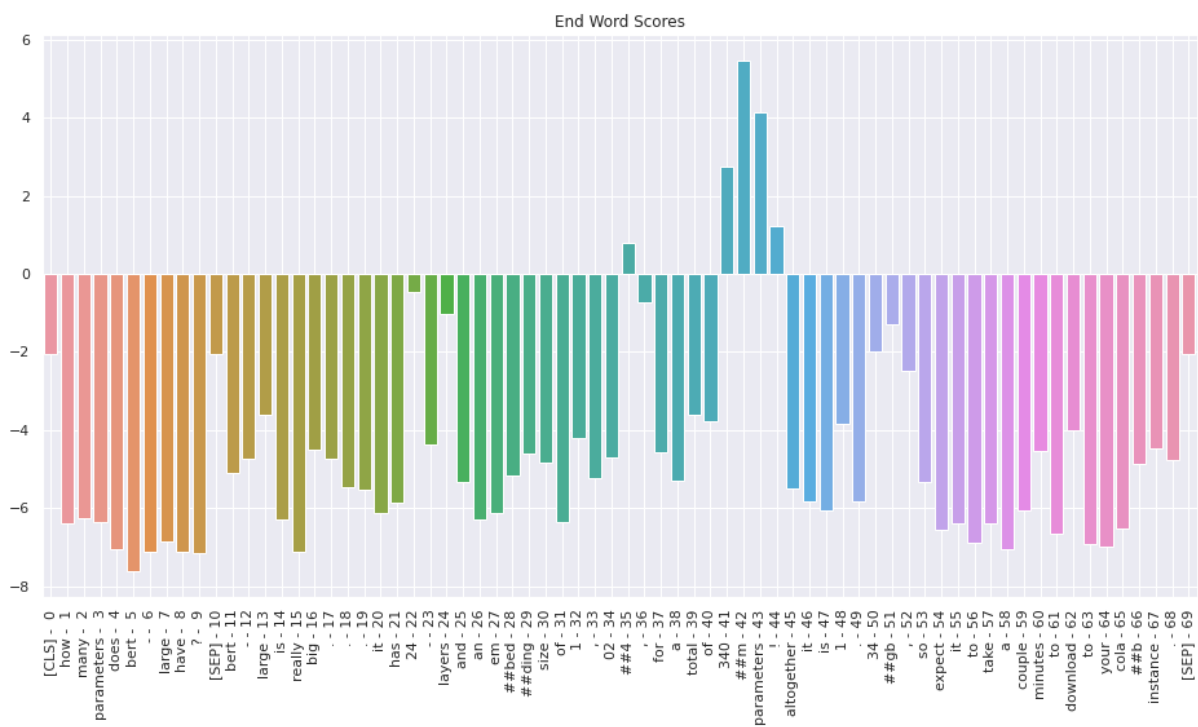
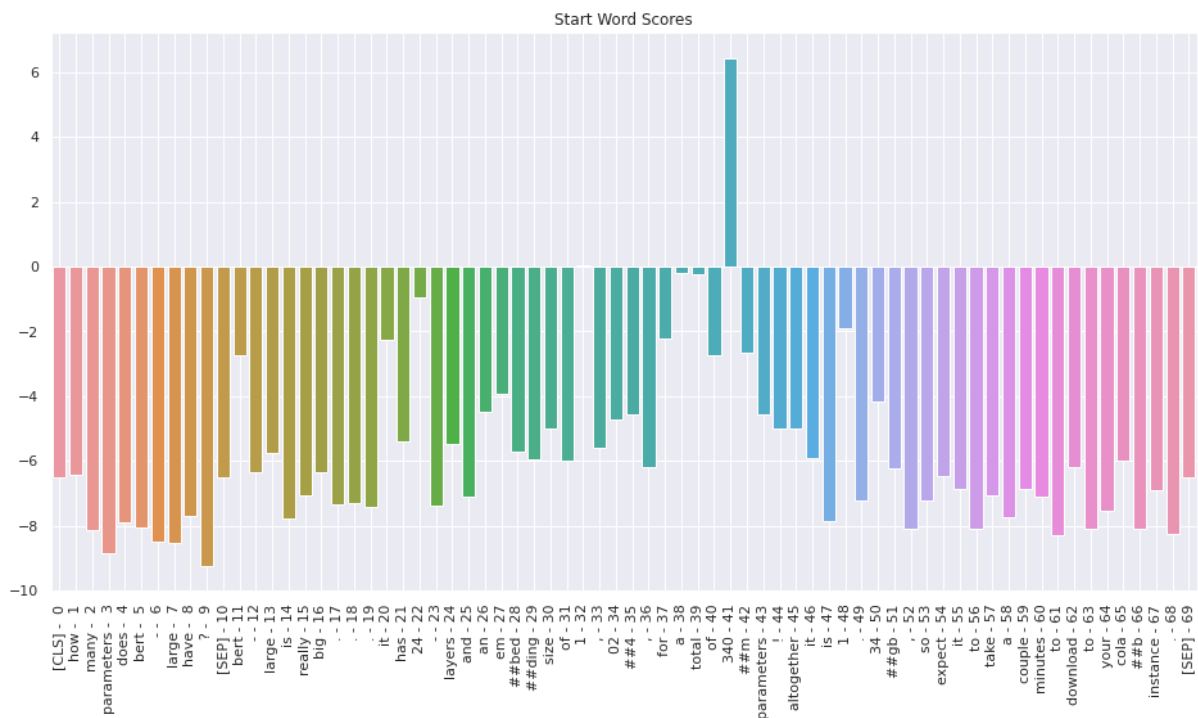
[12]: new_predictions = ensemble(preds_bidaf, preds_rnet)

[13]: # print(new_predictions)

[14]: evaluate_orig(dataset, new_predictions)
Exact Match: 60.2469135802
F1: 67.4230848772

```

Bar plot



CHAPTER 4

REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as computer system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: Minimum and Recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

4.2 SOFTWARE REQUIREMENTS

The software requirements are: Software Requirements deal with definition of software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software requirements for present project:

Operating system: - Windows 7/ XP/8/10

Language:- Python, Machine Learning, Deep Learning, Natural language processing

4.3 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in the case of operating systems. An HCL lists tested, compatible and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

Hardware requirements for this project:

Processor: - Intel dual Core i5/i7 Ram: - 1 GB HARD

Disk: - 80 G

CHAPTER 5

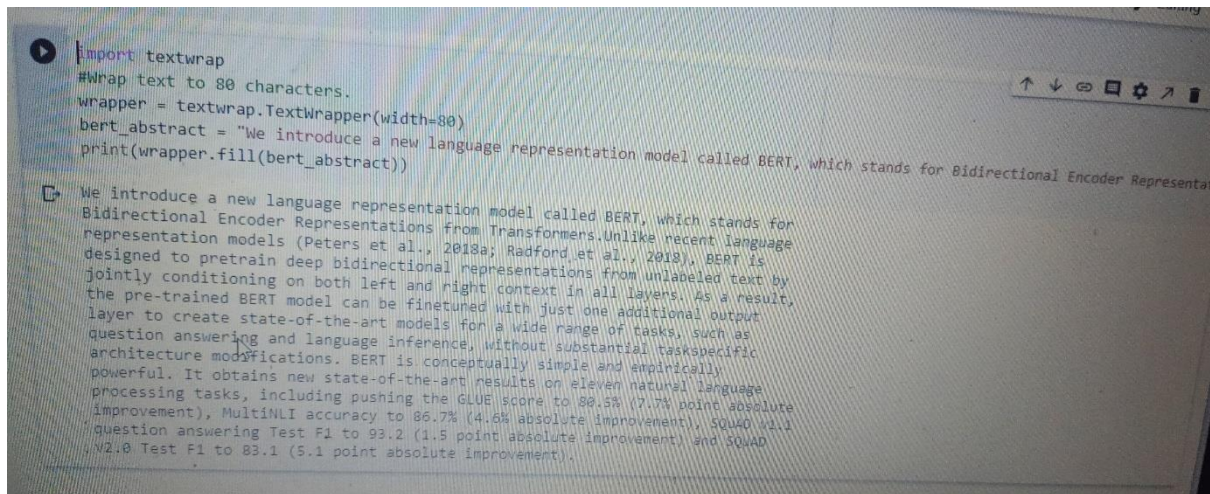
RESULTS AND DISCUSSION

5.1 RESULTS

5.1.1. SCREENSHOTS

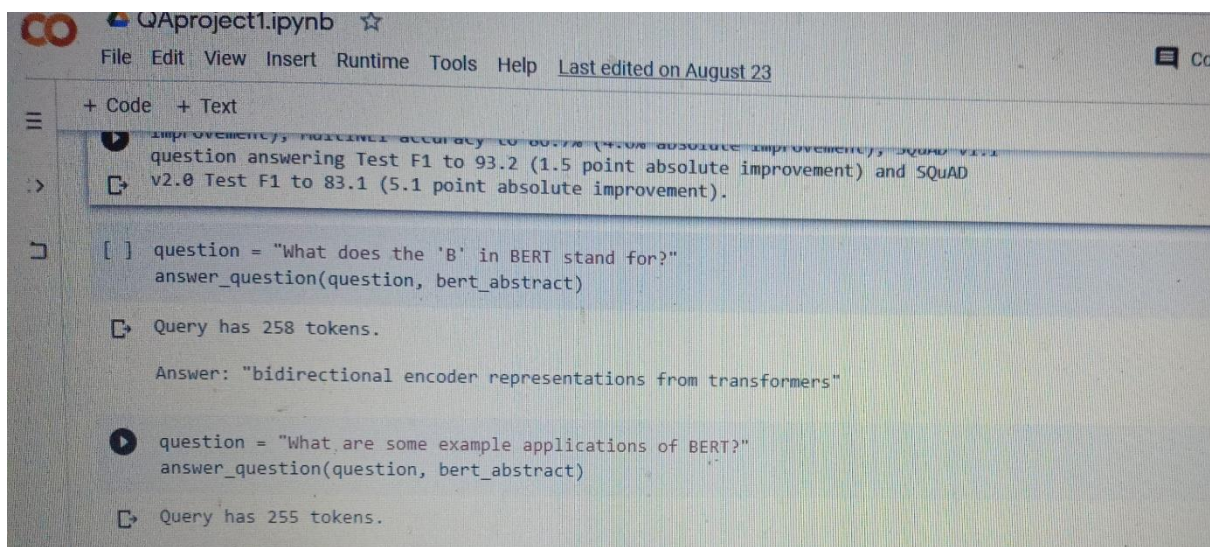
PARAGRAPH:

INPUT:



```
import textwrap
#Wrap text to 80 characters.
wrapper = textwrap.TextWrapper(width=80)
bert_abstract = "We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement)."
```

OUTPUT:



```
question = "What does the 'B' in BERT stand for?"
answer_question(question, bert_abstract)

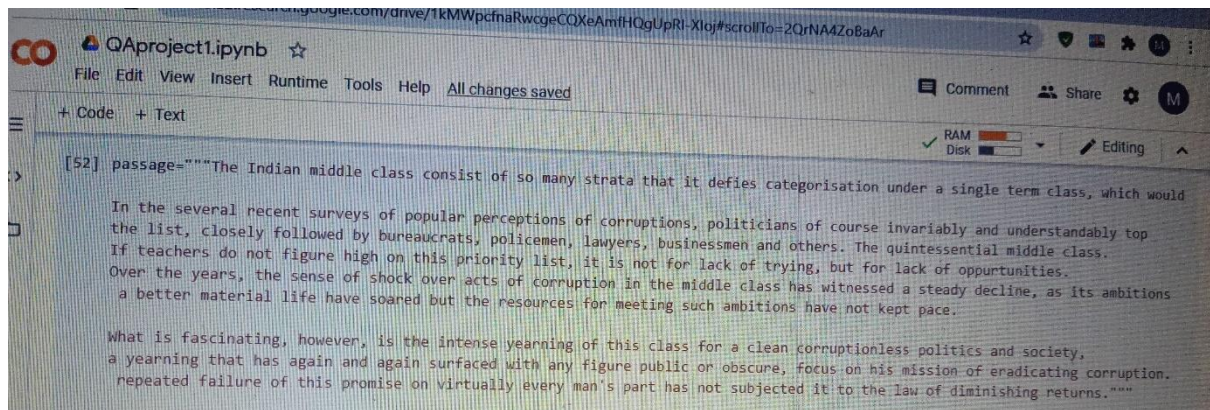
Query has 258 tokens.

Answer: "bidirectional encoder representations from transformers"

question = "What are some example applications of BERT?"
answer_question(question, bert_abstract)

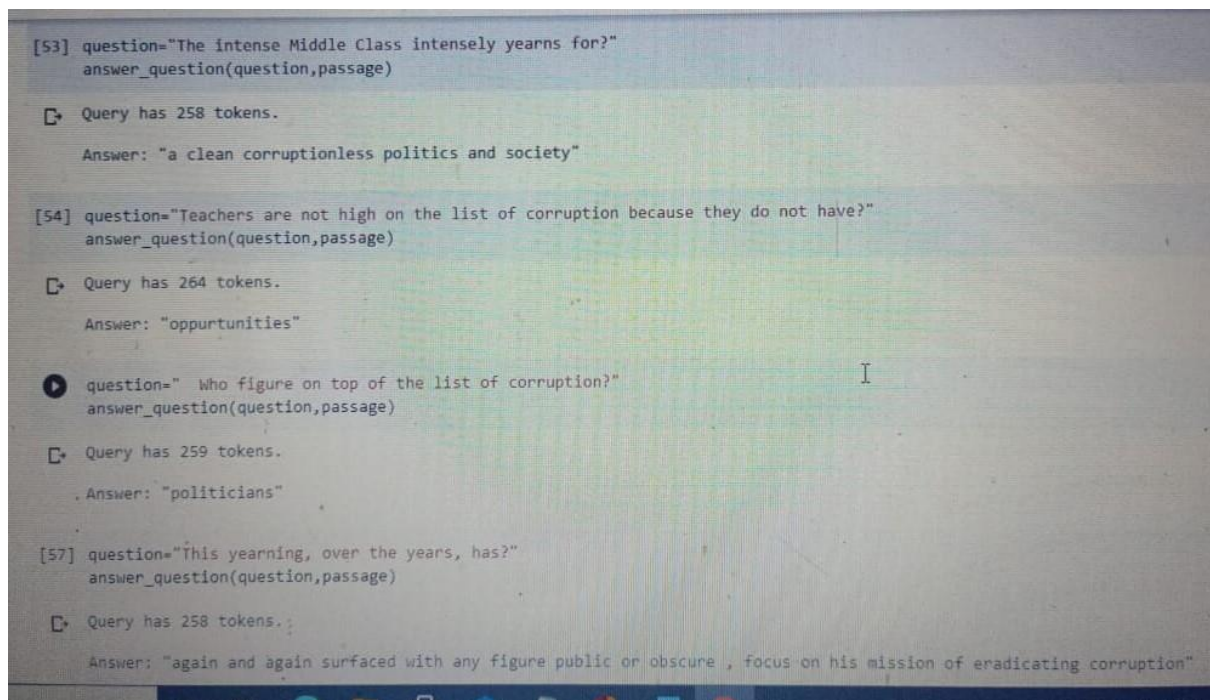
Query has 255 tokens.
```

INPUT:



```
[52] passage="""The Indian middle class consist of so many strata that it defies categorisation under a single term class, which would  
In the several recent surveys of popular perceptions of corruptions, politicians of course invariably and understandably top  
the list, closely followed by bureaucrats, policemen, lawyers, businessmen and others. The quintessential middle class.  
If teachers do not figure high on this priority list, it is not for lack of trying, but for lack of opportunities.  
Over the years, the sense of shock over acts of corruption in the middle class has witnessed a steady decline, as its ambitions  
a better material life have soared but the resources for meeting such ambitions have not kept pace.  
What is fascinating, however, is the intense yearning of this class for a clean corruptionless politics and society,  
a yearning that has again and again surfaced with any figure public or obscure, focus on his mission of eradicating corruption.  
repeated failure of this promise on virtually every man's part has not subjected it to the law of diminishing returns."""
```

OUTPUT:



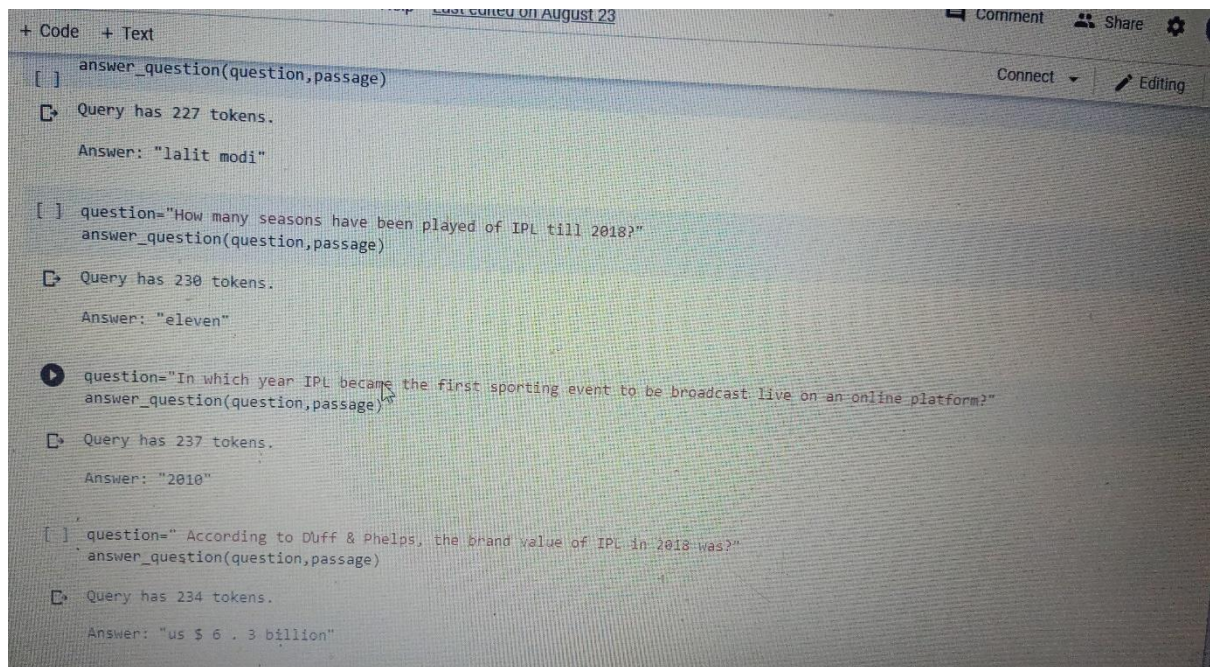
```
[53] question="The intense Middle Class intensely yearns for?"  
answer_question(question,passage)  
  
Query has 258 tokens.  
Answer: "a clean corruptionless politics and society"  
  
[54] question="Teachers are not high on the list of corruption because they do not have?"  
answer_question(question,passage)  
  
Query has 264 tokens.  
Answer: "opportunities"  
  
[55] question=" Who figure on top of the list of corruption?"  
answer_question(question,passage)  
  
Query has 259 tokens.  
Answer: "politicians"  
  
[57] question="This yearning, over the years, has?"  
answer_question(question,passage)  
  
Query has 258 tokens.  
Answer: "again and again surfaced with any figure public or obscure , focus on his mission of eradicating corruption"
```

INPUT:



```
Edit View Insert Runtime Tools Help Last edited on August 23
Comment Share M
Connect Editing
#example
passage=""The Indian Premier League (IPL) is a professional Twenty20 cricket league in India contested during April and May of ev
The IPL is the most-attended cricket league in the world and in 2014 ranked sixth by average attendance among all sports leagues.
There have been eleven seasons of the IPL tournament. The current IPL title holders are the Chennai Super Kings, who won the 2018
```

OUTPUT:



```
+ Code + Text
Last edited on August 23
Comment Share M
Connect Editing
[ ] answer_question(question,passage)
  Query has 227 tokens.
  Answer: "lalit modi"

[ ] question="How many seasons have been played of IPL till 2018?"
  answer_question(question,passage)
  Query has 230 tokens.
  Answer: "eleven"

[ ] question="In which year IPL became the first sporting event to be broadcast live on an online platform?"
  answer_question(question,passage)
  Query has 237 tokens.
  Answer: "2010"

[ ] question=" According to Duff & Phelps, the brand value of IPL in 2018 was?"
  answer_question(question,passage)
  Query has 234 tokens.
  Answer: "us $ 6 . 3 billion"
```

CHAPTER 6

CONCLUSION

6.1 CONCLUSION:

The rapid growth of the development of Internet-based information systems increases the demand for natural language interface that are easy to set up and maintain. It makes people life very comfortable by saving their time and also help them spell correction further. It helps learners to learn effectively by giving answers in no time or insteadly. This method of word2vec through natural language processing algorithm is a reliable and efficient methodology.

REFERENCES:

1) Basic deep learning

2) Sequential models(e.g. LSTM)

<https://www.coursera.org/learn/nlp-sequence-models/home/welcome>

3) Learn Attention, Transformers, Bert

<https://www.coursera.org/lecture/nlp-sequence-models/attention-model-intuition-RDXpX>

4) Understand attention is all you need paper

<http://jalammar.github.io/illustrated-transformer/>

<http://jalammar.github.io/illustrated-gpt2/>

<http://jalammar.github.io/illustrated-bert/>

5) <https://medium.com/@dhartidhami/understanding-bert-word-embeddings-7dc4d2ea54ca>

6) <https://israelg99.github.io/2017-03-23-Word2Vec-Explained/>