# QUIZ GENERATION CHATBOT

**A Project Report**

*Submitted by*


| | |
|---|---|
| **Vivek R. Bhave** | **111508015** |
| **Shreyansh R. Gopawar** | **111508073** |
| **Aditya D. Neralkar** | **111508076** |


*in partial fulfillment for the award of the degree*

*of*

## Information Technology

Under the guidance of

**Dr. Y. V. Haribhakta**

College of Engineering, Pune


**DEPARTMENT OF COMPUTER ENGINEERING**

**AND**

**INFORMATION TECHNOLOGY,**

**COLLEGE OF ENGINEERING, PUNE-5**



April, 2019

# DEPARTMENT OF COMPUTER ENGINEERING

# AND

# INFORMATION TECHNOLOGY,

# COLLEGE OF ENGINEERING, PUNE

## CERTIFICATE

Certified that this project, titled "QUIZ GENERATION CHATBOT" has been successfully completed by

| | |
|---|---|
| **Vivek R. Bhave** | **111508015** |
| **Shreyansh R. Gopawar** | **111508073** |
| **Aditya D. Neralkar** | **111508076** |

and is approved for the partial fulfillment of the requirements for the degree of "B.Tech. Information Technology".

SIGNATURE
**Dr. Y. V. Haribhakta**
**Project Guide**
**Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune - 5.**

SIGNATURE
**Dr. V. Z. Attar**
**Head**
**Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune - 5.**

# Abstract

Chatbots are proving themselves quite useful now a days. Chatbots like Siri(iOS), Google Assistant(Android), Google Allo (Cross Platform), Cortana (Windows), Amazon Alexa and many more, are used daily by many people, in various parts of world. These chatbots help us accomplish many daily tasks like finding a location, booking a table at restaurant, calling someone, etc.

These chatbots are all open domain chatbots, i.e. they can be used for general purpose tasks. But they are unable to perform specific tasks like answering questions which are specific to industry, education, or an organization. Such chatbots are called as closed domain chatbots. Many companies deploy such chatbots on their website to answer specific questions related to them. Most of the replies are pre-fed and if any new query comes to the chatbot, it either sends the query to the companys concerned authority for a reply or straight away ignores the question.

We will be focusing on how to build an open domain end-to-end conversation model for chatbots which shall test the users on their knowledge on any arbitrary topic they choose. The method used for open domain chatbots can be applied for closed domain chatbots as well if relevant data on that domain is available. The only problem faced by closed domain chatbots is that they lack datasets for development, whereas lots of data is available for open domain chatbots.

# Contents

# List of Tables

# List of Figures

# List of Symbols

# Chapter 1

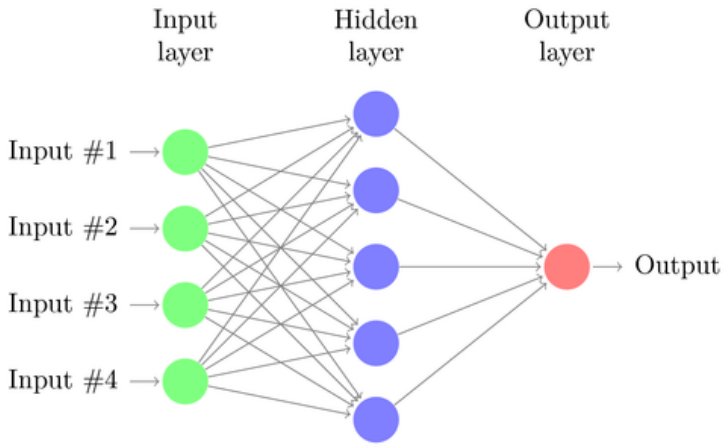
# Introduction

## 1.1 Importance of Automatic Question Generation (AGS) Systems

Gauging a students knowledge over a certain topic is one of the fundamental problems in education systems. Quizzing students has long been accepted as one of the best techniques to solve the problem. Studies conducted over the last several decades have found that providing students with frequent and good number of quiz questions leads to better understanding of the topics, than spending an equal amount of time studying notes or textbooks. Quizzes can be used at several places in the education domain. It can be used in MOOCs to test if a student has grasped the concept properly, they can be used as standalone tests in universities, or they can also be used as practice by students to discern their command over the subject.

However, with the volume of information available, generating questions from domain experts is an expensive and time consuming process. The task of Automatic Question Generation (AQG), in the field of Natural Language Processing, aims to generate questions from a given text, such that students are appropriately tested on their mastery over the subject.

Figure 1.1: Basic Neural Network Diagram




## 1.2 Neural Networks

A neural network is an interconnected collections of neurons. The connections between the neurons are modelled as weights and the final value that the neuron stores is the weighted summation of all the inputs. A neural network consists of an input layer, an output layer and multiple hidden layers.

## 1.3 Recurrent Neural Networks (RNN)

Recurrent neural network is a type of artificial neural network. The problem of remembering previous inputs to the network was solved by recurrent neural networks with the help of hidden layers. Because of their internal memory, RNN is able to remember important things about the input, which allows them to make precise predictions about whats coming next. This is why they are preferred algorithm for sequential data. Sequential data is just ordered data where related things follow each other.

Unlike the normal feedforward neural networks, recurrent neural networks feed the output of previous step to the current step. This helps the network to memorise the previous inputs. So basically RNN cycles through the information, i.e. when taking a decision it takes into consideration the current

Figure 1.2: Recurrent vs Feed-forward Neural Network

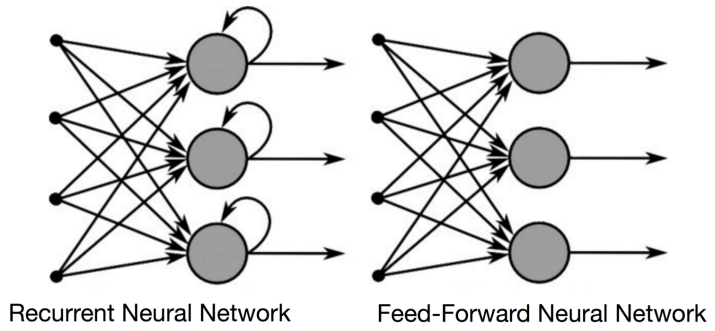
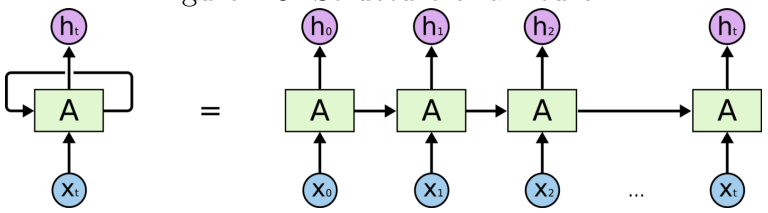



Figure 1.3: Structure of a Neuron




input as well as whatever it learned from the previous output. For example, consider the word Teacher, till the point a feed forward neural network reaches c it forgets about t, e and a. A RNN remembers exactly that. Recurrent Neural Network adds immediate past to the present. This is important since past data contains information about what is coming next.

RNN applies weight to current and also previous input and they tweak their weights through gradient descent and backpropagation through time. Gradient Descent is an algorithm that is used to iteratively minimize a given function. Backpropagation Through Time (BPTT) is basically just a fancy buzz word for doing Backpropagation on an unrolled Recurrent Neural Network. Unrolling is a visualization and conceptual tool, which helps you to understand whats going on within the network. RNN can be viewed as a sequence of Neural Networks that is trained one after another with backpropagation.

The above diagram shows the RNN being unrolled after the equal sign. The different timesteps are visualized and information gets passed from one

timestep to the next. Within BPTT the error is back-propagated from the last to the first timestep, while unrolling all the timesteps. This allows calculating the error for each timestep, which allows updating the weights.

### 1.3.1 Problems with RNN

1. Training RNNs is a difficult task.

2. **Exploding gradient problem :** Exploding gradient is when the algorithm assigns a stupidly high importance to the weights, without much reason. But fortunately, this problem can be easily solved if you truncate or squash the gradients.

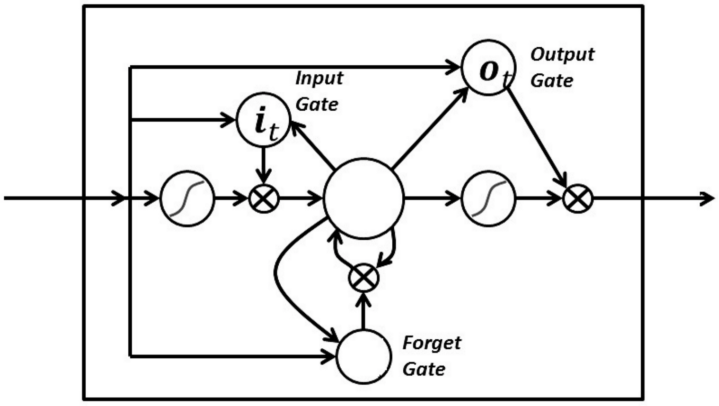
3. **Vanishing gradient problem :** Vanishing gradient is when the values of a gradient are too small and the model stops learning or takes way too long because of that. This problem is solved by the concept of LSTM.

## 1.4 Long Short Term Memory

Long Short Term Memory (LSTM) is an artificial recurrent neural network. Consider LSTM as an extension to the recurrent neural network, which basically extends its memory. Therefore they are capable of learning things that have very large time lags between them. LSTM can remember this because they contain their information in a memory, that is much like the memory of a computer because the LSTM can read, write and delete information from its memory.

The memory in LSTM are gated cells. Gated cell means that he cell decides whether or not to store or delete information based on the importance it assigns to the information. The importance assigning takes place through weights which are learned through the algorithm.

Figure 1.4: Structure of a Basic LSTM node




A basic LSTM node consists of input gate, output gate and forget gate. The input gate decides whether or not to let new input in. The forget gate deletes the information because it is not important. The output gate impacts the output at the current time step. The gates in a LSTM are analog, in the form of sigmoids, meaning that they range from 0 to 1. The fact that they are analog, enables them to do backpropagation with it.

The problem of vanishing gradients which is very common in recurrent neural networks is solved by LSTM as it keeps the gradient steep enough and therefore the training short and the accuracy high.

# Chapter 2

# Literature Survey

1. **Computational Intelligence Framework for Automatic Quiz Question Generation**

   The model described in this paper, uses a rule based approach and is able to generate three types of questions, namely, Fill in the Blank, True/False, and Wh Questions. The training phase of this algorithm involves, detecting topically important sentences, NER and POS Tagging, sanitizing the input sentence and then constructing a tree for further processing of each type of question. For Fill in the Blank, it gives weight to each word involved and marks important words for setting up blanks. Thr true or false sentence involves identifying modal verbs, and also factual type of sentences and further rule based processing. For Wh questions, patterns identified during the training phase are matched, and used to generate questions.

2. **Automatic Question Generation System**

   This is a traditional supervised learning approach and the model can be broken down into a sequence of processes. The text input is stemmed to get the meaning of each word and then passed to the Phrase Mapper. Then the key phrases are extracted using pre trained documents and the

document is summarized. In the end, Nouns are filtered in, on which questions can be generated.

3. **Automatic Question Generation for Intelligent Tutoring Systems**

   This system generates only Multiple Choice Questions, and has been trained on Wikipedia articles. The wikipedia articles are processed and unigrams, bigrams are extracted. The keywords from them are then extracted and their weights using the TF IDF weighting scheme are calculated. Using the above distractors can be generated. Whenever a query is fired to generate questions on, the wikipedia articles of the keyword is taken up, and rule based methods are applied to the sentences to create questions and presented to the users including the distractors as other viable options.

4. **Automatic Question Generation from Childrens Stories for Companion Chatbot**

   This paper was specifically targeted for generating questions from childrens stories, and was trained and tested on much lesser data than others. The model is said to work in two parts, that is one for question generation and the other for ranking of the questions. The model uses the part of speech tags and dependency parsing algorithms along with pre trained language rules for the question generation phase. For ranking of the questions, the model uses logistic regression to determine the acceptability probability of the question.

5. **Deep Guessing**

   Generating Meaningful Personalized Quizzes on Historical Topics by Introducing Wikicategories in Doc2Vec: The aim of this model is to load

all of the wikipedia articles, classify them into categories, and use the knowledge from the above, for generation of distractors in multiple choice questions. The model works on a novel idea of creation of paragraph vectors, whose advantage is that they are trained from unlabeled data and thus can work well for tasks which do not have enough labelled datasets. After the score of wiki categories is obtained, thresholds are decided to classify the records into categories and thus, are eventually used for the creation of meaningful options in the questions generated.

6. **Automatic question generation on the basis of the discourse connectives**

   This problem of question generation has been divided into two modules in this paper. The first part is that of Content selection and the next that of Question formation. The Content selection phase consists of recognizing the part in text that is relevant and important to generate questions on. The second phase of Question formation includes several subtasks like Word Sense disambiguation of the discourse connectives then the Identification of the type of question to be created and eventually applying syntactic transformations on the context. The paper mainly takes into consideration the seven main discourse connectives - although, as a result, because, for example for instance and since. Using the output generated so far, the type of question gets decided and then the question can be formed.

7. **Semantic Based Automatic Question Generation**

   The paper explains a system that applies two fundamental Natural Language Processing concepts namely Semantic Role Labeling and Named Entity Recognizer technique. These tasks are used to convert the in-

putted sentence to semantic pattern. The model described in the paper has developed a system which has identified patterns for each type of question. The question types under consideration here are all of the Wh-questions - who, when, why, where, what. The system for classification uses learning, storage memory, feature extraction, and associative Retrieval.

The sentence given as input will be first parsed using Named Entity Recognition and SRL technique. The output of these two algorithms has a direct correlation with the exact question type to be created. Thus after the question type identification, the question pattern is known using the pretrained rules.

8. **Automatic Multiple Choice Question Generation System for Semantic Attributes Using String Similarity Measures**

The paper has described a model which first selects a factual sentence and an important word to generate questions on from the text given as input. This selection is done on the basis of the semantic labels and named entities in the sentence. Then for actually generating a question the SRL and NER tag is used, which directly helps in finding out the type of question.

Then the model focuses on finding distractors i.e. the incorrect options of a question. For this task different similarity measure between sentences of the data set is taken into account. Eventually, when a question is generated, the measures of similarity between the actual question and the sentences of the input text is considered and sorted. The top three sentences obtained from the above procedure, are considered for finding a relevant important word to the question generated which are eventually

used as distractors.

Table 2.1: Comparison Table

| **Sr No.** | **Algorithm** | **Methodology** | **Type of Question** | **Evaluation of Result** |
|---|---|---|---|---|
| 1. | Cloze question generation | Sentence selection, key selection and distractor selection is domain specific and NER feature is used for key selection | Cloze | Manually Evaluation is done<br>1. Evaluation of the selected sentence<br>2. Evaluation of selected keyword<br>3. Evaluation of selected distractor. |
| 2. | Automatic question generation on the basis of the discourse connectives | Content selection and Question formation | Question generation like Why, when, where, in which | Manually evaluated for semantic and syntactic soundness of question by two evaluator |

| | | | | |
|---|---|---|---|---|
| 3. | Automatic Question Generation Using Software Agents for Technical Institutions | Document Processing, Information Classification and Question Generation. | Define, Describes, Give example, long descriptive questions | - |
| 4. | G-Ask | Citation Extraction, Citation Classification. and Generation | Long descriptive questions like Why, when, Does any.. | Compared questions generated by the system to those produced by humans. and Citation Classification performance is done through precision and recall. |

| | | | | |
|---|---|---|---|---|
| 5. | Automatic Multiple Choice Question Generation System | Extract sentence from Data Set, Prepare Question sentence, Measure the similarity between the question sentence and all sentences in the knowledge base, Return the three sentences that have the highest similarity values, three keywords of three sentences as distractor selection | MCQ | In this research out of nearly 145 parsed sentences, there were 109 considered good according to the keywords that are extracted from them. |

| | | | | |
|---|---|---|---|---|
| 6. | Semantic Based Automatic Question Generation | Input sentence, Feature Extraction through SRL, NER, Choose MCS, Test Sentence pattern and Test the Question type pattern | WH-questions like who, when, where, why, and how. | 170 sentences are extracted and mapped into 250 patterns using SRL and NER. The 250 patterns are used in training and testing. and Precision, Recall and F-measurement is used for classification of question type. The percentage of truly generated patterns increased 87% which appears to be promising ratio in this problem comparing it to other techniques used in generating questions automatically. |

| | | | | |
|---|---|---|---|---|
| 7. | Automatic Generation of Multiple Choice Questions From Domain Ontologies | Ontology-based strategies like class based, property based, terminology based strategies | MCQ (Choose the correct sentence) | The generated questionnaires were evaluated in three dimensions: Pedagogical quality, linguistic/syntactical correctness and number of questions produced. |
| 8. | Mind the Gap: Learning to Choose Gaps for Question Generation | 1)Sentence selection, 2)Question construction, 3)Classification/Scoring. | Fill in the blanks question | manually analyze the generated questions and rate the question |

# Chapter 3

# Problem Statement

The problem of question generation, is currently solved by domain experts, who create questions over text or topics by their intuition or knowledge. Automatic Question Generation can solve this laborious, time consuming task quite efficiently and much higher accuracy. Such an automatic question generation system can have applications in multiple domains.

## 3.1 Motivation

The main motivation of designing an Automatic Question Generation System, was contribution in the education domain. Generating quiz questions can be of utmost importance for teachers to ensure that students are able to understand the material in class, and at the same time, can become an easy way out for students to find out their preparedness over the subject, studying independently.

## 3.2 Objectives

The objective of the project is to take a topic as input from the user, and generate a set of questions related to the topic to present to the user. The questions generated through such a system, should be similar to questions

generated by a domain expert. Also the difficulty level of the questions, should be such that they can appropriately test the knowledge of the user.

# Chapter 4

# System Requirement Specification

The system uses the following technologies and libraries relating to them -

Technology:

- Python 3.6

Libraries:

- Numpy
- Scikit
- Keras
- TensorFlow
- Torch
- Pytorch

The hardware specifications of the system used for running experiments are -

- Nvidia GEFORCE RTX 2080Ti
- Ram 16 GB

- 512 SSD
- 2 TB HDD

The dataset used in training, validation and testing of the model was the SQUAD 2.0 (Stanford Question Answering Dataset). This is the most recent and state of the art dataset used as a benchmark for testing Automatic Question Generation Systems.

# Chapter 5

# System Design

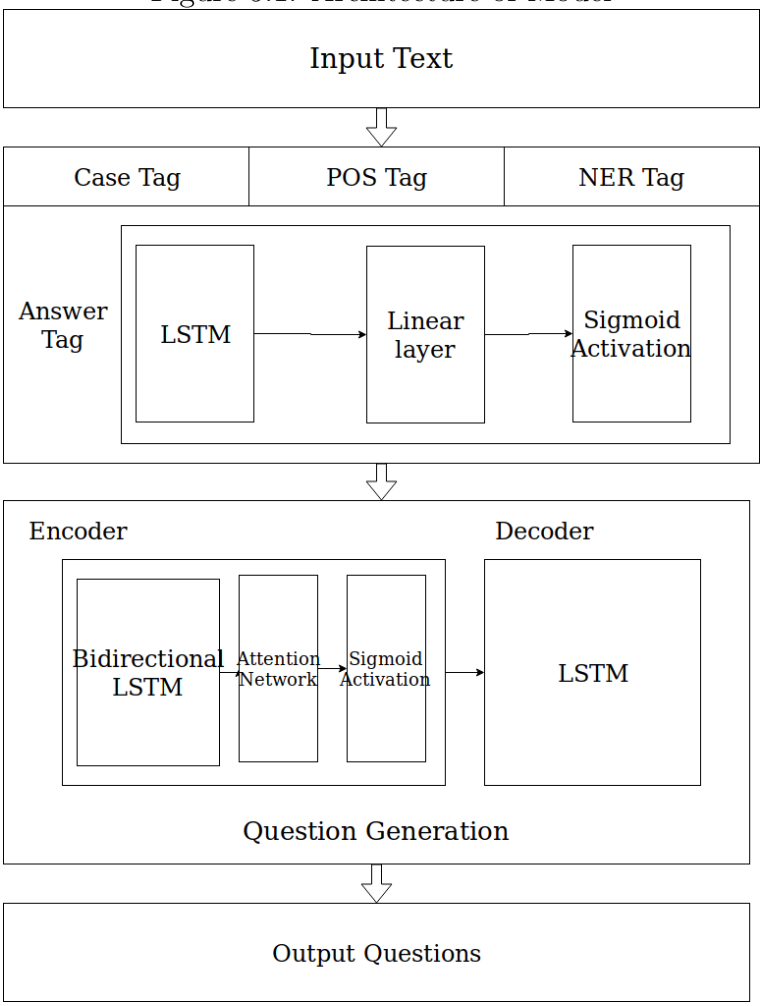
The complete work presented here can be divided into several parts right from taking in a topic as input, finding out relevant documents, tagging words as probable answer words, and eventually generating questions from it.

## 5.1 Topic Input & Data gathering

Initially, the user shall enter his/her topic of interest into the software. The software shall then crawl the web and gather links to various web pages relevant to the topic given by the user. On getting the links, further the software shall scrape those links and parse the HTML pages to get the information content in those web pages.

However, all the content received can be classified into three parts. The first part consists of markup language tasks and syntax, the second part shall contain data unrelated to the topic of interest, and the third shall be the data which is useful and relevant both to the topic of interest. The data classified into the first part, is removed by searching for markup language tasks in the obtained data. On finding such tags, the lines containing such tags shall be eliminated from the useful data text. For identifying data in the second part, the semantic meaning of the words and the web page from where the data

Figure 5.1: Architecture of Model



has been input is taken into consideration. Thus eventually the system has data which can be used for generating questions on the users topic of interest.

## 5.2 Pre processing block

The processing done at this stage can be broken down into 4 independent tasks. The output of each of these four tasks work as features for the question generation algorithm. The four tasks and their work can be summarized as follows -

1. Case of the word in context - This feature determines whether the first letter of the word is in lower or upper case. The output of this case thus determines whether the word is the starting word of the context. In quite a few cases, the starting word of the context dominates in meaning, and hence has been considered as a feature in the problem.

2. NER Tagging - The 7 set Stanford NER Tagging is used in the system. The NER Tagging helps in finding out a type of a noun the word is, like a place, a date or a person. As per the tag different Wh questions are created and hence the feature is of great significance in the problem.

3. POS Tagging - POS tags are important in generating patterns based on their frequency in the contexts. Frequently occuring series of consecutive POS tags that form answer words in the training sentences are most probable to be the best sources for generating questions in test contexts as well.

4. Answer Words - This feature is the most important of all the four features. This feature talks of whether the word in consideration is an answer word or not, in the question to be formed. Only those words

that are answer words are given paramount importance in the question generating process.

So to achieve this goal, a simple model containing a lstm and linear layer is created. The output of the model is binary string which is 1 for the Answer Tag and 0 elsewhere. The LSTM help in remembering the previous answer tags thus helping in predicting the current tag.

## 5.3 Question Generation - Algorithm

- Context Reader and EncoderContext reader is a bi-directional long short-term memory (bi-LSTM) network with 600 hidden layers. It processes the input words in both the forward and backward direction.
- Attention network
- Sigmoid activation for encoder
- Decoder - rnn lstm 600 hidden layer

## 5.4 Output

In the end, the system shall generate questions on either the topic of interest specified to the user or the text given as input to the system. The system can generate three types of questions. The first type is Fill in The Blank questions, where a factual or important word is left blank in a sentence from the input text. The second type is a one word answer, where an important sentence from the input text, is converted to a question and presented to the user. Finally, the system can also generate reading comprehension type of questions, wherein the input text on which questions are generated is made available to the user while he/she is answering the questions

# Chapter 6

# Timeline Required for Overall Implementation

**Project Stage I (Till December)** - Finalisation of Problem Statement, Literature Summary, Dataset Discovery

**January, February** - Implementation and Analysis of existing systems

**February, March** - Proposal of our own Model to solve the Problem Statement

**April** - Running experiments, Generating results, Publication Work,

# Chapter 7

# References

During the literature summary an exhaustive survey of the research already conducted in Automatic Question Generation Systems was conducted. Following are a few of the research papers, we referred -

1. Deep Guessing: Generating Meaningful Personalized Quizzes on Historical Topics by Introducing Wikicategories in Doc2Vec
2. Automatic Question Generation for Intelligent Tutoring Systems
3. Context Aware Restricted Tourism Domain Question Answering System
4. Automatic Question Generation System
5. Automatic Question Generation from Childrens Stories for Companion Chatbot
6. Computational Intelligence Framework for Automatic Quiz Question Generation

Among other things, we would also like to add a few of the online resources we used to learn concepts of Deep Learning in Natural Language Processing, which were of foremost importance in the design and implementation of the model we have proposed.

1. https://www.coursera.org/learn/neural-networks-deep-learning?specialization=d
learning

2. https://www.coursera.org/learn/nlp-sequence-models

3. https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/

4. https://cloud.google.com/storage/docs/gsutil_install

# Chapter 8

# Publication Details

Our research work over the year, has been drafted in the form of a journal paper, and has been submitted for review at the

Conference.

# Appendix A

# Algorithm

# Appendix B

# Flow Chart for Making Popcorn