ANSWER GENERATION

ADAPTIVE ARTIFICIAL INTELLIGENT QUESTION ANSWER SYSTEM

17-107

Preliminary Progress Review
(Preliminary Progress Review Documentation submitted in partial fulfilment of the requirement for the Degree of Bachelor of Science Special (Honours) In Information Technology)

Deleepa Perera (IT14103100)

Bachelor of Science (Honours) in Information Technology
(Specialization in Software Engineering)

Department of Information Technology

Sri Lanka Institute of Information Technology

May 2017

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

1.1 Purpose

The purpose of this PPR is to demonstrate the progress that has been made up to this point in the research project, Adaptive Artificial Intelligent Question Answer Platform. This document format was chosen over an SRS or DRD because it allows to capture information related to projects that are more skewed towards research than implementation. The QA domain has been primarily driven by researchers and a significant commercial product/service has not emerged yet during this period.

This document is intended to be read by persons that have a good understanding of deep learning, specifically deep neural networks. It is also important to keep in mind that this document is a part of a series of documents. To understand the context of this document better, it is advised to read the project proposal that was submitted for this topic and to read the other three PPR's associated with this document. This PPR will only concentrate on the Answer Generation mechanism of the Adaptive Artificial Intelligent Question Answer Platform.

1.2 Definitions, Acronyms and Abbreviations

DNN	Deep Neural Network
DRD	Design Review Document
GPU	Graphics Processing Unit
LSTM	Long Short Term Memory
MRNN	Multiplicative Recurrent Neural Network
PPR	Preliminary Progress Review
QA	Question Answer
RNN	Recurrent Neural Network
SRS	Software Requirements Specification

1.3 Overview

The rest of this document will be organised in the following way. In the second section a literature review will demonstrate the research gap that this project is attempting to fulfil. This will be done by doing a through analysis of current existing solutions in the QA domain and

revealing the advantages and disadvantages of those systems, and how the proposed system can improve on those failings. In the third section the research objectives and the methodology (problem solving approach) will be discussed. The next sections will discuss the data sources, anticipated benefits and the anticipated deliverables for this project.

2.0 LITERATURE REVIEW

2.1 Background

There is a wealth of information on the internet. For any given domain we are able to find a huge amount of information. However, to use this information effectively, there needs to be a system to process the data and extract out the meaningful information. Further it is important to provide a simple and seamless way of interacting with this data. This has given rise to the field to natural language question answering where a user must be able to ask a question in everyday language and receive a factually correct answer quickly. Deep learning is a subset of techniques of machine learning. Deep learning allows multiple processing layers to breakdown the given data into smaller parts and learn the representations of these data [1]. Deep learning is the state of the art in areas such as speech recognition, natural language understanding, visual object recognition, etc.

Traditionally QA systems use the tried methods of parsing, part-of-speech tagging, etc. that come from the domain of NLP research. While there is absolutely nothing wrong with these techniques, they do have their limitations. [2] W.A. Woods et al. shows how we can use NLP as a front end for extracting information from a given query and then translate that into a logical query which can then then be converted into a database query language that can be passed into the underlying database management system. The knowledge base is processed to an ontology that breaks it down into classes, relations and functions [3]. Natural Language Database Interfaces (NLDBIS) are database systems that allow users to access stored data using natural language requests. Some popular commercial systems are IBM's LanguageAccess and Q&A from Symantec [4].

Information retrieval (IR) is another technique that has been used to address the problem of QA. With IR systems pay attention to the organisation, representation and storage of information artifacts such that when a user makes a query the system is able to to return a document or a collection of artifacts that relate to the query [5]. Recent advances in OCR and other text scanning techniques have meant that it is possible to retrieve passages of text rather than entire documents. However IR is still widely seen as from the document retrieval domain rather than from the QA domain.

Template based question answering is another technique that has been used for QA and is currently being used by the START system which has answered over a million questions since 1993 [6]. START uses natural language annotations to match questions to candidate answers. An annotation will have the structure of 'subject-relationship-object' and when a user asks a question, the question will be matched to all the available annotation entries at the word level (using synonyms, IS-A, etc) and the structure level. When a successful match is found, the annotation will point to an information segment which will be returned as the answer. When new information resources are incorporated into the SMART system, the natural language annotations have to be composed manually [7].

The solution that we are proposing for this problem domain is one that is based on deep learning. Deep learning can be defined a subset of machine learning techniques that uses non-linear information processing to identify and extract features and patterns in data, classification and transformations. Since the purpose of this document is to discuss the Answer Generation component of this solution, the next section will look at some literature that discusses generic text generation using DNN's.

2.2 Answer Generation

For the answer generation component, the most important research area is text generation. Text generation it's self is an area that has received a lot of attention by deep learning researchers. This technique has become an extremely valuable tool in the field of publication where a DNN is able to produce and entire article from a set of keywords. This means that publications can cut down on the costs of human writers and can create articles at a faster rate. The most successful research in this domain has been done with relation to recurrent neural networks (RNN).

Due to the nature of RNN's these neural nets are capable of forecasting information instead of finding patterns in the given data set. This means a properly trained RNN will be capable of generating a sequence of text that will match the data set that it is trained on. However RNN's have a inherent problem called the vanishing/exploding gradients [8]. This causes an unstable relationship between the parameters and the hidden states, making RNN's notoriously difficult to train. However, a breakthrough in an optimisation technique called the Hessian-Free optimisation has made it cheaper (in terms of computing power) to train RNN's [9]. In order

to generate correct predictive words, the RNN's must be capable of remembering old word associations. For example, it would be common place for the the word 'milk' to be associated with the word 'cow', but beyond a certain point the neural net must be able to develop a deeper understanding of the meaning of the words to maintain that association. This problem has been tackled by using memory units in standard RNN's that can store information for a certain period of time and can access those units when required. These neural nets are known as 'Long-Short Term Memory' [10].

Recent research that attempted to create a character level language model has found that the performance of RNN's even with these advancements were not satisfactory enough. A new approach was to use multiplicative RNN's (MRNN's). The input to each step is a character and they would ideally be able to see the entire hidden-to-hidden weight matrix for each possible input [11].

Another possible approach is to use a recursive auto encoder model which will be fed with data formatted in a dependency parse tree [12]. In this approach the researchers have constructed a sentence level vector which can be decomposed when required and the same sentence can be retrieved again later on. However, by randomising the vector's at the output stage (decomposition) they have managed to generate paraphrased sentence structures from the data that they feed in. This approach requires some complex corpus pre-processing and is therefore a tedious task to carry out.

The approach for this research will comprise of a regular RNN that will be trained with the Hessian-Free optimisation technique. Depending on the accuracy and performance of the neural net, we will decide to implement LSTM's or to take the MRNN approach.

3.0 RESEARCH QUESTION

The primary overall research question that this project is trying to address is if there is a better way to build a QA system that utilises deep learning technologies. Due to the rapid growth rate of information, efficient QA systems will provide the ability for people to interact with large knowledge bases efficiently. While QA systems have caught the imagination of many researchers for many years now, until the maturity of deep learning technologies there hasn't been an effective way to implement such systems.

The answer generation component specifically is a key component in making the system user friendly. For example, when the user asks a question, the Adaptive Artificial Intelligent QA system should be capable of generating a human like answer. So to the user it will seem as if he/she is interacting with an actual person. Given this, a continued dialog is not a goal that we will be pursuing. Just a single answer that is generated for a given question.

In that context the research question is if it is possible to create an efficient neural network that is capable of generating text with a set of prime text words. This means that once the answer words have been extracted, the system must be capable of generating an answer sentence that matches the question asked, and contains the correct answer.

4.0 RESEARCH OBJECTIVES

The main research objective of the overall project is to create a Question Answer platform that utilises state of the art deep learning techniques to analyse a given data set and allow any user to extract information from that data set using the question and answer format. There are two type of data sets for such systems, open dataset and closed datasets. Open QA systems will utilise the information from the given knowledge base as well as any additional available knowledge.

To narrow down our scope we have chosen to build a question answer system for the **medical emergency domain.** As a case study and proof of concept we plan to implement a QA System that focuses only on answering questions related to medical emergency situations and not addressing open domain questions. The scope has been thus narrowed to, first, increase the accuracy of answers provided and second, to ensure that the project can be completed in the allocated time. The corpus will be the ECDS dataset provided by NHS, and the team is currently in the process of negotiating with the responsible people to obtain the dataset.

The end product would be a system that allows the user to ask medical emergency related questions in natural language form and the platform would find the most accurate answer and provide that answer in natural language form as well. The idea is to simulate a situation where the user is interacting with a person in the medical profession as close as possible. The accuracy of the answers will largely depend upon the accuracy of the data in the data set and therefore we cannot guarantee that this will be able to replace an actual medical professional. However, the goal in this research is to show that using deep learning techniques we are able to reduce some of the complexities and barriers that are present at the moment and are stopping QA systems from becoming mainstream products. The medical emergency situation was chosen purely out of convenience because of the availability of the dataset. It is only a proof of concept.

Answer generation is a component that is important to ensure that the user can interact with the platform as naturally as possible. It can be considered as a single-turn dialogue based mechanism for the platform. This means when a user asks a question, it will feel like the user is interacting with a real person. The answer the user gets must be both factually correct and linguistically and grammatically sound. This component will be handling the later part of that. So the basic research objective of this component is to generate an answer that is user friendly

using a character-level language model. There are several methods which this could be achieved in. As discussed in the literature review, there are RNN with LSTM, Char-RNN and MRNN's. These methods have been highlighted in the next section.

5.0 RESEARCH METHODOLOGY

We need to pre-process the answer that we give to the user, since a goal of this project is to make it extremely user friendly to interact with a large corpus, the answer generation component is necessary to ensure that user is able to interact with the system as naturally as possible.

The challenge with generating natural language from a single piece of text is that it is the reverse approach to what is taken by most other researchers. Usually deep neural networks are used to identify patterns and are used as predictive models. In this case the DNN must be used as a generative model instead. This means given some fragments of a sentence, a well trained model can create a proper sentence in natural language format.

The approach to this problem in this research will be to use recurrent neural networks (RNN). In a RNN the neurons and tensors connect together to form a directed cycle. This gives the neural network some memory properties. The neurons have an activation time after which they stop firing. This is in contrast to regular feed forward networks. However, RNN's have what is often referred to as the 'exploding gradients problem' because of the hidden states that the neurons can go into. This has made training RNN's tough. A recent development in the Hessian-Free optimization technique however has made it effective to train RNN's [9]. Sutskever et al. show how they have trained and optimised a RNN to predict the next character in a stream of characters [11]. The researchers developed a character level RNN model that was able to generate text with a high level linguistic structure and correct grammatical structure.

The most prominent research that has been done in this field is by a researcher named Andrej Karpathy, a PhD student from Stanford. He created a RNN called Char-RNN that is able to generate new text when a chunk of text is fed. Char-RNN works by analysing the previous characters in a string and guessing what the next characters should be. Therefore, by training a RNN with a large data set such as Project Gutenberg we will be able to train a good RNN model. Then the model can be fed with some prime-text which would come from the answer sentence selected in the previous component and the model should be able to generate the required sentence [13].

A RNN acts like a feedback loop where the output of one layer is added to the input and is fed back into the same neural net. This means a RNN can form a time series as shown below. So it is able to process information in a continuous series and this makes it ideal for forecasting information.

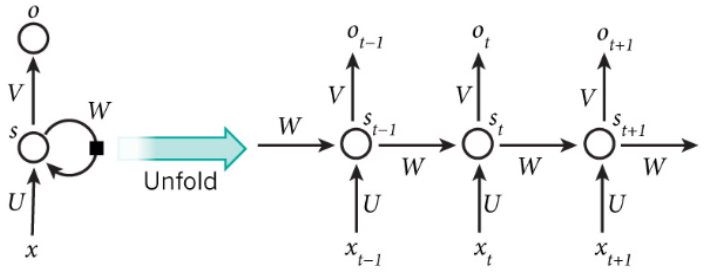

Figure 1.0: A time series RNN [14]

In the diagram above the output W of S_{t-1} becomes the input of the next iteration S_t, along with the new data, U. So the outputs form a time series. We can use this property of RNN's to generate new text from a model that has been trained using a vocabulary that is similar to the knowledge base that we are using. Existing research also shows that this method has proven to be successful were researches like Andrej Karpathy have been able to generate entire Shakespearean paragraphs through the RNN.

The expected outcome of this component would be something like follows:

Question	: When was Sri Lanka railways founded?
Identified answer sentence	: Sri Lanka railways has not been profitable since it was founded in 1858
Prime-text	: Sri Lanka railways founded 1858
Expected output	: Sri Lanka railways was founded in 1858

Overall the basic flow would be that RNN is trained using a question-answer data set which means that the resulting model would be able to produce answer texts. When vectorising the data set the same component that was built for the corpus pre-processing would be used. When the model is being used, parts of question sentence and the answer must be fed into the neural network as prime text.

First we would have to give the RNN a chunk of text and tell it to model the probability distribution for a character given a sequence of previous characters. One way of encoding the characters into a vector would be to use the 1-of-k encoding technique. This is a simple technique for vectorisation and during the research period we might have to consider a more efficient approach. Next we need to feed that data into the RNN and see what the probability distribution for each character is. Then depending on how we want to train our model we can perform a parameter update to nudge the gradient descent in the direction we want it to go.

From what is described above it must be obvious that the output of the model we train will be highly dependant on the training data set. This mean we need to be very careful when selecting the data set such that it fits the question answer format.

6.0 SOURCES FOR TEST DATA

This section will discuss the data set that is used for the answer generation component. Here a generic question answer dataset is expected to be sufficient. The sentences must be correctly structured and must compromise of the complete English alphabet. The dataset maybe from different text files or from a single text file. However, it is not practical to compile multiple data sources into a single text file. Also it is not possible to increase the data set over time in this manner. Instead the data could be in multiple files and each file would be processed one at a time and the vectorised data would be saved in a separate file. This would be the input for the neural net.

One of the dataset's used by Sutskever et al. is extracted from English Wikipedia. Since this research is the most similar to what we want to accomplish, this is probably the dataset that we would use as well [11]. A similar dataset that is available is the Stanford Question Answering Dataset (SQuAD) [15]. This dataset has been extracted from Wikipedia as well. The questions are extracted from crowdworkers on a set of Wikipedia articles and the answer text is extracted from the corresponding article. This dataset is distributed under the distributed under the CC BY-SA 4.0 license.

Another question answer dataset is provided for use in academic purposes by the Carnegie Mellon University [16]. This dataset consists of manually-generated fact-based questions from Wikipedia articles, and manually-generated answers to these questions. This data was collected over the period of 2008 – 2010.

Since this is for text generation, another good training dataset would be the Project Gutenberg dataset. This is a dataset that is comprised of 3,036 English books that have been written by 142 authors. There is a dataset that has been cleaned for all metadata, transcriber's notes and licensing information. Which means only the actual text data is present in that dataset.

7.0 ANTICIPATED BENEFITS

In terms of the the contribution to the body of knowledge in the QA domain, we expect that this research project will be able to show that deep learning techniques are now mature enough to be used for QA purposes. The availability of data and processing power has made this possible.

In terms of the answer generation component, it is expected that there is a contribution to the text generation research that is going on now. Text generation deals with the forecasting of a particular piece of information. In this case this would be a character. We hope to use the methods mentioned above to provide a more refined approach to the idea of text generation to prove that this method can be used for dialogue generation as well.

8.0 EXPECTED RESEARCH OUTCOME

The main benefit we expect from this project to the user's of the system is that it would make them more efficient when searching for information in the given knowledge base. Users would no longer have to rely on expert knowledge, which can sometimes be very expensive to access. Instead once that knowledge is collated into a process-able corpus, it can be processed by this platform and the information would be readily be accessible by the users. The design of the system also implies that no special knowledge is required to interact with the system. For example, there is no special query language that needs to be used. Instead they can rely on the everyday use of language to interact with this system.

The expected research outcome of the answer generation component is a software component that is capable of text generation. It must be able to take in a few words for an answer and warp those words into a proper answer sentence. The words that are fed in are called prime text. The prime text should be able to guide the RNN to generate the type of sentence that we want it to generate.

9.0 RESEARCH CONSTRAINTS

One of the main constraints that we have to face is the accessibility of the dataset. The main dataset that we would need to access is the ECDS dataset provided by NHS. This dataset has been used by Google's Deep Mind team with some success. While we have requested access to this dataset, or at least a part of this dataset, we haven't received any official confirmation yet. This is a research constraint for the project as a whole.

For the answer generation component, a research constraint is the training of the RNN. While the methods described above provide a vastly improved training mechanism (Hessian-Free optimisation), sometimes training the RNN can get unwieldy. Most of the research that we have read have used GPU's excessively for the purpose of parallelism and this has made the training process faster. Another challenge is that there must be continuous monitoring of the training process to avoid over fitting the model to the given dataset. The challenge is to balance over fitting and under fitting when training the model. This can be challenging sometimes with a neural net that is as complicated as RNN's.

10.0 PROJECT SCHEDULE

10.1 Overall Schedule

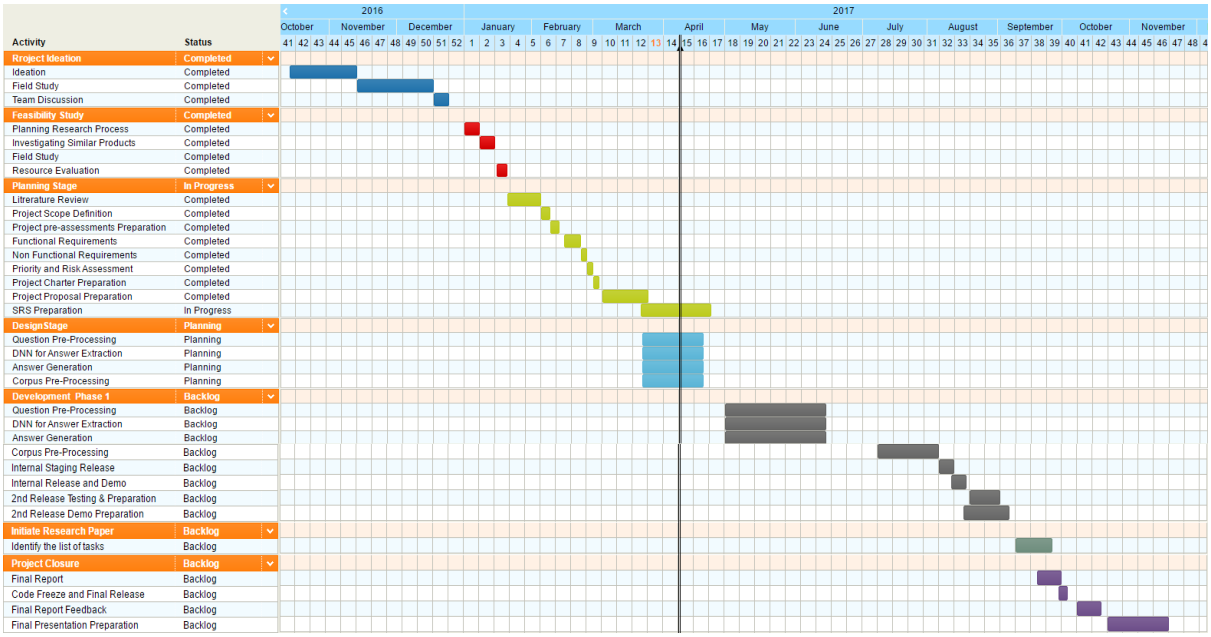

Figure 2.0: Gantt Chart

10.2 Answer Generation Schedule

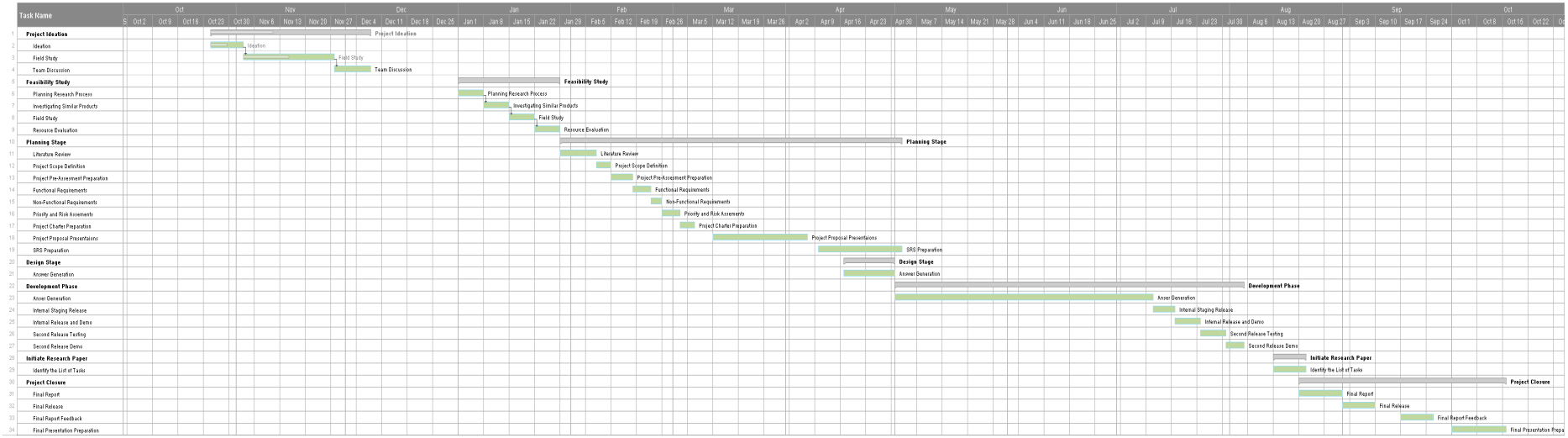

Figure 3.0: Component delivery schedule

11.0 REFERENCES

[1] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, vol. 521, pp. 436 – 444, 2015

[2] W.A Woods, R.M. Kaplan and B. Nash-Webber, “The lunar sciences natural language information system”, BBN Rep. 2378, Bolt Beranek and Newman, Cambridge, Mass., USA, 1977

[3] T.R. Gruber, “A translation approach to portable ontology specifications”, Knowledge Acquisition, 5 (2), 1993.

[4] R. Dale, H. Moisl and H. Sommers, Handbook of Natural Language Processing, 1st ed. New York: Marcel Dekker AG, 2006, pp. 215 - 250.

[5] L. Hirschman and R. Gaizauskas, “Natural language question answering: the view from here”, Natural Language Engineering, 7 (4), 2001, pp. 275-300.

[6] "The START Natural Language Question Answering System", Start.csail.mit.edu, 2017. [Online]. Available: http://start.csail.mit.edu/index.php. [Accessed: 26- Mar- 2017]

[7] B. Katz, G. Borchardt and S. Felshin, “Natural Language Annotations for Question Answering”, Proceedings of the 19th International FLAIRS Conference (FLAIRS 2006), 2006

[8] Y, Bengio, P. Simard, and P. Frasconi, “ Learning long-term dependencies with gradient descent is difficult.” *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[9] J. Martens, “Deep learning via Hessian-free optimization.” In *Proceedings of the 27th International Conference on Machine Learning (ICML)*. ICML 2010, 2010.

[10] S. Hochreiter, and J. Schmidhuber, ‘Long short-term memory’. *Neural Computation*, 9(8):1735–1780, 1997.

[11]I. Sutskever, J. Martens and G. Hinton, "Generating Text with Recurrent Neural Networks", *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[12] J. Boyd-Graber, H. Daumé & M. Iyyer, “Generating Sentences from Semantic Vector Space Representations”, 2014

[13] "The Unreasonable Effectiveness of Recurrent Neural Networks", Karpathy.github.io, 2017. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[14] "Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs", WildML, 2017. [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/.

[15] "The Stanford Question Answering Dataset", *Rajpurkar.github.io*, 2017. [Online]. Available: https://rajpurkar.github.io/SQuAD-explorer/. [Accessed: 01- May- 2017]

[16] "Question-Answer Dataset", *Cs.cmu.edu*, 2017. [Online]. Available: http://www.cs.cmu.edu/~ark/QA-data/. [Accessed: 01- May- 2017]

APPENDIX I: OVERALL SYSTEM ARCHITECTURE

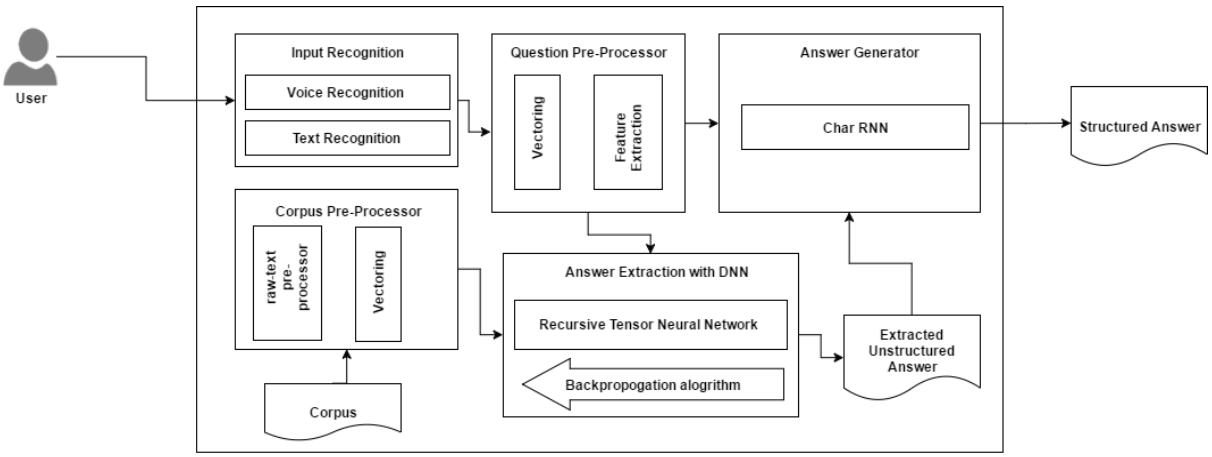

Figure 4.0: Architecture of Adaptive Artificial Intelligent Question Answer

APPENDIX II: PROJECT INFORMATION

The team involved with this particular project is as follows

Supervisor: Mr. Yashas Mallawarachi
External supervisor: Mr. Anupiya Nugaliyada

Implementation team of Adaptive Artificial Intelligent Question Answer System:
Akram M.R. (Leader)
Deleepa Perera
Singhabahu C.P.
Saad M.S.M.

This document is a part of a series of four documents. It only provides an insight into a single component (Answer Generation). For a complete overview of the project, please refer to the Project Proposal.