Final Report Draft

Report Compiled by : Akram M.R IT14109386

1.0 INTRODUCTION

The information growth rate in the world is increasing at a rapid rate. It has become impossible to keep up with the amount of information that is being added to any given domain. A regular human cannot keep up with all new that gets generated. Due to the wide availability of digital input and output devices and the ease of use of these devices people are creating more and more raw data. If we are able to process this data into meaningful information fast, it would be possible for people to become more productive and to get the information they want faster.

The "Adaptive QA Platform" is using a deep learning model for the QA system that will enable users to retrieve information in a fast and intuitive manner. Deep Learning/ Deep Neural Networks (DNN) has displayed great performance in tasks such as visual question and answering, object detection, classification etc. The project "Adaptive QA Platform" is built to be able to adopt to a multi-model approach where separately trained deep learning models can be plugged into the system for QA tasks. Since the system undertakes a multi model approach it was required by the question preprocessing component to able preprocess questions from a variety of domains.

The purpose of this report is to present the work of the question pre-processing component in the **Adaptive Artificial Intelligent QA System.** This report will give the reader a comprehensive understanding of the scope, research objectives, methodologies used, results and conclusion. The intended audience for this document is any party who has referred to the project proposal document of the platform and any other party interested in the projects project progress which falls under the CDAP module.

This report contains a further twelve sections. The next section in this report illustrates the literature review comparing existing solutions and methodologies in the QP domain. and the next three sections describes the **research gap, research problem, research objectives and methodology**. Research problem addresses three vital questions the Question Preprocessing component has answered, the research methodology addresses the methods and techniques used

to achieve the objectives mentioned in section four. The testing and implementation section describes the sources and the training dataset that was used to train and build the QP model.

The final sections of this report discusses regarding the findings, results and discusses the challenges faced during the research.

1.1 Research Gap

As explained in the introduction, 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. In the literature survey we discussed three different techniques that have been used to tackle this problem, NLP, Information Retrieval and Template based question answering. All three methods have flaws where either the accuracy is not high enough or it may take a lot of manual processing and so on. This has meant that while this is an important problem domain due to the high costs there haven't been any commercially viable solutions yet.

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. There are three key reasons that deep learning has become so popular in the recent past. First, the hugely increased processing abilities and availability of general purpose GPU's, the vast amount of training data that has become available and the many advances made in the recent past in the field of deep learning that has made the task of training artificial neural networks more efficient [9].

Deep learning makes extensive use of Artificial Neural Networks (ANN) to complete a given task. ANN's are inspired from the neural networks found in the human brain. The brain consists of an intricate network of neuron cells. Researchers have found success in trying to replicate this structure on silicon chip. Each neuron would consist of what is known as an activation function and the neurons would be connected to each other via connections called tensors [10]. The entire ANN would consist of an input layer of neurons, multiple hidden layers and an output layer.

Biases are assigned to neurons and weights are assigned to the tensors. These values influence what eventually becomes the output of the ANN. When training an ANN we will be adjusting these weights and biases to get the desired output for a known input data set and known scenario [11]. When working with a large ANN with several tens of neurons it can become a tedious task to adjust the weights and biases manually. There are algorithms we can use such as backpropagation to help us adjust these values automatically until we get to the desired output.

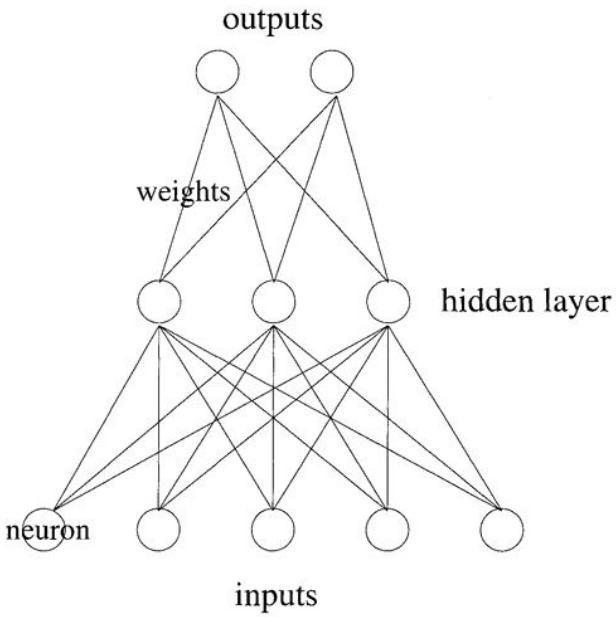


Figure 1.0 Basic representation of an ANN [10]

Since deep neural networks are exemplary at recognising patterns and processing data extremely fast, we believe that by applying deep learning techniques to this problem domain we will be able to overcome many of the drawbacks of the other approaches. We will be able to have a higher accuracy because of the state of the art neural network training paradigms, reduce manual tasks by allowing an ANN to process and structure the corpus and automatically extract the required features and we will not need to use an underlying database engine so therefore we will not need to adopt or develop a different query language.

1.2 Research Objectives

QA systems can take many different avenues in terms of its approach as illustrated in the literature review. Since we will be utilizing a neural network based approach the question presented by the user in natural language form will need to be represented in a vector space such that the vector space preserves the syntax and semantics of the question. In trying to achieve the aforementioned tasks we will need to identify and map these requirements to stable technical approaches. These approaches will also need to be tweaked in order to comply with the variation of the neural network model that will be deployed in the final system. Since the **Adaptive Artificial Intelligent QA System** targets the **Medical Emergency** domain the semantics of that domain needs to be studied and incorporated in the preprocessing stage.

Thus, we can conclude that the main objective of the QP component is to construct a vector space representing the question asked by the user. The vector should be constructed in a way it preserves and maps the semantic relationships of the words in close proximity in a low dimensional vector space. The importance of preserving the semantics is that it enables the neural network to identify the context of the question through the pattern it's represented in. Due to this a uniform technique needs to be identified that can also preserve the context and enables the neural network to perform efficient and effectively.

1.3 Literature Review

Artificial Intelligence has been gaining a lot of prominence in the current era. It is specifically to do with building systems that are able to learn by themselves without having to be programmed such that it ends up to be a giant truth statement. Deep learning is a subset of AI. Deep learning allows multiple processing layers to breakdown the given data into smaller parts and learn the representations of these data. Deep learning has been a breakthrough research area achieving high success rate in the areas of speech recognition, image processing etc.

QA is a well researched domain in the point of NLP and NLU. One of the significant tasks in a QA system is to represent the Natural Language question such as "What is the weather today ?" etc, in a machine readable/understandable format. Traditionally when QA systems use an Informational Retrieval (IR) based approach with an ontology built with significant human effort, the task of QP would be to generate a query out of the given question and the query can be executed in the underlying DBMS system [2]. The knowledge base in such system is always domain specific and further drilled down to categories within the domain. The primary task of QP in such an instance would be identify in what direction or category the question is presented by the user.

Another approach for QA system is closed domain QA systems. The question analysis in the closed domain system by A. Frank and co[1] uses a popular architecture known as Heart of Gold (HoG), they present an hybrid approach utilizing NLP techniques. A question is linguistically analysed by the Heart of Gold (HoG) NLP architecture, which flexibly integrates deep and shallow NLP components, In this architecture the initial stages of QP uses syntactic and semantic analysis.

Another approach for QA systems is the rule based approach. Rule based QA systems are an extended form of IR based systems. Rule Based QA doesn't use deep language understanding or specific sophisticated approaches [1]. A broad coverage of NLP techniques are used in order to achieve accuracy of the answers retrieved. Some popular rule based QA systems such as Quarc and Noisy channel generates heuristic rules with the help of lexical and semantic features in the

questions. For each type of questions it generates rules for the semantic classes like who, when, what, where and Why type questions.

The current trends in QA systems is to use a neural network based approach. Where the neural nets uses a preprocessed dataset to learn patterns and extract information accordingly. Pre-processing of the question and the dataset uses a popular NLP technique known as word embedding. Word embedding has the ability to map words with a semantic relationship in close proximity in a low dimensional vector space [3], this allows the neural network to understand the context of the question and the context of the dataset which is achieved in the preprocessing layer of the dataset. However this embedding layer in QP has several variations and includes different other techniques to enhance the representation of the vector.

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. In the literature survey we discussed few different techniques that have been used to tackle this problem, NLP, Information Retrieval and Information Extraction, Categorization. All the methods has its own flaws where either the accuracy is not high enough or it may take a lot of manual processing and so on.

2.0 RESEARCH METHODOLOGY

2.1 Research Problem

The key focus area in the question preprocessing component was to preprocess the question presented by the user in natural language form such that it is represented in a machine readable format. Since the **Adaptive Artificial Intelligent QA System** uses a neural network based deep learning approach, the task in QP is to generate a vector that the neural networks can understand. One of the main challenges in generating a vector is to maintain the syntax and semantics of the question asked and further map it in the most efficient and effective way such that the processing of the vector by the neural networks to extract the answer consumes as less time as possible and achieves the highest accuracy.

QP component is able to identify the most efficient representation of the vector space and it needs to follow strict process in order quantify the most effective and efficient representation.

In summary the research problem put forward for question preprocessing can be identified as the following

1. What is the most efficient machine readable representation of the natural language question ?
2. How can we preserve the syntax and semantics of the question within the representation ?
3. How can we ensure the representation is the most effective and efficient for the chosen neural network model for the system ?

Throughout the research of this component i was able address the above mentioned problems utilizing various deep learning and natural language processing techniques.

2.2 Methodology

In a Q&A based system understanding and processing a question is vital to provide the user with the most appropriate answer. Machines do not understand text as humans do thus questions inputted as texts need to be transformed into a vector that preserves the context of the question that was presented by the user. The objective of this area is to provide the neural network with the most effective vector format that would preserve and best represent the context of a question in a vector so that the DNN can utilize the vector to process and find the most appropriate answer.

For this purpose we will be using few natural language modelling techniques such as word embedding,syntactic analysis, Identification of question type such as a wh-question analysis.
Word Embedding is used to map words or phrases from a vocabulary to a corresponding vector. This type of representation has two important and advantageous properties:

1. It is a more efficient representation
2. It is a more expressive representation

Word embedding maps each word to a vector space.The Embedding layer will map each token from the question to its corresponding vector space, which preserves the contextual similarity of words in the vector space.

The embedding layer in the question processing component can be done through a popular pre-trained word embedding model known as word2vec or glove(exact model will be chosen based on further trial and error)[5]. Word2vec is a small two layer neural network. It contains two distinct models (CBOW and skip-gram), each with two different training methods (with/without negative sampling) and other variations [5]. To top that, it also contains a sharp pre-processing pipeline, whose effects on the overall performance is yet to be evaluated.

The other sub-component is Syntactic analysis of a question. Syntactic analysis is the process of identifying the structure of a sentence,The interplay of syntax and semantics of natural language questions is of interest for question representation. Researchers in the area of question understanding reccomend the use of a TreeLSTM neural network for this purpose[6], since it is capable to capture long distance interaction on a tree. The other option would be to go with a chain structured LSTM but the critical downfall of it for this specific task is that it fails to capture long distance interaction on a tree[6].

To obtain the parse tree information some of the available open source parsers such as the Noah's Ark parser [7], or the Standford Core Parser can be used.

Questions by nature are composed to fulfill different types of information. A what question and a how question requires different types of information [6]. In Order to incorporate this a Wh-Analysis will be required, thus an additional layer for question adaption will be required. Wh-word is basically the question word which is one of who, why, where, which, when, how, what and rest. “rest” are the questions that don't have any question word. Example :- Name of a disease that cause bowel bleeding?. This process segregates questions into different types and considers the type of question for answer generation, a recommended approach for this would be to encode question type information into a one hot vector which is a trainable embedding vector [6], and it is incorporated in the training process.

For the purpose of implementation the popular deep neural network library tensor flow along with the python programming language will be used. For the purpose of syntactic analysis the Standford core NLP parser will be used.Python enables us to carry out string manipulation easily unlike other programming languages since user inputs are text based it will be the preferred choice of language.The choice of the above mentioned technology is due to the widely available community support and free distribution of the software.

Work Flow diagram of the question preprocessing component is given below.

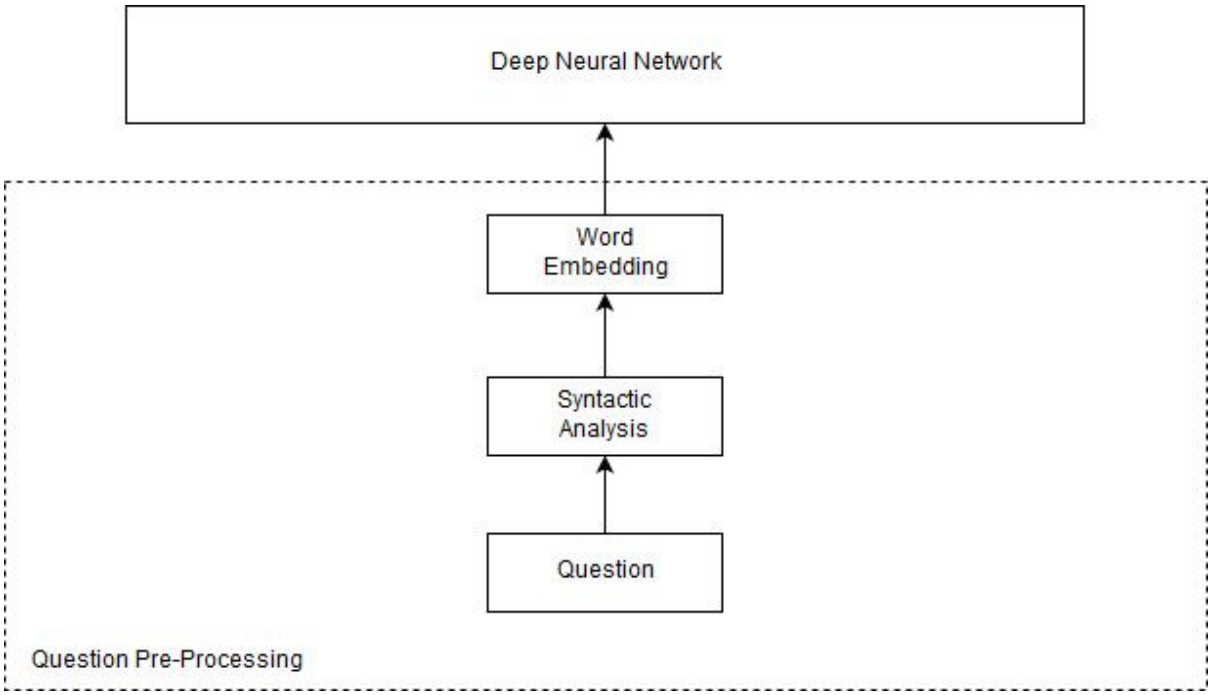


Figure 1.0 Question pre-processing module

2.3 Testing and Implementation

2.3.1 Sources for test data

For the purpose of the QA component the Cornell Movie Dialogue dataset was used to evaluate the pre-trained word embedding model and tweak and measure its performance although the Furthermore, a small test data set with domain specific questions was compiled to perform integration tests with the entire system.

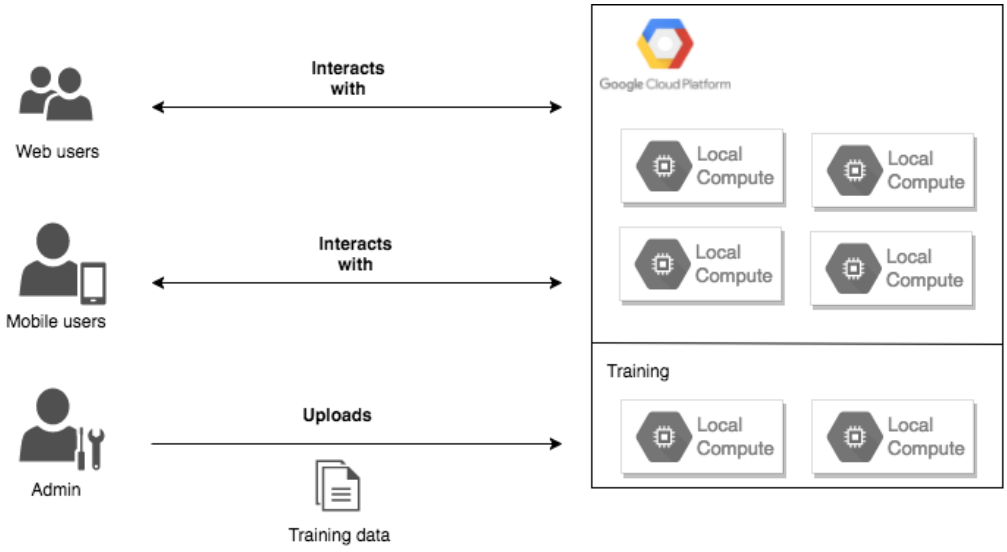
2.3.2 Evaluation and Analysis of results

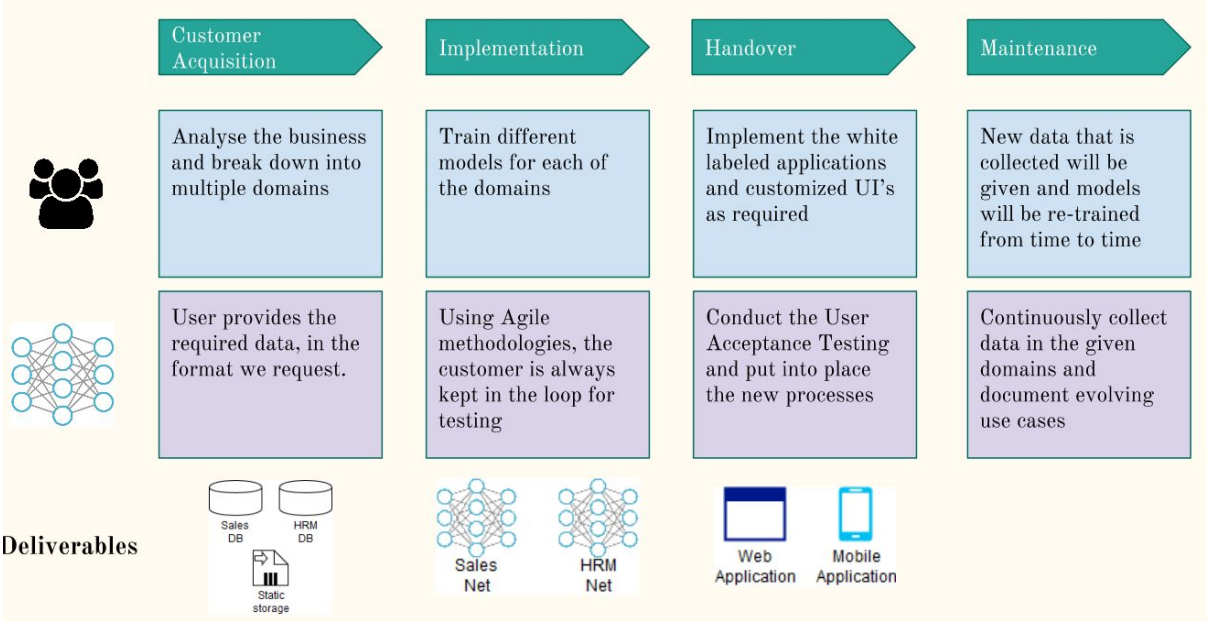
Since word embedding is one of the vital parts in QP it is relevant to focus on techniques that try to evaluate word embedding models. Thus, a paper by Tobias Schnabel, Igor Labutov. David Mimno, Thorsten Joachims presents a solid evaluation approach for word embedding layers [10]. They categorize the evaluation approach to four main types,

1. Relatedness: These datasets contain relatedness scores for pairs of words;
2. Analogy: This task was popularized by Mikolov et al. (2013a). The goal is to find a term x for a given term y so that x : y best resembles a sample relationship a : b.
3. Categorization: Here, the goal is to recover a clustering of words into different categories
4. Selectional preference: The goal is to determine how typical a noun is for a verb either as a subject or as an object (e.g., people eat, but we rarely eat people)

We can use the similar approach to analyze and evaluate our QP model.

2.4 Commercialisation





Todo:

- Discuss concepts for multiple models, uploading trainig data, automating the training process, white labeled applications, switching between models
- Discuss backend cloud architecture
- Add screenshots of application

3.0 RESULTS AND DISCUSSION

3.1 Results

3.2 Findings

3.3 Conclusion and future work

REFERENCES

[1] A. Frank, H. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg and U. Schäfer, "Question answering from structured knowledge sources", *Journal of Applied Logic*, vol. 5, no. 1, pp. 20-48, 2007.

[2] P. Gupta and V. Gupta,*p* "A Survey of Text Question Answering Techniques", *International Journal of Computer Aplications*, vol. 53, no. 4, pp. 1-8, 2012.

[3]"On word embeddings - Part 1", *Sebastian Ruder*, 2017. [Online]. Available: http://sebastianruder.com/word-embeddings-1/. [Accessed: 01- May- 2017].

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

[5] Ruder, Sebastian. "On Word Embeddings - Part 3: The Secret Ingredients Of Word2vec". *Sebastian Ruder*. N.p., 2017. Web. 26 Apr. 2017.

[6]J. Zhang, X. Zhu, Q. Chen, L. Dai and H. Jiang, "Exploring Quesiton Understanding and Adaptation in Neural-Network-Based Question Answering", 2017.

[7] "Noahs-ARK/semafor", *GitHub*, 2017. [Online]. Available: https://github.com/Noahs-ARK/semafor. [Accessed: 27- Apr - 2017].

[8] N. England, "NHS England » Emergency Care Data Set (ECDS)", *England.nhs.uk*, 2017. [Online]. Available: https://www.england.nhs.uk/ourwork/tsd/ec-data-set/. [Accessed: 01- May- 2017].

[9] "UCI Machine Learning Repository: Data Sets", *Archive.ics.uci.edu*, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html. [Accessed: 01- May- 2017].

[10] T. Schnabel, I. Labutov, D. Mimno and T. Joachims, "Evaluation methods for unsupervised word embeddings", 2017.

APPENDIX I: SYSTEM ARCHITECTURE DIAGRAM

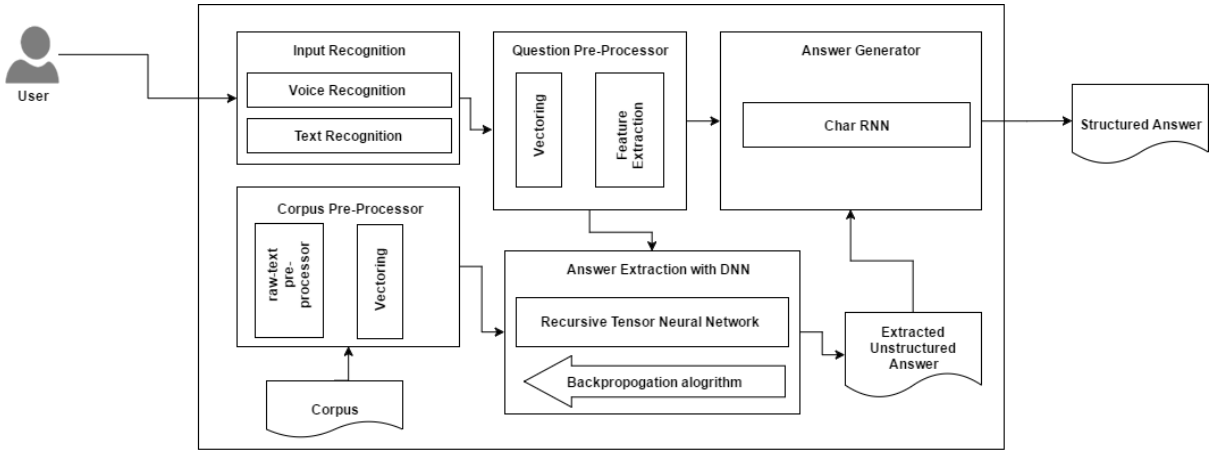


Figure 3.0 System Architecture of Adaptive Artificial Intelligent Question Answer

APPENDIX II: SPECIFIC TASKS TO BE FOCUSSED

1. Requirements Gathering
2. System Design
3. Research
4. Implementation
5. Web Interface
6. Mobile Application (Android only)
7. System Testing
8. Continuous Integration and Deployment

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.