Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 785–794

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Deep Learning Approaches for Question Answering System

Yashvardhan Sharma[a] , Sahil Gupta[a]

[a] Department of Computer Science Information System
Birla Institute of Technology Science, Pilani Campus Pilani-333031

Abstract

Question Answering (QA) System is very useful as most of the deep learning related problems can be modeled as a question answering problem. Consequently, the field is one of the most researched fields in computer science today. The last few years have seen considerable developments and improvement in the state of the art, much of which can be credited to upcoming of Deep Learning. In this paper, a discussion about various approaches starting from the basic NLP and algorithms based approach has been done and the paper eventually builds towards the recently proposed methods of Deep Learning. Implementation details and various tweaks in the algorithms that produced better results have also been discussed. The evaluation of the proposed models was done on twenty tasks of babI dataset of Facebook.

*Keywords:*coattention; deep learning; memory nets; neural networks; question answering; word vectors

Cossospending authors: yash@pilani.bits-pilani.ac.in

1877-0509 © 2018 The Authors. Published by Elsevier Ltd.

10.1016/j.procs.2018.05.090

1. INTRODUCTION

The problem of making a fully functional question answering system is one problem which has been quite popular among researchers. The new algorithms, especially deep learning based algorithms have made a decent progress in text and image classification. But, these machines have still failed to solve the tasks which involve logical reasoning. One best example of such problems is the question answering problem. It is only recently that with the introduction of memory and attention based architectures there has been some progress in this field. But still, with the ongoing development, there is much larger scope for improvement. In this paper, the analysis of, all the major deep learning algorithms for question answering, has been done.

1.1. Background

A question answering system implementation usually a chat bot can construct an answer when put up with a query. One of the earliest and most successful implementation of chat bot has been ALICE Bot, developed using AIML.

There were many similar applications released at that time but the major breakthrough was when Natural language Processing was used in order to solve the question answering task [1]. This solution proved to be a major improvement in the existing accuracy of the systems and since then all the models have tried to build on the basis of Natural Language Processing only. In the last few years, the Natural Language Processing solutions have been overshadowed by deep neural networks which essentially mimic the human brain and tend to produce better results.

1.2. Motivation

Question Answer systems are very useful as they allow users to enter a query based on some facts or stories and the system tries to use the context in the supporting facts and stories to answer the questions effectively instead of just giving out the best-suited keywords. Besides, most of the problems in artificial intelligence and Natural language processing can all be modeled as a question answering problems. For example, the task of text summarization can be modeled as a question answering task in the sense that if the user asks the system "What is the summary of the text?", it can answer the user by providing the appropriate summary. We aim to design a system that can basically be used to solve most of the machine learning problems. This serves as motivation to analyze research and improve all the solutions that have been proposed so far.

1.3. Objective

This work intends to use Deep Learning and other Artificial Intelligence algorithms in order to solve the Question Answering System problem. Here, we have analyzed and reviewed various algorithms in order to make a Question Answering System. The use of recurrent neural networks allows us to expand and apply this model to a variety of question answering tasks. For start we use a basic AIML chat bot which can easily be used to solve fact-based question answering task, then use LSTM models and finally use different types of memory networks and get an accuracy as high as 98 percent on some of the tasks of babI dataset [2].

2. RELATED WORK

2.1. Natural Language Processing

There are several NLP models that have been applied in this field. One of the major models is [3]. In this model, the authors have proposed to use POS tagging [4] and tf-idf [5] concept in order to match the query with questions already present in Yahoo Answers. One major issue with this is that all the questions may not be present on the internet and hence those questions cannot be answered. The work in [6] gives a score to each of the sentences in order to retrieve all the significant sentences from the text. They now do similarity matching between the retrieved sentences and user queries in order to solve the problem. This proves to be a fairly good approach but the answers are not always appropriately framed in order to solve the query

2.2. Deep Learning

Several Deep Neural Network models have been applied to NLP tasks. These models have mostly used Recurrent Neural networks like LSTMs and GRUs for text classification, summarization [7]. This work is inspired by the models in [8], [9] and [10]. In [8] baseline LSTM model has been proposed in order to solve the problem, the model has a very less accuracy but none the less provides a basis to build upon. The major breakthrough is Memory networks model (Weston et al.) [9] which proposed the use of memory in the system in order to effectively answer the questions. The model in [10] (Dynamic Memory Networks) combines the paradigms of memory networks and attention mechanism in order to overcome the shortcomings of the memory networks in [9]. Very recently the model in Xiaong et al [11] was proposed in order to use the Dynamic memory Networks of [10] for Visual Question answering tasks.

Besides the NLP and Deep Learning techniques of solving the question Answering tasks specifically, the work in Mikolov et al [12], the model also known as word2vec is perhaps the most significant model [16]. It proposes a way to map words into usable data structures called word vectors while also catching their context correctly.

3. DATASET

For our system, we have used the babI dataset by Facebook in order to train our deep neural network models. The set has been divided into 20 different files corresponding to the types of questions as described in [2].

The twenty types of questions include:

- **Single Supporting fact**- This includes questions where each question has an answer based on only a single supporting fact. e.g. "Mary traveled to the office. Where is Mary?"
- **Two or three supporting facts**- This includes questions where each question has an answer based on two supporting facts. e.g. To answer “Where is the football?” one has to combine information from two sentences “John is in the playground” and “John picked up the football”.
- **Two or Three argument relations**: We need to be able to recognize a relation between subjects and objects so as to be able to answer the questions which are based on these relations. e.g. "What is the bedroom north of?"
- **Yes/No Questions**- This category includes the questions which have answers of the form Yes/No. e.g. “Is John in the playground?”
- **Counting and Listing tasks**- This category includes the questions related to counting. e.g. “How many objects is Daniel Holding?"
- **Simple Negation and Indefinite knowledge**- This category includes questions about facts which are of negative form or the questions about abstract knowledge of things like using the words like "maybe"
- **Time Reasoning**- These questions include time expressions. E.g. “Where was Julie before the park?”
- **Basic Deduction and Induction**- This category includes questions like "Sheep are afraid of wolves. Gertrude is a sheep. What is Gertrude afraid of?"
- **Positional and Size Reasoning**- This includes questions which have positional and size based data. E.g. “Is Football bigger than a tennis ball?"
- **Path Finding**- The questions of this task includes finding the path between two points. These questions effectively involve a search problem
- **Agent's Motivations**- To answer why an agent performs an action. For example, the system must learn that hungry people might go to the kitchen.

This dataset is generated synthetically and divided into various tasks so as to comprehensively develop the understanding and logical reasoning of our machine. Each model should be evaluated on each of the tasks and should never be tuned per task to obtain definite results.

In addition to providing these 20 tasks in the English language, the tasks were also provided in Hindi in order to train our model in the multilingual fashion which is out of the scope of this paper.

4. METHODOLOGY AND ANALYSIS

The work started with the basic model of a simple chat bot which can be effectively used to answer fact-based queries or the queries that have already been answered on the internet. This was implemented using AIML (Artificial Intelligence Markup Language) Chat bot. Then the actual babi dataset by Facebook was explored and we started using deep Learning Architectures. First of all, we use LSTM baseline model and then move on to improve it by using memory networks and dynamic memory networks.

4.1. AIML Chat Bot Implementation

AIML chat bot, also known as alice bot is a very simple to use chat bot which could be customized to operate for answering simple FAQ type of questions. It has the following basic tags.

- **<aiml>**: This tag begins and ends the AIML chatbot document
- **<Category>**: This tag marks a "unit of knowledge" in the bot's knowledge base.
- **<Pattern>**: This tag is used to contain a simple pattern that matches user input given to an Alicebot. These pattern tags store the context of the conversation and are really useful for a successful chat bot implementation.
- **<Template>**: This tag contains the response to a user input.

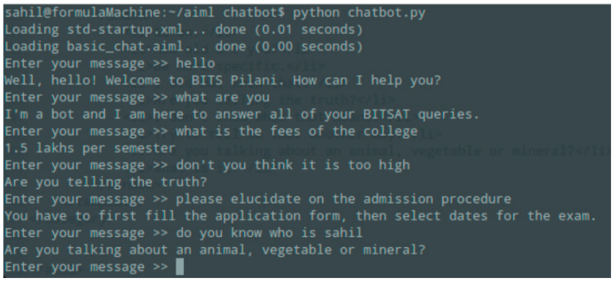

Figure 1: Implementation of AIML chatbot for university queries

4.2. LSTM Baseline

Before going into LSTMs or any NLP(Natural language Processing Architecture) it is important to understand the model of word vectors. The model Word2Vec has been used in this work [12]. As described in [12], "*Word2Vec is a semantic learning framework that uses a shallow neural network to learn the representations of words/phrases in a particular text. Simply put, it is an algorithm that takes in all the terms (with repetitions) in a particular document (divided into sentences) and outputs a vector form of each.*" The main benefit of word2vec model was that the word vectors are able to capture the context of the text very effectively. So, when implementing the complex deep learning models we don't need to worry about these aspects of our model. Word2Vec comes in two flavors:

- **Continuous Bag of Words (CBOW)**: The input to the model could be w_{i-2}, w_{i-1}, w_{i+1},w_{i+2} i.e. the preceding and following words of the current word we are at and the output of the neural network will be w_i. Hence we can think of the task as predicting the word given its context.
- **Skip Gram:** The input to the model is w_i and the output could be w_{i-1},w_{i-2},w_{i+1},w_{i+2}. So the task is to predict the context given a word.

Now coming onto LSTM (Long Short-Term Memory networks).Here is an excerpt about LSTM from [13]"*LSTMs are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter Schmidhuber (1997) and were refined andpopularized by many people in the following work. They work tremendously well on a large variety of problems and are now widely used. LSTMs [13] are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.The following figure shows the exact architecture of LSTM networks".*

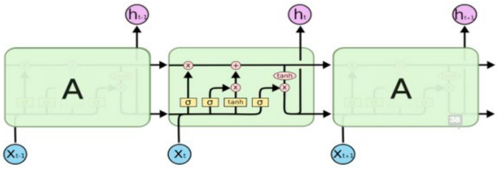

Figure 2: LSTM Architecture(source:[13])

The LSTM baseline model information flow: The stories and questions are first passed through word2vec models and their respective embedding is obtained. The question embedding is then passed through a recurrent neural network and is further added to the embedding of stories. The added information is then passed through another LSTM the encoded vectors from which are finally passed through softmax layer that makes answer predictions. The following figure pictorially describes the flow of information.

The implementation details for the model are described in section 5 and the results of same are described in Section 6. The LSTM Baseline model provides an accuracy of around 53% on single fact-based questions and 20% on 2 or more facts based questions. It was noticed that the proposed baseline model had an asymmetric nature, so a modification was proposed to make the model symmetric. As it turned out, the modification did prove to be an improvement. But, as evident from the results, the accuracy was found not to be good enough and needed an improvement. So, memory networks were proposed in order to increase accuracy.

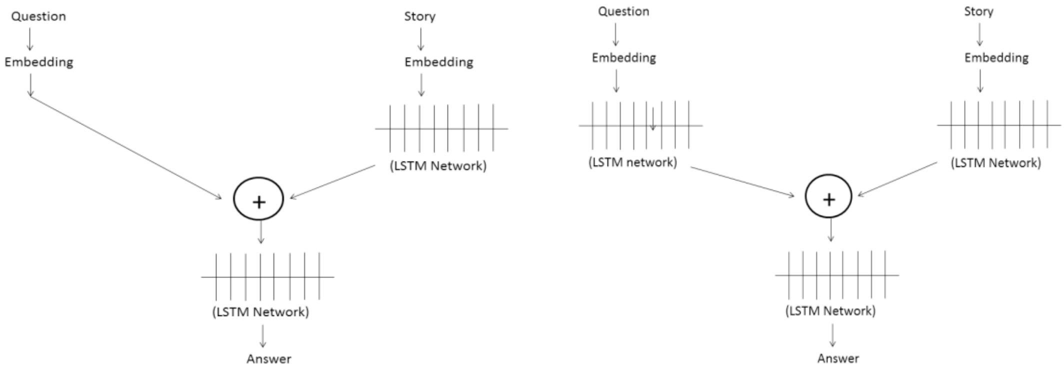

Figure 3: (a) LSTM baseline architecture as in [1], (b) Modified LSTM Baseline architecture

4.3. Memory Networks

The biggest problem with the normal LSTM model was that the neural network did not have any place where it could store all the information from the text. So, Weston et al proposed memory networks that incorporate this problem by essentially providing the memory to the network. The description of the network is as follows.

Input Representation: Lets assume we are given an input set $x_1, x_2, \ldots\ldots, x_i$ which is needed to be put in memory. The whole set of the x_i vectors is converted into the desired vectors of memory m_i, let say of a of dimension d. These memory vectors are computed by embedding each of the x_i in a Cartesian space, in the simplest case, by using an embedding matrix A (of size $d*V$). The query vectors q are also embedded (via another embedding matrix B with the dimensions same as A) to obtain an internal state u. In the embedding space, the match between u and each of the memory vector m_i is computed by taking the inner product of these vectors followed by a *softmax*.

Output memory representation: Each of the x_i has a corresponding output vector c_i (given by another embedding matrix C). The response from the memory, o is a sum over the transformed inputs c_i, multiplies by weights corresponding to the probability vector from the input.

$$O = \sum p_i . c_i \quad (1)$$

Generating the final prediction: In the single layer case, the sum of the output vector o and the input embedding u is then passed through a final weight matrix W (of size V X d) and a *softmax* to produce the predicted label.

$$\text{Output} = Softmax(W(o+u)) \tag{2}$$

The full architecture of memory networks is given in Figure 4. The model proved to be a great success of single fact-based questions and provided around 96 % accuracy on single fact-based questions but this accuracy drops drastically for complex tasks. To solve this, an improved memory network i.e. Dynamic Memory Network [10] has been proposed.

4.4. Dynamic Memory Networks

The problem with the memory networks was that they only allowed a single pass at the memory before answering the question. This is like learning all the things and then trying to answer everything in one go. The Dynamic memory networks addressed this issue and provided the provision of having multiple passes at the input memory (known as episodes) before answering the query and hence giving the model to answer more effectively. The description of the network is as follows.

Input Module: The input module of the network essentially encodes text inputs from the babI task into a distributed vector representation. Here, the input may be anything-a long story, a sentence, a news article, a movie review, or several Wikipedia articles. Word embedding obtained from the embedding matrix is given as input to the recurrent network. At each time step t, the hidden state of the network is updated. Here L is the embedding matrix and w_t is the index of the word$t$ of the input sequence.

$$h_t = \text{GRU}(L[w_t], h_{t-1}) \tag{3}$$

Question Module: Just like in the case of input story, the question is also given as a words sequence in NLP problems. Again just like in the case of input, the question is encoded via a recurrent neural network. When queried with a question of T_Q words, the question encoder hidden states at any time$t$ is given by

$$q_t = GRU(L[w^Q_t], q_{t-1}) \tag{4}$$

L represents the word embedding matrix and w^Q_t represents the index of the $t$he word in the question.

Episodic Memory Module: The episodic memory module iterates over the output of the input module, while updating its internal episodic memory. The episodic memory module comprises of an attention mechanism as well as a GRU with which it updates its memory. For each iteration, this attention mechanism attends over the fact representations c while taking into consideration the question representation q and the previous memory m_{i-1} to produce an episode e_i. The episode is then used, alongside the previous memories m_{i-1} to update the episodic memory.

$$m_i = GRU(e_i, m_{i-1}) \tag{5}$$

It is usually beneficial for the episodic memory module pass over the input multiple times. After T_M passes, the final memory left is given to the answer module. The detailed architecture is given in figure 5. The dynamic memory networks give the highest accuracy and the least error among all the models represented above. The further implementation and results are discussed in section 5 and 6 respectively

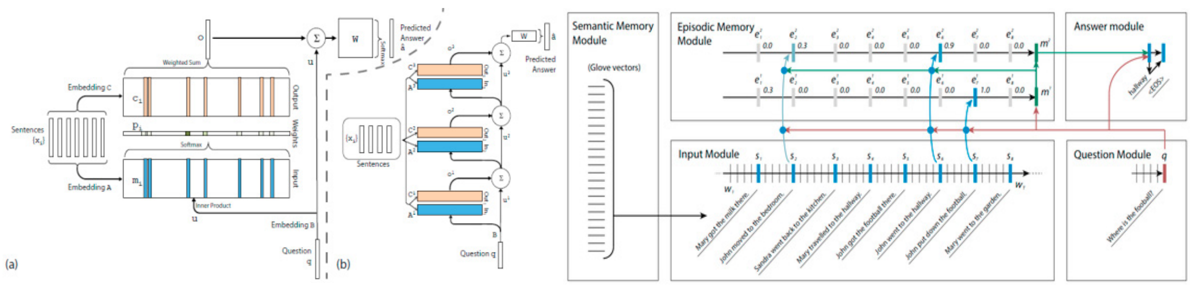

Figure 4: (a) Memory Network architecture(source:[9]); (b) Dynamic memory network architecture(Source: [10])

5. IMPLEMENTATION

The dataset used in our implementations of the model as mentioned earlier is the babI dataset. Preprocessing of the data was done in MATLAB and python. Python was used as the main language to implement the models along with the general deep learning libraries namely keras, pytorch, and tensorflow. Some tweaks were made to the already defined models while training in order to obtain better results.

5.1. Preprocessing

The babi dataset is a natural language dataset and can't be used in its exact form. It needs to be converted in a suitable data structure in order to use it for further computation and processing.

Tokenize- First step in the pre-processing is to tokenize the sentences into different words. For example, 'Bob dropped the apple. Where is the apple?' is tokenized to ['Bob', 'dropped', 'the', 'apple', '.', 'Where', 'is', 'the', 'apple', ' ?']

Splitting into Story, Questions, and answers: Next, the sentences were split into stories, questions and answers so that they can be fed to the proposed models.

Combining all the stories- All the stories were then combined up to the point that the question was asked. This finally becomes the story for that particular question.

Indexing the stories, questions, and answers- Finally, the questions and stories are indexed according to their time of occurrence and are eventually processed via word2vec model. The answers are transformed to one hot encoded vector.

5.2. LSTM Implementation

The Keras library was used in order to implement the architecture mentioned in the previous section and in [2]. The numbers of LSTM neural network blocks used were 50. Besides this, while training a dropout was used of 50% in order to avoid overfitting. The dataset was trained on 40 epochs and we got an accuracy of roughly 48%.

Next, an asymmetric nature of the model was noticed wherein the questions are encoded using LSTMs but the stories were not. The architecture was manipulated a little bit and just like the question vectors, the story embedding were also passed through an LSTM network to obtain symmetry. This improved accuracy to around 53%.

5.3. Memory Networks

The implementation of the memory networks [8], [9] was again done on Keras. The layers used were word embedding, recurrent LSTMs, dense layers, softmax, dropout of 50%. The network was trained for 120 epochs with Adam optimizer and categorical cross entropy Loss Function. There were tries to change the optimizer to RMS Prop and the loss function to sparse categorical cross entropy but the results obtained did not show any major improvement. In the original paper the final memory output o and u have a summation relationship before going into the softmax function; in this work, LSTM network was added after the summation which resulted in an improvement in accuracy.

5.4. Dynamic Memory Networks

For the implementation of dynamic memory networks [10], [11], the library pytorch was used as keras cannot handle the dynamic nature of memory. The network was agained for 120 epochs with Adam Optimizer and categorical cross entropy function. But for this, a callback of early stopping was used which if there was no change in accuracy for 10 epochs stopped the execution. While training, the callback was called at 45th epoch. The error was as low as 0.006 and the accuracy of 99.2% was achieved on single fact-based questions.

It was observed that the implementation results proved our hypothesis of the difference in models. It was hypothesized that Memory Networks are an improvement over LSTM Baseline and the Dynamic memory Networks are a further improvement over memory Networks which proved to be a correct hypothesis. The possible reasons for these results are discussed in the next section.

6. RESULTS AND DISCUSSIONS

The specifications of the models are described in table 1.

POS tagging(+tf-idf) was used for the first task and gave an accuracy of 37% which significantly dropped to about 4-5% for the other tasks, and so, the results haven't been included in the test results table. Table 2 shows the test accuracies of all the models on the tasks of babI dataset

The modified LSTM version used in this work are an improvement over other LSTM baselines. The drastic improvement achieved in memory networks from LSTM can be attributed to the fact that the memory networks have the memory where they can store stories and then answer questions. Mathematically, it was observed that in LSTM, the error function while training got stuck in a local minimum and never came out of it. The same happened with the memory network as it gave the same accuracy for about 15 consecutive epochs but due to the better capabilities of memory networks, they were able to get out of the minimum at some point of time. The improvement from memory Network to Dynamic memory Network can be solely credited to having multiple episodes i.e. giving multiple chances to our models to have a look at the input story.

Table 1: Specification of the various models used

Algorithm	Questions (Type)	Error Function	Optimizer
POS Tagging+ tf-idf	Fact-Based	NA	NA
LSTM Baseline	Single Fact Support	Categorical Cross Entropy	Adam
Memory Networks	Easy tasks of BabI	Sparse Cross Entropy	rmsProp
Dynamic Memory Networks	Most of the tasks of BabI	Categorical Cross Entropy	Adam

7. LIMITATIONS AND SCOPE

In this paper, neural network based framework for general question answering tasks have been proposed that are trained using raw input-question-answer triplets. Generally, these frameworks can solve sequence tagging tasks,classification problems, sequence-to-sequence tasks andquestion answering tasks that require transitive reasoning.

Out of the three architectures highlighted in this study, the Dynamic memory network architecture is a potentially general architecture for a various NLP applications, applications that include classification, question answering and sequence modeling [14]. A single architecture is a first step towards a single joint model for multiple NLP problems. The DMN is trained end-to-end with

The models and the study is currently restricted to small question answer, and fails to perform well on large information based questions, answering which include reasoning. Also, in this paper, the babI dataset was used in order to evaluate all the models. The babI dataset is an artificially generated dataset by Facebook which is very general in nature. So, the architectures may give some weird result on the use of natural dataset.

8. CONCLUSIONS

This work focused on analyzing, implementing and improving the various popular methodologies that have been proposed in the field of Question Answering. Such analysis is important for anyone looking to venture into the field.

The dataset (babI dataset which is a standard synthetic dataset released by Facebook) was used and thoroughly analyzed as the dataset is the most important thing in any machine learning application. It has been quite evident that the Deep Learning is the way forward for the field to improve as they prove to be a major improvement over the general pre-deep learning era techniques.

In this paper, we observe that LSTM baseline models are only useful for the tasks which are simple, generic, and do not require much of processing. Giving the memory to the network, as done in memory networks, serves as a major improvement, but the model has its shortcomings because it is allowed to have only a single pass over the story for answering questions and just as with human brain, the network struggles with tasks that require too much of knowledge from the story. Dynamic memory networks which allow multiple passes over story solve this problem and hence solve most of the babI tasks. It can be observed that even the DMNs struggle with tasks 17 and 19 which require extensive reasoning ability

A lot of work still remains to be done in the field so as to achieve full reasoning capability. There is a need to look at ways to solve ambiguous questions because they are actually very common in real life scenarios. There isalso a need to device a model that can serve as multilingual question answering system so as to handle the queries in all the languages. Besides this, the developments in the field have been rapid and there is a need tocarefully analyze all that is going on. Xiaong et al have proposed the advanced Dynamic Memory Networks also known as DMN+. These networks also have the ability to solve visual data-based questions.

Table2: Test accuracies on the different tasks of the babI dataset

Task Number	**LSTM Baseline**	**Memory Networks**	**DynamicMemory Networks**
1: Single Supporting fact	52%	96%	99.2%
2: Two Supporting facts	26%	67%	94%
3: Three Supporting facts	23%	42%	90%
4: Two Argument Relations	64%	94%	99.4%
5: Three Argument Relation	66%	86%	97%
6: Yes/No Questions	44%	74%	99.1%
7: Counting	47%	81%	93%
8: Lists/Tests	42%	79%	91%
9: Simple Negation	68%	83%	97%
10: Indefinite Knowledge	44%	67%	92%
11: Basic Coreference	63%	88%	99.6%
12: Conjunction	76%	92%	99.6%
13: Compound Coreference	91%	91%	99.8%
14: Time Reasoning	29%	74%	96%
15: Basic Deduction	23%	98%	100%
16: Basic Induction	24%	44%	92%
17: Positional Reasoning	50%	56%	57%
18: Size Reasoning	54%	90%	95%
19: Path Finding	6%	7%	31%
20: Agent's Motivation	89%	99.1%	100%
Mean Accuracy	49%	75%	91%

References

[1]Meera Udani, Avinash Shrivas, Vaibhav Shukla (2013) "Question Answering System Based on Artificial Intelligence for Restricted Domain", *International Journal of Engineering Research & Technology*

[2]J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint: 1502.05698, 2015.*

[3]Tait Larson, Johnson(Heng) Gong, Josh Daniel "Providing a Simple Question Answering System By Mapping Questions to Questions.",*Technical report, Department of Computer Science,Stanford University, 2006*

[4]Leon Derczynski, Alan Ritter, Sam Clark, Kalina Bontcheva,(2013) "Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data " *Proceedings of Recent Advances in Natural Language Processing, pages 198–206*

[5]J. Ramos(2003) "Using TF-IDF to Determine Word Relevance in Document Queries"*Technical report, Department of Computer Science, Rutgers University, 2003*

[6]Sanglap Sarkar, Venkateshwar Rao, Baala Mithra SM, Subrahmanya VRK Rao(2015)"NLP Algorithm Based Question and Answering System*", Proceedings of 2015 Seventh International Conference on Computational Intelligence, Modelling and SimulationPages 97-101*

[7]M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. 2017. Text Summarization Techniques: A Brief Survey. *ArXiv e-prints (2017). arXiv:1707.02268*

[8]Darshan kapashi, Pararth Shah(2014)"Answering Reading Comprehension Using Memory Networks", *Technical report, Department of Computer Science, Stanford University*

[9]Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus(2015) "End to End Memory Networks", *Arxiv e-prints(2015) arXiv:1503.08895v5 [cs.NE]*

[10]Ankit Kumar, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher (2016)"Ask Me Anything, Dynamic MemoryNetworks", *Arxiv e-prints(2016)- arXiv:1506.07285v5 [cs.CL]*

[11]Caiming Xiong, Stephen Merity, Richard Socher(2016) "Dynamic Memory Networks for Visual and Textual Question Answering", *Arxiv e-prints (2016)- arXiv:1603.01417*

[12]Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean(2013) "Efficient Estimation of Word Representations in Vector Space", *Arxiv e-prints (2013). arXiv:1301.3781v3 [cs.CL]*

[13] H. Sak, A. Senior, K. Rao, F. Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition", *INTERSPEECH, 2015*.

[14]Andreas, J., Rohrbach, M., Darrell, T., and Klein, D.Learning to Compose Neural Networks for Question Answering. *arXiv preprint (2016) arXiv:1601.01705*.

[15]M. Malinowski and M. Fritz. "A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input". *NIPS, 2014.*

[16]Pennington, Jeffrey, Richard Socher, and Christopher D Manning. (2014)"Glove: Global vectors for word representation". *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*

[17]Samuel R. Bowman, Christopher Potts, and Christopher D. Manning. (2015)"Recursive neural networks can learn logical semantics". *Proc. of the 3rd Workshop on Continuous Vector Space Models and their Compositionality.*