International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 982–995

A Bi-model MemN2N Network for Complex Question Answering Task

R.Poonguzhali[1], Dr.K.Lakshmi[2]
[1]Research Scholar, [2] Professor
Department of Computer Science and Engineering
Periyar Maniammai Institute of Science and Technology.

Abstract

Question Answering (QA) system is a field of Natural language processing, in which the users can post query in their own languages. The system also gives precise answer instead of list of documents. A memory network has the ability to perform reasoning with inference components and long-term memory component. The two components are used efficiently to find the answers from the story context for a given query. In our earlier work [26] we evaluated the performance of MemN2N network with complex and easy question answering tasks and found that the MemN2N fail to produce good results with some complex QA tasks of bAbI dataset. This work intends to improve the performance with a state of the art Bi-Model end to end memory network (BiMemN2N_I) model for such complex QA tasks and compare its performance with the standard MemN2N model and MemNN models.

In this work, a Bi-model MemN2N Network based question answering system is implemented and its performance is evaluated with a complex question answering tasks from bAbI dataset. In addition, the performance of training and testing with suitable metrics are studied and identified the difference in the performance of two question answering tasks.

Keywords: *NLP, QA, Deep learning, RNN, LSTM, GRU, Memory Networks, MemNN, MemN2N, End to End Memory Networks, bAbI Tasks.*

I. INTRODUCTION

Natural language processing (NLP) is a sub-division of artificial intelligence in which machines interpret, recognize and infer meaning from human language. Human language is very difficult to understand due to its ambiguity. This ambiguity of languages makes NLP a challenging problem in computer science [1]. Using NLP; we can make machines think like human with deep understanding of language.

Question Answering (QA) system is a field of Natural language processing, which returns precise answer to the user's question. This system is useful because the users can give the question in natural language. Problems that are discussed mostly in natural language processing are represented as QA problems [8]. The challenge of existing QA systems is to understand the natural language questions and context correctly, to give the exact answer. The complexity of QA system is due to the dynamic nature of languages [11].

There are four major approaches in the implementation of Question Answering Systems. They are Rule-Based, Statistical, Machine learning, Deep learning. The implementation of NLP methods increasingly depends on data - driven strategies that help to create a more efficient and scalable model[12].

Rule-based approach is the earliest type of QA systems. This approach needs a tremendous human effort to devise rules and linguistic tools [13]. These rules are manually written based on lexical and semantic features. A deep knowledge about the language is also required to write the rules. The main disadvantage of this method is time consuming. [14] [15].

In the recent days, there has been a significant increase in web documents or online repositories. This gives way to statistical approaches. This approaches work with vast volume of data and different types of resources [16]. These algorithms requires large amount of data for training the model. Once the model gets trained, it produces better performance than other models. In addition, the learned model can be used for any other applications i.e. independent of any language. The main disadvantage of this method is to recognize the linguistic features for sentences [17].

Machine learning has the ability to learn easily towards the QA systems. These systems build the taxonomy from the training data and this taxonomy is used to give the answer. Thus this system shows its independence among other systems like Rule based approaches and Statistical approaches. In addition, these systems are enhancing over time, it becomes one of the successful solution for QA [18].

Recently Neural Networks provide the promising results in QA systems that brings machine Intelligence and human cognition very closer [19].

First Question answering systems - BASE BALL (1961) and LUNAR (1972) which answer the questions in structured database. The Question Answering domain started emerging in the year 1999. Due to the challenge for Open domain question answering in Text Retrieval Conference (TREC), there comes lot of opportunities for QA systems. Even though Current QA systems deal with simple factual questions, more system needed to answer complex questions. One such question is temporal question [7].

QA system answers the questions in natural language automatically and accurately. The user gets the answer directly from the system instead of searching in the documents. The QA system may be categorized as Restricted Domain Question Answering (RDQA) and Open Domain Question Answering (ODQA). Restricted Domain QA system answer questions from any pre- defined knowledge base. Open Domain QA system focuses on answering questions from a vast amount of documents from any domain which may be semi-structured or unstructured [3].

Natural language processing (NLP)

The field of NLP is a sub division of AI and computational linguistics. It is an emerging area of research as it deals with Human Computer Interaction. The problem of understanding the human language is to understand the concepts delivered from the combination of words. It is very difficult for the machines to understand. The ambiguity and indefinite characteristics of natural languages make NLP difficult to implement. This area was dominated by rule - based methods implemented by linguistic features, then statistical methods and in recent days deep learning are very successful in this field.

Natural Language Processing involves following techniques, lexical analysis, parsing, semantic analysis, discourse integration, and pragmatic analysis. Still NLP is facing lot of challenges in Human Computer Interaction and it has opened to many opportunities for using the techniques in robotics, chat bots, automation and digital transformation [1].

The popular applications of NLP are Question answering, Automatic summarization, Natural language generation, Natural language understanding, Speech recognition.

Deep Learning

Deep Learning is a subdivision of machine learning which with a collection of algorithms which has the ability to find a best solution to any problem given a sufficiently large input dataset.

In [22], the authors present the power of deep learning approaches in the field of NLP. They are:

- The deep learning approaches in natural language processing can be applied to replace existing linear models with replacement models that can achieve commensurate or better performance.
- The deep learning approaches build completely new models to solve the problems easily in NLP.
- The deep learning approaches extract and learn the features from natural language automatically without any manual intervention.

- There is continuous increase in performance of deep learning models in natural language processing and also rapid improvement on challenging tasks.
- Huge end-to-end trained deep learning models give better performance for natural language problems.

The bAbI tasks for NLP Research

It is a collection of 20 tasks and the individual task is able to test the ability of different learning model. Basically it is done by testing the different aspects of text understanding and reasoning. The task contains training and test data. So that the test data is used to test the performance of the model. Refer [4] for more information about these tasks. This dataset contains several directories such as English, Hindi and Shuffled letters. Each directory contains 1000 and 10000 training and testing data.

Each task has the following file format:

```
ID sentence
ID sentence
ID sentence
ID question[tab]answer[tab]supporting fact IDS.
...
…
```

About this work

This work intends to improve the performance with a state of the art Bi-Model MemN2N (BiMemN2N) model for such complex QA tasks and compare its performance with the standard MemN2N model and MemNN models.

In this work, a Bi-model MemN2N network based question answering system is implemented and its performance with a complex QA tasks is evaluated using bAbI dataset. The performance of training and testing is studied with suitable metrics and also the performances for two QA tasks are shown.

About Keras

Keras is an API executed on the top of Tensorflow. The code was written with python. This API was designed by a google engineer François Chollet .It is very easier to debug. This API is user friendly as it reduces the number of user actions. It is designed in modularized fashion which can be extended to create new models. This makes Keras suitable for advanced research.

The keras can be used for the following reasons:

- Easy to build the model
- It supports neural networks.
- The same code can be executed both in CPU and GPU flawlessly.

The high-Level API keras deals with the models created, sharing layers, multiple input and multiple output models. Backend engine performs computational graph with the help of other libraries such as Tensorflow or Theano. Tensorflow is the default "backend engine" in keras but it can also be changed in the configuration.

About Tensor Flow

Tensor-Flow is an end to end open source platform. It has set of tools and libraries which can be used for easy deployment of Ml applications. The architecture is flexible. It can be executed easily through a range of platforms like CPUs, GPUs, and TPUs. It was designed by Google Brain team members. This platform helps to work more in the area of machine learning and deep learning. Tensor-Flow is optimized to run faster on GPUs than CPUs.

II. MODELING

Modeling a QA system

Training a Typical QA System

The following diagram shows the typical process of training a questing answering system. Using a word-vector dictionary, the training story texts, question texts and their corresponding answers texts will be vectorized and a deep learning network will be trained with the vectorized training data.

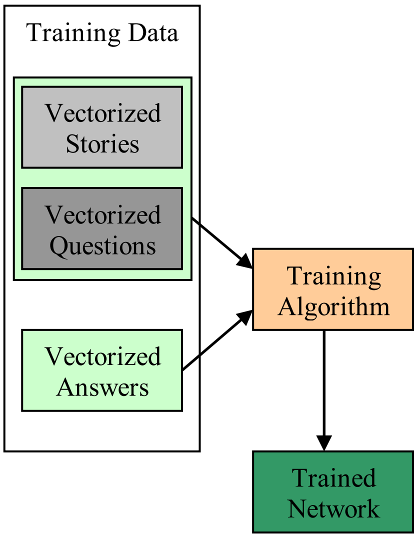

Figure 1. The Training Process

Testing a Typical QA System

The following diagram shows the typical process of testing or validating a questing answering system. The test story texts, and question texts will be vectorized and fed in to the trained network and the network will predict the possible answer vectors. The actual answer test from the vectorized answers will be created using reverse lookup in the word-vector dictionary.

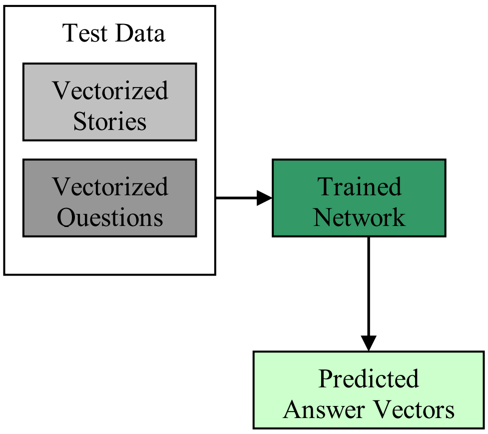

Figure 2. The Testing Process

The Memory Network (MemNN) [6]

In [6] the authors proposed a memory network. Memory networks combine inference components and long-term memory component efficiently. The long-term memory can be used for prediction. This model solves the problem of long term dependencies faced by RNN.

Memory network consists of a long term memory m and four learnable components. The four components are Input, Generalization, Output, and Response. Consider the input which is given to the chosen model. The steps are:

- The input x is converted into I(x) i.e. vector representation using standard processing techniques.
- The Generalization component adds the new input in the old memories mi :

  m_i = G (m_i, I (x), m) for all i.
- The output is generates the new output: o = O (I (x), m).
- At last, the output features o is decoded into desired format: r = R (o).

The above steps are applied both at training and testing time, if there is a difference between two phases, then the four components are not changed.

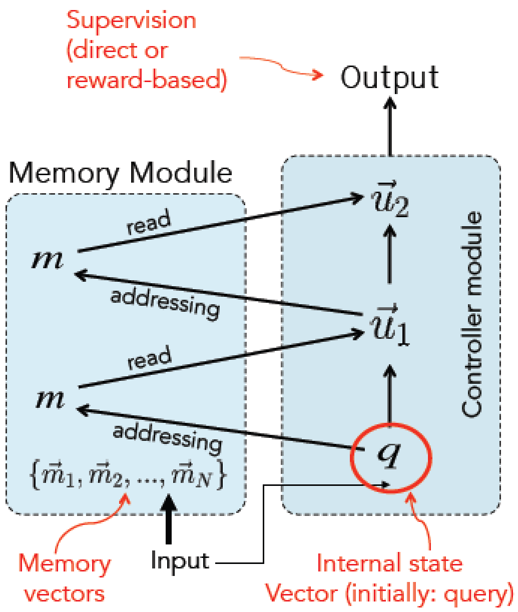

Figure 3. MemNN Model

MemNN Implemention

I : Input is converted to vector reprentation using bag-of-word-embeddings x.

G : x is stored in next free slot m_N.

O : Continue looping with all memories k=1 or 2 times:

1st loop max: finds the best match m_i with x.

2nd loop max: finds the best match m_J with (x, m_i).

The output o is represented with (x, mi, m_J).

R : ranks all words seen by the model o and returns highest scoring word.

Training :

Training in this model is performed with a margin ranking loss and stochastic gradient descent. Particularly, for a given query x with answer r and supporting sentences m_{O1} and m_{O2} (when k = 2), the model parameters are minimized U_O and U_R:

Minimize:

$$\sum_{\bar{f}' \neq m_{O_1}} \max(0, \gamma - s_O(x, m_{O_1}) + s_O(x, \bar{f})) +$$
$$\sum_{\bar{f}' \neq m_{O_2}} \max(0, \gamma - s_O([x, m_{O_1}], m_{O_2}]) + s_O([x, m_{O_1}], \bar{f}'])) +$$
$$\sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, m_{O_1}, m_{O_2}], r) + s_R([x, m_{O_1}, m_{O_2}], \bar{r}])) + \quad \text{……………………..(1)}$$

Where:

S_O is the matching function for the Output component.

S_R is the matching function for the Response component.

x is the input question.

m_{O1} is the first true supporting memory (fact).

m_{O2} is the first second supporting memory (fact).

r is the response

True facts and responses m_{O1}, m_{O2} and r should have higher scores than all other facts and responses by a given margin.

γ is the margin and $\bar{\mathbf{f}}$, $\bar{\mathbf{f}}'$ and $\bar{\mathbf{r}}$ are all predicted labels. At each step of SGD, $\bar{\mathbf{f}}$, $\bar{\mathbf{f}}'$, $\bar{\mathbf{r}}$ are sampled instead of computing the whole sum for each training sample.

Suppose if RNN is used for the R component of MemNN the final term was changed with the standard log likelihood like a language modeling task.

The End to End Memory Network (MemN2N)

Input memory representation:

This model is described using a single layer. The inputs x_1, .., x_i are stored in memory. Then the inputs are converted into vector representation $\{m_i\}$ of dimension d using an embedding matrix. Similarly the query q is also converted using another embedding matrix to get u which is the internal state. Next the match is determined by finding the product between u and mi. Then softmax function is applied.

$$p_i = Soft\max(u^T m_i) \quad \text{……………………………….(2)}$$

Where

$$Soft\max(z_i) = e^{z_i} / \sum_j e^{z_j} \quad \text{…………………(3)}$$

and *p* is a probability vector of the inputs.

Output memory representation:

The response vector o is determined by the sum of transformed inputs c_i and probability vector from the input.

$$0 = \sum_i p_i c_i \quad \text{………………………………..……(4)}$$

It is easy to compute gradients and back-propagate through it.

SSN: 2233-7857 IJFGCN

Final prediction:

The final label is determined using the weight matrix W with sum of the output and the input. Finally softmax function is applied.

$$\hat{a} = Soft\max(W(o+u)) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

Multiple Layers:

The multiple layers are designed to handle multiple K hop operations. The input to the layer is the sum of o^k and u^k from layer k:

$$u^{k+1} = u^k + o^k \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

The individual layer has particular embedding matrices A^k and C^k .The final answer is predicted by combining the o^k and the u^k of the top layer.

$$\hat{a} = Soft\max(Wu^{k+1}) = Soft\max(w(o^k + u^k)) \ldots(7)$$

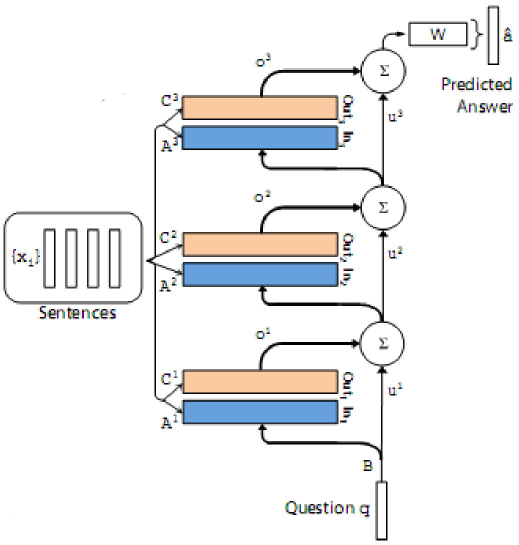

Figure 4. A 3 Layer MemN2N Model

The above figure depicts the three-layer memory model. The difference between the Memory Network model in MemNN[6] and MemN2N is the hard max operations is replaced with the softmax operations.

The Proposed Bi-model MemN2N network (BiMemN2N_I).

The following diagram shows the design of proposed bi-model MemN2N network.

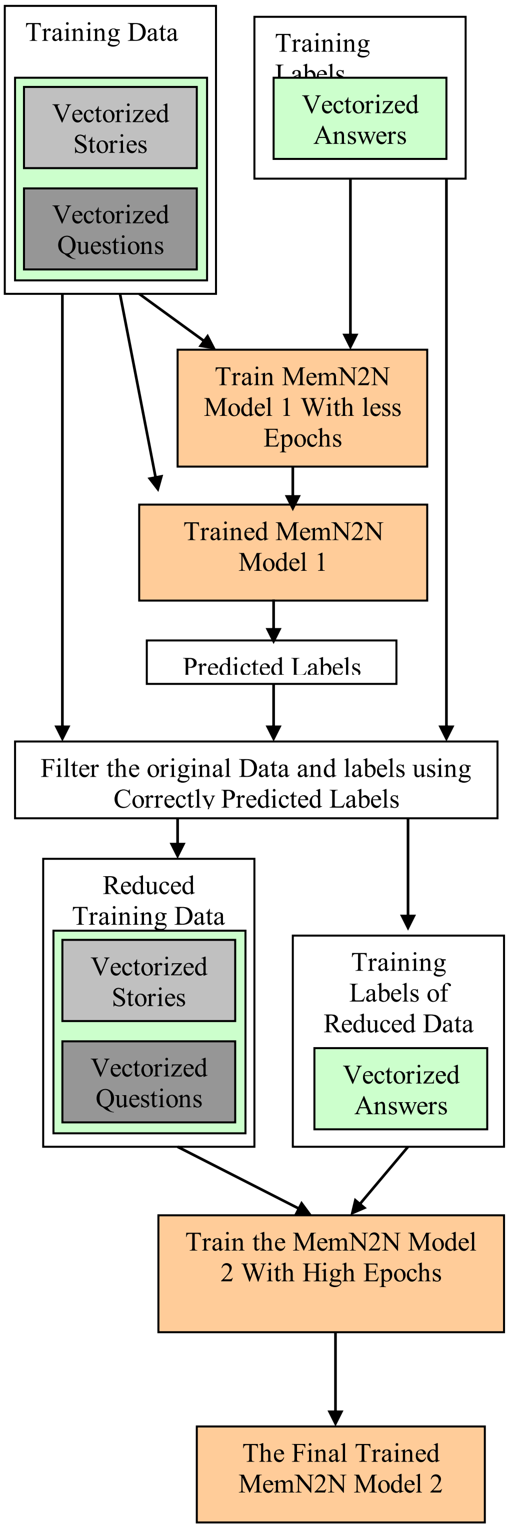

Figure 5. The Proposed Bi-model MemN2N Network

The Working of Bi-model MemN2N network.

In the proposed Bi-model MemN2N network (BiMemN2N_I), there will be two MemN2N networks involved in the training phase of the network. Both MemN2N Model 1 and the MemN2N Model 2 will have same network configuration but the first model will be training only for few epochs to achieve low accuracy and the second model will be trained to achieve maximum accuracy.

Steps involved in the process:

- First the whole training data is used to roughly train the first network MemN2N Model 1 only for 10% of original total epochs.
- Using the less trained MemN2N Model 1 predict the labels of the training data. (because of poor training, the network will be able to predict very low number of original data correctly)
- Using the index of the correctly predicted labels, separate only the data that belongs to the correctly predicted labels,
- Now, only using that portion of the data train the second model (MemN2N Model 2) upto to a maximum number of high epochs (here 150 epochs *).
- Now it will give a well trained MemN2N Model 2 which will capable of predicting the answers with improved accuracy.
- Using the MemN2N Model 2,predict the answers of the test dataset.
- Measure the testing performance in terms of accuracy and loss.

* Note: If the compared standard MemN2N Model consumes T seconds for N epochs of training, in this work, the MemN2N Model 2 will be trained upto that maximum time of T second to balance the time complexity. So in this case, we can run more than N epochs with in that time of T seconds.

III. THE RESULTS AND DISCUSSION

The experiments and evaluations are carried out on a laptop with Intel Core i7 Processor with 16 GB RAM. We didn’t use any GPU/TPU during training. We developed all the code in Python language (ver 3.5) with Keras and Tensorflow.

Performance Metrics Used:

Accuracy

It is a metric used to estimate the performance of a machine learning algorithm. The algorithm is trained using the training data and a classifier is developed. Testing data is used to test the classifier. The accuracy of a classifier on a given set tuples that are correctly classified by the classifier.

$$Accuracy = \frac{1}{n}\sum_{k=1}^{|G|}\sum_{x:g(x)=k} I(g(x)=\hat{g}(x)) \qquad \text{................(8)}$$

where I denotes the indicator function. If the classes match, this function gives 1 and 0 otherwise.

Results with Low Number of Training Samples

Training and Testing Parameters:

Training Data Directory : data/tasks_1-20_v1-2/en
Total Training Samples :1000
Total Testing/Validation Samples :1000
No Epochs :100
Training Batch Size : 32

Evaluating Performance with Small Training Data

SSN: 2233-7857 IJFGCN

The networks were trained with 1000 samples and tested with another 1000 samples. The following table shows the testing performance with 1000 samples of the complex QA Task ID- 16, 17 and 19 of bAbI dataset.

Overall Results

Training Parameters :

Total Training Samples :1000

Total Testing/Validation Samples :1000

Training Batch Size : 32

The following table shows the overall results of Task-19 from English text corpus.

Table 1. The Results with The bAbI tasks ID: 19

Model	No Trainable Parameters	Time Taken for Training (s)	Accuracy
MemNN With LSTM	23556	108	0.1
MemNN with Simple RNN	10404	68	0.097
MemN2N	14800	211	0.388
BiMemN2N_I	14800	200	0.510

The following table shows the overall results of Task-16 from English text corpus.

Table 2. The Results with The bAbI tasks ID: 16

Model	No Trainable Parameters	Time Taken for Training (s)	Accuracy
MemNN With LSTM	20984	69	0.44
MemNN with Simple RNN	7736	50	0.48
MemN2N	10800	182	0.715
BiMemN2N_I	10800	212	0.792

The following table shows the overall results of Task-17 from English text corpus.

Table 3. The Results with The bAbI tasks ID: 17

Model	No Trainable Parameters	Time Taken for Training (s)	Accuracy
MemNN With LSTM	19467	106	0.565
MemNN with Simple RNN	7851	62	0.564
MemN2N	8400	117	0.551
BiMemN2N_I	8400	135	0.582

The following bar chart compares the number of trainable parameters used in different network.

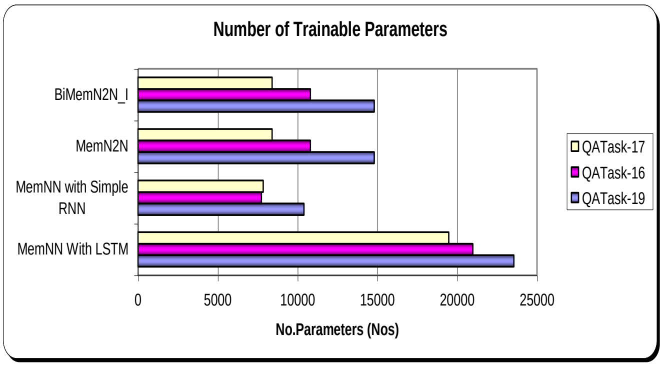

Figure 6. The Comparison of No. Parameters

The Network parameter size of the proposed BiMemN2N_I model is equal to that of standard MemN2N model. Even though, the Network parameter size of SimpleRNN based MemNN model is low, its performance is in term of accuracy is not better than that of BiMemN2N_I model

The following bar chart compares the performance in terms of training time.

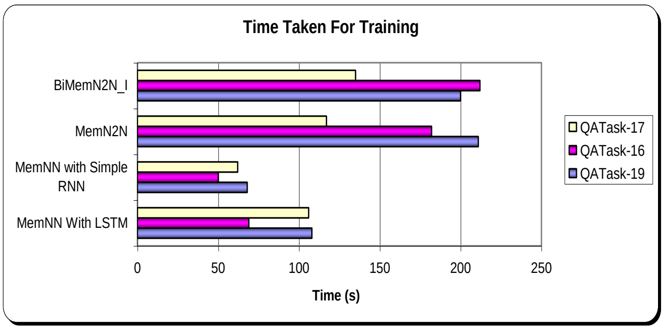

Figure 7. The Comparison of Training Time

The average time consumed for training the BiMemN2N_I is almost equal to that of the standard MemN2N model.

The following bar chart compares the performance in terms of accuracy.

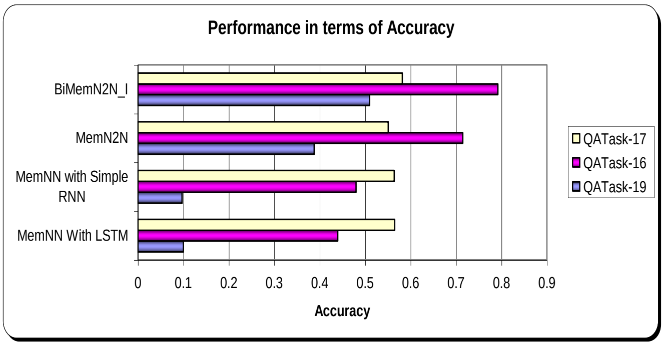

Figure 8. The Comparison of Accuracy

As shown in the above bar chart, the performance of BiMemN2N_I in terms of accuracy is higher than all other compared models. In the case of much complex QA Task 16 and QA Task 19, the acheived improvement was obvious and much good.

IV. CONCLUSION

From the results, we can clearly understand that the performance of BiMemN2N_I is getting improved significantly with the bi model training approach. With the BiMemN2N_I model, we achieved 5 to 10 % improvement in accuracy on some complex QA tasks(16, 19) at almost equal training time of standard MemN2N.

The proposed BiMemN2N model was able to select fine training data for achieving better training performance. The reduction in the data size and the selection of quality input data are the reason behind the improvement in performance in our model. The reduction in training data size opens much hope on better training within the same duration (of training a Standard MemN2N model).

So, in our future works, we will study more about other possibilities improving the bi-model memory networks and try to improve their performance on different kinds of complex QA tasks.

V. ACKNOWLEDGEMENT

We thank the Management of Periyar Maniammai Institute of Science & Technology (Deemed to be University) for providing us support and facilities to complete this work successfully.

REFERENCES

[1] Deeksha Dwivedi, Mahendra Singh Sagar, "Overview of Natural Language Processing", 4th International Conference on System Modeling & Advancement in Research Trends (SMART), 2015.

[2] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria,"Recent Trends in Deep Learning Based Natural Language Processing", arXiv:1708.02709v8 [cs.CL], 25 Nov 2018.

[3] Kolomiyets Oleksander and Marie-Francine Moens, A survey on question answering technology from an information retrieval perspective, in: Journal of Information Sciences volume 181, 2011.

[4] Jason Weston, Antoine Bordes, Sumit Chopra, Tomas Mikolov, Alexander M. Rush, "Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks", arXiv:1502.05698v10 [cs.AI] 31 Dec 2015, (conference paper at ICLR 2016).

[5] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus, "End-To-End Memory Networks", arXiv:1503.08895v5 [cs.NE] 24 Nov 2015.

[6] Jason Weston, Sumit Chopra & Antoine Bordes, "Memory Networks", arXiv:1410.3916v11 [cs.AI], 29 Nov 2015, (conference paper at ICLR 2015).

[7] Poonguzhali, K Lakshmi, "Temporal Question Answering System: A Survey", JETIR June 2019, Volume 6, Issue 6, ISSN-2349-5162.

[8] Yashvardhan Sharmaa, Sahil Guptaa, "Deep Learning Approaches for Question Answering System", International Conference on Computational Intelligence and Data Science (ICCIDS 2018).

[9] K.S.D. Ishwari, A.K.R.R.Aneeze, S.Sudheesan, H.J.D.A. Karunaratne, A. Nugaliyadde, Y.Mallawarrachchi, "Advances in Natural Language Question Answering: A Review", arXiv:1904.05276 [cs.CL], Apr 2019.

[10] Emily M. Bender, "Linguistically Na¨ive != Language Independent: Why NLP Needs Linguistic Typology", Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics, pages 26-32, Athens, Greece, 30 March, 2009.

[11] L. Kodra and E. Kajo, "Question Answering Systems: A Review on Present Developments, Challenges and Trends", International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, 2017

[12] E. Brill, J. Lin, M. Banko, S. Dumais and A. Ng, "Data-Intensive Question Answering", Trec.nist.gov, 2018.

[13] H. Madabushi and M. Lee, "High Accuracy Rule-based Question Classification using Question Syntax and Semantics", Aclweb.org, 2018.

[14] E. Riloff and M. Thelen, "A Rule-based Question Answering System for Reading Comprehension Tests", 2018.

[15] S. Humphrey and A. Brownea, "Comparing a Rule Based vs. Statistical System for Automatic Categorization of MEDLINE Documents According to Biomedical Specialty", 2018.

[16] S. K. Dwivedia and V. Singh, "Research and reviews in question answering system," in Proceedings of International Conference on Computational Intelligence: Modeling Techniques and Applications, 2013, pp. 417 - 424 .

[17] D. Cohn, Z. Ghahramani and M. Jordan, "Active Learning with Statistical Models", Journal of Artificial Intelligence Research, vol.4, 1996.

[18] X. Li and D. Roth, "Learning Question Classifiers", Dl.acm.org, 2018.

[19] Raghuvanshi, A., & Chase, P., "Dynamic Memory Networks for Question Answering".

[20] Ramesh Sharda et.el, "Business Intelligence and Analytics: Systems for Decision Support", Copyright© 2015, 2011, 2007 by Pearson Education, Inc., ISBN 10: 0-13-305090-4, ISBN 13: 978-0-13-305090-5

[21] Elvis, "Deep Learning for NLP: An Overview of Recent Trends", https://medium.com/dair-ai

[22] Yoav Goldberg, "A primer on neural network models for natural language processing", arXiv:1510.00726[cs.CL],
2 Oct 2015.

[23] R Poonguzhali, Dr K Lakshmi, " Evaluating the Performance of Recurrent Neural Network based Question Answering System with Easy and Complex bAbI QA Tasks", International Journal of Advanced Science and Technology, Vol. 29, No. 5s, ISSN: 2005 – 4238 pp. 1389-1402, (2020).

[24] R Poonguzhali, K Lakshmi, "Analysis on the Performance of Some Standard Deep Learning Network Models for Question Answering Task", Manuscript accepted for publication.

[25] R Poonguzhali, K Lakshmi, "Evaluating the Performance of Keras Implementation of MemNN Model for Simple and Complex Question Answering Tasks", In: Test Engineering and Management, Volume 82, ISSN: 0193 - 4120 pp. 9620-9629, (2020).

[26] R Poonguzhali, K Lakshmi, "Evaluating the Performance of MemN2N Model for a Complex Question Answering Task", Manuscript submitted for publication.

[27] R Poonguzhali, K.Lakshmi , "Analysis on the Language Independent and Dependent Aspects of Deep Learning based Question Answering Systems" , In: International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-9 Issue-6, April 2020.