Classification of Question using Attention Based Convolution & GRU Neural Network

BITS ZG628T: Dissertation

by

Shamal Winchurkar

2014HT13001

Dissertation work carried out at

NVIDIA India Graphics Pvt. Ltd, Pune

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

PILANI (RAJASTHAN)

April 2019

Classification of Question using Attention Based Convolution & GRU Neural Network

BITS ZG628T: Dissertation

by

Shamal Winchurkar

2014HT13001

Dissertation work carried out at

NVIDIA India Graphics Pvt. Ltd

Pune

Submitted in partial fulfillment of M.Tech. Software Systems degree programme

Under the Supervision of

Nikhil Joshi,

NVIDIA Graphics India Pvt. Ltd

Bangalore

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

PILANI (RAJASTHAN)

April 2019

CERTIFICATE

This is to certify that the Dissertation entitled Classification of Question using Attention based Convolution & GRU Neural Network and submitted by Shamal Winchurkar having ID-No. 2014HT13001 for the partial fulfillment of the requirements of M.Tech. Software Systems degree of BITS, embodies the bonafide work done by him under my supervision.

[illegible]

Place : Bangalore

Date : 08/04/2019

Signature of the Supervisor

Nikhil Joshi,

Engineering Manager,

Compute Software,

NVIDIA India Pvt. Ltd., Bangalore

Birla Institute of Technology & Science, Pilani

Work-Integrated Learning Programmes Division

Second Semester 2018-2019

BITS ZG628T: Dissertation

ABSTRACT

BITS ID No.	**:** 2014HT13001
NAME OF THE STUDENT	**:** Shamal Winchurkar
EMAIL ADDRESS	**:** 2014ht13001@wilp.bits-pilani.ac.in
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	**:** NVIDIA India Pvt. Ltd., Pune
SUPERVISOR'S NAME	**:** Nikhil Joshi
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	**:** NVIDIA India Pvt. Ltd., Pune
SUPERVISOR'S EMAIL ADDRESS:	nikhilj@nvidia.com
DISSERTATION TITLE	**:** Classification of Question using Attention Based Convolution & GRU Neural Network

Broad Academic Area of Work: Machine Learning

[illegible]

Signature of the Student
Name: Shamal Winchurkar
Date: 08/04/2019
Place: Pune

[illegible]

Signature of the Supervisor
Name: Nikhil Joshi
Date: 08/04/2019
Place: Bangalore

ABSTRACT

Classifying question into right category is crucial step in any Question-Answer System. There are many models build on various paradigms to solve this problem like SVM based models, manual feature extraction based models and deep learning based models. This work explores the various deep learning models based on Convoulation Neural Network and GRU with Dynamic Attention to solve this problem and build models to design classifier which effectualy classify questions into right category with maximum accuracy. CNN are very good in extracting local features from the input and representating higher level features of the given input features. GRU are very good in encoding long term dependency information into feature vector. Dynamic attention focus on particular feature/features of the encoded feature vectors which plays important role in classifying given features vectors in particular catagory.

KEY WORDS

- Convolution Neural Network (CNN)
- Gated Recurrent Unit (GRU)
- Deep Neural Network (DNN)
- KERAS
- Tensorflow
- Python
- Attention
- MaxPooling
- Lambda
- Tanh
- Google News Embeddings (GN)
- Dimensions
- Dense Layer
- Softmax

ABBREVIATIONS

- CNN: Convolution Neural Network
- GRU: Gated Recurrent Unit
- DNN: Deep Neural Network
- GN: Google News Embeddings

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor/mentor Mr. Nikhil Joshi and my additional examiner Mr. Davendra Rawat for the continuous support of my dissertation study and research, for their patience, motivation, enthusiasm, encouragement, insightful comments and immense knowledge. Their guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better supervisor/mentor and additional examiner. Last but not the least, I would like to thank my family: my wife and son for their patience and support.

TABLE OF CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

NVIDIA is the world leader in Deep Learning and Artificial Intelligence Computing. It has released CUDA, cuDNN, TensorRT and DIGITS SDKs for Deep Neural Network and Machine Learning acceleration on GPU Parallel computing in recent years.

The project under discussion has explored the specific NLP application "Question type classification" of Deep Neural networks like CNN, GRU/LSTM and Dynamic Attention and harness the computing power and enhancements of GPU Parallel computing by implementing the model in Tensorflow Keras APIs.

For any Question & Answer type application, classifying the question into right category is crucial and directly affects the quality of answers. Following are the broad categories of questions.

Table 1: Question Categories

Primary Classes	Sub Classes
ENTITY	TECHNIQUE, TERM, SUBSTANCE, RELIGION, SYMBOL, WORD, LANGUAGE, COLOR, SPORT, BODY, FOOD, PRODUCT, PLANT, CREATIVE, LETTER, VEHICLE, EVENT, DISEASE, INSTRUMENT, ANIMAL
NUMERIC	WEIGHT, TEMPERATURE, DISTANCE, COUNT, CODE, MONEY, PERCENT, PERIOD, DATE, SPEED, SIZE
LOCATION	CITY, COUNTRY, MOUNTAIN, STATE
HUMAN	INDIVIDUAL, TITLE, GROUP
DESCRIPTION	DEFINITION, REASON, MANNER
ABBREVIATION	EXPRESSION

Any question always has some focus words for example "What, Why, How, Where, When, Which, date, money, time, location, etc" which plays important role in understanding the ask in the question. We can use Dynamic Attention mechanism to focus on those word-features in question context in combination with CNN and GRU/LSTM to classify question type along with other features. We have used NVIDIA's GPU Cloud Platform to accelerate these models on Tesla V100 GPU.

In present work, we build and train the following models step by step on top of word vectors obtained from an unsupervised neural language model. These vectors were trained by Mikolov et al [2013b] on 100 billion words of Google News and are publicly available.

- CNN.
- Attention Aware BGRU Network
- Attention Aware CNN-BGRU Network

2 Related Work

In recent years, many models based on deep neural network are constructed upon input word sequences for text classification. Among these models, CNN and RNN (LSTM/GRU) are most popular ones. CNN networks are very good in extracting local correlation and high-level features and LSTM/GRU networks are very good in encoding long term temporal dependency information.

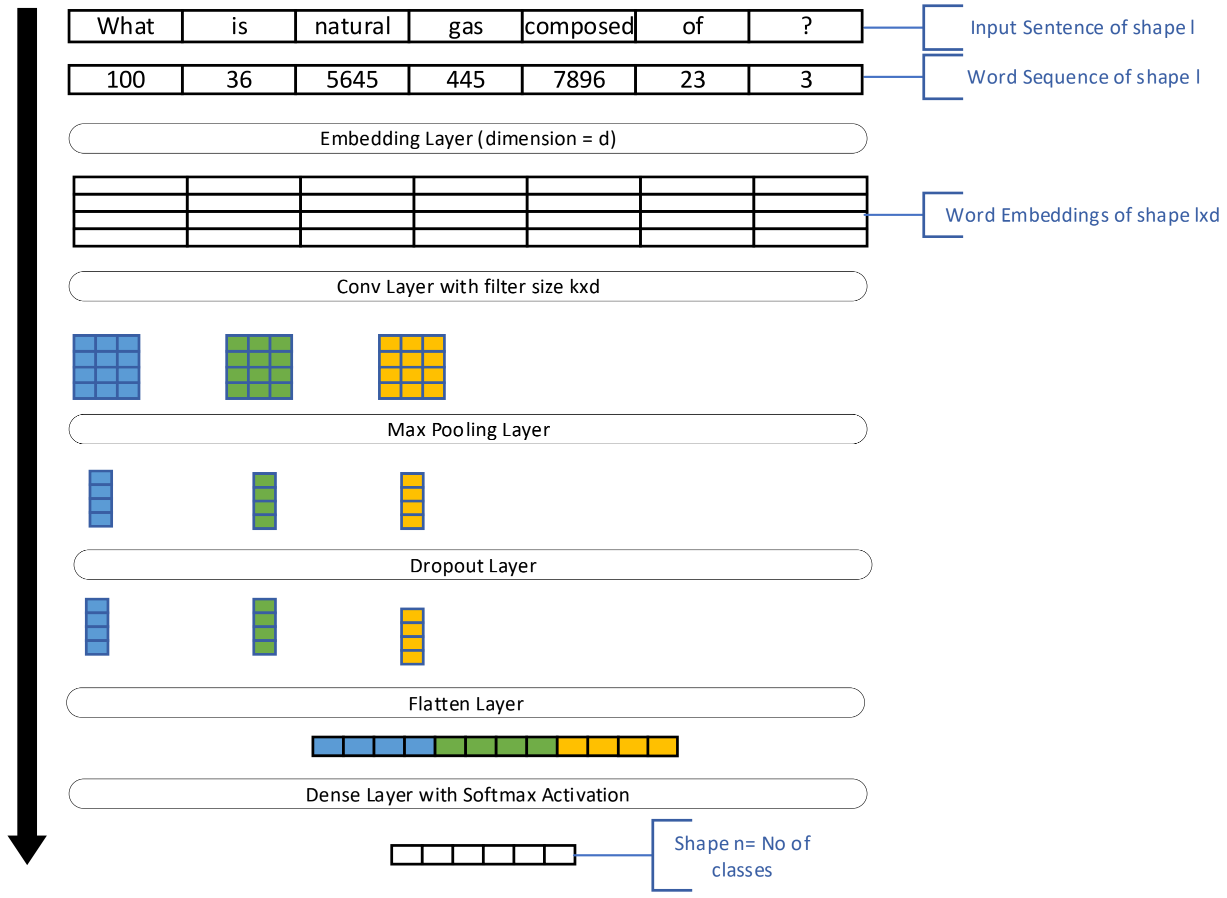
CNN model for text classification is presented by Yoon [2014] and C-LSTM model for text classification is presented by Chunting Zhou et al [2015]. C-LSTM model combines CNN with LSTM to extract local correlation and high-level features and encode their long-term temporal dependencies and use dense layer to classify encoded vectors into desired categories.

Attention-Based Bidirectional LSTM model for relation classification is presented by Peng Zhou et al [2016]. This model implements dynamic attention layer after LSTM layer to focus the attention of the network on words features which are playing important role in classifying the relation in particular category.

3 MODELS

3.1 CONVOLUATION NEURAL NETWORK

The model architecture is shown in figure 1.



What
is
natural
gas
composed
of
?
100
36
5645
445
7896
23
3
Input Sentence of shape l
Word Sequence of shape l
Embedding Layer (dimension = d)
Word Embeddings of shape lxd
Conv Layer with filter size kxd
Max Pooling Layer
Dropout Layer
Flatten Layer
Dense Layer with Softmax Activation
Shape n= No of classes

Figure 1: Convolution Neural Network

3.1.1 Embedding Layer

Let S be the input question sentence padded to make all sentences to fixed length of l and $x_i \in R^d$ be the d-dimentional word vectors for the i-th word in a word sequence. Let $x = \{ x_i \mid x_i \in R^d, 1 \leq i \leq l\}$ denotes the embedding vectors of input word sequence as output of embedding layer.

3.1.2 Convolution Layer

The 1-D convolution layer uses filter vector sliding over a word vector sequence and detects features at different positions. Let k be the length of the filter and vector $m \in R^{k \times d}$is a filter. For each position j in the sequence, we have a window vector $w_j = \{x_j, x_{j+1}, \dots, x_{j+k-1}\}$ with k consecutive word vectors.

The filter m convolves with the window vectors at each position and generates a feature map $c = \{c_j | c_j \in R^d, 1 \leq j \leq g\}, where\ g = l - k + 1$.

Each element c_j of the feature map is calculated as follows:

$$c_j = f\left(w_j{}^\circ\, m + b\right), 1 \leq j \leq g$$

Where ° is element-wise multiplication and addition, $b \in R$ is the bias term and f is nonlinear activation function ReLU.

The feature map $c = \{c_j | c_j \in R^d, 1 \leq j \leq g\}$ is the output of Convolution layer and it is feed to MaxPooling Layer.

3.1.3 MaxPooling Layer

MaxPooling layer applies a formula $p_{i\times d} = \max(c_j, c_{j+1}, \ldots c_{j+i-1})$ and generates the output $p = \{p_{ixd}\}$. It is feed to Dropout Layer.

3.1.4 Dropout Layer

Dropout layer randomly sets the weights of the neurons to 0 at different positions at every step in the layer to avoid over fitting.

3.1.5 Flatten Layer

Flatten layer takes the input of shape nxd and convert it into a one-dimension shape t = nd. This 1-D shape is feeds to Dense layer for regression computation.

3.1.6 Dense Layer with Softmax Activation

This layer calculates the actual predictions probabilities using fully connected layer with Softmax activation function.

$$o = f(dot(w^T, t) + b)$$

$$w = \{w_j\} \ are\ weights$$

$$t = \{t_i\} \ is\ the\ input\ vector$$

$$b\ is\ the\ bias$$

$$f\ is\ the\ nonlinear\ softmax\ function$$

$$o\ is\ the\ output\ predection\ vector\ of\ dimension\ j = no\ of\ classes$$

3.2 Attention aware GRU Network

The model architecture is shown in figure 2.

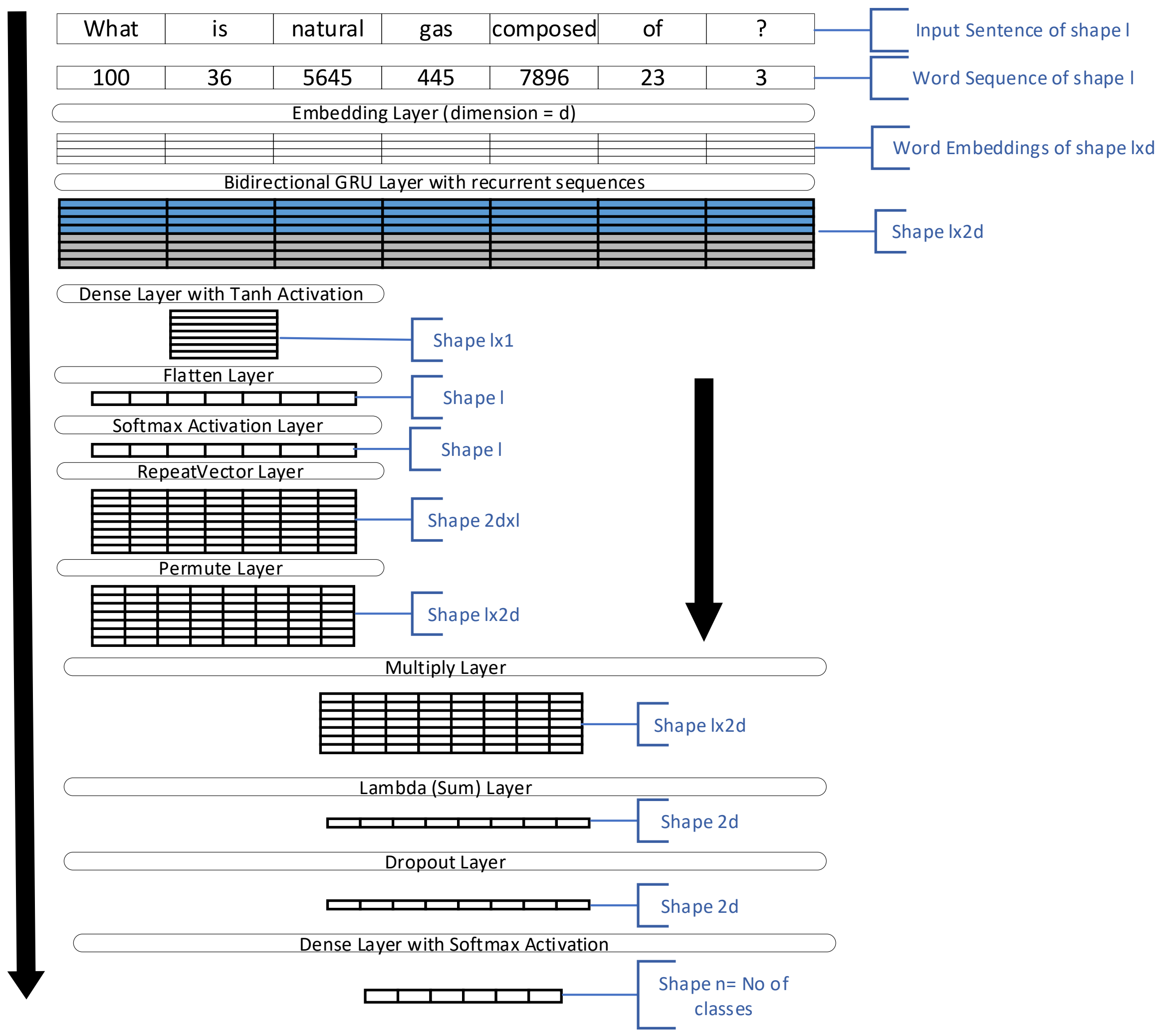


Figure 2: Attention Aware BGRU Network

3.2.1 Embedding Layer

Let S be the input question sentence padded to make all sentences to fixed length of l and $x_i \in R^d$ be the d-dimentional word vectors for the i-th word in a word sequence. Let $x = \{ x_i \mid x_i \in R^d, 1 \leq i \leq l\}$ denotes the embedding vectors of input word sequence as output of embedding layer.

3.2.2 Bidirectional GRU Layer

GRU is explicitly designed for time-series data for learning long-term dependencies. Bidirectional GRU with recurrent sequences layer process input from both the directions and generate $l \times 2d$ output tensor, where d is the embedding dimension and l is length of word sequence.

The output of BGRU layer is feed to Attention Layer (sequence of layers) for Attention vector calculation and to Multiply Layer for inference calculation.

3.2.3 Attention Layer

Let $H = \{h_i | \ h_i \in \ R^{2d}, 1 \leq i \leq l\}$ that BGRU layer produced, where l is the word sequence length.

We generate attention vector $\propto$ by applying first dense layer with tanh activation function and then softmax activation layer.

$$M = \tanh(H)$$

$$\propto = softmax(w^T M)$$

3.2.4 Multiply and Lambda Layer

Multiplying layer multiplies $H = \{h_1, h_2, h_3, \dots, h_l\}$ with $\propto$ element wise and Lambda layer sum over l dimension to produce final tensor which it feeds to dropout layer.

$$r = H \propto^T$$

3.2.5 Dropout Layer

Dropout layer randomly sets the weights of the neurons to 0 at different positions at every step in the layer to avoid over fitting.

3.2.6 Dense Layer with Softmax Activation

This layer calculates the actual predictions probabilities using fully connected layer with Softmax activation function.

$$o = f(dot(w^T, t) + b)$$

$$w = \{w_j\} \; are \; weights$$

$$t = \{t_i\} \; is \; the \; input \; vector$$

$$b \; is \; the \; bias$$

$$f \; is \; the \; nonlinear \; softmax \; function$$

$$o \; is \; the \; output \; predection \; vector \; of \; dimension \; j = no \; of \; classes$$

3.3 Attention aware CNN-BGRU Network

The model architecture is shown in figure 3.

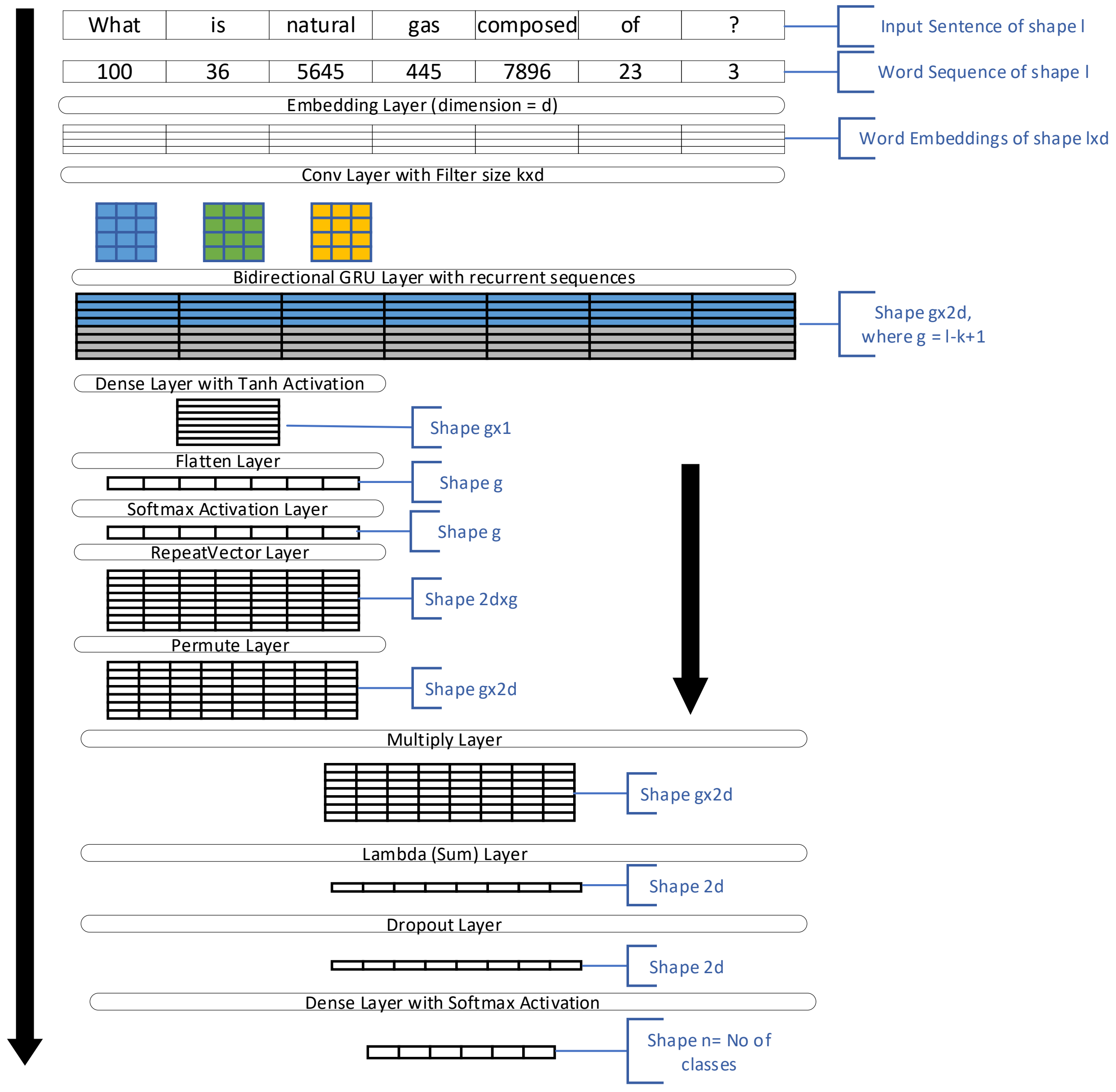


Figure 3: Attention Aware CNN-BGRU Network

3.3.1 Embedding Layer

Let S be the input question sentence padded to make all sentences to fixed length of l and $x_i \in R^d$ be the d-dimentional word vectors for the i-th word in a word sequence. Let $x = \{ x_i \mid x_i \in R^d, 1 \leq i \leq l\}$ denotes the embedding vectors of input word sequence as output of embedding layer.

3.3.2 Convolution Layer

The 1-D convolution layer uses filter vector sliding over a word vector sequence and detects features at different positions. Let k be the length of the filter and vector $m \in R^{k\times d}$is a filter. For each position j in the sequence, we have a window vector $w_j = \{x_j, x_{j+1}, \dots, x_{j+k-1}\}$ with k consecutive word vectors.

The filter m convolves with the window vectors at each position and generates a feature map $c = \{c_j \middle| c_j \in R^d, 1 \leq j \leq g\}, where\ g = l - k + 1$.

Each element c_j of the feature map is calculated as follows:

$$c_j = f\left(w_j{}^{\circ}\, m + b\right), 1 \leq j \leq g$$

Where ° is element-wise multiplication and addition, $b \in R$ is the bias term and f is nonlinear activation function ReLU.

The feature map $c = \{c_j \middle| c_j \in R^d, 1 \leq j \leq g\}$ is the output of Convolution layer and it is feed to BGRU Layer.

3.3.3 Bidirectional GRU Layer

GRU is explicitly designed for time-series data for learning long-term dependencies. Bidirectional GRU with recurrent sequences layer process input from both the directions and generate $g \times 2d$ output tensor, where d is the embedding dimension and g is length of feature map c.

The output of BGRU layer is feed to Attention Layer (sequence of layers) for Attention vector calculation and to Multiply Layer for inference calculation.

3.3.4 Attention Layer

Let $H = \{h_i | \ h_i \in \ R^{2d}, 1 \leq i \leq g\}$ that BGRU layer produced, where g is the length of feature map c.

We generate attention vector ∝ by applying first dense layer with tanh activation function and then softmax activation layer.

$$M = \tanh(H)$$

$$\propto = softmax(w^T M)$$

3.3.5 Multiply and Lambda Layer

Multiplying layer multiplies $H = \{h_1, h_2, h_3, \ldots, h_g\}$ with ∝ element wise and Lambda layer sum over g dimension to produce final tensor which it feeds to dropout layer.

$$r = H \propto^T$$

3.3.6 Dropout Layer

Dropout layer randomly sets the weights of the neurons to 0 at different positions at every step in the layer to avoid over fitting.

3.3.7 Dense Layer with Softmax Activation

This layer calculates the actual predictions probabilities using fully connected layer with Softmax activation function.

$$o = f(dot(w^T, t) + b)$$

$$w = \{w_j\}\ are\ weights$$

$$t = \{t_i\}\ is\ the\ input\ vector$$

$$b\ is\ the\ bias$$

$$f\ is\ the\ nonlinear\ softmax\ function$$

$$o\ is\ the\ output\ predection\ vector\ of\ dimension\ j = no\ of\ classes$$

3.4 Loss Function

We are using Cross-Entropy Loss function to calculate the loss during training and minimizing the Cross-Entropy Loss. The error is defined as:

$$L(x^i, y^i) = \sum_{j=1}^{k} 1\{y^i = j\} \log(y_j^{\sim i}),$$

Where x^i is the training sample, $y^i \in \{1,2,3, \dots, k\}$ is the number of possible labels and $y_j^{\sim i} \in \{0,1\}$ for each label $j \in \{1,2,3, \dots, k\}$.

3.5 Optimizer

We are using RMSProp, a variant of SGD algorithm for back propagation. RMSprop divides the learning rate by an exponentially decaying average of squared gradients.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)(\frac{\delta C}{\delta w})^2$$

$$w_t = w_{t-1} - \frac{n}{\sqrt{E[g^2]_t}} \frac{\delta C}{\delta w}$$

Where $E[g]$ is moving average of squared gradients, $\frac{dC}{dw}$ is gradient of the cost function with respect to the weight, n is learning rate and β is moving average parameter.

4 DATASET

We have downloaded TREC Question Type classification from the following source http://cogcomp.org/Data/QA/QC/. TREC divides all questions into following 6 categories and 50 sub-categories. It is divided into two sets, Training dataset and Test dataset.

Table 2: Question Categories

Primary Categories	Sub Categories
ENTETY	TECHNIQUE, TERM, SUBSTANCE, RELIGION, SYMBOL, WORD, LANGUAGE, COLOR, SPORT, BODY, FOOD, PRODUCT, PLANT, CREATIVE, LETTER, VEHICLE, EVENT, DISEASE, INSTRUMENT, ANIMAL
NUMERIC	WEIGHT, TEMPERATURE, DISTANCE, COUNT, CODE, MONEY, PERCENT, PERIOD, DATE, SPEED, SIZE
LOCATION	CITY, COUNTRY, MOUNTAIN, STATE
HUMAN	INDIVIDUAL, TITLE, GROUP
DESCRIPTION	DEFINITION, REASON, MANNER
ABBREVIATION	EXPRESSION

Training dataset has total 5447 sample questions. We are using 4955 samples for training and 492 samples for validation. Test dataset has total 500 sample questions. Following is the statistic of the training, validation and test dataset.

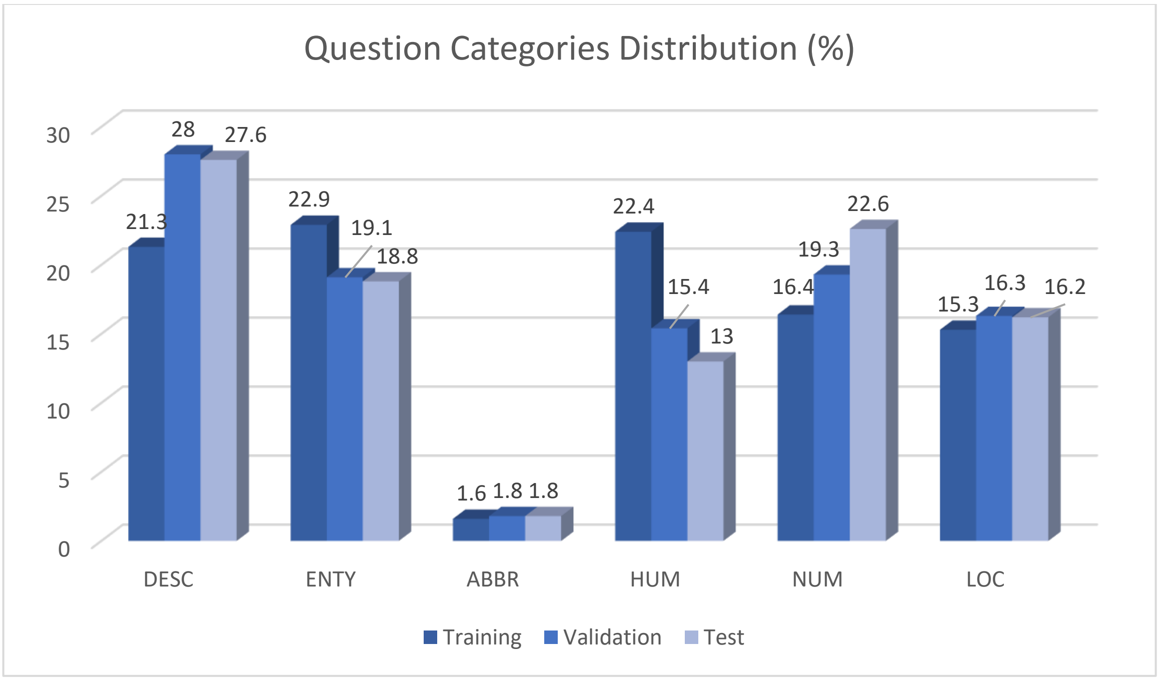


Figure 4: Question Categories Distribution

5 PREPROCESSING OF DATASET

We are cleaning the data by removing punctuation marks, commas, ‘s and converting string into small characters. As convolution layer needs fixed input length so we are finding out maximum sentence length let say l in the dataset and making all sentences to length l by padding sentences with special word <PAD/>.

We are generating word sequences by splitting sentences into words and building word vocabulary of size 8774 words and then using that word vocabulary to sequence words in the sentences.

6 WORD EMBEDDINGS

We have tried first GLoVe word embeddings with given dataset and with our models but results we not promising then we switched to Google News word2vec word embeddings of dimension 300 which gave us promising results with given dataset with our models.

Google News embeddings contains 100 billion words vectors so its size is very big and demands very large amount of memory signification amount of processing time to load word embeddings in memory at runtime. TREC dataset has only 8774 words in its vocabulary so we took this factor in consideration and generated a new subset of word embeddings of size 8774 vectors each of size 300 dimensions. This helped in reducing memory requirements and processing time during training and testing of the models.

7 EXPERIMENTS

7.1 HYPER PARAMETERS

We have choose following hyper parameters to train and test the models discussed in section 3 with TREC dataset discussed in section 4.

- Word Embedding Dimension: 300
- Sentence Length: 34
- Dropout Rate: 0.20
- Epochs: 30
- Batch Size: 50
- Learning Rate: 0.001
- Loss Function: Categorial Crossentropy
- Optimizer: RMSProp
- Training Samples: 4955
- Validation Samples: 492
- Test Samples: 500
- Metric: Categorial Accuracy

7.2 Convolution Neural Network

7.2.1 Training and Validation Measurements

The model achieved best validation accuracy of 0.892 and validation loss of 0.3668 during training and validation phase. Please find the following Accuracy and Loss curves.

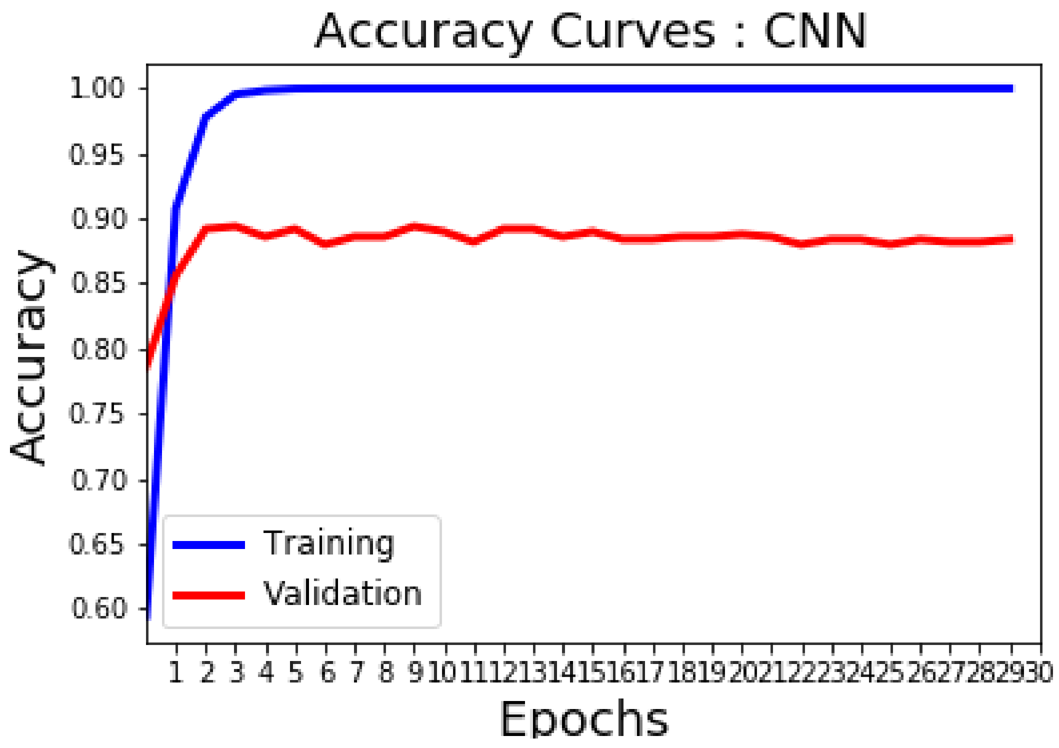


Figure 5: CNN Training & Validation Accuracy Curves

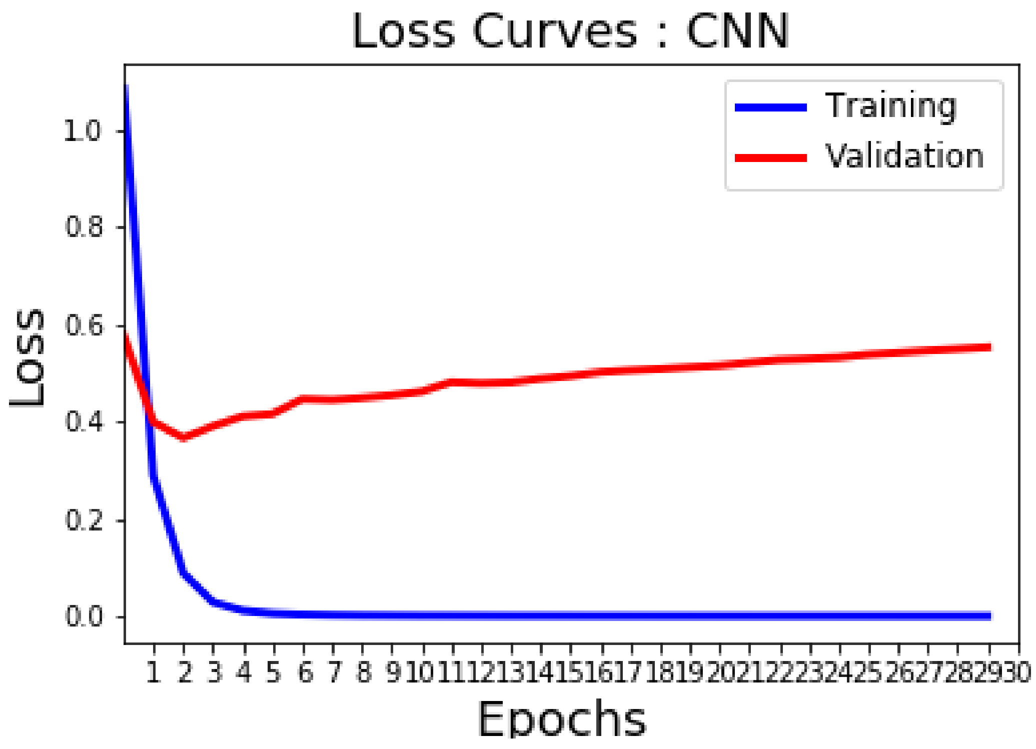


Figure 6: CNN Training & Validation Loss Curves

7.2.2 Test Measurements

The model achieved the best Accuracy of 0.918 and loss of 0.082 on test dataset. Accuracy and loss for test data is calculated using Confusion Matrix method. Please find the following Categorical Accuracy Chart.

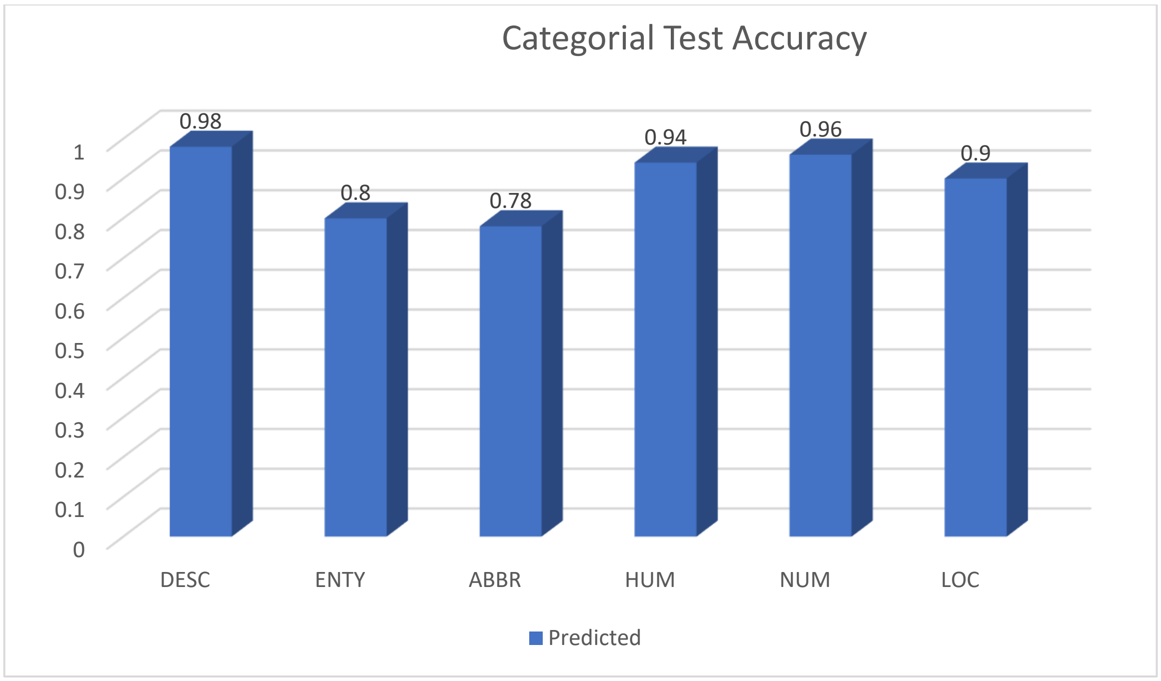


Figure 7: CNN Categorial Test Accuracy

7.3 Attention Aware GRU Neural Network

7.3.1 Training and Validation Measurements

The model achieved best validation accuracy of 0.896 and validation loss of 0.365 during training and validation phase. Please find the following Accuracy and Loss curves.

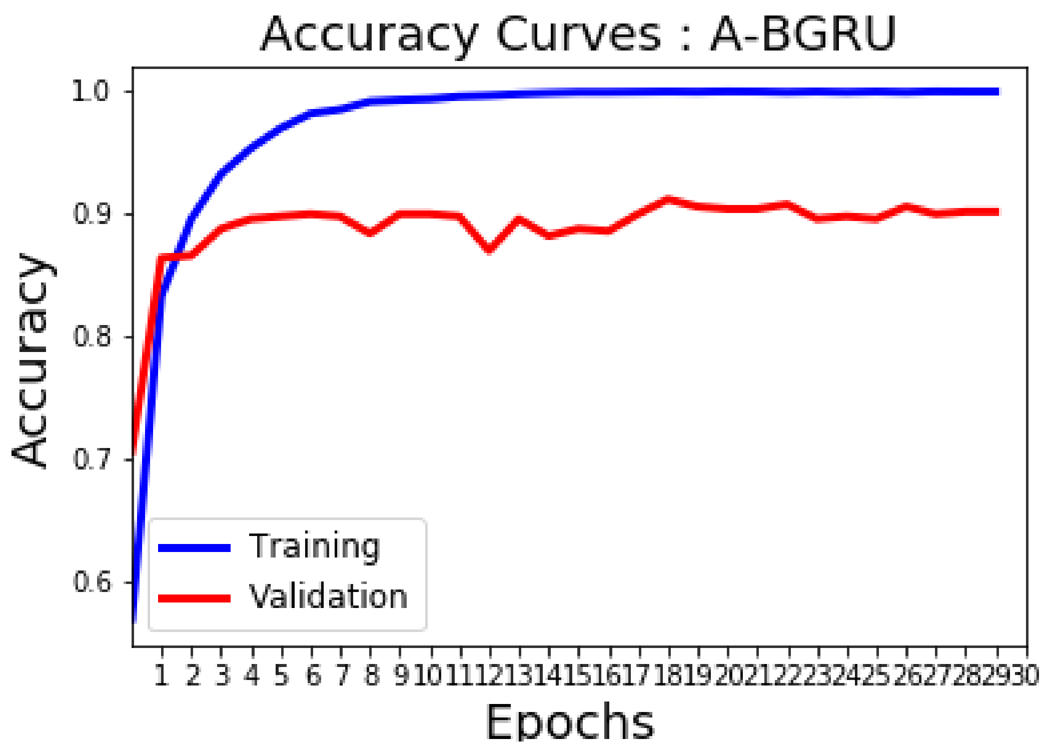


Figure 8: Attention Aware BGRU Training & Validation Accuracy Curves

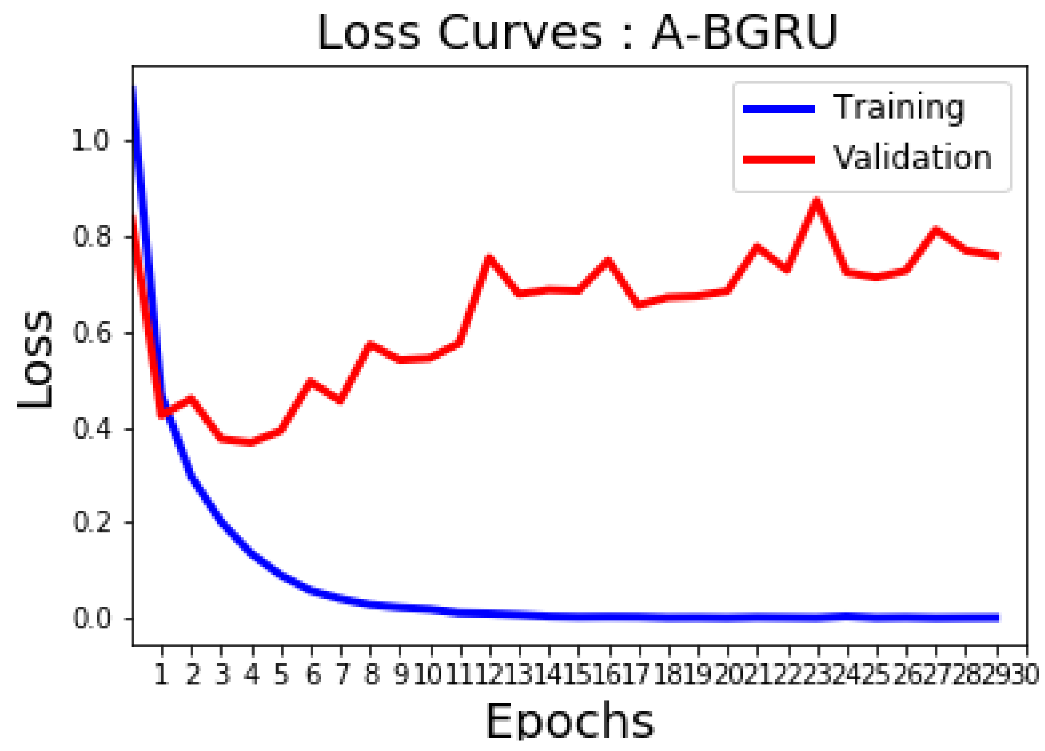


Figure 9: Attention Aware BGRU Training & Validation Loss Curves

7.3.2 Test Measurements

The model achieved the best Accuracy of 0.924 and loss of 0.076 on test dataset. Accuracy and loss for test data is calculated using Confusion Matrix method. Please find the following Categorical Accuracy Chart.

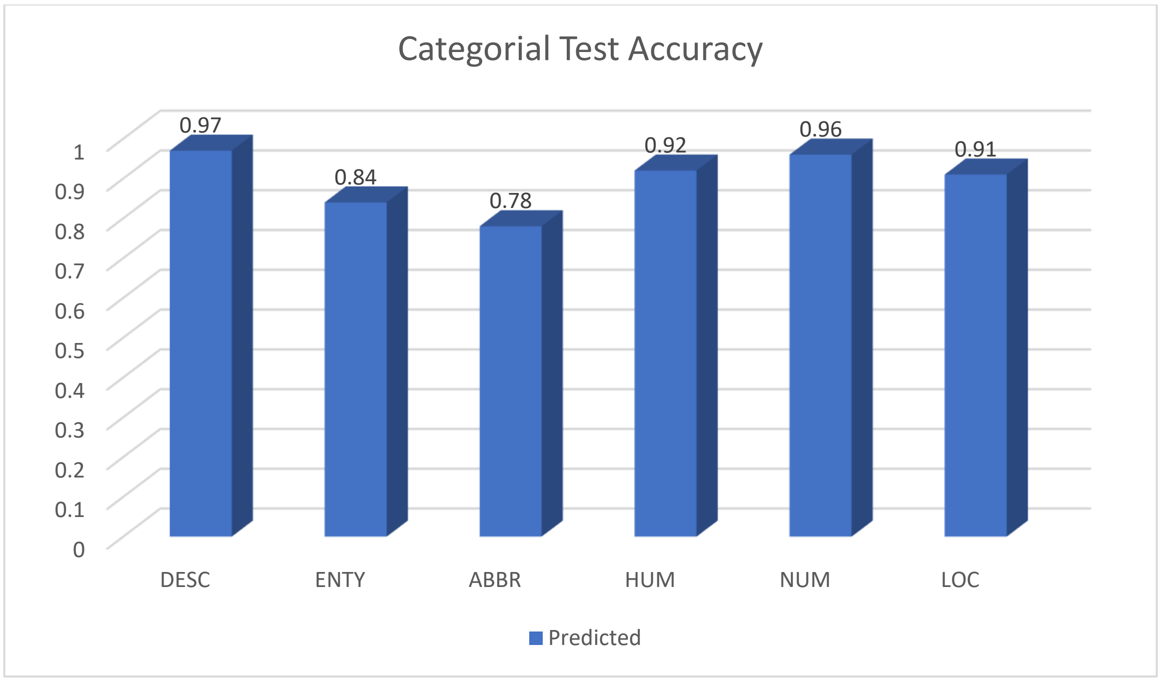


Figure 10: Attention Aware CNN-GRU Test Categorial Accuracy

7.4 ATTENTION AWARE CNN-GRU NEURAL NETWORK

7.4.1 Training and Validation Measurements

The model achieved best validation accuracy of 0.894 and validation loss of 0.3625 during training and validation phase. Please find the following Accuracy and Loss curves.

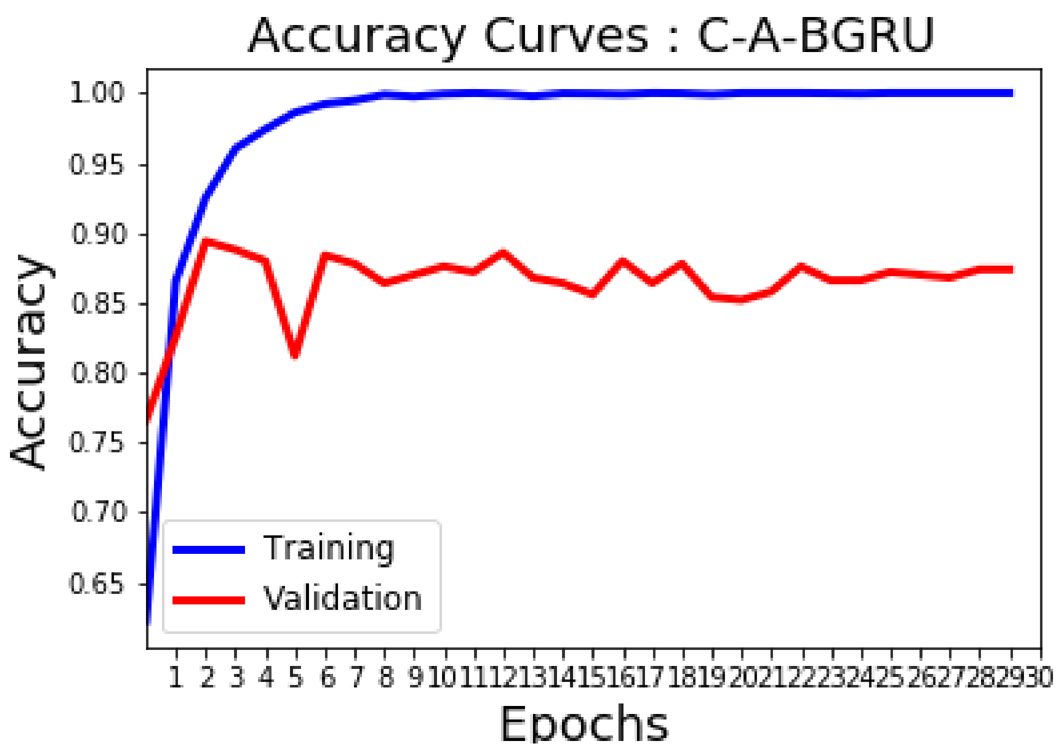


Figure 11: Attention Aware CNN-BGRU Training & Validation Accuracy Curves

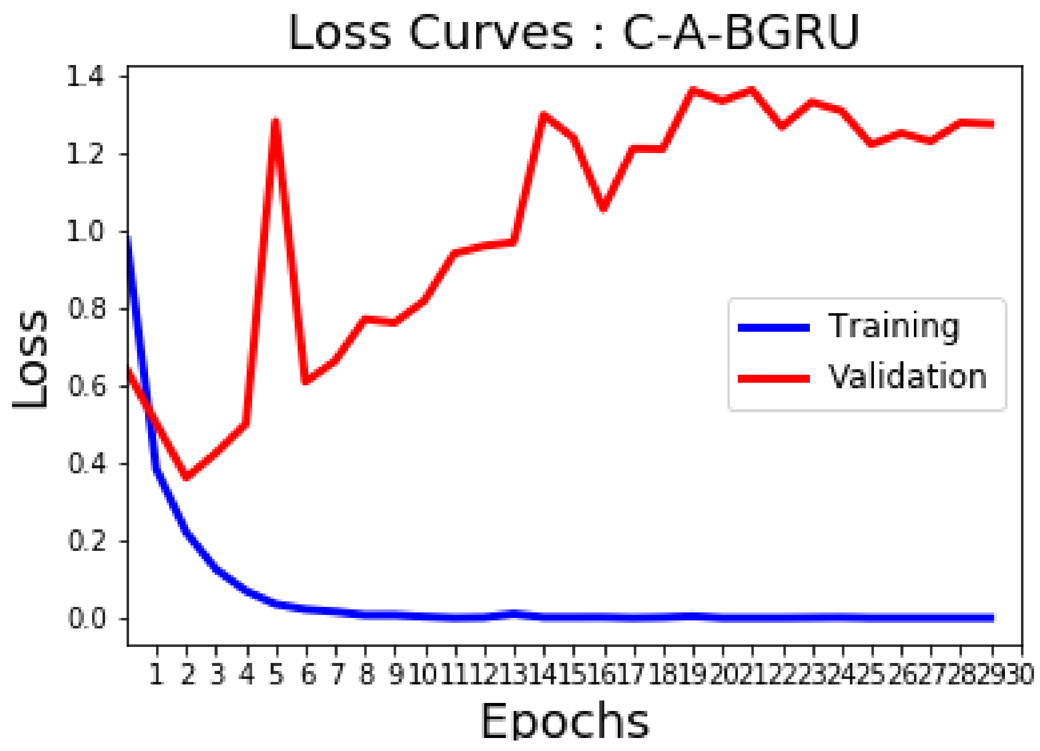


Figure 12: Attention Aware CNN-BGRU Training & Validation Loss Curves

7.4.2 Test Measurements

The model achieved the best Accuracy of 0.922 and loss of 0.078 on test dataset. Accuracy and loss for test data is calculated using Confusion Matrix method. Please find the following Categorical Accuracy Chart.

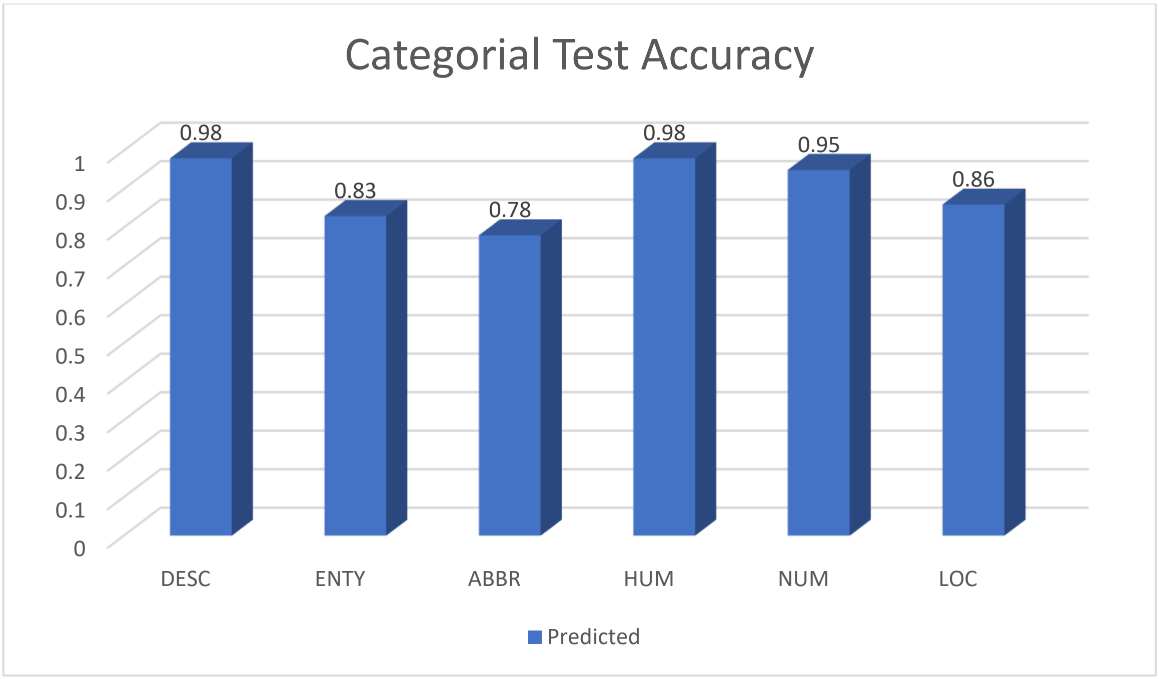


Figure 13: Attention Aware CNN-BGRU Categorial Test Accuracy

8 SUMMARY

We have summarized and compared the performance of all the models. All the models have performed similarly and achieved the accuracy results of more than 0.91 on test data. Attention aware networks performed slightly (1%) better than Non-Attention aware networks. Please find the Categorial Accuracy performance comparison on test data across all the models.

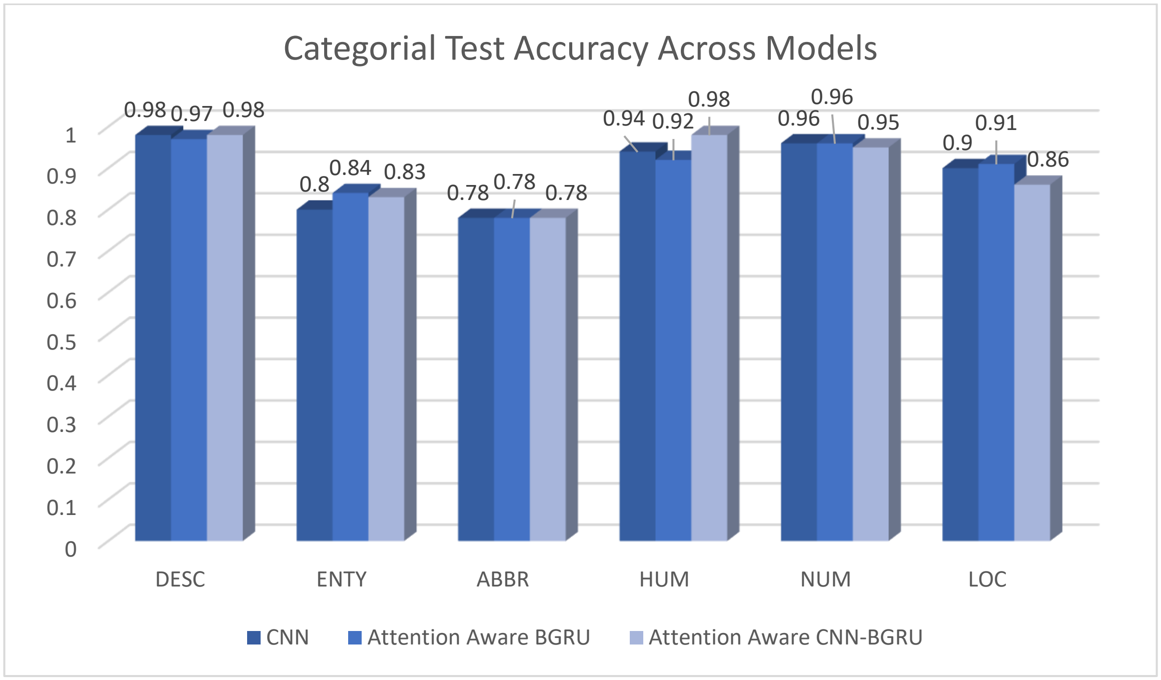


Figure 14: Categorial Test Accuracy Across Models

Please find the overall Test Accuracy performance comparison on test data across all the models.

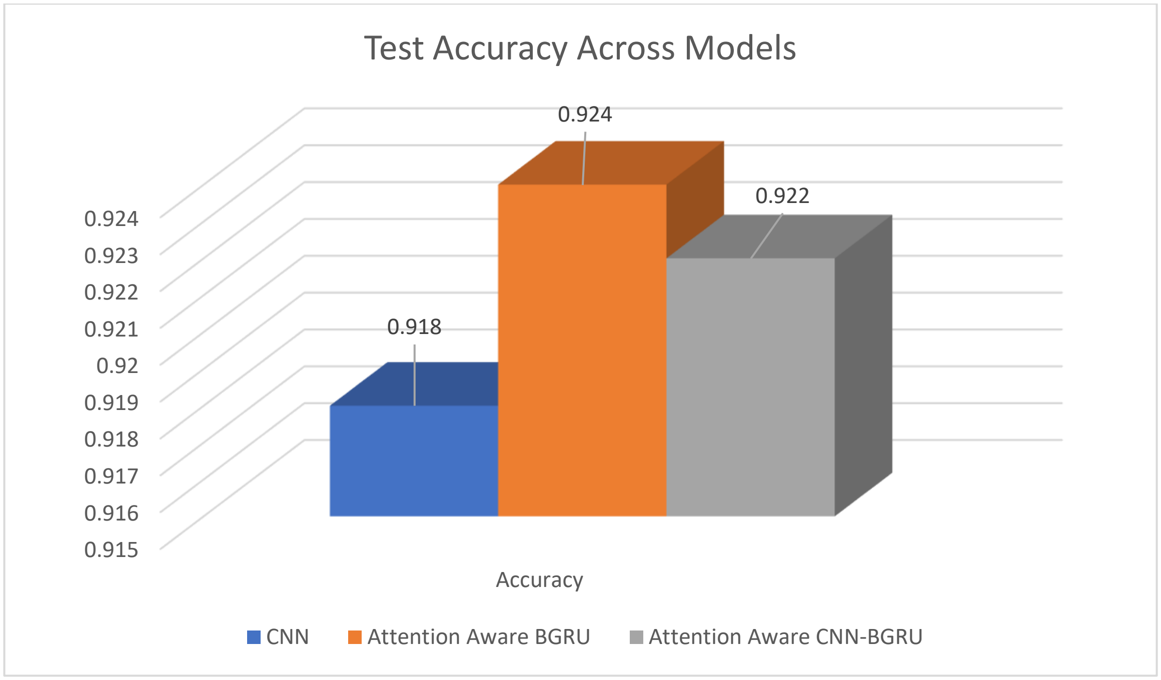


Figure 15: Test Accuracy Across Models

9 SOFTWARE

We have used Tensorflow 1.12 and Keras 2.1.6-tf Python APIs to implement these models. Source code is available at github: https://github.com/shamalwinchurkar/question-classification.git.

Following are the addition efforts we took while implementing these models other than just using standard Keras APIs.

- Design and implemented custom TF-Keras callback to monitor global maximum of validation accuracy and global minimum validation loss at AT_EPOCH_END event during training. When we get global maximum, we store the weights of the model and remember that state. We use stored model weights to do inference on test data.
- Dropout rate is set to 0.20 during training and set to 0 during testing.
- Designed and implemented Dataset class to abstract data operations, like preprocessing of data, building vocabulary, converting sentences into word sequences, extracting target classes and their labels and splitting data into 3 sets, training, validation and test.
- Implemented custom EmbeddingWeights Initializer class to incorporate Google New Embedding Matrix into TF-Keras Embedding Layer as initial weights for Embedding word vectors.
- Implemented Embeddings class to abstract loading of Embeddings from file and generating Embedding Matrix.
- Designed and implemented TF-Keras model classes to encapsulate various layers under model class for cleaner implementation.

10 FUTURE WORK

In this dissertation, we have implemented Dynamic Attention after BGRU layer where model is deciding which feature vectors are important in classifying particular question in desired category. We can think of implementing Attention Layer before Convolution Layer which will incorporate domain knowledge in the form of focus word dictionary and give more weightage to those focus words during training and inference.

Due to current limitation of Keras APIs and time constraints, we decided to not implement this idea in this dissertation and considered as future work to implement custom Keras Layer to perform focus word based Attention calculations.

11 REFRENCES

[Mikolov et al.2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

[Yoon.2014] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification.

[Chunting Zhou et al.2015] Chunting Zhou1, Chonglin Sun2, Zhiyuan Liu3, Francis C.M. Lau1. A C-LSTM Neural Network for Text Classification.

[Peng Zhou et al.2016] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, Bo Xu. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification.

12 Checklist of Items for the Final Dissertation Report

This checklist is to be duly completed, verified and signed by the student.

1.	**Is the final report neatly formatted with all the elements required for a technical Report?**	Yes
2.	Is the Cover page in proper format as given in Annexure A?	Yes
3.	Is the Title page (Inner cover page) in proper format?	Yes
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes Yes
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes Yes
6.	Is the title of your report appropriate? **The title should be adequately descriptive, precise and must reflect scope of the actual work done.** Uncommon abbreviations / Acronyms should not be used in the title	Yes
7.	Have you included the List of abbreviations / Acronyms?	Yes
8.	Does the Report contain a summary of the literature survey?	Yes
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate	Yes Yes Yes Yes Yes Yes
10.	Is the conclusion of the Report based on discussion of the work?	Yes
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes Yes Yes

12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	Yes

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.

Place: Pune

Date: 08/03/2019

[illegible]

Signature of the Student

Name: Shamal Winchurkar

ID No.: 2014HT13001