

A Sequence-Based Machine Comprehension Modeling Using LSTM and GRU



Sujith Viswanathan, M. Anand Kumar and K. P. Soman

Abstract Machine comprehension deals with the idea of teaching machines the ability to read a passage and provide the correct answer to a question asked from it. Creation of machines with the ability to understand natural language is the prime aim of natural language processing. A machine comprehension task is an extension of question answering technique which provides the machines an ability to answer questions. This task revolutionizes the way in which humans interact with machines and retrieve information from them. Recent works in the field of natural language processing reveal the dominance of deep learning technique in handling complex tasks which suggest the use of neural network models for solving machine comprehension tasks. This paper discusses the performance of code-mixed Hindi data for handling machine comprehension using long short-term memory network and gated recurrent unit. A comparative analysis on the basis of accuracy is performed between the two sequence models to determine the best-suited model for handling this task.

1 Introduction

The process of providing systems the ability to interact with texts given by humans can be considered as one of the most improving areas of artificial intelligence (AI). The area that provides an interaction between systems and texts is referred to as natural language processing (NLP). The two major variant areas under NLP are natural language understanding (NLU) and natural language generation (NLG). NLU deals with the extraction of information from a text [4], and NLG is a domain that deals

S. Viswanathan (✉) · M. Anand Kumar · K. P. Soman
Center for Computational Engineering and Networking (CEN), Amrita School of Engineering,
Coimbatore, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: cb.en.p2cen16016@cb.students.amrita.edu

M. Anand Kumar
e-mail: m_anandkumar@cb.amrita.edu

K. P. Soman
e-mail: kp_soman@amrita.edu

© Springer Nature Singapore Pte Ltd. 2019
V. Sridhar et al. (eds.), *Emerging Research in Electronics, Computer Science and Technology*, Lecture Notes in Electrical Engineering 545,
https://doi.org/10.1007/978-981-13-5802-9_5

with processes similar to the one that can make a system write or report on something in the form of texts [8]. Question answering (QA) is a technique that falls under NLU which focuses on building systems that can automatically answer questions posted by humans in natural language [7]. The initial works on QA systems were based on two types of methodologies, information retrieval (IR)-based QA, and knowledge-based QA. Recent works in QA systems focuses on factoid-based QA in which the questions will be answered based on certain available facts. Question classification is another important aspect of traditional QA systems which deals with categorizing the type of questions [1]. One of the major applications of QA system is conversational models referred to as chatbots; it has a vital role in business perspective as they provide customer assistance [9].

Machine comprehension (MC) can be considered as an extended version of QA task. An MC technique deals with the process of creating a model that can read a set of sentences given to them in the form of a passage and provide a correct answer to a question asked from the given passage. This technique needs more focus on NLU perspective as there should be more focus given to each passage passed to the model. Deep learning technique has evolved as one of the best-suited means for performing language modeling tasks as it outperformed various machine learning algorithms. Recent researches depict the advantage of deep learning technique in handling complex tasks on natural languages this indicates the scope of improving MC tasks using deep learning as some deep learning networks have the ability to capture sequential information, whereas other approaches fail to perform this functionality. In this paper, we focused on comparing the performance of code-mixed Hindi language in handling MC tasks, so we incorporated two sequence models for this comparison. The two networks used are:

- Long short-term memory network (LSTM)
- gated recurrent unit (GRU)

The two networks are selected by considering the fact that recurrent neural network (RNN) family of networks performs well in handling sequential information. An LSTM cell has the functionality to include or exclude information according to their requirements as it handles long-term dependencies efficiently. A GRU cell is another variant of LSTM, but it is computationally more efficient as it has one gate lesser when compared to the structure of LSTM, and they are capable of making QA systems [13]. Even though the two networks are subparts of RNN, these two networks are superior to it as RNN poses a problem of vanishing gradient during backpropagation. The LSTM is one of the most widely used neural network models for language modeling tasks, and it performs well in handling QA tasks [11].

A QA dataset can be generally classified into two types: open and closed datasets. In open QA datasets, the appropriate response relies upon general information's other than the details given in texts. In closed QA datasets, the response can be retrieved from the texts provided in the dataset itself. There is a wide number of research carried out in the field of QA in English [6]; however, these technologies are not widely applied to Indian languages which indicate the huge scope for researches related to QA and MC tasks in Indian languages. The dataset we used includes twenty different

classes of tasks in code-mixed Hindi which constitutes twenty different ways in which answers are retrieved from supporting sentences in a passage according to respective questions. The dataset we used can be considered as a closed QA dataset. The prime aim of handling this dataset is to create a single model that handle a majority of twenty tasks. There are a number of NLP tasks carried out by researchers in code-mixed Indian languages [5], but implementing a new model for solving MC in code-Mixed Hindi language can help in introducing other code-mixed Indian languages to the area of MC. The performance of both the networks with respect to the twenty tasks is analyzed in this paper to identify the best-suited model for code-mixed Hindi.

2 Dataset

The dataset incorporated for this work is obtained from Facebook research, and it is available in code-mixed Hindi language. The dataset contains twenty different tasks that depict different ways in which answers can be retrieved from a set of given sentences in the form of a passage. This is an ideal dataset to start any MC research in code-mixed Indian languages as it considers different answer retrieval approaches. The tasks under this dataset are not including any noises. Humans can obtain an accuracy of 100% in answering questions based on the sentences. The dataset we used is referred to as the (20) QA bAbI tasks; it includes a training and testing dataset in separate [12]. This paper uses 10,000 examples provided as stories for training and 1000 for testing. Each story after a certain number of lines includes a question to be answered from the above sentences. An answer followed after a question includes a label indicating the line from where the answer is retrieved. The dataset includes twenty machine comprehension tasks, and they are represented in Table 1. The dataset tests the performance of a model in various tasks by considering supporting facts, arguments, conjunction, etc. The important aspect in using this dataset is that it helps in identifying an MC model in code-mixed Hindi language that can solve a majority of the above mentioned tasks.

The training dataset includes answers with a label given to them, indicating the line from where the information related to the answer is obtained. This label depends on the tasks, and according to it, the number of labels varies. The examples of sample dataset with stories, question, answers, and their labels for two tasks are shown below:

- **Single supporting Fact:**

- 1 Justin daftar mein gaya.
- 2 Tina vidyalay mein gayi.
- 3 Tina is samay kahan hai? vidyalay, 2.

- **Three supporting Facts:**

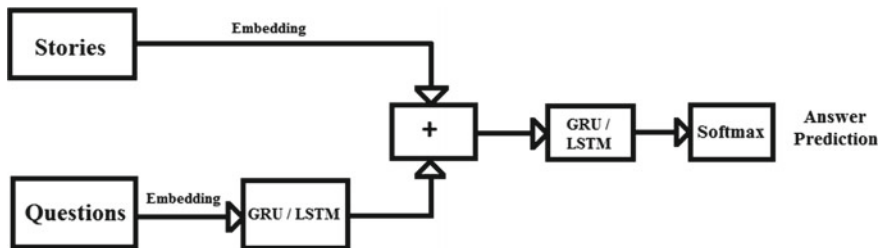
- 1 Rohan gusalkhaney mein chala gaya.
- 2 Rohan rasoi ghar mein gaya.
- 3 Rohan gayend chod aaya.
- 4 Rasoi ghar se pehle gayend kahan per thi? gusalkhana, 3 2 1.

Table 1 Twenty machine comprehension tasks [10]

Tasks [1–10]	Tasks [11–20]
Single supporting fact	Basic coreference
Two supporting facts	Conjunction
Three supporting facts	Compound coreference
Two argument relations	Time reasoning
Three argument relations	Basic deduction
Yes or no questions	Basic induction
Counting	Positional reasoning
List sets	Size reasoning
Simple negation	Path finding
Indefinite knowledge	Agents motivations

3 Methodology

A machine comprehension model proposed by us to handle the twenty different tasks in code-mixed Hindi language includes two variant networks that belong to RNN family, LSTM, and GRU networks. This section explains the way in which the two networks are used to handle the process of providing machines the ability to read a passage in the form of sentences and answer a question asked from the passage. In this work, we considered the stories and questions separately in different blocks as shown in Fig. 1. The predicted answers are compared with the actual answer to analyze the efficiency of the model. In order to perform the training of our model, we give stories and questions followed by answers as inputs for each task. These inputs are made to undergo few preprocessing techniques. The working of our model in handling machine comprehension task is explained in the following sections.

**Fig. 1** Architecture of the proposed model

3.1 Preprocessing

The preprocessing techniques include tokenization, a process in which a particular sentence is separated into various tokens including punctuations as a separate token. A dictionary constituting all the words in the inputs passed to the model is created, and using this, word indexes are generated in such a way that each word in the vocabulary is assigned to a number. Each sentence inputted to the model has to be converted to a standard numerical form using the word indexes. The maximum word length of the stories and questions is found out for each task. The words in the sentences are replaced by the indexes they possess in the vocabulary, and they are padded with zeros to make each sentence vector of numerical form into the same size, that is, story maximum length for story sentences and question maximum length for questions. A validation set is created to perform cross-validation in such a way that from a total of 10,000 training examples, 9500 examples will be used for training and remaining 500 will be used for validating the model before testing.

3.2 Long Short-Term Memory Network

An LSTM network is a type of RNN which helps in handling long-term dependencies in data. LSTM is made up of three gates and one cell state. The gates and cell state together provide interactions. LSTM includes the creation of special modules designed to allow information to be gated in and gated out when needed. In the intermediate period, the gate will be closed such that the things coming during the intermediate period does not interfere with the remembered states in LSTM. Hochreiter and Schmidhuber [3] in 1997 solved the problem of providing RNN the ability to remember things for a long time. This was done by designing a memory cell using logistic and linear units with multiplicative interactions. The three gates in LSTM are input gate, forget gate, and output gate. Information gets into the cell whenever its 'input' gate is on. The information stays in the cell until a 'forget' gate is on. Information can be read from the cell when an 'output' gate is on.

3.3 Gated Recurrent Unit

GRU is another variant of LSTM introduced in [2] by Cho et al. It is similar to LSTM, but it is more computationally efficient than LSTM because it has only two gates and does not use any memory unit. The two gates present in GRU are update and reset gates. In GRU, the forget and input gate's functionality found in LSTM are combined to form an update gate. The update gate characterizes the amount of past memory to be kept in GRU.

3.4 Answer Prediction

The prime aim of a machine comprehension model is to predict the correct answer to a question asked of it based on the story provided earlier. To handle this task, we have three sections, in which the **story block** handles the stories. For each sentence in a story, embeddings are generated by the model with respect to a fixed embedding dimension by providing inputs as the vectors, which was 50 considered by us. The **question block** handles the questions where the embeddings are generated by the model with respect to the same embedding dimension given for story sentences. These embeddings of questions are passed through an GRU/LSTM layer, which performs the functionality of an **encoder** that encodes required information in encoder–decoder architecture. The **answer block** deals with the prediction of answers. In this sections, the embeddings generated by the first and second section are added together to form a merged representation, and this is passed through another GRU/LSTM layer which performs the functionality of a **decoder** that decodes required information in an encoder–decoder architecture. This section includes a final output layer that includes softmax which gives a probability score over the vocabulary that helps in the retrieval of answer from the vocabulary using the score given. The encoder GRU captures the necessary information associated with the question when the input sentences are added to it; this allows the decoder to fetch that information.

4 Result

The machine comprehension model built by us on code-mixed Hindi language accurately provided answers for the questions asked based on a given story in most of the instances. The dataset was released with tasks having twenty class to analyze the performance of a model in various ways of answer retrieval. The two variants of architecture used by us include one with LSTM network and other with GRU. From our observations, we find that both networks solved the majority of the twenty classes of tasks when a few remained as failed. We have incorporated two variant networks of RNN family to identify the best-suited and computationally efficient approach for solving machine comprehension tasks. The comparison study based on the accuracy we obtained for both networks indicated that the number of failed tasks in LSTM is high when compared to GRU. The variation in the performance of both models is graphically analyzed with respect to the test accuracy obtained for both networks in twenty different machine comprehension tasks as depicted in Fig. 2. The architecture of two networks we used is less complex when compared to many existing approaches as it takes less time to execute. The performance analysis of the twenty tasks shows that the machine comprehension dataset on code-mixed Hindi is better handled by GRU. The validation and test accuracy obtained for both networks are shown in Table 2. The results we obtained for both networks suggest how the two

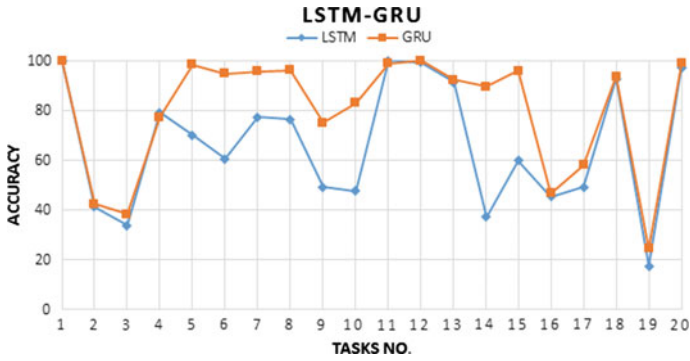


Fig. 2 Graph representing the test accuracy for LSTM and GRU

Table 2 Accuracy of twenty tasks in LSTM and GRU

Task No	LSTM		GRU	
	VAL	Test	VAL	Test
1	99.8	100	100	100
2	40.7	41.3	40.9	42.5
3	33.5	33.9	37.6	38.5
4	79.6	79.3	78.4	77.4
5	67.2	70.1	99.4	98.6
6	64.2	60.7	95.2	94.9
7	78.4	77.5	96.1	95.8
8	78.2	76.5	96.0	96.2
9	48.2	49.4	78.2	75.1
10	51.2	47.7	85.2	83.2
11	99.6	100	99.3	98.9
12	99.4	99.5	100	100
13	93.8	91.5	94.2	92.5
14	32.8	37.3	91.6	89.6
15	62.6	59.9	98.8	96.0
16	43.2	45.4	48.4	46.9
17	48.6	49.4	62.8	58.2
18	92.3	92.8	93.5	93.8
19	17.2	17.5	24.1	24.7
20	97.0	97.3	98.6	98.9

networks can handle each unique type of answer retrieval mannerisms for twenty tasks. These results are having higher accuracy in most cases when compared to the other similar existing approaches.

5 Conclusion

The paper is focused on the incorporation of machine comprehension technique into code-mixed Indian languages. The system built by us performed equally well on both LSTM and GRU networks. The accuracy obtained for all twenty tasks on the two models depicts the success of both networks in handling a majority of the given tasks. The accuracy of models, when compared to each other, led us to the conclusion that GRU network has a superiority in handling code-mixed Hindi dataset used by us. The results obtained by us hence prove the advantage of using GRU network for solving these tasks. GRU yielded higher accuracy on most of the tasks when compared to LSTM. The computational complexity of GRU is very less when compared to LSTM due to the absence of a gate in GRU. This paper thus leads to the conclusion that usage of GRU network will be the best solution for solving question answering or machine comprehension-related tasks available in code-mixed Indian languages. The results we obtained for both networks are better when compared to other existing models for machine comprehension in code-mixed Indian languages.

6 Future Work

The scope of artificial intelligence-based natural language processing tasks in Indian languages is widely increasing due to the availability of a variety of Indian regional languages. Our approach can be considered as one such initial steps for handling machine comprehension-based tasks in other code-mixed Indian languages which open up a wide scope for researchers to explore. Incorporating similar machine comprehension datasets in pure scripts of Indian languages will be a remarkable achievement, which can be solved using a number of existing models similar to the ones we implemented. The code-mixed Hindi dataset we used can be incorporated into another model, and the performance can be explored. The failed tasks among the twenty classes used can be given special focus. Incorporating certain pre-trained embedding techniques can be considered as the future scope for the improvement of failed tasks with respect to this model.

References

1. Anand Kumar M, Soman K (2016) Amrita-CEN@MSIR-FIRE2016: code-mixed question classification using BoWs and RNN embeddings. In: FIRE (Working notes), pp 122–125
2. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. [arXiv:1412.3555](https://arxiv.org/abs/1412.3555)
3. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
4. Loatman RB, Post SD, Yang CK, Hermansen JC (1990) Natural language understanding system. US Patent 4,914,590

5. Rahul Venkatesh K, Anand Kumar M, Soman K (2015) Amrita-CEN_NLP FIRE 2015 language identification for Indian languages in social media text. In: FIRE workshops, pp 26–28
6. Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016) Squad: 100,000+ questions for machine comprehension of text. [arXiv:1606.05250](https://arxiv.org/abs/1606.05250)
7. Ravichandran D, Hovy E (2002) Learning surface text patterns for a question answering system. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, pp 41–47
8. Reiter E, Dale R (2000) Building natural language generation systems. Cambridge University Press
9. Shawar BA, Atwell E (2007) Chatbots: are they really useful? LDV Forum 22:29–49
10. Sujith Viswanathan AKM, Soman K (2018) A deep learning approach to machine comprehension in code-mixed Hindi language. Int J Pure Appl Math (in press)
11. Wang D, Nyberg E (2015) A long short-term memory model for answer sentence selection in question answering. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, vol 2: short papers, pp 707–712
12. Weston J, Bordes A, Chopra S, Rush AM, van Merriënboer B, Joulin A, Mikolov T (2015) Towards AI-complete question answering: a set of prerequisite toy tasks. [arXiv:1502.05698](https://arxiv.org/abs/1502.05698)
13. Zhang J, Zhu X, Chen Q, Dai L, Jiang H (2017) Exploring question understanding and adaptation in neural-network-based question answering. [arXiv:1703.04617](https://arxiv.org/abs/1703.04617)