INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
*(Volume 9, Issue 6 - V9I6-1209)*
Available online at: https://www.ijariit.com

# Convolutional Recurrent Neural Network Model for Question Answering Task

*Gelvesh G.*
gelveshg@gmail.com
*Dayananda Sagar College of Engineering, Bengaluru, Karnataka*

***ABSTRACT***

***Question answering is a crucial task in natural language understanding, as it can be applied to a wide range of natural language processing challenges. Recurrent Neural Networks (RNNs) are commonly used as a baseline model for various sequence prediction tasks, including question answering. While RNNs excel at capturing global information over a long span of time, they may not effectively retain local information. To address this limitation, we propose a model that combines both recurrent and convolutional neural networks, allowing for end-to-end training using backpropagation. Our experiments on the bAbI dataset show that this model can significantly outperform the RNN model in question answering tasks.***

***Keywords*** *—Deep Learning, NLP, neural network, CNN, RNN*

## I. INTRODUCTION

Question answering (QA) is a challenging task in natural language processing, involving the understanding of context and reasoning through a series of sentences. QA systems are given a story and a question, and their goal is to find the correct answer by extracting relevant information from the story. Many NLP tasks can be framed as QA problems, and QA is crucial for developing dialogue systems and chatbots, making personal assistants like Siri and Cortana, more robust and helpful.

QA has been a topic in information retrieval for a long time, but early datasets for evaluating QA systems were small, and traditional approaches often relied on rule-based algorithms and linear classifiers with hand-engineered features. With the rise of more expressive models like deep neural networks, larger datasets became necessary. Hermann et al. introduced a large-scale QA dataset, and subsequent efforts have focused on creating datasets to assess a computer's critical reasoning abilities. Weston et al. introduced a set of artificial tasks called "bAbI tasks" for model evaluation.

Recurrent Neural Networks (RNN) are neural networks with recurrent connections between hidden layers at consecutive time steps, making them suitable for sequential and temporal data prediction, such as speech and text. RNNs and their variants have become standard for various NLP tasks. It is possible to use a Recurrent Neural Network Language Model to predict answers from the vocabulary based on a joint representation of the story and question.

 

Convolutional Neural Networks (CNN) are feed-forward neural networks inspired by the human visual cortex, with consecutive convolutional and subsampling layers. While CNNs have had a significant impact on computer vision, they have also begun to influence natural language processing. Dos Santos and Gatti achieved state-of-the-art results in sentiment analysis of single sentences using CNNs. CNNs have been used for sentence classification and modeling.

Combining RNN, which handles long-term dependencies, with CNN, which excels at detecting localized features, has shown promise in various applications like machine translation and music classification. We propose a similar model for question answering, which outperforms single RNN models.

In this paper, we provide relevant background information in Section II, discuss the dataset with examples in Section III, present our proposed question answering model in Section IV, describe the experimental results in Section V, and conclude in Section VI..

## II. BACKGROUND STUDY

### Word Embedding

Word embedding is a technique that generates distributed word representations, where words with similar meanings are mapped to nearby points in a vector space. In contrast, traditional methods like tf-idf or n-grams create fixed-length vector representations for words. More recently, word embedding models like Word2Vec and Glove have gained popularity for generating low-dimensional word vectors.

The concept of distributed representation of words is inspired by human cognition, and it was proposed during the resurgence of connectionist ideas in the 1980s. Researchers, including Bengio et al., demonstrated that neural networks, along with distributional representations, can outperform standard n-gram models in language modeling tasks. Mikolov et al. introduced the Continuous Bag-of-Words (CBOW) and Skip-Gram models for learning word embeddings from raw text.

In Figure 1, a well-known framework for word embedding is depicted. Each word is represented by a column in the matrix W, with the ith column corresponding to the ith word in the vocabulary. The primary objective is to predict the next word based on context. To make predictions, the matrices associated with words in the context are either summed or concatenated. Once the model is trained, words with similar meanings tend to have similar values in multiple dimensions of the vector.

For example, word vectors can capture semantic similarity, and differences in word vectors can represent various relationships between words, such as gender, number, or degree. These characteristics have led to a wide range of applications for word vectors in language modeling and understanding tasks.

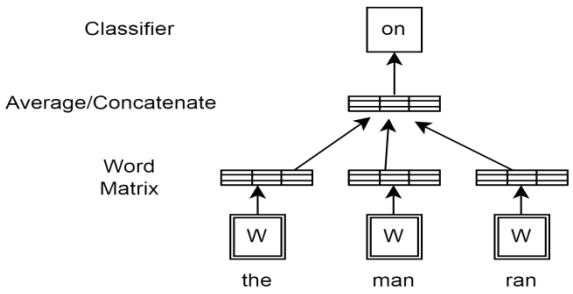


**Fig.1:** A general approach for learning Word Embedding

### Recurrent Neural Network

A neural network is termed as recurrent when it possesses one or more cycles, allowing for the possibility to trace a path from a unit back to itself. Typically, contemporary recurrent networks employ an architecture known as the Elman network, as depicted in Figure 2. This network enables the transmission of hidden layer information from previous time steps to subsequent hidden nodes.
The hidden state at time step 't' is determined by a function involving the preceding hidden state and the current input.

$s_t = f(U_{x\,t} + W_{st-1})$ ------- (1)
The output(y) at time step t is calculated as
$o_t = f(V_{st})$ ----------(2)

Here,
xt = Input at t timestep
st = Hidden state at t timestep
ot = Output at t timestep

 

f = an activation function like o or tanh
U = Weight between input and hidden state
W = Recurrent weight between hidden states
V = Weight between hidden state and output

As expressed in equation (1), the hidden state is a function of both the previous context and the current input. This characteristic theoretically allows Recurrent Neural Networks (RNNs) to capture long-term dependencies across different time steps. However, in practice, the gradient tends to become very small after a certain number of time steps. This issue was addressed by Hochreiter and Schmidhuber, who introduced a novel approach for calculating the recurrent unit known as Long Short Term Memory (LSTM).

RNN Language Model
Language modeling is a crucial problem in the field of Artificial Intelligence (AI). Statistical language models aim to capture the underlying statistical patterns in the distribution of word sequences within a document. These models find significant applications in areas such as speech recognition, machine translation, and natural language processing. The primary objective of a word-level language model is to estimate the probability of the next word in a sequence, given all the preceding words.

$P(w_n \mid w_{n-i}, \ldots\ldots\ldots\ldots, w_2, w_i)$.

To illustrate, For example, P ( blue | The color of berries is) =?

A seminal paper by Bengio and colleagues introduced a neural network-based language model that utilized a feedforward neural network [20]. Subsequently, Mikolov and his team proposed a Recurrent Neural Network-based Language Model [18]. This model and its variations are now widely employed in numerous sequence prediction tasks, including various natural language applications like translation and text classification.

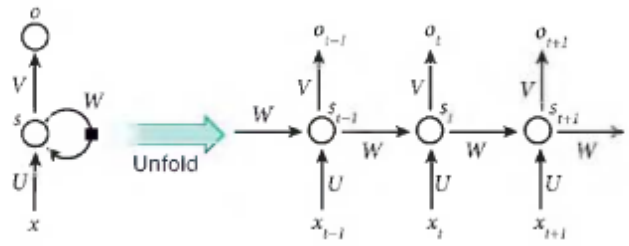


**Fig. 2**: A recurrent neural network unfolded through time [27]

**Convolutional Neural Network**
Convolutional networks, also known as CNNs, are a type of neural network inspired by the human visual system. They are particularly well-suited for handling data with a grid-like structure, such as images. The concept of CNNs traces its roots back to the groundbreaking experiments of Hubel and Wiesel, who demonstrated the hierarchical architecture of the visual cortex and its ability to create meaningful representations from raw visual data. CNNs, trained end to end, were initially employed for tasks like document recognition in the 1990s, highlighting their utility. Over recent years, CNNs have evolved to become a standard in the field of computer vision.

To clarify the workings of CNNs, let's first explore them in the context of image processing, which tends to be more intuitive. Later, we will delve into their application in language processing. Typically, convolutional networks consist of two main types of layers, which we will detail below:

1) Convolution layer: A convolutional layer takes an input and performs a convolution operation with various kernels or filters to generate a new activation map, also known as a feature map. These kernels have trainable weights, and the convolution operation involves summing up the element-wise products of the kernel and small overlapping portions of the input (of the same size as the kernel) as it slides across the input. This process is depicted in Figure 3. Subsequently, the convolved features pass through a non-linear function before being forwarded to the next layer. When working with convolution, there are four essential hyperparameters that need to be determined.

These hyperparameters include:

K = Number of filters
F = Spatial extent of the filters
S = Stride number
P = The amount of zero padding

 

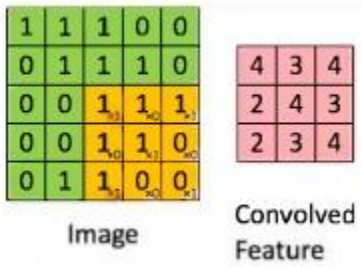


**Fig. 3.** Convolution operation using 3x3 kernel [34]

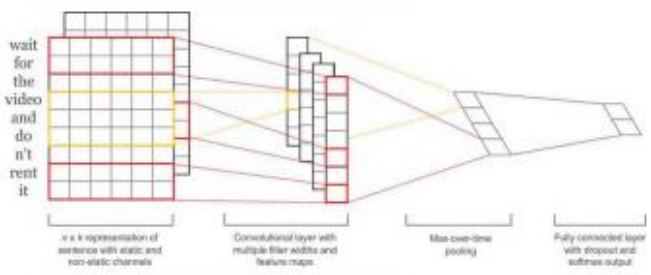


**Fig. 4.** Convolutional Neural Networks for sentence classification [25]

2) Pooling Layer: The pooling layer is often referred to as a subsampling layer because its primary purpose is to reduce the spatial dimensions of its input, thereby decreasing the number of parameters. This reduction in parameters helps keep computational complexity low and also aids in preventing overfitting during the training process.

The pooling layer has two non-trainable hyperparameters:

F, which represents the spatial extent of the filters.
S, which indicates the stride number.

Typically, max pooling or average pooling is used in this layer, meaning that it involves either selecting the maximum or calculating the average value within a specific window.

When applying CNN to natural language processing (NLP), documents or sequences of sentences are usually represented as matrices where each row corresponds to a token (e.g., words). Each row is a word vector typically obtained using a word embedding technique. The number of columns in the matrix corresponds to the dimension of the word vector. The filter's width matches the number of columns in the input matrix, while the filter's length can be a small odd number. Figure 4 illustrates how CNN is applied to NLP.

## III. THE DATASET

We conducted experiments on the synthetic question-answering tasks found in the Facebook bAbI dataset. This dataset assesses a model's capability to retrieve pertinent information and perform reasoning tasks. It comprises 20 distinct tasks, each designed to evaluate specific abilities such as counting, time reasoning, or basic induction/deduction.

Each task within the dataset includes a set of statements (referred to as a "story"), a question, and an answer, which is typically a single word. While the answer is provided during training, it must be predicted during testing. Only a subset of the statements contains essential facts, while others are just noise. The model must be able to filter out the noise to arrive at the correct answer. There are two versions of the training data: one with 1,000 training samples per task and a larger one with 10,000 samples per task. Our model was trained using the larger version.

It's important to note that excelling on this synthetic dataset doesn't guarantee the same level of performance on real-world datasets. Nevertheless, mastering these tasks is a prerequisite for a question-answering model to perform effectively in more complex challenges.

## IV. THE MODEL

The CNN model utilizes local connections to process a fixed number of words, allowing it to capture local information and create higher-level features. It also generates semantic representations that are invariant to rotation and translation. However, it has a fixed context size and cannot capture long-term dependencies. In contrast, RNNs, particularly those with LSTM or GRU cells, maintain global information by updating their state based on current inputs and previous states, making them suitable for handling text data.

 

To harness the strengths of both CNN and RNN, we propose a novel model. In this model, we first convert the story and question into word vectors using an embedding layer. These embeddings are then fed into an RNN with 50 LSTM units, and their outputs are combined into a single vector. Another RNN takes this merged vector as input and produces a fixed-length vector. This vector passes through a convolutional layer with 50 filters of size 3 and a max-pooling layer. Finally, a fully connected softmax layer is applied to the entire vocabulary to determine the answer's index. We've also applied dropout to the embedding and convolutional layers to prevent overfitting, although this is not depicted in the figure.

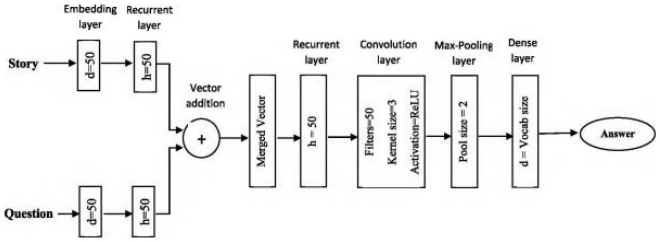


**Fig. 5.** Proposed model for question answering.

## V. EXPERIMENTS

In our evaluation, we used the bAbl question answering dataset, employing 10,000 training samples (consisting of story, question, and answer tuples), and tested the model on 1,000 samples for each task. All experiments were conducted solely on a CPU (Intel Core i5, 2.5GHz), without utilizing a GPU. We implemented the model using the Keras deep learning library with a TensorFlow backend and trained it using backpropagation with the Adam optimizer.

### Quantitative Results

Table 1 presents the outcomes of our experiments. Notably, our proposed Convolutional Recurrent Neural Network (C-RNN) outperformed the LSTM baseline from a previous study [6] on all tasks. The mean accuracy for the LSTM model was 49%, whereas the C-RNN achieved a mean accuracy of 74.95%, which is 25.95% higher than the LSTM model. In the context of the task, achieving an accuracy of 95% or higher is considered successful [39]. Using this criterion, our model successfully solved 7 out of the 20 tasks.

The performance across the 20 tasks is also visually represented in Figure 6, which displays the corresponding accuracy scores. Notably, for tasks 13, 19, and 20, both models perform similarly. However, for multiple tasks, including tasks 1, 6, 7, 17, and 18, our model significantly outperforms the LSTM baseline.

## VI. CONCLUSION

In our research, we introduced a convolutional recurrent neural network model designed for the task of question answering. This model incorporates the use of LSTM outputs fed into a convolutional layer to enhance the detection of short-term dependencies between words. Our experiments, which we conducted using the bAbl question-answering dataset, clearly indicate that our proposed model offers improved accuracy when compared to a standalone LSTM model. In fact, we observed a substantial increase of 25.95% in average accuracy on the test set.

Nonetheless, there are ongoing areas for improvement. While our model performs exceptionally well with larger training sets, it does require more time for training. The integration of external memory and attention mechanisms alongside this model could prove highly effective in addressing questions posed within longer narratives. Additionally, such enhancements may contribute to a reduction in the required size of the training data. We are eager to explore various memory-augmented architectures, such as those mentioned in references [39] and [1].

**Table 1:**Test set accuracy(%) of LSTM and our proposed model

| **Task** | **LSTM** | **CRNN** |
|---|---|---|
| Single Supporting Fact | 50 | 88.9 |
| Two Supporting Facts | 20 | 45.0 |
| Three Supporting Facts | 20 | 36.9 |
| Two Argument Relations | 61 | 77.1 |
| Three Argument Relations | 70 | 99.2 |
| Yes/No Questions | 48 | 83.0 |
| Counting | 49 | 93.8 |
| Lists/Sets | 45 | 78.2 |

 

| | | |
|---|---|---|
| Simple Negation | 64 | 88.5 |
| Indefinite Knowledge | 44 | 71.1 |
| Basic Co reference | 72 | 95.1 |
| Conjunction | 74 | 95.0 |
| Compound Co reference | 94 | 95.2 |
| Time Reasoning | 27 | 41.5 |
| Basic Deduction | 21 | 54.2 |
| Basic Induction | 23 | 48.6 |
| Positional Reasoning | 51 | 95.0 |
| Size Reasoning | 52 | 96.4 |
| Path Finding | 8 | 17.8 |
| Agents Motivations | 91 | 98.4 |
| Mean Accuracy | 49 | 74.95 |

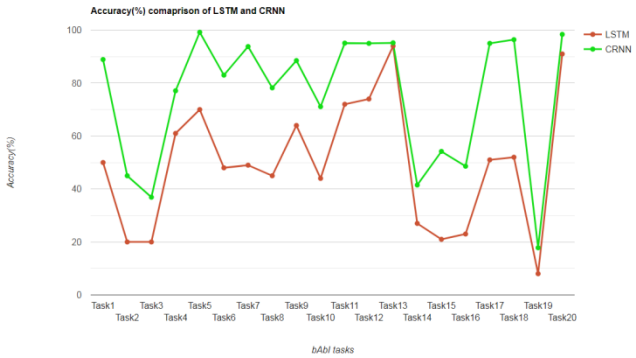


**Fig. 6.** The plot shows the accuracy(%) comparison of the LSTM baseline and our proposed model for every task in the bAbl dataset

## VII. REFERENCES


[1] YuL, Deep learning for answer sentence selection (2014)

[2] CongG, Finding question–answer pairs from online forums Proceedings of the Thirty-first International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)

[3] HuangJ, Extracting chatbot knowledge from online discussion forums Proceedings of the Twentieth International Joint Conference on Artifical Intelligence (IJCAI'07)

[4] YaoX,Answer extraction as sequence tagging with tree edit distance. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies (NAACL-HLT)

[5] WangB,Extracting chinese question–answer pairs from online forums Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC)

[6] A. Barrón-Cedeño,Thread-level information for comment classification in community question answering. Proceedings of the 2015 Association for Computational Linguistics (ACL)

[7] M. Heilman ,Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies (NAACL-HLT)

[8] A. Moschitti ,Exploiting syntactic and shallow semantic kernels for question answer classification. Proceedings of the 2007 Association for Computational Linguistics (ACL)

[9] N. Kalchbrenner ,A convolutional neural network for modelling sentences Proceedings of the Fifty-second Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)

[10] A.M. Schaefer ,Learning long-term dependencies with recurrent neural networks Neurocomputing (2008)

[11] LaiS. ,Recurrent convolutional neural networks for text classification Proceedings of the 2015 AAAI Conference on Artificial Intelligence (AAAI)(2015)

[12] DingS.,Using conditional random fields to extract contexts and answers of questions from online forums.

[13] C. Shah ,Evaluating and predicting answer quality in community QA Proceedings of the Thirty-third International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)


 

[14] A. Kumar, O. Irsoy, R Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in International Conference on Machine Learning, pp. 1378-1387, 2016
[15] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, 'Teaching machines to read and comprehend," in Advances in Neural Information Processing Systems, pp. 1693-1701,2015
[16] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," arXiv preprint arXiv:1606.05250, 2016
[17] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification.," in EMNLP, pp. 1422-1432,2015.
[18] T. Mikolov, M. Karafiat, L. Burget, J. Cemocky, and S. Khudanpur, "Recurrent neural network based language model.," in Interspeech,vol. 2, p. 3, 2010.
[19] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," arXiv preprint arXiv:1404.2188,2014.
[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," Journal o f machine learning research, vol.3,no. Feb, pp. 1137-1155, 2003.
[21] P. Dakwale and C. Monz, "Convolutional over recurrent encoder for neural machine translation," The Prague Bulletin o f Mathematical Linguistics, vol. 108, no. 1, pp. 37-48, 2017.
[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'Tmagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, pp. 1097-1105, 2012.
[23] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts.," in COLING, pp. 69-78, 2014.
[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean,"Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, pp. 3111—3119, 2013
[25] Y. Kim, "Convolutional neural networks for sentence classification,"arXiv preprint arXiv:1408.5882, 2014.
[26] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on,pp. 2392-2396, IEEE, 2017.
[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521,no. 7553, pp. 436^44, 2015.
[28] M. I. Jordan, "Serial order: A parallel distributed processing approach,"Advances in psychology, vol. 121, pp. 471-495, 1997.
[29] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14,no. 2, pp. 179-211, 1990.
[30] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks,vol. 5, no. 2, pp. 157-166, 1994.
[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[32] D. H. Hubei and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," The Journal o f physiology, vol. 148, no. 3,pp. 574-591, 1959.
[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings o f the IEEE, vol. 86,no. 11, pp. 2278-2324, 1998.
[34] D. Britz, "Understanding convolutional neural networks for nip." http://www.wildml.com/2015/ll/understanding-convolutional-neural-networks-for-nlp/.
[35] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings o f the 25th International Conference on Machine Learning, ICML'08, (New York, NY, USA), pp. 160-167, ACM, 2008.
[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," Journal o f machine learning research, vol. 15, no. 1,pp. 1929-1958, 2014.
[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization,"arXiv preprint arXiv:1412.6980, 2014.
[38] J. Weston, S. Chopra, and A. Bordes, "Memory networks," arXiv preprint arXiv:1410.3916, 2014.
[39] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merrienboer,A. Joulin, and T. Mikolov, 'Towards ai-complete question answering: A set of prerequisite toy tasks," arXiv preprint arXiv:1502.05698, 2015

 