Intelligent Learning Assistant using BERT and LSTM

Conference Paper · July 2021

DOI: 10.1109/ICCCNT51525.2021.9579531

4 authors, including:

Lekshmi S Nair
Amrita Vishwa Vidyapeetham Kollam
18 PUBLICATIONS **51** CITATIONS

Intelligent Learning Assistant using BERT and LSTM

Sriharsha C, Rithwik SRK, Prahlad Kumar P, Lekshmi S Nair
Department of Computer Science and Engineering,
Amrita Vishwa Vidyapeetham,
Amritapuri, India.

***Abstract*—Learning and preparing for the examinations have been a tough task according to the students' perspective, especially when the students are learning remotely. Subjects like languages or history require a lot of reading and remembrance skills associated with it. In this paper, we propose a question-answering model based on BERT (Bidirectional Encoder Representations from Transformers) that can assist learners of all ages with immediate feedback from the topic. We incorporate LSTM (Long Short-Term Memory) along with BERT into our application. This has helped in achieving a language translation from English to Hindi that helps non-native English users understand the context. Our model can be applied in assisting learners of any age.**

***Index Terms*—BERT, LSTM, Natural Language Processing, Translation, Question-Answering**

I. Introduction

During the pandemic situation, education is continuously evolving and moving towards remote learning. This has caused many students to drastically change the methods of studying. In the field of learning and education, many researchers have made contributions by using machine learning methods. One of the examples is that the student is assessed on some parameters and prospective career options are provided based on their learning style [1]. All the methods have both advantages and disadvantages. The vast amount of educational data can be leveraged to increase the efficiency of remote learning. We have come up with a solution that makes it easier for students to efficiently learn new topics remotely.

BERT (Bidirectional Encoder Representations from Transformers) [2] is introduced by Google for Search Engine Modelling. Compared to directional models, BERT accepts a sequence of words which helps to understand the context of the word based on both the directions(i.e. left and right). This property of BERT makes it better when compared to other uni-directional models based on RNNs. In this paper, we prototype a model that inputs a contextual passage followed by questions. Using the BERT architectural model, we are able to generate answers to the questions from the passage. This feature can be utilised by the learners. The learners can feed in the desired topics while studying and clarify doubts in the form of questionnaires. The answers generated by the model can help students to quickly assess their topic and clear their doubts.

The proposed model is further supported with LSTM [3], which helps in the translation of sentences in the passage to the native language. We have utilised English to Hindi translation in the implementation. LSTM is a slightly tweaked version of RNN developed to address the issue of the vanishing gradient problem. The *cell state* in LSTM helps in attaining the information flow. The advantage of this state is that the model can remember or forget the information that has contextual importance and prioritize the various values stored in the memory. The LSTM architecture model is able to translate the answer from English to Hindi. This translation helps the learners who are relatively new to the English language. They can understand the context better and also improve their vocabulary and communication skills.

The rest of the paper is divided into different sections. Section II represents related work, Section III represents background study, Section IV represents experimentation and results, followed by conclusion and future work in Section V.

II. Related work

Question-answering has been one of the most useful and challenging deep learning problems which can solve many issues and simply many things which machines could not do to date. In the last few years, there has been considerable development in this aspect due to improvement in hardware which helped us to train Deep Neural Networks.

There are numerous NLP models which can be applied in this field. Some are based on POS tagging [4] and TF-IDF [5] concepts in order to match the queries with questions already present on Search Engines. One major flaw of these concepts is, if a certain question is not already present on the internet then we won't get a corresponding answer. This is a fairly good approach of the questions are already present and we can score similar questions [6] and match retrieved sentences to get appropriate answers.

Several Deep Learning Techniques have been applied for this task. These models have mainly used RNN(Recurrent Neural Networks) like LSTMs and GRUs(Gated Recurrent Units) for classification and summarizing [7] tasks. In [8] there is a base model based on LSTM which has low accuracy but has a solid foundation to build upon. The breakthrough is the Memory networks model [9] which uses the memory in the system to synthesise a relevant answer from the context. The model [10] Dynamic Memory Networks uses an attention mechanism and combines memory networks to overcome the shortcomings of the previous model.

In recent years, BERT has replaced all these unidirectional models because it can understand the context [11] more

properly due to its bi-directional aspects and also extract the most amount of information due to the application of word embeddings, word2vec converts each word to an n-dimensional vector to catch the context correctly.

The challenge of translating languages using natural language processing has been a major concern. In recent times, the translation models have been surpassing even human translation. Many language modelling architectures in Indian languages [12] are evolved. LSTM has been widely used to translate many languages. for example, the paper [13] exhibits a translation system for natural language processing for a language called Wolof. It is a Niger-Congo language and has significantly low resources. French was used as a source language to train the model. They used vanilla LSTM based encoder-decoder architecture and was further developed and modified to bidirectional LSTMs with attention mechanisms. Even with high resource constraints, the model was able to produce outstanding results achieving a BLEU score of 47%. The deep learning concepts and techniques are being used right from the start. [14] explains very articulately the usage of these techniques in the field of NLP. Before LSTM became popular, Recursive recurrent neural network (R2NN) was one of the most used techniques for machine learning. The R2NN was combined from RNN and recursive neural network e.g. Recursive auto-encoders.

In the paper, [15] the authors have tried to investigate the possibility of a model to learn language modelling without parallel corpora but perform a translation with another state-of-the-art model that has been trained using parallel corpora. This was achieved by taking the sentences from monolingual corpora in two different languages. These sentences are mapped into the same latent space. The model effectively learns to translate in the absence of labelled data due to the ability of the model to take advantage of the shared feature space to reconstruct in both languages. It strikingly boasts of a 32.8 BLEU score on the Multi30k and 15.1 BLEU score on WMT English-French datasets. There is also a possibility to introduce better word embeddings [16] and feature vector representations [17] to the model which can improve the overall performance.

The SCN-LSTM (Skip Convolutional Network and Long Short Term Memory) is a deep learning model built for language translation. It was constructed by learning and training the real dataset and the public PTB (Penn Treebank Dataset) [18] with an accuracy score of 95.21%. The feasibility of the performance of the model, the quality of translation, and the accessibility and adaptability in practical teaching are explored and scrutinized to deliver a conceptual basis for the research and application of the SCN-LSTM translation model specifically in English teaching.

In the past, there has been very few experiments that integrate multiple architectures to tackle problems on language modelling. Using two different architectures suited for their special purposes, we have accomplished an efficient way to combine the two models and enable smooth data flow to solve a leading problem in the field of education. This model can also be scaled to several languages.

III. Background Study

A. *sQuAD*

Stanford Question Answering Data-set [19] abbreviated as SQuAD is a corpus of reading comprehension data-set that consists of 100,000+ questions. The passages are from Wikipedia articles. The answer to each question is a segment of the text from the corresponding passage. The traditional datasets consist of close-style queries with single-word answers. On the contrary, the answers to questions from SQuAD can include long phrases. This enables the model to extract more features from the sentences leading to a better convergence towards the answer. This data set is obtained by curating passages and crowd-sourcing question-answers on these passages. It maintains the diversity in answer types and difficulty level in the questions. It also maintains a degree of syntactic divergence between the question and answer sentences. We have leveraged these properties of SQuAD to improve the performance of our Question-answering model.

B. *Eng-Hin dataset*

The IIT Bombay English-Hindi Parallel Corpus [20] is the largest publicly available corpora consisting of 14.9 lakh parallel segments. In each segment, there is an English sentence and its corresponding Hindi translation. This data set has been enhanced for machine translation. It has been used in tasks of Asian Language Translation(2016 and 2017). This corpus is a compilation of a variety of existing corpora. Some of the corpora include Judicial Domain corpus - I, Judicial Domain corpus - II, Mahashabdkosh, Indian Government corpora, Hindi-English Lined Wordnet and Gyaan-Nidhi Corpus. We are using this data set in our translation model.

C. *BERT*

BERT stands for Bidirectional Encoder Representations for Transformers. BERT is trained on unlabelled data by jointly conditioning on both sides. This gives it a state-of-the-art performance in a wide variety of tasks like question answering and language modelling without a major change of the architecture. BERT has an edge over the other models due to the Bi-directional aspect. It helps find the proper relation and context [21] between the words of each sentence which makes it accurate. It also achieves state-of-the-art performance in eleven natural language processing tasks. The transformer encoder reads the whole sentence and processes it in both directions. BERT alleviates unidirectional constraint by leveraging Masked Language Model(MLM) [22] and Next Sentence Prediction(NSP) [23].

In Masked Language Model, 15% of the words are masked for the model to predict the words from the sequence and vocabulary. This is achieved by adding a classification layer on top of an encoder output. It is then multiplied by the output vector from the classification layer with the embedding matrix and transformed into a vocabulary dimension. In the end, the probability of each word in the vocabulary is calculated in place of the mask using the Softmax activation function.

For Next Sentence Prediction, the BERT model is fed with a pair of sentences as input which helps it to predict the second

subsequent sentence based on the first sentence. The model is trained with 50% data in which the second sentence is the subsequent sentence of the first in the inputted document. In the other 50% of the input data, the second sentence is picked at random or with no context with respect to the first sentence. This helps the model to distinguish and differentiate whether there is a context between the two sentences. To fine-tune BERT for Question-Answering we make the first sentence as the context/text from which we want the answers and the subsequent sentence as a question which we are asking. For the model to distinguish between two sentences, the [CLS] token is added before the first sentence and a [SEP] token is added at the end of each sentence. Then the entire sentence is inputted and the output token is transformed using a simple classification layer. This helps in calculating the probability of the next sequence using the Softmax activation function. BERT's performance in question answering is incorporated into our proposed model.

D. LSTM

LSTM stands for Long Short-Term Memory. They are inspired by RNNs and are used in applications like handwriting recognition, speech and language recognition [24], learning grammar, music composition etc. They can also be used for standard classification/regression problems like the stock price prediction [25].

RNNs can solve the purpose of sequence handling and NLP based problems. They are great for short contexts but cannot remember big paragraphs that contain a lot of information. For such large contexts, we need the model to remember the various associations in the paragraph and understand the context that makes sense. For example, given a sentence, *India has 29 states*, the RNN can answer the question, *How many states India has?* But for the sentence *I love to eat samosa but I'm living in the USA*, RNN fails to understand the question: *What is your favourite cuisine?* This is because RNN fails to remember the context. A Vanilla RNN fails to understand the context of a long paragraph due to the problem of vanishing gradient. During back-propagation, the weight gets updated with the error term. The error term is the product of the learning rate, the error term of the previous layer and the input to the current layer. As we move towards the initial layer of the architecture, this derivative of the error function keeps on multiplying until it gradually diminishes and approaches zero. This makes the process of training the initial layers very difficult. This is why RNNs can remember short contexts but fail when a lot of words are fed in. This issue led to the development of the LSTM, which is a tweaked version of RNNs. The LSTM makes some modifications to the information fed into it. The model selectively remembers certain things from the information that it assumes to be important for the context and forgets the things that don't contribute much to the context. RNNs are not capable of this prioritization. LSTMs are able to achieve this by various addition and multiplication mechanisms while making the information flow through the cell states.

The Seq2Seq LSTM translation model requires a final dense layer with a Softmax activation function. The outputs from the decoder will be one-hot encoded vectors. The decoder LSTM takes the cell state and hidden state vectors from encoder LSTM and the input sentence. The output of the decoder LSTM is passed through the final dense layer to get decoder outputs. So the input to the encoder LSTM which is the embedded vector is fed to get the encoder outputs, hidden state outputs, and cell state outputs. we pass the hidden state and cell state outputs to the decoder LSTM as encoder states. The decoder LSTM also takes output sentences with START token appended. The outputs from the decoder LSTM are passed through the dense layer. Considering the advantages of LSTM, we have included LSTM into our model to achieve our English-Hindi translation.

IV. EXPERIMENTATION AND RESULTS

Our model is the combination of the two aforementioned architectures. We use BERT for question-answering and LSTM for the translation. The input to the system would be a passage and some questions related to the passage. The user can enter any passage and there is no restriction to the number of questions.

The following sequence of steps are followed for generating answers from an input passage:

- Convert the passage-question pair into a SQuAD format. All the unnecessary indentation and punctuation are removed.
  Now tokens are added to differentiate between paragraph and question.
  The SQuAD format object contains five parameters; question id, question text, list of tokens in the passage (doc-tokens), start and end position.
- After converting the input to a SQuAD example, the features are extracted.
  Tokenize the passage and the query by the framework tokenizer to get query tokens and all passage tokens.
  We limit the length of query token length to 64 and passage token length to 384.
  Passages longer than the passage tokens length can also be inputted.
  To achieve this, a sliding window approach is incorporated. We take chunks up to our max length of 128 tokens which are specified as document stride length.
- Next, convert the data into segments for the Next Sentence Prediction. Each segment has 2 sentences. We made sure 50% are consequent sentences so that they have the same context and the other 50% of the segments are randomly selected.
  After Next Line Prediction, every segment is prefixed by the CLS token and at the end, every sentence is added by a SEP token.
- Now we mask 15% of the words in the paragraph randomly to perform Masked language modelling. Finally, we get the input features object. We then convert the input ids, input mask and segment ids to tensors. We create a tensor data-set and feed it in batches to the pre-trained BERT model.

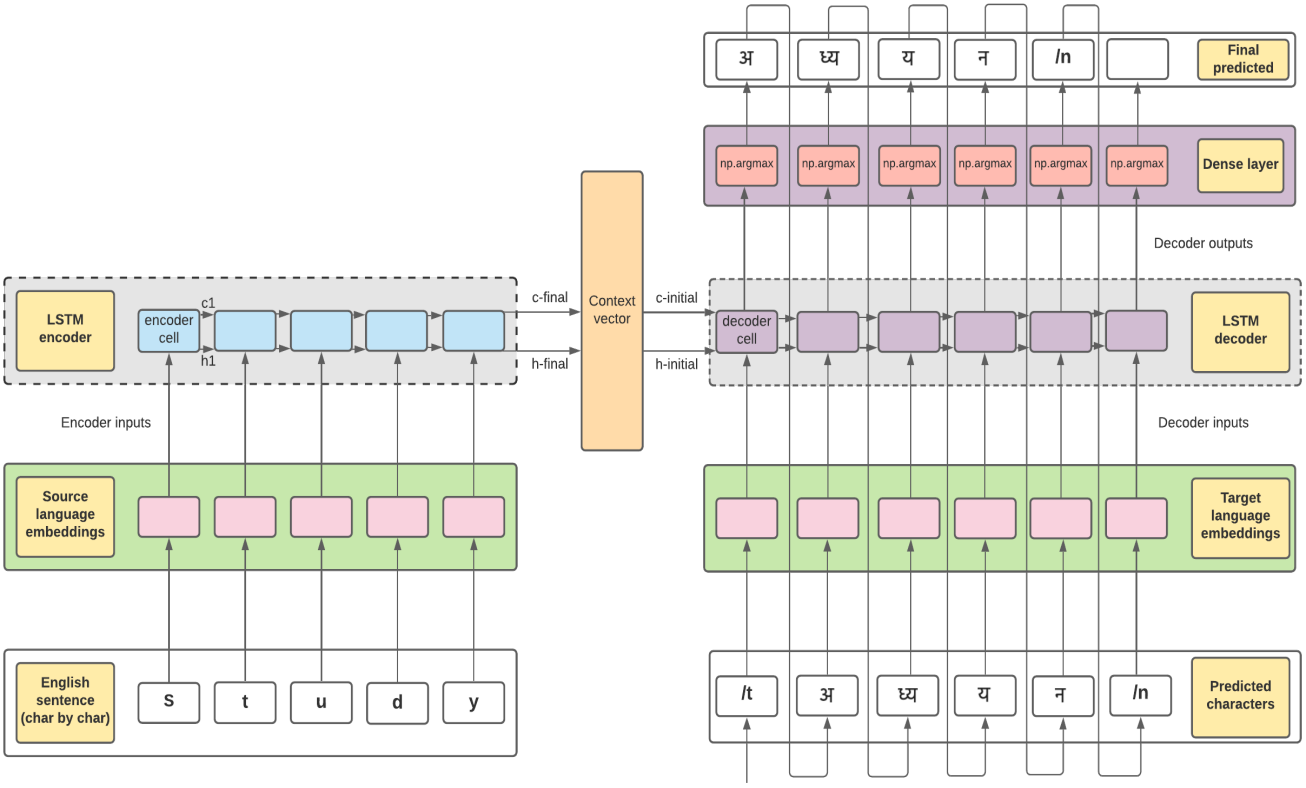

Fig. 1: LSTM Architecture

To fine-tune the model for Question-answering, we make the first sentence as context and the subsequent sentence as the questions.

- After running the data through the model, we get many results with different probabilities. Now, we have to process the raw results to get the final answer.
- We parse all the results and discard the results with invalid parameters and tokens. We sort the sequence of predictions based on the sum of start index logit and end index logit. Index logit is the likelihood of the index being the answer. Before we output the result, we need to remove all the added tokens of the result and clean the white space to get the final answer text.

Now, the answer generated answer from the BERT model has to be translated to Hindi. Various pre-processing techniques are performed for training the translation model. The following sequence of steps are followed for generating translation from an input sentence:

- We convert the data in English into lowercase and remove punctuation and spaces in both languages. Nextly the unwanted stop words are removed. Stemming is performed, where all the words in the sentence are replaced by the root words by removing all the suffixes. Then, the sequence is lemmatized. It groups the inflected forms of the word together that can be analysed to a single word (lemma). Finally the START and the END tokens to each sentence.
- Now we compile all words together from the dataset in both the languages and build a vocabulary of it. We sort the words and give every word a unique index. Using the vocabulary that is built, we tokenize all the sentences for training.
- Now randomise the data and then spilt it in the ratio of 8:2 for training and testing respectively. Batches of size 128 for training are created.
- We then designed a model architecture of 7 layers consisting of 2 input layers, 2 embedding layers, 2 LSTM layers and 1 dense layer as shown in Figure 1. The purpose of our model is to turn encoder input data and decoder input data into decoder target data.
  From Input Layer 1, the English sequence is sent to the encoding embedded layer where the word is transformed into a vector representation. This vector is used by the encoder LSTM(LSTM layer 1) which finds the context of English words in the sentence. Encoder LSTM generates encoder outputs, hidden state and cell state. The decoder LSTM (second layer) takes the hidden states, cell states and the tokenized input sentence. These input sentences are passed through the embedding layer before passing as an input to the decoder LSTM. The outputs from decoder LSTM(second layer) are sent to the final dense layer where the sequences are translated to Hindi. These sequences are in a one-hot encoded format.
  Finally, we decode the Hindi sequence into the Hindi text using the Hindi vocabulary we initially made. As a final processing, unwanted tokens and spaces are removed to

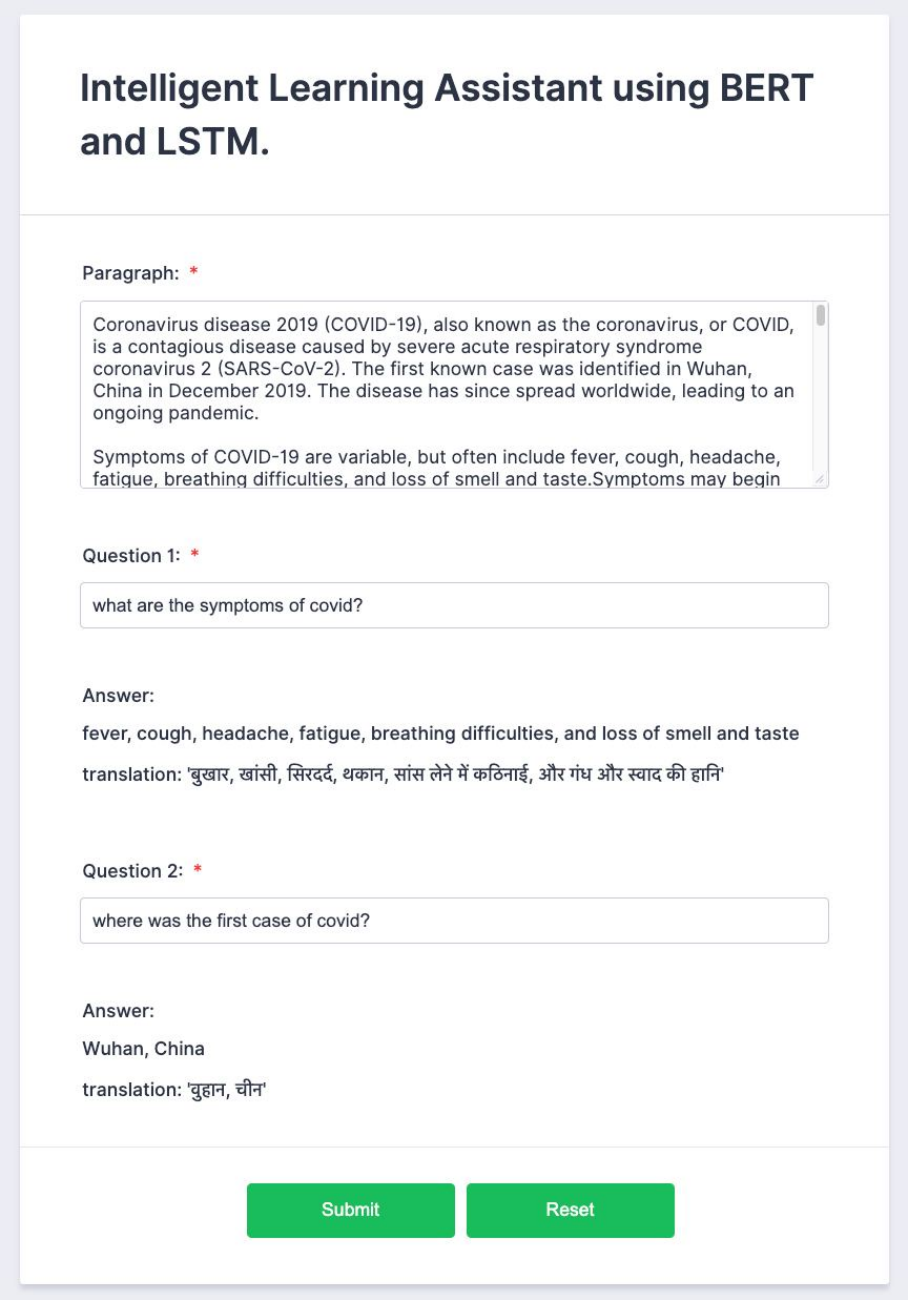

Fig. 2: Learning Assistant Output

get the final Hindi sentence output.

For an ease of use, a simple interactive page is developed. The user can enter the passage and the required questions into it. The answers in English and Hindi would appear underneath each question. The sample is as shown in Figure 2.

V. Conclusion and Future Work

We have introduced a novel and interactive way to make the students' learning experience better by addressing a few problems they face while learning remotely and digitally. The model proposed will help the learners understand the gist of the topics that is read and help answering any questions that is fed using the BERT architecture. The LSTM network also translates the answers for the user's convenience so they can understand the answers in their mother tongue. Many learners can be benefited by using our model to learn and revise the chapters. The model can be extended to support translation into different regional languages. Enabling text to speech also can be achieved to aid visually disabled people. We can reduce the size and increase the speed of training using methods such as distillation and quantization. This will make our system more accessible for mobile device users.

REFERENCES

[1] *A. S. Kuttattu, G. S. Gokul, H. Prasad, J. Murali and L. S. Nair, "Analysing the learning style of an individual and suggesting field of study using Machine Learning techniques." 2019 International Conference on Communication and Electronics Systems (ICCES), 2019, pp. 1671-1675, doi: 10.1109/ICCES45898.2019.9002051*

[2] *Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).*

[3] *Ralf C Staudemeyer, Eric Rothstein Morris. "Understanding LSTM, a tutorial into Long Short-Term Memory Recurrent Neural Networks." arXiv:1909.09586, 2018.*

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

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

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

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

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

[9] *Sukhbaatar, Sainbayar, et al. "End-to-end memory networks." Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2. 2015.*

[10] *Kumar, Ankit, et al. "Ask me anything: Dynamic memory networks for natural language processing." International conference on machine learning. PMLR, 2016.*

[11] *Akbik, Alan, Duncan Blythe, and Roland Vollgraf. "Contextual string embeddings for sequence labeling." Proceedings of the 27th international conference on computational linguistics. 2018.*

[12] *Sanjanasri, J. P., M. Anand Kumar, and K. P. Soman. "Deep learning-based techniques to enhance the precision of phrase-based statistical machine translation system for Indian languages." International Journal of Computer Aided Engineering and Technology 13.1-2 (2020): 239-257.*

[13] *Lo Alla, Dione Cheikh Bamba, Nguer Elhadji Mamadou, Ba Sileye O. Ba, Lo Moussa. "Using LSTM to Translate French to Senegalese Local Languages." ICLR AfricaNLP workshop, 2020.*

[14] *Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, Shikha Jain. "Machine translation using deep learning: An overview." International Conference on Computer, Communications and Electronics, 2017.*

[15] *Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato. "Unsupervised Machine Translation Using Monolingual Corpora Only." ICLR, 2018*

[16] *Joseph Turian, Lev Ratinov, and Yoshua Bengio. "Word representations: A simple and general method for semi-supervised learning." In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010.*

[17] *Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents." In International Conference on Machine Learning, 2014.*

[18] *Beibei Ren. "The use of machine translation algorithm based on residual and LSTM neural network in translation teaching." In the journal PLOS ONE, 2020*

[19] *Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).*

[20] *Kunchukuttan, Anoop, Pratik Mehta, and Pushpak Bhattacharyya. "The IIT Bombay English-Hindi Parallel Corpus."*

[21] *Lo, Chi-kiu, and Michel Simard. "Fully unsupervised crosslingual semantic textual similarity metric based on BERT for identifying parallel data." Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL). 2019.*

[22] *Nozza, Debora, Federico Bianchi, and Dirk Hovy. "What the [mask]? making sense of language-specific BERT models." arXiv preprint arXiv:2003.02912 (2020).*

[23] *Shi, Wei, and Vera Demberg. "Next sentence prediction helps implicit discourse relation classification within and across domains." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.*

[24] *Aparna, C., and M. Geetha. "CNN and Stacked LSTM Model for Indian Sign Language Recognition." Symposium on Machine Learning and Metaheuristics Algorithms, and Applications. Springer, Singapore, 2019.*

[25] *S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1643-1647, doi: 10.1109/ICACCI.2017.8126078.*