IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 2, June 2024, pp. 2333~2341
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i2.pp2333-2341

A multilingual semantic search chatbot framework

Vinay R., Thejas B. U., H. A. Vibhav Sharma, Poonam Ghuli, Shobha G.
Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, India

Article Info

Article history:

Received Jun 23, 2023
Revised Oct 10, 2023
Accepted Jan 6, 2024

Keywords:

Bidirectional encoder representations from transformers
Chatbot
Cross-lingual question answering dataset
Natural language processing
Stanford question answering dataset
Universal sentence encoder

ABSTRACT

Chatbots are conversational agents which interact with users and simulate a human interaction. Companies use chatbots on their customer-facing sites to enhance user experience by answering questions about their products and directing users to relevant pages on the site. Existing chatbots which are used for this purpose give responses based on pre-defined frequently asked questions (FAQs) only. This paper proposes a framework for a chatbot which combines two approaches-retrieval from a knowledge base consisting of question answer pairs, combined with a natural language search mechanism which can scan through the paragraphs of text information. A feedback-based knowledge base update is implemented which provides continuous improvement in user experience. The framework achieves a result of 81.73 percent answer matching on stanford question answering dataset (SQuAD) 1.1 and 69.21 percent answer matching on SQuAD 2.0. The framework also performs well on languages such as Spanish (67.32 percent answer match), Russian (61.43 percent answer match), and Arabic (51.63 percent answer match). By means of zero shot learning.

Corresponding Author:

Vinay R.
Department of Computer Science and Engineering, RV College of Engineering
RV Vidyanikethan Post, Mysore Road, Bengaluru 560059, Karnataka, India
Email: rvinay73@gmail.com

1. INTRODUCTION

Chatbots play a major part in enhancing user experience on websites due to ease of access and round the clock availability. They can be useful in fast website navigation by directing the user to relevant parts of the website based on the user's query. Chatbots which answer queries about products on a website can also significantly reduce the time and effort needed by the user to contact customer support. There are two broadly used approaches to design chatbots for the above stated purpose. One approach involves comparing the user's query to the queries stored in the frequently asked questions (FAQ) database and retrieving the response for the query most similar to the query. This is achieved by generating a vectorized representation of questions using techniques such as bag of words [1], term frequency-inverse document frequency (TF-IDF) [2], Word2Vec [3], [4], global vectors for word representation (GloVe) [5], FastText [6] and then using a similarity metric such as cosine distance or euclidean distance to find the most similar question. The other approach involves training a deep learning model by using the question answer pairs and then using this model to answer user queries based on the relationships it has learnt between the questions and answers in the training set. Recent trends in implementing this approach involve the use of recurrent neural networks (RNNs) [7], [8], sequence to sequence (seq2seq) neural models [8], [9] and long short-term memory networks (LSTMs) [10], [11]. Both the approaches require a very large set of FAQs to cover the entire site's contents. In the second approach, a large FAQ set is also necessary for training the model effectively since deep neural networks (DNN) require large amounts of data to provide accurate results. Modern enterprises have websites which contain many products with detailed information about each of them. Defining FAQs and their answers for

each product is a very exhaustive process and it may not cover all the contents of the site. Another drawback is that in case the website is updated frequently, additional FAQs need to be created for each update. In case of the second approach, the model would also need to be retrained every time new FAQs are added. Both these tasks would result in frequent updates to the chatbot's modules which requires extra human and computational resources.

To solve these issues, we propose a framework which combines the first approach (using vectorized representations of questions to find the most similar question in FAQ database) with a natural language search mechanism. The chatbot answers queries by first going through a knowledge base consisting of question answer pairs, and if a satisfactory answer is not retrieved, it would then perform a search on the entire website's contents to give a relevant answer. The user is also presented with alternative answers at every stage. We have implemented a feedback system wherein if the user finds the response satisfactory after the search phase, the user query and answer are added to the knowledge base. This ensures that the computationally intensive search phase is used less frequently as the usage increases, providing for a self-improving nature. Another objective of the framework is to ensure support for multiple languages to support a use case where enterprises have websites in native languages for users in countries in which English is not the primary language. The system is built completely from open-source technologies such as tensorflow hub, huggingface transformer library and MongoDB.

The concept of a chatbot was first realised when ELIZA [12], a rule based chatbot which used pattern matching, was introduced in 1966. It was used to simulate conversations with a psychotherapist. ALICE was another rule based chatbot which showed improvements over ELIZA by using a simple pattern matching algorithm [13]. Both chatbots were rule based which meant that they wouldn't work well for any query which was outside the rules. To build a chatbot without training from scratch, we would have to make use of transfer learning. The advantages of using sentence level embeddings instead of word level embeddings for transfer learning tasks has been shown in recent works [14]. Universal sentence encoder was introduced, which constructs sentence-level embeddings [15]. There were two variants introduced, one used a transformer architecture, the other used a deep averaging network (DAN). The transformer architecture [16] is superior to RNNs, LSTMs in machine translation tasks and reduces training time. It is mostly reliant on an attention mechanism which allows the architecture to formulate relationships between words in an input sequence and gets rid of recurrences and convolutions. The sentence encoder model that uses the transformer architecture generates sentence embeddings that include context aware word representations which take into consideration the identity and order of other words. The DAN [17] based sentence encoder produces sentence embeddings which are produced when a feed forward DNN is used to process the averaged word and bi-gram embeddings. The transformer-based model showcases higher accuracy whereas the DAN based model is faster. The addition of multilingual convolutional neural network (CNN) and transformer-based variants of the Universal Sentence Encoder [18], [19] which have robust performance in tasks like cross-lingual semantic retrieval makes it a very versatile model. All the variants generate a vector of length 512 for a variable length input text sequence.

Text-to-text-transfer-transformer (T5) framework was introduced in [20] which converts every natural language processing (NLP) task to a simplified common format which involves taking input text strings and genrating target text as output. The massive clean crawled corpus (C4) text dataset, which is twice the size of Wikipedia, was used in training of the T5 model. The model produced class-leading results on tasks such as machine translation, text classification. This common NLP task modelling format allows for very simple application of the framework to every problem, with the hyperparameters, model format remaining unchanged irrespective of the type of NLP task. This aspect of T5 enabled it to be used in the process of question generation (QG). As stated earlier, the process of manually defining FAQs can be very time consuming. Therefore, QG can form a basis for chatbot data augmentation. Bidirectional encoder representations from transformers (BERT) [21] and generative pre-trained transformer 2 (GPT-2) [22] are some of the latest models which were successful in the task of generating questions given a context. Stanford question answering dataset (SQuAD) [23] is a prominent dataset used for fine tuning models for this task. The format of the dataset is such that it consists of paragraph (context), questions about the paragraph content and answer to each question as spans within the context. The dataset consists of over 500 paragraphs and 100,000 question answer pairs. The way this format is used to finetune a model for QG is by giving the paragraph (context) as input and the questions which are based on that paragraph as a target output [24]. Proposes a T5 based QG model using the same procedure, which produced results similar to previous class-leading QG models, with very less training. To augment a chatbot's training data we would require not only QG but question answer pair generation. This can be achieved by using T5 in a multi task variation, where its is would first select spans of text as answers, then generate questions based on the answer, and finally it would act as a question answering system by finding the answer for the generated query and evaluate the correctness of the query by comparison the obtained answer with the actual answer [25].

The ability of a chatbot to search through text and not just rely on FAQ pairs can be implemented

using the advancements made in machine reading comprehension (MRC) that involves a machine combing through a set of paragraphs to answer questions about these paragraphs. SQuAD [23] is one of the benchmark datasets used in MRC. The structure of SQuAD is described in the previous paragraph. It was improved over earlier available datasets [26]-[29] because none of them had the combination of high quality and a large size, which SQuAD showcases. Some researchers describe the class-leading techniques to perform MRC on SQuAD [21], [30]-[33]. Out of these, BERT [21] had the best performance. BERT utilizes the transformer architecture proposed in [16] in addition to bidirectional training mechanism which uses masked language modelling (MLM) to randomly mask words and provide sentence word predictions using the words present before and after it. This simultaneous consideration of left and right surroundings of a word leads to better contextualised representations. Another training strategy used is next sentence prediction where two statements are provided as input and the model tries to forecast if the second statement follows right after the first in the real context. This leads to the model learning relationships at a sentence level. The pretrained model is then fine tuned for different tasks. The details of fine-tuning BERT on SQuAD for MRC are described in [21]. Pires *et al.* [34] showcase that multilingual BERT [21] (pre-trained on text from 104 different languages) performs very well when fine tuned in one language for a given operation and evaluated on another language (zero shot cross lingual model transfer). The tasks used were part-of-speech tagging (POS tagging) and named entity recognition (NER). Konovalov *et al.* [35] shows multilingual BERT's promising performance in zero shot cross lingual model transfer for MRC task.

This survey on chatbot history, benchmark datasets in textual domain, class leading approaches in a variety of NLP operations has been crucial in building a chatbot framework which can be used in enterprise websites. The chatbot framework we propose leverages sentence level embeddings, question answer pair generation, MRC techniques. The survey has also helped us build our framework using only open-source technologies and ensure multilingual support. The related works were especially useful in modelling the requirements for an enterprise website chatbot in terms of different NLP tasks.

2. METHODOLOGY

In this section, a two-stage chatbot framework is proposed that can evolve with experience. The proposed framework can work well for many languages. The discussion will be based on the design of each of these stages and how each stage contributes to answering the questions posed by the user.

2.1. System overview

The system involves two stages-database search stage and paragraph search stage. Figure 1 showcases a high-level overview of the framework. Once a question is asked based on the similarity score of the question and user feedback the system will try to get the answer for the query asked by the user. The database search stage searches whether an answer exists in the database. If a similar question is not present or the user is not satisfied with the answer given, paragraph search stage is invoked. These two stages will be discussed in detail in the further sections. The system works on user feedback and hence we assume that the user doesn't make any irrational choices that can affect the overall performance of the system.

2.2. Database search

The database search is the first stage of the system. This stage consists of an existing database that tores some predetermined or already queried questions along with the appropriate answers. A few question answer pairs have been added to the database by the use of the T5 model [20]. When a question is asked, the system first converts the question to its corresponding question embedding using the multilingual CNN based Universal Sentence Encoder model [15], [18], [19] and we get the question embedding q for the question. Embeddings v_i of all the questions present in the database are obtained and then the embeddings are compared using the similarity function [15] given in (1). Then the most relevant answers are sorted in a data structure D based on similarity. The data structure D is then returned. The algorithm used to calculate similarity is given in (1):

$$sim(q, v_i) = 1 - arccos((\frac{q.v_i}{|q||v_i|}))/\pi \quad (1)$$

Algorithm 1: Computing similarity scores

ComputeScores(V)

1) Input the question Q
2) Find the embeddings of the question Q. Let it be q
3) for all the values V_i present in V do
4) v_i = embeddings of V_i

5) score = sim (q, v_i)
6) Store it in a data structure D
7) [End of for]
8) Sort D
9) return D

After calculating the similarity scores, the answer for the question corresponding to highest similarity score is selected as the answer for the query. Here a limiting threshold is considered below which a question is declared not present in the database and paragraph search is invoked. The limiting threshold is set to 0.75. The limiting threshold is found by testing the model on the quora question pair similarity dataset and values ranging from 0.70 to 0.80 were checked. The model performed best on 0.75. Now if the response satisfies the user, he will proceed with the next question or else he will ask for other answers. Then the system presents the user with three different questions in order to understand if the question asked is similar to any of the next most similar questions that are stored in the data structure. If the user finds any of the questions similar, the system then retrieves the answer for the chosen question. In case the user is still dissatisfied with the answer, paragraph search stage is invoked.

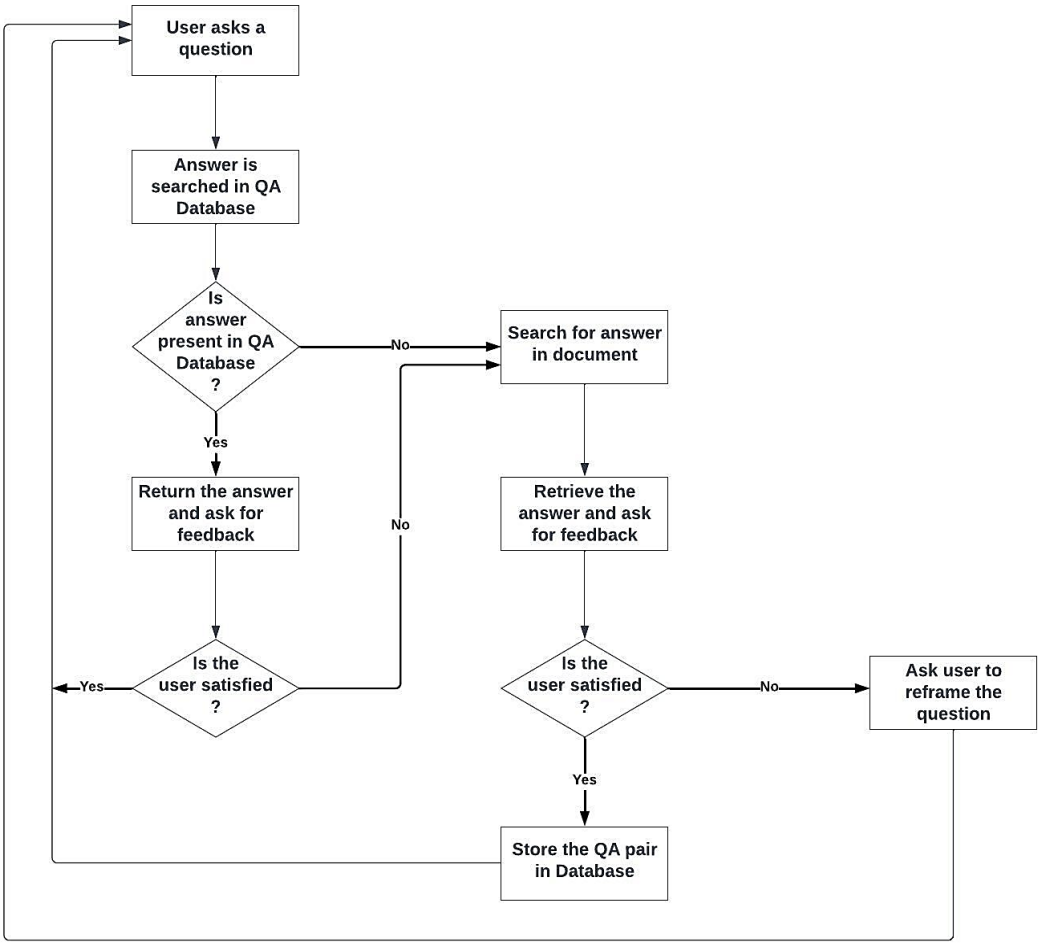

Figure 1. System overview

2.3. Paragraph search

In this stage, the answer for the question is searched in paragraphs. At first the embeddings pi of all the paragraphs are obtained using the multilingual CNN based Universal Sentence Encoder model [15], [18], [19]. Then the similarity scores are obtained using (1) and Algorithm 1 is employed to compute the similarity scores of all paragraphs. Once the data structure D is obtained, the paragraph with highest similarity is taken and passed onto the multilingual BERT model fine tuned on SQuAD [21] which retrieves the answer given a

question and paragraph. Here, zero shot cross lingual transfer [34], [35] is utilized for response prediction if the paragraph or queries are in languages other than English. Now the answer obtained is sent to the user for his feedback. If the user is satisfied, he asks the next question or else the system returns the next best answers. If the user is still dissatisfied, the system might not have the required information or requests the user to reframe the question. Based on user feedback, the answer to a given query is either stored in the database or discarded completely.

3. RESULTS AND DISCUSSION

This section discusses the results of the proposed method. This experimental setting, the metrics and comparison between various standard class-leading methods are discussed in the proceeding sub sections. The 2 experminets conducted for performance evaluation of the chatbot framework are the question pair similarity test, paragraph search and answer retrieval task. These experiments evaluate the database search and paragraph search portions of the framework separately.

3.1. Experimental setup

The experiments involved in the proposed work are conducted on a Windows machine with Intel i7 8th Gen at 3.2 GHz coupled with an NVIDIA RTX 2060 GPU. The Universal Sentence Encoder is used from Tensorflow Hub. The embedding size is 512. For paragraph search, SQuAD 1.1 fine tuned multilingual BERT base model is used which is available on HuggingFace transformers. The embedding size of this model is 512. The maximum sequence length to train the BERT model was 512 and the maximum expected answer length was 64.

3.2. Dataset

Since the BERT model has been fine-tuned and pre-trained on SQuAD 1.1, we didn't retrain the model further on other datasets. Instead, the model is directly tested on SQuAD 2.0 and cross-lingual question answering dataset (XQuAD). The model was fine tuned on English alone but due to BERT's inherent capacity to generalise (i.e., zero shot learning) it was decided to test it on XQuAD without training. The Quora question pair similarity dataset is used to understand how well Universal Sentence Encoder can compare the questions as similar based on a predefined threshold.

3.3. Evaluation metrics

We evaluate the framework in both stages. The evaluation of question similarity in database search is done using F1 scores. For paragraph search, evaluation is done on two things-firstly whether the paragraph chosen is right and secondly if the final answer obtained is right. The paragraph exact match is used [36]-[38] to evaluate whether the paragraph chosen is the right one among four best paragraphs considering the user feedback which takes into account the four best paragraphs to choose answers. Exact match and F1 scores [39] for evaluating if the answer chosen is right from the chosen four paragraphs.

3.4. Question pair similarity

The question pair similarity test is done on the quora question pair similarity dataset by using the Universal Sentence Encoder. A total of 15000 question pairs are chosen out of which 9409 are not similar pairs while 5591 questions are similar. The threshold for a question pair to be marked as similar is set to 0.75. With the given threshold, the model obtains a 0.70 F1 score and a 69.91 accuracy. The test results are showcased in Table 1. The false positive count is comparable to the true positives and true negative count. This might lead to unnecessary feedback in the database search stage of the framework.

Table 1. Quora question pair similarity results

	Predicted NO	Predicted YES
Actual NO	5329	4080
Actual YES	433	5158

Table 2 gives a performance analysis of the approach adopted in this framework versus the other models. The results in the table are taken from [40]. Due to the zero-shot nature, the approach performs fairly well. The accuracy when compared with other transformers models [20], [41], [42] is lower. Although the results obtained are not the best, the model is still persisted with due to its multilingual capacity and DAN connections, as it can understand and correlate complex sentences. This is further discussed in the next section where the results for the paragraph search is evaluated.

Table 2. Performance of various models on quora question pair similarity dataset

Model	Accuracy (%)	F1 score (%)
Logistic regression (LR) with Unigrams	75.4	63.8
LR with Bigrams	79.5	70.6
Support vector machine (SVM) with Unigrams	75.9	63.7
SVM with Bigrams	79.9	70.5
Decision tree	73.2	65.5
Random forest	75.7	66.9
Gradient boosting	75.0	66.5
Continuous bag of words (CBOW)	83.4	77.8
LSTM	81.4	75.4
LSTM + Attention	81.8	75.5
Bi-directional long short-term memory network (BiLSTM)	82.1	76.2
BiLSTM + attention	82.3	76.4
Proposed model	69.9	70.0

3.5. Paragraph search and answer retrieval

The performance evaluation for paragraph search and answer retrieval is done using SQuAD 2.0 and XQuAD dataset. For this purpose, a total of 25 paragraphs and 315 related questions are chosen from SQuAD 2.0 dataset and 25 paragraphs and 153 questions are chosen from XQuAD dataset. The outcomes obtained are showcased in Tables 3 to 5. The exact match metric is used for evaluating the results obtained from Universal Sentence Encoder and BERT multilingual model. The Universal Sentence Encoder performs very well on most of the languages in recognizing the exact paragraph that contains the answer. The accuracy obtained is pretty high as shown in Tables 3 and 4. Due to the zero-shot nature of the system, the model obtains high accuracy without any fine tuning. So, this model acts as the suitable agent for recognizing the paragraph that contains the answer.

Table 3. Exact match testing results on XQuAD dataset

Language	Paragraph exact match (%)	Answer exact match (%)
ES	100.0	67.32
AR	94.12	51.63
DE	99.34	64.05
EL	60.13	29.41
HI	36.60	18.95
RU	98.69	61.43
EN	98.03	75.16

Table 4. Testing results on SQuAD 2.0 dataset

Paragraph exact match (%)	Answer exact match (%)	F1 Score (%)			
		First	Second	Third	Fourth
80.0	69.21	52.70	26.43	24.20	17.65

Table 5. F1 scores obtained while testing on XQuAD dataset

Language	F1 score (%)			
	First answer	Second answer	Third answer	Fourth answer
ES	71.85	41.69	28.46	25.18
AR	46.92	26.93	18.28	11.30
DE	67.75	42.41	35.35	22.79
EL	22.56	15.79	11.15	4.83
HI	14.74	2.98	2.68	2.38
RU	62.70	32.15	24.90	20.74
EN	79.42	48.02	39.15	30.12

The F1 score and answer exact match is used to assess the answers obtained from the multilingual BERT model. The evaluation results are showcased in Tables 4 and 5 for XQuAD and SQuAD 2.0 datasets respectively. Here, four different F1 scores correspond to the response that is obtained based on the best paragraph selected and also the answers retrieved from the feedback mechanism. The multilingual BERT model achieves an 81.73 percent exact match score and an 89.009 F1 score on SQuAD 1.1 dataset without the paragraph searching technique. This result is comparable to many class-leading models trained and tested on

SQuAD 1.1 dataset. These results are compared in Table 6. The values obtained here are taken from [21] Although the BERT multilingual model is not fine tuned on SQuAD 2.0 or on XQuAD, the model achieves pretty good results comparable to some pretrained models [43].

Table 6. Performance of various models fine tuned on SQuAD 1.1

Model	Exact match	F1 score
BERT large (single+TriviaQA)	85.1	91.8
BERT large (ensemble+TriviaQA)	87.4	93.3
BERT large (single)	84.1	90.9
BERT large (ensemble)	85.8	91.8
Proposed model	81.73	89.009

4. CONCLUSION

In this paper, a chatbot framework is introduced which can be used in enterprise sites containing a lot of information. The ability of our framework to perform natural language search on text such as entire website content is useful in the above-mentioned use case because FAQs have limitations of covering entire content. Use of QA pair generation as a data augmentation technique aims to eliminate the effort required for manual FAQ generation. The feedback-based update facilitates the continuous improvement of performance of the chatbot by storing back the new QA pair back to the database and hence, reducing usage of computationally intensive paragraph search phase. The paragraph search phase relies on embeddings obtained from the Universal Sentence Encoder for comparing similarities. In case the embeddings are not stored, the time taken to get the embeddings of the data present increases. In order to avoid this over reliance, data can be grouped based on similarities so that when querying a specimen of that grouped data will give a similarity score which will be similar to most of the data in the group. Using TF-IDF and other document reranking techniques to query data can also speed up the process and reduce dependency on Universal Sentence Encoder for obtaining similarities. Fine tuning multilingual varaints of Universal Sentence Encoder and BERT on datasets such as XQuAD can result in performance improvement of the proposed framework.

REFERENCES

[1] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1–4, pp. 43–52, 2010, doi: 10.1007/s13042-010-0001-0.

[2] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp. 1-12, 2013.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality. arXiv e-prints, page," *Advances in neural information processing systems*, vol. 26, pp. 1–9, 2013.

[5] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1532–1543, 2014, doi: 10.3115/v1/d14-1162.

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, Aug. 2018, doi: 10.1109/MCI.2018.2840738.

[8] J. Cahn, "CHATBOT: Architecture, Design, and Development," Senior Thesis, Department of Computer and Information Science, University of Pennsylvania, Pennsylvania, USA, 2017.

[9] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP 2014-2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734, 2014, doi: 10.3115/v1/d14-1179.

[10] K. Ramesh, S. Ravi shankaran, A. Joshi, and K. Chandrasekaran, "A survey of design techniques for conversational agents," *Communications in Computer and Information Science*, vol. 750, pp. 336–350, 2017, doi: 10.1007/978-981-10-6544-6_31.

[11] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modelling," *Fifteenth Annual Conference of the International Speech Communication Association*, pp. 338–342, 2014.

[12] J. Wizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36-45, 1966.

[13] B. A. Shawar and E. Atwell, "ALICE chatbot: Trials and outputs," *Computacion y Sistemas*, vol. 19, no. 4, pp. 625–632, 2015, doi: 10.13053/CyS-19-4-2326.

[14] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 670–680, 2017, doi: 10.18653/v1/d17-1070.

[15] D. Cer *et al.*, "Universal Sentence Encoder," *EMNLP demonstration, Association for Computational Linguistics*, pp. 1–7, 2018, doi: 10.48550/arXiv.1803.11175.

[16] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5999–6009.

[17] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé, "Deep unordered composition rivals syntactic methods for text classification," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, pp. 1681–1691, 2015, doi: 10.3115/v1/p15-1162.

[18] Y. Yang *et al*., "Multilingual universal sentence encoder for semantic retrieval," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 87–94, 2020, doi: 10.18653/v1/2020.acl-demos.12.
[19] M. Chidambaram *et al*., "Learning cross-lingual sentence representations via a multi-task dual-encoder model," *ACL 2019 - 4th Workshop on Representation Learning for NLP, RepL4NLP 2019-Proceedings of the Workshop*, pp. 250–259, 2019, doi: 10.18653/v1/w19-4330.
[20] C. Raffel *et al*., "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.
[21] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, 2019.
[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," Technical Report Open Al, pp. 1-24, 2019.
[23] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuad: 100,000+ questions for machine comprehension of text," *EMNLP 2016-Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 2383–2392, 2016, doi: 10.18653/v1/d16-1264.
[24] K. Grover, K. Kaur, K. Tiwari, Rupali, and P. Kumar, "Deep learning based question generation using T5 transformer," *Communications in Computer and Information Science*, vol. 1367, pp. 243–255, 2021, doi: 10.1007/978-981-16-0401-0_18.
[25] F. Ç. Akyön, D. Çavuşoğlu, C. Cengiz, S. O. Altinuç, and A. Temizel, "Automated question generation and question answering from Turkish texts," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 5, pp. 1931–1940, 2022, doi: 10.55730/1300-0632.3914.
[26] M. Richardson, C. J. C. Burges, and E. Renshaw, "MCTest: A challenge dataset for the open-domain machine comprehension of text," *EMNLP 2013-2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 193–203, 2013.
[27] J. Berant *et al*., "Modeling biological processes for reading comprehension," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1499–1510, 2014, doi: 10.3115/v1/d14-1159.
[28] K. M. Hermann *et al*., "Teaching machines to read and comprehend," *Advances in Neural Information Processing Systems*, vol. 2015, pp. 1693–1701, 2015.
[29] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The Goldilocks principle: Reading children's books with explicit memory representations," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1-13, 2016.
[30] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bi-directional attention flow for machine comprehension," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1-13, 2017.
[31] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 845–855, 2017.
[32] M. E. Peters *et al*., "Deep contextualized word representations," *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 2227–2237, 2018, doi: 10.18653/v1/n18-1202.
[33] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, "Reinforced Mnemonic reader for machine reading comprehension," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 4099–4106, 2018, doi: 10.24963/ijcai.2018/570.
[34] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 4996–5001, 2020, doi: 10.18653/v1/p19-1493.
[35] V. P. Konovalov, P. A. Gulyaev, A. A. Sorokin, Y. M. Kuratov, and M. S. Burtsev, "Exploring the bert cross-lingual transfer for reading comprehension," in *Proceedings of the International Conference*, pp. 1–9, 2020, doi: 10.28995/2075-7182-2020-19-445-453.
[36] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over Wikipedia graph for question answering," *8th International Conference on Learning Representations (ICLR)*, pp. 1-20, 2020.
[37] Y. Nie, S. Wang, and M. Bansal, "Revealing the importance of semantic retrieval for machine reading at scale," *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 2553–2566, 2019, doi: 10.18653/v1/d19-1258.
[38] K. Nishida *et al*., "Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 2335–2345, 2020, doi: 10.18653/v1/p19-1225.
[39] Y. Zhang, P. Nie, A. Ramamurthy, and L. Song, "Answering any-hop open-domain questions with iterative document reranking," *SIGIR 2021 - Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 481–490, 2021, doi: 10.1145/3404835.3462853.
[40] L. Sharma, L. Graesser, N. Nangia, and U. Evci, "Natural language understanding with the quora question pairs dataset," *arXiv-Computer Science*, pp. 1-10, 2019.
[41] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: decoding-enhanced bert with disentangled attention," *ICLR 2021 - 9th International Conference on Learning Representations*, pp. 1-21, 2021.
[42] Z. Lan *et al*, "Albert: A lite bert for self-supervised learning of language representations," *ICLR*, pp. 1-17, 2020.
[43] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1-11, 2019.

BIOGRAPHIES OF AUTHORS

Vinay R. graduated from RVCE, Bangalore in 2022. His areas of interest include NLP and software engineering applications. He is currently working at Akamai Technologies. He can be contacted at email: rvinay73@gmail.com.

Thejas B. U. graduated from RVCE, Bangalore in 2022. His areas of interest include NLP, image processing, and software engineering applications. He is currently working at Saigeware Technologies Pvt. Ltd. He can be contacted at email: thejas.bu@gmail.com.

H. A. Vibhav Sharma graduated from RVCE, Bangalore in 2022. His areas of interest include NLP and software engineering applications. He can be contacted at email: vibhavsharmaha@gmail.com.

Poonam Ghuli worked as Assistant Professor at RVCE Bangalore, India from 2005 to 2022. Her main area of interest lies in the implementation of machine learning algorithms. She can be contacted at email: poonamghuli@rvce.edu.in.

Shobha G. is professor at RVCE with close to 30 years of experience. Her areas of interest include data mining, data warehousing, and NLP. She can be contacted at email: shobhag@rvce.edu.in.