Automated Customer Consultation System For Pastry Shops

Trung Quoc Nguyen
Department of Information Technology
FPT University
Ho Chi Minh City, Vietnam
trungng46@fpt.edu.vn
Department of Cybernetics and Biomedical Engineering
VSB–Technical University of Ostrava
17. Listopadu 15, 708 33 Ostrava, Czech Republic
quoc.trung.nguyen.st@vsb.cz

Vinh Truong Hoang
Faculty of Information Technology
Ho Chi Minh City Open University,
Ho Chi Minh City, Vietnam
vinh.th@ou.edu.vn

Tuan Le-Viet
Faculty of Information Technology
Ho Chi Minh City Open University
Ho Chi Minh City, VietNam
tuan.lv@ou.edu.vn

***Abstract*—Automated customer consulting is a form of automatic customer care and consulting through texting and chat functions to replace human presence. This research improves the Bi-LSTM language model. We want to increase an automated customer consultation system's accuracy and applicability, which might affect enterprises and traders. Our question-answer system uses querying the entity and model textual similarity to match models. Automated customer care systems use computers or other technology to help customers. It empowers clients to address problems without human assistance in customer care. Human resources can address complex requests or high-value consumers since automation handles many simple and repetitive activities. Many firms utilize it, especially fast-growing ones that need to arrange support.**

***Index Terms*—LSTM, Bleu Core, Q-A systems**

I. Introduction

In certain cases, the sales staff at the pastry shops cannot respond quickly to a large number of customers at the same time; in other cases, customers need advice but do not have time to go to the bakery to learn about the types of cakes; and in other cases, when the sales team at the bakery cannot satisfy many customers at once. Our team felt the need to come up with a solution to the problem described above, hence we came up with the idea of a Question Answering System (QAS) technique that can be used in pastry shops. Question-answering systems (QAS) are extremely beneficial because most deep-learning problems can be thought of as question-answering problems. As a result, this topic is now one of the most researched in computer science. The system can also be taught to understand many different languages, allowing it to reach a larger audience. One of the most difficult aspects of designing a QA system is accurately understanding user queries or questions. Users can ask questions in a number of ways and use informal language that may or may not follow conventional grammar rules. The system must be able to determine the underlying intent of the question and receive appropriate responses from the knowledge base. NLP algorithms are used to analyze the user's words and determine the main topic of the question. The algorithm will then use this data to provide appropriate and accurate answers. Three steps make up the QAS process:
1) Question analysis: analyze questions, classify and refine questions
2) Data analysis: extract potential data and identify potential answers
3) Answer analysis: extracting potential answers and ranking the best answers.
Together, these elements process queries and documents at multiple levels until the correct response is found. For example, if the results of analyzing the questions are poor, then analyzing the answers will certainly yield poor results. Similarly, a good question analysis result does not guarantee a good answer analysis result, and vice versa. As a result, many scientists only pay attention to one QAS component.

We used a hybrid quality assurance technique that combines Query the entity [1] with an NLP model, which was novel. The entity will be queried in response to a user query. If it fails, a text similarity model will be used to discover answers from a large pastry shop QA dataset. Deep learning and similarity comparison are used to represent the text in this article, which is its best work. The model is HBAM. HBAM has two layers: Bi-LSTM and word attention. The Bi-LSTM layer collects sentence forward and backward orientation data. The attention layer identifies keywords in

a phrase.

In business pastry consulting, the Siamese frame and Manhattan distance are used to calculate semantic similarity. The Siamese framework is popular in metric learning [2], [3]. Comparing text cosine similarity index [2] to Manhattan city distance. Our HBAM outperformed MaLSTM [4] and Bi-LSTM [5] in various testing methods and data sets.

II. RELATED WORKS

In the following, we define two problems that are at the center of the chatbot system. Realizing the capacity to understand natural language, or creating the required mechanisms to enable a software system to comprehend natural language queries in the same way that a person would, is the first challenge. The second challenge seeks to get pertinent data from a domain-specific database in order to provide solutions that may be returned to the user:

- Understanding user questions (Intent Detection): To comprehend and handle a user's inquiry, natural language processing (NLP) and natural language understanding (NLU) are used.
- Knowledge base retrieval and storage: the ability to store and query medical queries and answers using a domain knowledge database.

We examine previous research that addresses the two issues mentioned above. Chatbots. Joseph Weizenbaum created Eliza, the world's first chatbot, at the MIT Artificial Intelligence Laboratory in 1966 [6]. But Eliza doesn't grasp the user's inquiry. Psychiatry professor Kenneth Colby's Turing Test was first successfully completed by Parry in 1972 [7]. However, just 48% of psychiatrists are able to accurately identify the true patient based only on their discussion. The multiple-turn dialog decision tree was attempted to be used by Ni et al. [8], [9] to reach a decision on behalf of a patient. According to Helen et al. [10], the accuracy of the model on shorter context talks may be successfully increased by using transfer learning to transfer typical cases from SQuAD to Bible QA. Using N-gram approaches, which effectively minimize the data noise, Dai, Z., et al. [11] developed a "focused pruning method" to limit the candidate result space and make some improvements. "APVA" was introduced by Wang, Y., et al. [12] in order to precisely forecast the relationship between the question and response entities. A novel framework for semantic analysis was suggested by Yih, S., et al. [13]; after the question has been translated and examined in query language, the new inquiry will be connected to the knowledge base. In order to increase performance in 2017, Yu, M., et al. [14] created a hierarchical RNN network employing residual learning. It is able to identify the relationship inside the knowledge base when an input query is present. In addition, they created a straightforward KBQS system that incorporates connection detection and entity linking.

A. Siamese based Semantic sentence similarity

A Siamese Long Short-Term Memory (LSTM) network has been suggested by Mueller et al. [4] to calculate the semantic similarity between two variable-length phrases. LSTM, however, is unable to identify keywords inside a phrase. A Siamese design with bidirectional long short-term memory (LSTM) networks and an attention mechanism was suggested by Baziotis et al. [15]. To capture both two-direction contexts, the model makes use of bidirectional long short-term memory (LSTM).To classify, they take into account the fully connected (tanh) in the last layer, which may lead to overfitting.

B. Word Embedding

One-hot encoding [16] encodes categorical data for machine learning algorithms, boosting model predictiveness. Categorical values are numerical representations of dataset categories. The one-hot encoder displays categories as binary, with 0 indicating no feature and 1 indicating its presence. The model is trained using this binary feature vector. A machine situation is often represented by a one-hot encoding. A decoder determines the machine's binary or gray code status. One-hot machines don't need a decoder to identify their state; hence, this doesn't apply. A bit set to a high value defines the n-th state of an element.

A one-hot vector in the corpus or dictionary is comparable to a feature vector where each feature is assigned a value of 0 or 1, indicating the presence or absence of a word based on its dictionary word number. The feature vector contains all dictionary terms and their indexes. Words and indexes are retained at the same feature vector index. Dictionary embedding of feature vector yields a binary vector.

This project uses the skip-gram model [17] from the **word2vec method**. Data was used to train Gensim's word2vec Python package. Specific hyperparameters were focused on to improve word embeddings. The parameters include training method, dimensionality, context window, and subsampling. Negative sampling was used for this project since it is more computing efficient than hierarchical softmax. The hidden layer of the neural network was expanded to 300, improving word embeddings. A sub-sampling rate of 1e-3 was used to balance the sample's uncommon and common words.

Given a large amount of data, a minimum count of 1 was established to ensure that every word in the corpus was taken into account throughout the training process. The word2vec model underwent training using the processed data, employing the hyper-parameter values indicated earlier. The resulting model was then stored in a file format. The word vectors generated by the word embedding model possess a dimension of m*n, where m represents the size of the dictionary and n represents the size of the hidden layer.

III. AUTOMATED CUSTOMER CONSULTATION SYSTEM FOR PASTRY SHOPS

A. Material

Virtual Sales-Assistant Systems In general, a virtual sales assistant system (Figure 3) is a way to act as a guide when navigating many online resources. The content of the current

website or catalog is presented in response to the input sample.

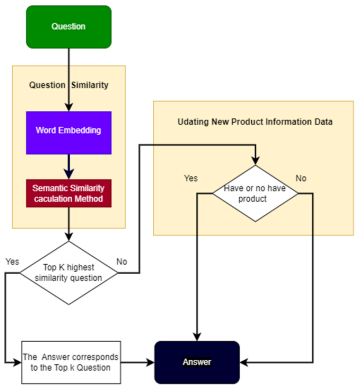

Figure 1. Virtual Assistant Bakery Architecture

Customer input is a customer inquiry. The semantic similarity computation uses the word embedding from the consumer's inquiry. It will discover the highest K rating index and deliver the results. No results will be returned if the customer's question cannot be determined or the rating index is too low. New product data will be updated via inquiries. It allows customers to retrieve bakery product data using short queries concerning pricing, pictures, and key information. The customer will be routed back to the original site to re-enter the search query if it cannot be resolved. Catalog extensions provide extra product information and recommendations to help buyers decide.

Automated customer consulting for pastry shops is an attempt to help human involvement in customer service and consultation by means of automated texting and chatting services. This project will use the Bi-LSTM language model to improve word production with accurate contextual meanings. We want to improve the precision and application of an effective automated customer consultation system, which might impact commercial organizations and trade firms. Our chatbot system uses Query the entity and model textual similarity to match models. We also used F1 Score, Bleu Score, Precision, and Recall to evaluate our technique. The Question_Duplicated data collection contains almost 337,103 similar questions collected via data augmentation.

B. Methods

Processing data: we created our own dataset because we couldn't find one for a pastry business, and then utilized data augmentation to boost density. data for model training. First, we manually produced 5230 question-and-answer combinations for Cake Question and Answer. Customers ask store personnel questions about buying, selling, and consulting cakes. The replies are in Vietnamese. We carefully pick and evaluate question-answer pairings to ensure they meet Vietnamese demands and have the best semantics. We used data augmentation to enhance data density to the greatest achievable level for model training because this data set is too tiny and may not be able to make the model clever after training. We employ Data Augmentation to boost density from the original Question and Answer Cake dataset to create Question Duplicate Data, which our team uses to train models using 337103 question and answer pairings. Using the Data Argument approach, we will build 5 questions with comparable meanings to a question in the original Cake Question Answer data collection and utilize concordance to enhance the data.

We crawled data from the bakery's website; this dataset comprises 17 cake kinds, including their names, prices, images, descriptions, etc. A store-specific cake and sales conversation data collection was built using pastry data. Our data collection is carefully selected to meet client cake consulting needs and accurately reflect the Vietnamese setting.

Data Augmentation (DA) generates several copies of current datasets to increase training data size without further data collection. To improve classification performance, the data must be modified to preserve class categories. Computer vision and natural language processing (NLP) use data augmentation technologies to overcome data availability and diversity issues. Augmented photos are easy to make, but Natural Language Processing (NLP) is challenging because of language. Since it would change context, replacing every word with its synonym is not an option. Data augmentation increases the training dataset, improving the model's performance. An enhanced data distribution should balance similarity and dissimilarity to the original data. Data augmentation methods should strike a balance to avoid overfitting and poor performance.

Manhattan LSTM Model The structure of the proposed Manhattan LSTM (MaLSTM) model is depicted in Figure 5. In this study, we only concentrate on Siamese architectures with linked weights, namely LSTMa and LSTMb. These two networks individually analyze one phrase from a given pair. It is important to note that LSTMa and LSTMb are identical in this context, meaning that they have the same weights. However, the overall unbound version of this paradigm may be more advantageous for applications with asymmetrical domains, such as information retrieval, where search queries have distinct stylistic differences from stored texts.

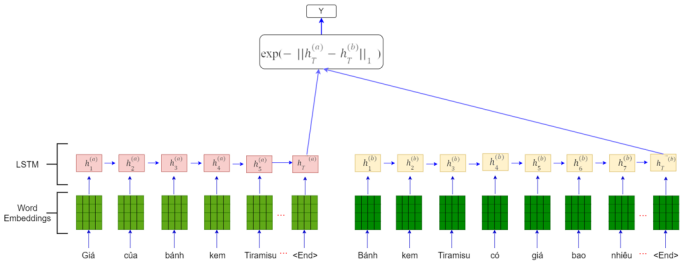

Figure 2. Manhattan LSTM (MaLSTM) model

Hierarchical Bi-LSTM Attention model The schematic

illustrating our newly suggested hierarchical Bi-LSTM Attention model [18] is presented in Figure 6. Its purpose is to facilitate the comparison of semantic similarities. The entire architecture is built upon a Siamese LSTM framework [4]. The entire architecture is built upon a Siamese LSTM framework [4]. The Siamese structure incorporates a single Bi-LSTM layer and a single word attention layer. The sentences on the bottom left and right show the user's input query and the question from the QA dataset. The two inquiries will be shown by employing word embedding first, followed by utilizing Bi-LSTM [19] to construct the whole phrase embedding, taking into account the surrounding context. Subsequently, each Bi-LSTM encoder will be multiplied by a word attention value, which may be regarded as a weight to emphasize the crucial aspect of a phrase. The context vector will be merged with attention to comprehend the sentence representation u_w [20]. Finally, the similarity value will be calculated by taking the weighted total of each hidden state value h_i) and multiplying it by its corresponding attention value. The specific information will be displayed in the subsequent subsections.

The LSTM-based sequence encoder, known as Long Short Term Memory networks (LSTMs), was introduced by Hochreiter and Schmidhuber [1]. The cell state is a crucial component of LSTMs. The state of the cell may be conceptualized as a type of conveyor belt. Through a series of linear exchanges, it moves directly downward throughout the whole chain. Information may flow seamlessly without any alterations. The fundamental concept of LSTM may be categorized into three sequential stages. The initial stage reveals the specific information that will be disregarded by the cellular state. The "forget gate layer" determines this selection by combining the h_{t-1} and x_t representing the hidden layer value at time $t-1$ and the input layer value at time t, respectively. The weight matrix between the hidden layer and output layer is denoted by W_g, while the bias vector is represented by b_g. The desired value, denoted as g_t, may be obtained using this formula, which effectively eliminates irrelevant data.

The next stage is selecting the specific data that will be stored within the cell's state. The input gate layer is responsible for determining the values that will be updated. A tanh layer generates a vector that is used to determine if the new candidate values C_t should be updated in the state. Subsequently, these two values will be combined to provide a state update.

$$j_t = \sigma(W_j.[h_{t-1}, x_t] + b_j) \tag{1}$$

$$\widetilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{2}$$

The determination of whether the prior cell state C_{t-1} will be updated by the new cell state C_t is contingent upon the preceding stages. The product of the previous state and the time increment t, along with the selection of items to be forgotten at an earlier time, will be performed. Subsequently, the product of J_t * C_t will be included. Therefore, a new candidate value will be determined based on the frequency of selection for each state value to be updated. Finally, the outcome will be determined. The result will be refined based on the cell status. Next, a sigmoid layer will be used to determine which components of the cell state will be output. The cell state will undergo a hyperbolic tangent function, which will limit its values between -1 and 1. This result will then be multiplied by the output of the sigmoid gate. This process determines the final output based on the selected component.

$$q_t = \sigma(W_q.[h_{t-1}, x_t] + b_q) \tag{3}$$

$$h_t = q_t \times \tanh(C_t) \tag{4}$$

Word Attention Given a sentence w_{it} , t [0,T]. Firstly, each word of the sentence will be embedded by using an embedding matrix W_e

$$x_{it} = W_e w_{it}, t \in [1, T] \tag{5}$$

We use Bidirectional LSTM [21] to collect the information of each word in both the forward and reverse directions. The bidirectional LSTM consists of a forward LSTM $f^{->}$ and a reverse LSTM $f^{<-}$.

$$\overrightarrow{h}_{it} = LS\overrightarrow{T}M(x_{it}), t \in [1, T] \tag{6}$$

$$\overrightarrow{h}_{ti} = LS\overrightarrow{T}M(x_{ti}), t \in [1, T] \tag{7}$$

In order to represent those keywords in a sentence. We strive to employ attention. Initially, we input the hit into the tanh function to obtain u_{it} as a concealed representation of h_{it}. Next, we compute the significance of each word u_{it} and get a normalized weight a_{it} by using a softmax function. Next, we compute the sentence vector s_i by summing the weighted values of each word.

$$u_{it} = \tanh(W_w h_{it} + b_w) \tag{8}$$

$$\alpha_{it} = \frac{\exp(u_{it})}{\Sigma_t \exp(u_{it})} \tag{9}$$

$$s_i =_t \alpha_{it} h_{it} \tag{10}$$

$$f\left(s_i^{(a)} s_i^{(b)}\right) = \exp\left(-\left\|\sum_i a_i^{(a)} h_i^{(a)} - a_i^{(b)} h_i^{(b)}\right\|\right) \in [0, 1] \tag{11}$$

The calculation relies on the Manhattan distance. The formula that allows for the expression of two sentences can be represented by $s_i^{(a)} = \Sigma_i a_i^{(a)} h_i^{(a)}$ and $s_i^{(b)} = \Sigma_i a_i^{(b)} h_i^{(b)}$. $a_i^{(b)}$ means the attention value is present in both directions. $h_i^{(a)}$ and $h_i^{(b)}$ mean the hidden state value in both directions.

IV. EXPERIMENTS AND RESULTS

A. Dataset

Cake Question and Answer is the first dataset we created manually with 5230 question and answer pairs. The questions and answers are all in Vietnamese and revolve around questions often asked by customers to store staff for the purpose of buying, selling, and consulting cakes. The question and answer pairs are carefully selected and tested by us to help the answers have the most appropriate semantics for Vietnamese people and to help the answers satisfy the needs of Vietnamese clients. After processing and removing duplicate words in sentences, we obtained 5230 pairs of questions and answers of consulting data.

Question Duplicate Data is a set of sentence pairs in the open domain. It has 337103 pairs of sentences tagged with a format like "text1 text2 is_duplicate" which means whether the two sentences are semantically similar. If they have equal semantic meaning, then the tag will be "1"; otherwise "0".

The dataset was collected by us from the official website of the bakery, https://thienthuanphatbakery.com/. The collected data includes 17 types of cakes and all information about each type of cake, such as cake name, cake price, cake image, and cake description. We have consolidated a store-specific cake and sales conversation data set. The data set is carefully checked and selected by us to help customers satisfy their cake consulting requests and is true to the Vietnamese context.

B. Train HBAM, MaLSTM and Bi_LSTM model

a) ***Description of presentation****:* We run 3 models on the Question_Duplicated data set containing 337,103 pairs of questions labeled 1 (same), 0 (different). However, we adjust batch-size, epoch, max_seq_length to be compatible with Question_Duplicated dataset.

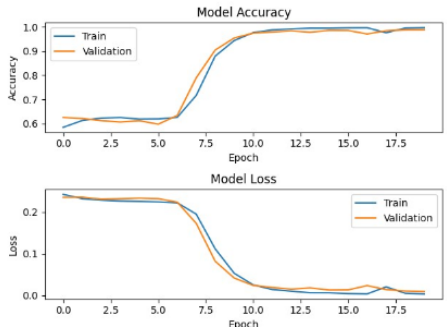

Figure 3. Overview about HBAM Model

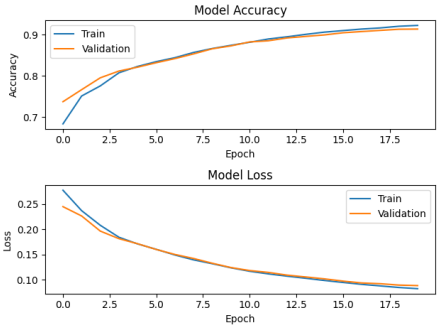

Figure 4. Overview about MaLSTM Model

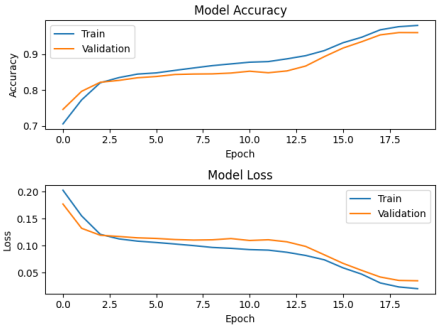

Figure 5. Overview of Bi_LSTM Model

b) ***Setting parameters of training model****:* When setting up the parameters to fine-tune the HBAM, MaLSTM, and Bi_LSTM model, we choose Epochs of 20 for training and set the batch size for both evaluation and n_hidden to 50. Additionally, we set the parameter to increase the batch size of the model to 1024 by setting the Gradient Accumulation parameter. Instead, we increase the batch size via Gradient Accumulation of Direct Increasing because directly increasing the batch size will result in increased GPU memory usage. In this training session, we used a GPU:T4 from Google Colab whose GPU memory ranges from around 13GB. This means we cannot increase the batch size further without running into memory overflow problems. Therefore, the Gradient Cumulative setting ensures that the batch size can increase up to a maximum of 1024. Additionally, we also want to ensure that the batch size is not too low, as too low a batch size will lead to problems with loss of function during training.

C. Final Results of the Similarity Comparison

BLEU Score: BLEU Score is an evaluation metric for Machine Translation tasks. It is calculated by comparing the n-grams of machine-translated sentences to the n-gram of human-translated sentences. Usually, it has been observed that the BLEU score decreases as the sentence length increases. This, however, might vary depending on the model used for translation. Here, we use BLEU Score to compare the answer results between customers and actual question

sets. **BP stands for Brevity Penalty**, which penalizes the score when the Machine Translation is too short compared to the reference (correct) translations.

*a) **Comparison results between models**:*

Dataset_Test: We selected a totally separate testing data set from the training data set in order to conduct an objective model evaluation process. There are 2840 pairs of similar and distinct texts in our test data set. Table 14 illustrates the comparison results that we generated using the test data set.

Model Name	Precision	Recall	F1- Scores	Bleu Score
MaLSTM	0.88	0.87	0.86	0.85
BiLSTM	0.96	0.95	0.95	0.95
HBAM	0.97	0.97	0.97	0.96

Table 4: Evaluation result for the three model

The result: After testing three models, we discovered that the HBAM model is97% more accurate than the other two models and has nearly identical precision, recall, and F1-Score. MaLSTM, with an accuracy of 86%, is the least accurate model. Based on the aforementioned findings, we think that the HBAM model will assess data more accurately, compare comparable words, and provide the most accurate results for users.

V. CONCLUSIONS

We have achieved success in providing solutions with the achievement of creating a consulting data set about project-specific cakes with 337,103 question pairs, more than 5,230 question and answer data, and creating a Q&A system to help advise on cakes for the Cake shop. From this project, our team learned a lot of new knowledge and skills in this field, through the difficulties of creating, selecting, and checking each sentence in the data set for accuracy and precision. Compatible with the Vietnamese language context, our team has gained more knowledge in collecting data and building a Q&A system. From this solution, cake shops can apply this method to their stores to help customers consult cakes faster and meet customers' consulting needs. In the future, we will try to develop projects that have higher accuracy and can be applied to many other fields related to product consulting for customers.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] J. C. P. Wen-tau Yih, Kristina Toutanova and C. Meek, "Learning discriminative projections for text similarity measures," in *Proceedings of the fifteenth conference on computational natural language learning*, 2011, pp. 247–256.

[3] K. Chen and A. Salman, "Extracting speaker-specific information with a regularized siamese deep network," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[4] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[5] K. L. Jacob Devlin, Ming-Wei Chang and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] W. Joseph, "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.

[7] V. Cerf, "Parry encounters the doctor," Tech. Rep., 1973.

[8] L. Ni and J. Liu, "A framework for domain-specific natural language information brokerage," *Journal of Systems Science and Systems Engineering*, vol. 27, pp. 559–585, 2018.

[9] L. Ni, C. Lu, N. Liu, and J. Liu, "Mandy: Towards a smart primary care chatbot application," in *International symposium on knowledge and systems sciences*. Springer, 2017, pp. 38–52.

[10] H. J. Zhao and J. Liu, "Finding answers from the word of god: Domain adaptation for neural networks in biblical question answering," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[11] Z. Dai, L. Li, and W. Xu, "Cfo: Conditional focused neural question answering with large-scale knowledge bases," *arXiv preprint arXiv:1606.01994*, 2016.

[12] Y. Wang, R. Zhang, C. Xu, and Y. Mao, "The apva-turbo approach to question answering in knowledge base," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1998–2009.

[13] X. H. Scott Wen-tau Yih, Ming-Wei Chang and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*, 2015.

[14] M. Yu, W. Yin, K. S. Hasan, C. d. Santos, B. Xiang, and B. Zhou, "Improved neural relation detection for knowledge base question answering," *arXiv preprint arXiv:1704.06194*, 2017.

[15] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 747–754.

[16] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine learning and deep learning for phishing email classification using one-hot encoding," *Journal of Computer Science*, vol. 17, pp. 610–623, 2021.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[18] Q. Bao, L. Ni, and J. Liu, "Hhh: an online medical chatbot system based on knowledge graph and hierarchical bi-directional attention," in *Proceedings of the Australasian computer science week multiconference*, 2020, pp. 1–10.

[19] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[20] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[21] M. Tan, C. Dos Santos, B. Xiang, and B. Zhou, "Improved representation learning for question answer matching," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 464–473.