Enhancing Medical Question Answering with LSTM-Based Recurrent Neural Networks and Integrated Multi-Task Learning

Dedeepya Sai Gondi, Veera Venkata Raghunath Indugu, Hemanth Volikatla and Vamsi Krishna Reddy Bandaru

March 14, 2025

Enhancing Medical Question Answering with LSTM-Based Recurrent Neural Networks and Integrated Multi-Task Learning

1st Dedeepya Sai Gondi
AI/ML, Simplyturn Technologies,
Princeton, TX, USA
sai.gondi@ieee.org

2nd Veera Venkata Raghunath Indugu
Data Science and Cloud Technologies,
Mutual of Omaha,
Exton, PA, USA
veera.indugu@ieee.org

3rd Hemanth Volikatla
Independent Researcher,
Alpharetta, GA, USA
hemanthvolikatla@gmail.com

4th Vamsi Krishna Reddy Bandaru
Independent Researcher,
The Colony, TX , USA
bvkrba@simplyturn.com

***Abstract*— The purpose of this study is to examine the use of Long-Short Term Memory (LSTM)-based Recurrent Neural Networks (RNNs) for the short text conversation problem—specifically targeted to generate solutions to medical questions. User-generated questions that have been answered by professionals form the training data bases—which are sourced from various online services. WebMD, HealthTap and iCliniq are some of the online services that provide data on which models are trained and evaluated before an optimal dataset is selected. However, neural machine translation models served as the foundation for the models created for this task—with extensions that included transfer learning and multi-task learning. Every model adheres to the encoder-decoder paradigm. In this sense, an encoder creates a latent vector representation of a question, and after being trained end-to-end, it initializes the state of a decoder to generate an answer for that question. A model architecture that accepts a binary input and controls two "modes" for its decoder RNN is one theory. While the latter trains its decoder to generate replies to questions encoded with answers—the former trains it on generic medical/health related text using a "language-model" mode. Therefore, this study proposed model architecture that integrates the work of question category classification with the goal of answer production. In this case, the network uses the encoder's final state to classify the query category and provides the decoder with further input in the form of anticipated class. Finally, a unique RNN-based language system trained on general medical/health-related literature has been taught to support these suggested models during interpretation by aggregating their probabilities at every time-step to enhance the accuracy of the resulting response.**

***Index Terms*— Medical, Language-Model, Long-Short Term Memory, Recurrent Neural Networks**

I. INTRODUCTION

Many spheres have profited from the use of Deep Learning (DL) [1]–[5]—especially Deep Neural Networks (DNNs) [6]–[10]—such as object detection or image classification with Convolutional Neural Networks (CNNs), speech recognition using Recurrent Neural Networks (RNNs) and Named Entity Recognition (NER) using RNNs. This has greatly improved Artificial Intelligence (AI) capabilities in these areas. However, achieving high performance on Statistical Machine Translation (SMT) tasks with DNNs is still challenging. There has been some progress lately by applying RNNs to this task. A particular end-to-end technique called “Neural Machine Translation” (NMT) [11]—which follows an encoder-decoder scheme introduced in [12]—has shown very good results. In NMT, sequences are encoded into fixed-sized latent vector representations, which a decoder then generates new sequences from. This has outperformed state-of-the-art phrase-based translation systems and may change SMT forever. Despite these advancements, another complex issue related to SMT remains unsolved using DL—the "Short Text Conversation" (STC) problem [13]–[19]. STC involves generating an appropriate human-like response to a single—relatively short statement, post, or question. This task can be considered a sub-task of conversational modeling [20]—which is essential in developing chatbots and other interactive AI systems. Therefore, this study focuses on the STC problem within the context of health and medical Questions and Answers (Q/As)—specifically studying the generation of responses to medical queries. However, the issue at hand is the insufficient availability of answers to medical and health-related questions. While the internet hosts numerous forums where such questions can be asked and occasionally receive excellent responses—finding a relevant question with a satisfactory answer can be a difficult and time-intensive task. Posting the question yourself can be even more frustrating due to the delay in receiving an appropriate response. Automating the generation of answers to these questions, effectively creating a "digital doctor"—could resolve this issue. The problem, therefore, aligns with the STC problem, where a single suitable answer must be generated for a given medical or health-related question. However, as discussed in [21] and compared with general conversations such as Twitter[1] and Weibo[2]—which have more limited sources of Q/A data on medical and health subjects—the goal of attaining a highly effective “digital doctor” is very ambitious—given the large amount of general medical text available from sources like

[1] https://twitter.com/?lang=en

[2] https://en.wikipedia.org/wiki/Weibo

Wikipedia[3], it should be possible to create useful answers for few questions.

Therefore, we aim to develop an end-to-end model using RNNs [22]–[25] and the encoder-decoder framework that generates sensible or somewhat useful answers for a small set of health and medical related questions. For training purposes we will obtain data from sites where professionals provide medical Q/As. The strength of this approach lies in RNN's ability to handle sequential data coupled with encoder-decoder framework's strength in giving coherent and contextually appropriate responses. There are several important steps in training an end-to-end system. First collect a wide range of medical Q/A pairs from trusted sources into one dataset. This dataset will serve as the basis for training our model. Then clean up any inconsistencies or noise within this data which could negatively affect how well the model performs; these steps are necessary if high quality results are desired. Once our dataset has been prepared we can begin training our model using an encoder-decoder framework. The encoder portion transforms input questions into fixed-size latent vector representations during training while decoder generates correct answer by leveraging these latent vectors together with its own internal states about previously generated words. The answers sequence is produced word-by-word until special finish token is outputted. However, to evaluate the performance of our trained models—Bilingual Evaluation Understudy (BLEU) score and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score are used as metrics to gauge generated answer qualities—they give information about accuracy, fluency and relevance among others things—thus reflecting how well did these systems respond based on their inputs.

This study is as follows; similar papers are shown in the following section. The methods and materials are detailed in Section III. The experimental analysis is carried out in Section IV, and in Section V, we provide some conclusions and future directions for the study.

II. Related Works

The problem of STC has been solved in different ways, such as Information Retrieval (IR) approaches— [26], [27], SMT methods—[28], [29] and a more holistic end-to-end approach with an encoder-decoder framework—[30], [31]. In [26], they used an IR method to solve the STC problem. In general, IR tries to extract appropriate responses from a fixed set of post-response pairs. They rank them using a linear ranking Support Vector Machine (SVM) [32]–[35]—where the index of post-response pairs is built from data collected from Twitter. However, one major limitation is that this method relies on a fixed set of responses—meaning—it cannot generate responses not present in its dataset. Another limitation is that the IR approach requires manually designed matching features which can be difficult and time-consuming. The initial attempt at generating conversational responses using SMT involved training an open-source phrase-based decoder "Moses" [36] on 1.3 million post-response pairs collected from Twitter. In this context, each post along with its corresponding response forms a single post-response pair. The SMT approach was found to perform better than IR methods where its responses were preferred over human ones in 15% cases. Compared to generating translations from bilingual data which is what SMT does primarily—there are several challenges when trying to generate conversational response from post-response data. [29] proposed a "Neural Responding Machine" (NRM) for generating responses to posts from the Twitter-like platform Weibo—which uses an encoder-decoder framework with RNNs for both the encoder and decoder. The model was trained on about 4.5 million post-response pairs collected from Weibo. The NRM outperformed retrieval-based models as well as SMT-based models including the one used in [36]. However, the models proposed in this study for solving STC problem are based on the methods used in [29]. Therefore, in this study, we train an RNN-based language model on general text from medical and health-related Wikipedia articles and papers—then use this model to assist an answer-generating model in producing better responses. We leverage the knowledge embedded in general text to improve quality and relevance of model-generated responses. By using transfer learning—a model can take advantage of pre-trained language models that have seen a large amount of text data and hence improve the quality of its responses by making them more coherent and contextually relevant. Conversely, multi-task learning teaches a model to perform several tasks at once which often results in stronger generalization abilities and performance across different domains. By incorporating these advanced techniques, the study aims to overcome the limitations of previous methods and provide a more effective solution to the STC problem. The RNN-based language model trained on a diverse set of text data ensures that the model can generate responses that are not only relevant but also informative, particularly in the context of medical and health-related conversations.

III. Materials and Methods

The proposed and tested models were applied with the TensorFlow library. It is a powerful framework for deploying very fast Graphics Processing Unit (GPU) optimized mathematical operations through python code. In the script, TensorFlow makes a computational graph which is a set of statements that specify all required operations to perform certain computations. Then users define tensors within this framework—these are data structures operated on by TensorFlow, and they are evaluated as part of the computational graph. The availability of GPU acceleration in TensorFlow is one of its biggest advantages. This allows to process big data sets and complex neural network architectures more efficiently than it can be done using just Central Processing Unit (CPU). It's especially important for DL tasks where training time can be decreased significantly [37]–[42]. However, even if multiple GPUs can be utilized by TensorFlow to speed up calculations further—only one GPU was used in this study for all experiments. This simplifies implementation and avoids extra complexity associated with managing multiple GPUs but still provides noticeable

[3] https://www.wikipedia.org/

performance improvements—when GPU was employed training time of our models reduced three times approximately. But it should be mentioned that throughout the whole study this speedup wasn't measured accurately enough so different computations or configurations may lead to another result. To generate visualizations and illustrations we have employed matplotlib library together with Inkscape[4] tool. Matplotlib is a comprehensive plotting library for creating static, animated, and interactive visualizations in Python which was used here to produce plots included into study—similarly Inkscape was chosen as vector graphics editor similar to Adobe Photoshop because such editors enable creation detailed scalable graphics necessary for clear representation complex information or results. It's important to note that CPUs were necessary for general computational tasks as well as data pre-processing power while Titan X GPU provided capabilities required by models during their intense training phase. Such combination ensured that experiments could be performed with both efficiency and effectiveness realized simultaneously.

A. *Data Analysis*

For this study, we used four different text corpora—the Q/A dataset, the extended Q/A dataset, the PubMed[5] dataset, and the Europarl English-French dataset[6]. Each of these corpora contributed to making our work more comprehensive and robust. The Q/A dataset contain questions and answers taken from WebMD[7] Answers. People ask health-related questions on this website, and experts or other users answer them. However, in order to ensure that the quality of answers was high enough for our purposes—we filtered user responses according to some criteria such as number of answers written (more than 50), votes received (more than 10), followers (more than 5), or if other users marked their answer as useful or not. But even after such measures were taken there is no guarantee about quality because these thresholds were based mainly on intuition gathered during browsing the forum. At first there were 23,437 Q/A pairs in the data which had about 18 million characters after removing duplicates. Categories for questions were assigned using tags matched with records from "Unified Medical Language System" (UMLS) database—where symptoms and diseases are grouped into general categories. The similarity between tags and symptom/disease names was measured by calculating "Levenshtein distance[8]". To enrich Q/A dataset more pairs were extracted from eHealth[9] forum, HealthTap[10], iCliniq[11] websites as well as Question Doctors[12] site among others which altogether gave us additional 166804 Q/A pairs thus increasing total size by around seven times up to approximately 66 million characters in length. This made data much more diverse because now we had different sources represented by questions of various lengths with answers of different lengths too. It turned out that eHealth forum's questions were usually longer than those from other sources while response length stayed more or less constant across all sources. PubMed dataset includes text extracted from medical papers and health-related Wikipedia articles. We took 53541 medical papers from PubMed which were either abstracts or full texts and also included 7077 Wikipedia articles belonging to category "Health and fitness" including its subcategories. So in total it was approximately 600 million characters worth of text. This corpus provided a lot of scientifically valid content for our study. Europarl English-French dataset is a benchmarking dataset for SMT consisting of proceedings from the European parliament translated into multiple languages with corresponding English translations available as well. For our purposes we mostly used French translations here to test network's performance against known state-of-the-art benchmarks—since there are much fewer resources in this language compared to English translations—which tend to dominate many tasks like SMT evaluation etc. There are altogether 2007723 translation pairs between these two languages which gives us around 630 million characters. Since data taken from online forums tends to be quite noisy, significant preprocessing was required before using them as inputs for training models etc. Ads and spam posts were filtered out based on checking fraction of uppercase letters in a post; questions containing only links were removed. Preprocessing steps applied per QA pair were as follows:

- Convert text to lowercase.
- Remove Hyper Hypertext Markup Language (HTML)/ Extensible Markup Language (XML) tags.
- Replace links with "*<link>*".
- Convert happy/sad emoticons to "*<positive_smiley>*" / "*<negative_smiley>*".
- Manually remove specific introductory phrases like "I understand your pain...".
- Reduce multiple whitespaces to single whitespace.
- Simplify repeated punctuation marks down to single instance.

The above steps were meant to ensure the data is at its cleanest and uniform possible state—thus enhancing quality and dependability of analysis and other uses. Altogether, these four datasets combined to form an extensive text corpus with a wide range of topics which can be used for strong medical or health-related research.

B. *Model Analysis*

This section outlines the various model architectures tested in the study—all based on an "encoder-decoder" setup but with different configurations in the encoder or decoder components. Each model operates at the character level by learning character embeddings—employing a fixed alphabet instead of a large word vocabulary. The group covers 99.9% of all characters and results in an alphabet size of 53—which includes tokens for "padding", "start-of-sequence", "end-of-sequence", and "unknown-character". The character embeddings are batch-normalized before being fed into the

[4] https://inkscape.org/

[5] https://pubmed.ncbi.nlm.nih.gov/

[6] https://datarepository.wolframcloud.com/resources/Europarl-English-French-Machine-Translation-Dataset-V7

[7] https://www.webmd.com/

[8] The Levenshtein distance calculates the least amount of changes (replacements, insertions, and deletions) required to change one string into another.

[9] https://ehealth.eletsonline.com/

[10] https://www.healthtap.com/

[11] https://www.icliniq.com/

[12] https://my.clevelandclinic.org/patients/information/questions-to-ask-your-doctor

encoder. The models are—the standard LSTM cells which are used in generally both of the encoder and decoder. BLSTM—at the encoder—bidirectional LSTM cells are typically used, however, regular LSTM cells have been used in the decoder. BLSTM Attention—in addition to use of the BLSTM cells in the encoder—this model has also featured to include the attention mechanism in the decoder. BLSTM Attention C2W—hierarchical char2word encoder is utilized by employing BLSTM cells with attention in the decoder—adding an extra layer of encoding on top of the character-based RNN encoder. For each time-step corresponding to the end of a word—the hidden state is passed to the word-level encoder. Mixed Language Model (LM) —the decoder has two modes—a language model mode and an answer generation mode—with a binary mode input to switch between them. The Mixed LM model can be any of the previous models with an additional binary mode input for the decoder. Training involves a mix of Q/A dataset and PubMed dataset observations, with a parameter mix determining the ratio of Q/A observations in each batch. Validation loss is computed only on Q/A observations to ensure focus on the main task. Assisted LM—during prediction—a pretrained language model is employed to assist by merging probabilities from both the language model and answer generation model. The Assisted LM model leverages a pretrained language model to enhance predictions. At each time step, predictions from both the language model and the answer generation model are combined by multiplying their probabilities and renormalizing to sum to one. This model is architecture-independent and can utilize any of the previous configurations. Multi-task Classifier—this model simultaneously classifies question category as well as generates answers. This model classifies the question category based on the final encoder state and generates answers. It uses an additional dense layer with ReLU activation to project the encoder state to logits for category prediction. During training, a loss function combining classification and answer generation is used, weighted by a hyperparameter. For observations with multiple categories, one category is sampled each time the observation is shown to the network, acting as a regularizer. For observations without a category, category probabilities are concatenated to the decoder input, and the category prediction term in the loss function is set to zero.

IV. Experimental Analysis

Different numbers of cells (recurrent layers), state sizes, and character embedding sizes were used to train LSTM, BLSTM, and BLSTM Attention models with and without dropout on the Q/A dataset in order to determine which network architecture is best. Models were evaluated according to their BLEU score. Figs. 1 and 2 show the resulting BLEU scores; in these plots state size refers to both the dimensionality of the state as well as character embeddings. No cross-validation was performed on these results because of time constraints and lack of requirement for statistical significance in determining an optimal configuration. Based on inspection of Figs. 1 and 2—it seems that generally the BLSTM Attention model performs better — this is expected since this architecture achieves superior performance on the SMT task.

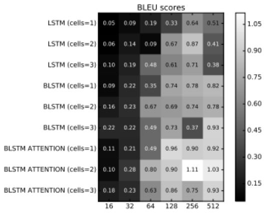

Fig. 1. BLEU ratings for several models with different cell counts and state sizes that don't have any dropouts

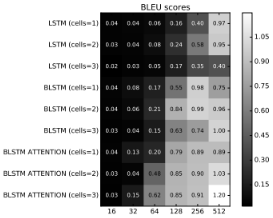

Fig. 2. BLEU ratings for the various models with 50% dropout and different cell and state sizes

Also models tend to achieve higher BLEU scores when you increase their state size (i.e., number of units). However there appears to be a sweet spot around 256 units if no dropout is applied during training. Looking at convergence plots for BLSTM Attention models—which show training and validation loss over time with/without dropout (refer to Figs. 3 and 4)—it seems like overfitting could happen because validation loss tends to go up for models with state size = 256 or 512 when no dropout is used. Notice that using dropout makes smaller-state-size models perform much worse than those without it; indeed comparing convergence plots between these models in Figs. 3 and 4 reveals that final validation loss with dropout > without it—suggesting that dropout hurts performance of less complex models—this might be due to their limited expressiveness from fewer parameters, where disabling half during training prevents them from learning underlying patterns. Convergence plots suggest most models converge well when dropout is used; however validation loss frequently shows small jagged spikes indicative of possibly too high learning rate that could be fixed by decreasing the learning rate in subsequent tests. Another general trend is that increasing state size and character embeddings dimensions speeds up model convergence.

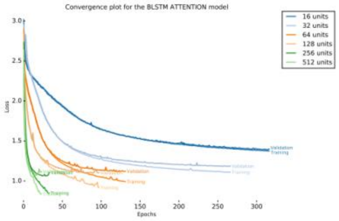

Fig. 3. BLSTM Attention model convergence without dropout utilizing various numbers of units for the network's state size and character embeddings

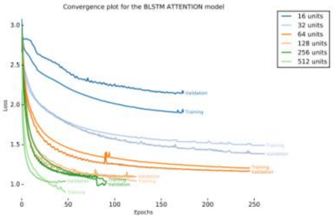

Fig. 4. BLSTM Attention model convergence with 50% dropout utilizing various units for character embeddings and network state size

While it’s true that the BLSTM Attention model with three cells, state size = 512, and dropout = 50% achieves the highest BLEU score—the optimal architecture has two cells, state size = 256, and no dropout. This configuration gives similar performance but with significantly fewer parameters. Training time is highly dependent on encoded sequence length—especially when attention is used—so the effect of maximum encoding length on performance was tested—Fig. 5 shows BLEU scores for BLSTM Attention model trained on Q/A dataset with different maximum encoding lengths. Performance increases as maximum encoding length increases but beyond 200 characters there’s no significant further improvement, indicating most information about question is contained within first 200 characters of input sequence.

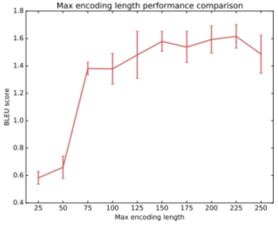

Fig. 5. 5-cross-validated BLEU scores with different maximum encoding lengths for the BLSTM Attention model

Despite the potential for better performance with longer maximum encoding lengths—the remaining runs will use a maximum encoding length of 100 characters. Similarly, the performance effect of the embedding size, i.e., the dimensionality of character embeddings, was examined. Fig. 6 shows the cross-validated performance of the BLSTM Attention model on the Q/A dataset with varying embedding sizes. The plot indicates that performance improves with dimensionality, and the optimal embedding size is found to be 256. One might expect an embedding size of 512 to perform at least as well as 256 since each character vector can hold the same amount of information, if not more. The performance decline may be due to the increased connections between embeddings and hidden layers, emphasizing the need for proper regularization, which might not have been adequately performed in this examination. The network may have overfitted the data with an embedding size of 512. To minimize the importance of proper regularization, an embedding size of 256 is used in the remaining runs.

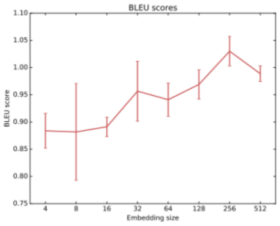

Fig. 6. BLEU scores that were cross-validated using different embedding sizes for the BLSTM Attention model (3-fold CV was employed)

A. Additional Experiments

The assessment of the suggested models—as shown in Table I—gives interesting insights about their performance when trained on complete Q/A pairs. One of the most noticeable findings is that there exists a difference between final test loss and validation loss reported through convergence plots which can be attributed to maximum likelihood training difficulties. What is interesting—however is that even though these two values do not match up it becomes obvious that BLEU scores are inversely related with final loss thereby indicating that better outputs are obtained by models that overfit more. This can be because the dataset has many similar questions and answers which may prompt the model to remember only few responses and give slight variations for all queries. Assisted LM models lower down final loss acquired by integrating predicted probabilities with those from language model but this extension does not increase BLEU score under any model. Though multiplying probabilities was the only method used to combine them different approaches could have been applied so as to reduce impacts caused by language model probabilities on performance yet none were explored in this case. Moreover if character distributions helpful for answer generating models were supported then characterizing Pubmed datasets during training rather than using Q/A data set would have provided better knowledge about such distributions Each model's performance is summarized as follows:

- LSTM: Achieved a BLEU score of 1.11 with a corresponding loss of 4.38. Category accuracy is not applicable for this model.
- LSTM (Mixed LM) and LSTM (Assisted LM): Yielded lower BLEU scores of 0.54 and 0.24 respectively, with associated losses of 3.45 and 2.77. Category accuracy is not applicable.
- BLSTM: Demonstrated a higher BLEU score of 1.76 with a loss of 4.92. Category accuracy is not applicable.
- BLSTM (Mixed LM) and BLSTM (Assisted LM): Obtained BLEU scores of 0.64 and 0.32 respectively, with losses of 4.74 and 2.88. Category accuracy is not applicable.
- BLSTM Attention: Scored a BLEU of 1.05 with a loss of 4.46. Category accuracy is not applicable.
- BLSTM Attention (Mixed LM) and BLSTM Attention (Assisted LM): Showcased BLEU scores of 1.18 and 0.30 respectively, with losses of 4.81 and 2.76. Category accuracy is not applicable.
- BLSTM Attention C2W: Achieved a BLEU score of 1.51 with a loss of 4.79. Category accuracy is not applicable.
- BLSTM Attention C2W (Mixed LM) and BLSTM Attention C2W (Assisted LM): Produced BLEU scores of 1.34 and 0.46 respectively, with losses of 4.99 and 2.91. Category accuracy is not applicable.
- Multi-task classifier: Exhibited a BLEU score of 0.47, a loss of 3.66, and a category accuracy of 0.38.
- Multi-task classifier (Mixed LM) and Multi-task classifier (Assisted LM): Their BLEU scores were 0.38 and 0.11 respectively, with losses of 4.07 and 2.49. Category accuracy was consistent at 0.36 and 0.38 respectively.

The BLEU score is questionable as a performance metric for Bilingual SMT because there are many valid answers to a question compared to source/translation pairs in SMT systems. But manual evaluation by human assessors would be resource-intensive given the number of possible translations; therefore we have used a metric like BLEU that can be computed automatically. Furthermore, recent work on multi-task learning has shown that training networks on multiple objectives could help with data scarcity issues—although this did not seem to be true here.

TABLE I
FINAL RESULTS APPLYING THE SUGGESTED MODELS ON THE Q/A DATASET

Model	BLEU score	Loss	Category accuracy
LSTM	1.11	4.38	-
LSTM (Mixed LM)	0.54	3.45	-
LSTM (Assisted LM)	0.24	2.77	-
BLSTM	1.76	4.92	-
BLSTM (Mixed LM)	0.64	4.74	-
BLSTM (Assisted LM)	0.32	2.88	-
BLSTM Attention	1.05	4.46	-
BLSTM Attention (Mixed LM)	1.18	4.81	-
BLSTM Attention (Assisted LM)	0.30	2.76	-
BLSTM Attention C2W	1.51	4.79	-
BLSTM Attention C2W (Mixed LM)	1.34	4.99	-
BLSTM Attention C2W (Assisted LM)	0.46	2.91	-
Multi-task classifier	0.47	3.66	0.38
Multi-task classifier (Mixed LM)	0.38	4.07	0.36
Multi-task classifier (Assisted LM)	0.11	2.49	0.38

V. CONCLUSION AND FUTURE WORKS

The study implies that end-to-end networks training are extremely difficult in generating medical question answers using the encoder-decoder framework which is similar to SMT. It is possible for a bidirectional RNNs to learn some fixed template responses to certain type of questions, and also learn somehow responses to specific types of questions that require more information from the user. In addition, many of the generated answers are spelled correctly and understandable ones; hence it shows that the model has learned some syntax and semantics understanding. However these answers were manually chosen from test samples, with most cases not being useful as answers to the question asked by users but loops repeating themselves multiple times were often encountered during modelling process. The approach did not yield better results than any other method used before despite its effectiveness in various translation works. There was no significant improvement observed when a network was trained as part language model and answer generator or even predicting sequences at once by splitting them into two separate models each working on different task since BLEU score is not suitable for evaluating performance of STC problem where many valid answers are likely to exist for each given query compared with bilingual translations. No improvements were found on final performances after training a few epochs while part of it was done specifically like language model thus acting both roles simultaneously; this only helped speed up convergence slightly during initial stages but slowed down considerably afterwards per epoch due extra data involved—decoder dealing with switching between modes which include generating mode. Also when trying predict classifications along with producing outputs jointly from inputs so far constructed an architecture that is capable doing these two tasks concurrently—although accuracy declined bit classification tasks if network had to also predict sequences may be because there aren't enough number parameters within current setup though this issue shall be looked into later on.

VI. DECLARATIONS

A. ***Funding:*** No funds, grants, or other support was received.

B. ***Conflict of Interest:*** The authors declare that they have no known competing for financial interests

C. ***Data Availability:*** Data will be made on reasonable request.

D. Code Availability: Code will be made on reasonable request.

REFERENCES

[1] V. Kanaparthi, “Transformational application of Artificial Intelligence and Machine learning in Financial Technologies and Financial services: A bibliometric review,” Jan. 2024, doi: 10.1016/j.jbusres.2020.10.012.

[2] V. Kanaparthi, “Exploring the Impact of Blockchain, AI, and ML on Financial Accounting Efficiency and Transformation,” Jan. 2024, Accessed: Feb. 04, 2024. [Online]. Available: https://arxiv.org/abs/2401.15715v1

[3] V. Kanaparthi, “Credit Risk Prediction using Ensemble Machine Learning Algorithms,” in *6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings*, 2023, pp. 41–47. doi: 10.1109/ICICT57646.2023.10134486.

[4] V. Kanaparthi, “Examining Natural Language Processing Techniques in the Education and Healthcare Fields,” *International Journal of Engineering and Advanced Technology*, vol. 12, no. 2, pp. 8–18, Dec. 2022, doi: 10.35940/ijeat.b3861.1212222.

[5] V. Kanaparthi, “Evaluating Financial Risk in the Transition from EONIA to ESTER: A TimeGAN Approach with Enhanced VaR Estimations,” Jan. 2024, doi: 10.21203/RS.3.RS-3906541/V1.

[6] Celada-Bernal, S., Pérez-Acosta, G., Travieso-González, C., Blanco-López, J., & Santana-Cabrera, L. "Applying Neural Networks to Recover Values of Monitoring Parameters for COVID-19 Patients in the ICU." DOI: 10.3390/math11153332.

[7] Yu, Y., Lin, H., Meng, J., Wei, X., & Zhao, Z. "Assembling Deep Neural Networks for Medical Compound Figure Detection." DOI: 10.3390/info8020048.

[8] Sha, A., Krishna, E. R. A., Inture, A. R., Menon, D. S., Joseph, J., & A. T. "Deep Reinforcement Learning for Brain Aneurysm Segmentation in 3D TOF MRA Images: A Comparative Study using 3D U-Net, 3D ResNet, and LSTM Networks." DOI: 10.1109/ICACRS58579.2023.10404944.

[9] Nanthini, K., Sivabalaselvamani, D., Selvakarthi, D., Pavethran, D., Srinaath, N., & Vignesh, K. S. "Performance of Recurrent Neural Networks in Liver Disease Classification." DOI: 10.1109/ICEARS56392.2023.10085202.

[10] Kessler, S., Schroeder, D., Korlakov, S., Hettlich, V., Kalkhoff, S., Moazemi, S., Lichtenberg, A., Schmid, F., & Aubin, H. "Predicting readmission to the cardiovascular intensive care unit using recurrent neural networks." DOI: 10.1177/20552076221149529.

[11] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2015. Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/1409.0473v7

[12] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems*, Sep. 2014, vol. 4, no. January, pp. 3104–3112. Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/1409.3215v3

[13] G. S. Kashyap, A. E. I. Brownlee, O. C. Phukan, K. Malik, and S. Wazir, “Roulette-Wheel Selection-Based PSO Algorithm for Solving the Vehicle Routing Problem with Time Windows,” Jun. 2023, Accessed: Jul. 04, 2023. [Online]. Available: https://arxiv.org/abs/2306.02308v1

[14] S. Naz and G. S. Kashyap, “Enhancing the predictive capability of a mathematical model for pseudomonas aeruginosa through artificial neural networks,” *International Journal of Information Technology 2024*, pp. 1–10, Feb. 2024, doi: 10.1007/S41870-023-01721-W.

[15] G. S. Kashyap, A. Siddiqui, R. Siddiqui, K. Malik, S. Wazir, and A. E. I. Brownlee, “Prediction of Suicidal Risk Using Machine

Learning Models." Dec. 25, 2021. Accessed: Feb. 04, 2024. [Online]. Available: https://papers.ssrn.com/abstract=4709789

[16] G. S. Kashyap *et al.*, "Detection of a facemask in real-time using deep learning methods: Prevention of Covid 19," Jan. 2024, Accessed: Feb. 04, 2024. [Online]. Available: https://arxiv.org/abs/2401.15675v1

[17] G. S. Kashyap, D. Mahajan, O. C. Phukan, A. Kumar, A. E. I. Brownlee, and J. Gao, "From Simulations to Reality: Enhancing Multi-Robot Exploration for Urban Search and Rescue," Nov. 2023, Accessed: Dec. 03, 2023. [Online]. Available: https://arxiv.org/abs/2311.16958v1

[18] H. Habib, G. S. Kashyap, N. Tabassum, and T. Nafis, "Stock Price Prediction Using Artificial Intelligence Based on LSTM– Deep Learning Model," in *Artificial Intelligence & Blockchain in Cyber Physical Systems: Technologies & Applications*, CRC Press, 2023, pp. 93–99. doi: 10.1201/9781003190301-6.

[19] M. Kanojia, P. Kamani, G. S. Kashyap, S. Naz, S. Wazir, and A. Chauhan, "Alternative Agriculture Land-Use Transformation Pathways by Partial-Equilibrium Agricultural Sector Model: A Mathematical Approach," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.11632v1

[20] O. Vinyals and Q. Le, "A Neural Conversational Model," Jun. 2015, Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/1506.05869v3

[21] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2011, pp. 583–593. doi: 10.5555/2145432.

[22] V. Kanaparthi, "Robustness Evaluation of LSTM-based Deep Learning Models for Bitcoin Price Prediction in the Presence of Random Disturbances," Jan. 2024, doi: 10.21203/RS.3.RS-3906529/V1.

[23] V. Kanaparthi, "AI-based Personalization and Trust in Digital Finance," Jan. 2024, Accessed: Feb. 04, 2024. [Online]. Available: https://arxiv.org/abs/2401.15700v1

[24] V. K. Kanaparthi, "Navigating Uncertainty: Enhancing Markowitz Asset Allocation Strategies through Out-of-Sample Analysis," Dec. 2023, doi: 10.20944/PREPRINTS202312.0427.V1.

[25] V. K. Kanaparthi, "Examining the Plausible Applications of Artificial Intelligence & Machine Learning in Accounts Payable Improvement," *FinTech*, vol. 2, no. 3, pp. 461–474, Jul. 2023, doi: 10.3390/fintech2030026.

[26] Z. Ji, Z. Lu, and H. Li, "An Information Retrieval Approach to Short Text Conversation," Aug. 2014, Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/1408.6988v1

[27] V. Blinov, K. Mishchenko, V. Bolotova, and P. Braslavski, "A pinch of humor for short-text conversation: An information retrieval approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10456 LNCS, pp. 3–15. doi: 10.1007/978-3-319-65813-1_1.

[28] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing Statistical Machine Translation for Text Simplification," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, Dec. 2016, doi: 10.1162/tacl_a_00107.

[29] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-Text conversation," in *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, Mar. 2015, vol. 1, pp. 1577–1586. doi: 10.3115/v1/p15-1152.

[30] M. Zou, X. Li, H. Liu, and Z.-H. Deng, "MEMD: A Diversity-Promoting Learning Framework for Short-Text Conversation." pp. 1281–1291, 2018. Accessed: May 28, 2024. [Online]. Available: https://aclanthology.org/C18-1109

[31] G. Pandey, D. Contractor, V. Kumar, and S. Joshi, "Exemplar encoder-decoder for neural conversation generation," in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, vol. 1, pp. 1329–1338. doi: 10.18653/v1/p18-1123.

[32] Liu, L., Sun, X., Li, C., & Lei, Y. "Classification of Medical Text Data Using Convolutional Neural Network-Support Vector Machine Method." DOI: 10.1166/jmihi.2020.3042.

[33] Ramachandran, R., & Arutchelvan, K. "Optimized Version of Tree based Support Vector Machine for Named Entity Recognition in Medical Literature." DOI: 10.1109/ICISS49785.2020.9316051.

[34] Jamaluddin, M., & Wibawa, A. "Patient Diagnosis Classification based on Electronic Medical Record using Text Mining and Support Vector Machine." DOI: 10.1109/iSemantic52711.2021.9573178.

[35] Marafino, B. J., Davies, J., Bardach, N. S., Dean, M. L., & Dudley, R. "N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit." DOI: 10.1136/amiajnl-2014-002694.

[36] H. Hoang and P. Koehn, "Design of the moses decoder for statistical machine translation," in *ACL-08: HLT - Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 2008, pp. 58–65. doi: 10.3115/1622110.1622120.

[37] S. Wazir, G. S. Kashyap, and P. Saxena, "MLOps: A Review," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.10908v1

[38] S. Wazir, G. S. Kashyap, K. Malik, and A. E. I. Brownlee, "Predicting the Infection Level of COVID-19 Virus Using Normal Distribution-Based Approximation Model and PSO," Springer, Cham, 2023, pp. 75–91. doi: 10.1007/978-3-031-33183-1_5.

[39] G. S. Kashyap *et al.*, "Revolutionizing Agriculture: A Comprehensive Review of Artificial Intelligence Techniques in Farming," Feb. 2024, doi: 10.21203/RS.3.RS-3984385/V1.

[40] N. Marwah, V. K. Singh, G. S. Kashyap, and S. Wazir, "An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning," *International Journal of Information Technology (Singapore)*, vol. 15, no. 4, pp. 2317–2327, May 2023, doi: 10.1007/s41870-023-01264-0.

[41] P. Kaur, G. S. Kashyap, A. Kumar, M. T. Nafis, S. Kumar, and V. Shokeen, "From Text to Transformation: A Comprehensive Review of Large Language Models' Versatility," Feb. 2024, Accessed: Mar. 21, 2024. [Online]. Available: https://arxiv.org/abs/2402.16142v1

[42] G. S. Kashyap, K. Malik, S. Wazir, and R. Khan, "Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing," *Multimedia Tools and Applications*, vol. 81, no. 25, pp. 36685–36698, Oct. 2022, doi: 10.1007/s11042-021-11558-9.