Employing RNN Encoder-Decoder for Text Generation in Deep Learning for Q&A

Bhimsen Moharana, Bivas Ranjan Parida
College of Engineering Bhubaneswar, Biju pattnaik University of Technology, Odisha, India

ABSTRACT: Conversational modeling is an important task in natural language understanding and machine intelligence. Deep Neural Networks (DNNs) are powerful model that achieve excellent performance on difficult learning tasks. Although DNNs work well with availability of large labeled training set, it cannot be used to map complex structures like sentences end-to-end. Existing approaches for conversational modeling are domain specific and require handcrafted rules. In this paper, we propose a simple neural network based approach based on recent proposed sequence to sequence framework. Our model generates reply by predicting sentence using chained probability for given sentence(s) in conversation. This model is trained end-to-end on large dataset. Primary findings show that model shows common sense reasoning on movie transcript dataset. We use Attention to focus text generation on intent of conversation as well as beam search to generate optimum output with some diversity.

KEYWORDS: NLP, Deep Learning, Language Modeling, Text Generation

I. INTRODUCTION

An intelligent system with ability to comprehend and generate context aware responses can provide solution to complex problem like customer care. Natural Language Processing and Deep Learning plays important role in building such system which show intelligence. Deep Neural Networks (DNNs) are powerful models which have already achieved good performance for different machine learning tasks like speech recognition [1], image captioning [2], image classification [3, 4], text classification [5, 6, 7, 8], etc. Convolution Neural Network (CNNs) are very powerful at handling multidimensional data such as image, video, audio where input and output dimensions are known but they perform so well when applied to text where input and output dimensions are unknown prior to execution. Recurrent Neural Networks (RNNs) is another type of neural network which shows ability to map complex structures. This ability of RNNs can be used for Language Modeling tasks like machine translation, text generation, conversation modeling.

Advances in field of machine learning with deep learning and neural networks have led to remarkable progress towards solutions for many complex learning tasks such as speech recognition, computer vision, and language processing. Recent advances in deep learning shows that neural networks are much more capable than just classification and can be used for mapping complex structures such as sentences. A good example of this ability is the sequence-to-sequence framework [9] used for machine translation. This feature of sequence-to-sequence framework to map complex structures end to end allows us to tackle difficult tasks where domain knowledge is not available and/or it's very difficult to model rules manually [10].

Sequence-To-Sequence [9] model shows that RNNs with memory cells like LSTM [11] or GRU [12] can successfully map source language sentence to target language and can be trained in end-to-end fashion achieving good results for tasks like machine translation, sentence memorization. This approach can be extended for purpose of conversation modeling [10], by making task of conversation modeling into task of translating questions into answers. This baseline approach can successfully map dialogues and responses for simple tasks bus fails at complex tasks and lengthy sequences. Sequence-To-Sequence model can be further extended by using neural attention [13, 2, 14, 15] to keep text generation focused on the intent of the conversation as well as using beam search [16, 17] to select one of many possible sequences allowing model to be more diverse and human like at text generation task.

This approach uses vocabulary and word-embedding provided by word-to-vector model. But use of this dictionary approach limits the learning capacity of model to max word size defined by vocabulary size hyper parameter. This can be overcome with use of character level generative model instead of word level by using CNN softmax or character LSTM in combination with word level LSTM.

II. LITERATURE REVIEW

Artificial Intelligence (AI) have been a hot topic for past years and lot of research has been done in fields like machine translation, computer vision, pattern recognition, etc. Some of the work related to NLP, Deep Learning and Neural Networks is noted in this section.

Cho, van Merrienboer, Gulcehre, et al. [12]proposed new kind of neural network based on encoder and decoder for machine translation and new type of cell called Gated Recurrent Unit (GRU). The neural network proposed in article encodes source language sentence in fix length vector using encoder and decoder decodes this fix-length vector into variable length target language output sequence. The model proposed leads to improved BLEU score for statistical machine translation. Yao, Zweig, and Peng [13] proposed improvement over traditional encoder-decoder RNN models for conversation modeling by adding an attention network. Proposed model consists three RNNs, encoder encodes the input sentences and decoder that generates responses then there is newly added attention network that models intention of the conversation over time.

Sutskever, Vinyals, and Le [9] illustrated ability of multilayer LSTM RNN to achieve good performance on Machine Translation tasks. Article also shows that reversing the input sequences yields in better representations of the dependencies in input sequence and expected output sequence as well as better mapping of source symbols to target symbols. Proposed approach produces good BLEU scores by itself and state-of-art results when coupled with other baseline systems. Sukhbaatar, Weston, Fergus, et al.[18] proposes a recurrent memory-based model with multihops and trains the same with standard gradient descent. Author then evaluates the model for question-answer task. Model attends to sequences in timely fashion by considering next relevant piece of information at each time step. Though the model outperforms the baseline unsupervised approaches, it is far inferior than supervised approach.

Bahdanau, Cho, and Bengio[19] proposed a novel ”attention” mechanism for improvements in standard sequence-tosequence models. Since not all information can be encoded in a single vector, author proposes an approach to overcomes this by introducing an attention vector based on weighted sum of the input hidden states. Then the attention weights are learned along with rest of the weights and biases in the network. Approach proposed in article enables model to focus on more important part of input sequence to generate output sequence. Luong, Pham, and Manning[15] evaluates effect of various attention mechanisms for task of Machine Translation. The author proposes “global” and “local” attention models where attending over all source words and subset of source words respectively. Ioffe and Szegedy[20] proposed a technique to normalize unit activation and unit variance within network. Author shows that Batch Normalization leads to faster training and better accuracy for convolutional networks. It also reduces the need of dropout.

Rush, Chopra, and Weston[14] Extends sequence-to-sequence model for task of abstractive sentences summarization. Neural attention is added for soft alignment. Vinyals and Le[10] applies sequence-to-sequence model for modeling conversations instead of Machine translation like in base paper. The proposed approach exploits the ability of RNNs to map complex structures for purpose of modeling conversations. Author then trains model on IT-Helpdesk and OpenSubtitles dataset.

Xu, Ba, Kiros, Cho, Courville, Salakhudinov, Zemel, and Bengio[2] tries to improve image captioning by allowing decoder to focus on specific part of image than entire image and finds correspondence between words and image patches. The RNN uses underlying CNN outputs as input to map objects in image patch with captions in knowledge base. Zhang, Zhao, and LeCun[8] evaluated deep Convolutional Neural Network (CNN) on large-scale text classification using one-hot encoding to achieve competitive performance.

Chung, Cho, and Bengio[21] illustrates LSTM model, the model unlike baseline LSTM model models makes use of per word character-level CNN outputs and highway layer. Since word embedding are completely avoided the resulting model has significantly fewer parameters while achieving better performance. Kim, Jernite, Sontag, and Rush[22] evaluates use of character-level decoder in Natural Machine Translation, also proposes a bi-scale architecture with slow and fast layers in decoder. Both biscale and base character-level decoder models perform better than word-level models at Machine Translation. Lee, Cho, and Hofmann[23] proposed a character-level Neural Machine Translation model. Unlike other RNN based encoder decoder models for NMT, this model uses CNN with max-pooling for encoder while using highway layer to reduce size of source representation. Standard RNN is used as decoder.

Ghosh, Vinyals, Strope, et al.[24]proposes a Contextual LSTM (CLSTM) model which makes use of both the input word and context vector to predict next word. This model performs better at selecting next sentence and predicting next topic than the baseline models for same tasks. Chung, Ahn, and Bengio[25] proposed a new hierarchical RNN which learns both temporal and hierarchical representations without prior knowledge of structure or timescale of hierarchy. To achieve this binary boundary detectors are used at each layer which control propagation of information between neighboring layers.

III .MODEL

Figure 1 illustrates high level diagram of the model in proposed work. It uses word embedding to represent relations between words and this is used to compute possible candidate word while generating output.

Proposed approach makes use of sequence-to-sequence framework [9]. The model is based on Recurrent Neural Network (RNN) which reads input at one token at a time while generating output at one token at time. To speedup training and obtaining better accuracy the true output sequence is fed to decoder for training and learning happens by back propagation. The model is trained to maximize cross-entropy of correct sentence. During inference greedy approach is used, where instead of true output sequence the output generated in previous step is fed to decoder as next token. Less greedy approach in form of Beam-Search is used to provide better output by taking into consideration multiple output paths instead of going with local maximum at each step like in greedy approach of vanilla decoder.

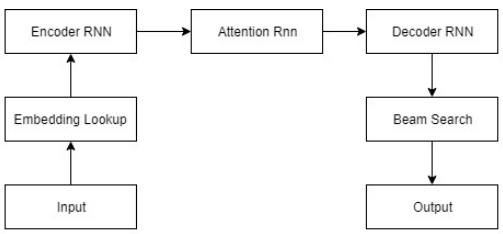

Figure 1: High level diagram of proposed model.

The Seq2Seq framework relies on the encoder-decoder paradigm. The encoder encodes the input sequence, while the decoder produces the target sequence. For example, consider dialogue pair is "ABC", "WXYZ". Then neural network can be used to map "ABC" to "WXYZ" as shown in figure 2.

A. Encoder

Each word from input sequence is associated to a vector$w \in R^d$. Then, we simply run LSTM over this sequence of vectors and store last hidden state of LSTM which will be encoder representation of input vector.

B. Decoder

Now that we have encoded vector representation of input sequence i.e. e, we'll use it to generate target sequence word by word. Last hidden state of encoder e along with start of sentence indicator are given as input to decoder.

Decoder LSTM computes next hidden state$h_0 \in R^h$. Then we apply some function g so that $s_0 := g(h_0) \in R^V$is vector of same size as vocabulary. Then, apply Softmax to normalize it into vector of probabilities$p_0 \in R_V$. Each entry in p measures likelihood of each word in vocabulary being output token.

$$h_t = \text{LSTM}(h_{t-1}, w_{lh}) \quad (1)$$

$$s_t = g(h_t) \quad (2)$$

$$p_t = Softmax(s_t) \quad (3)$$

$$i_t = argmax(p_t) \quad (4)$$

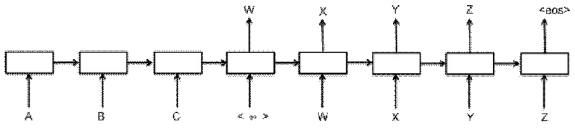

Figure 2: Example, use of seq2seq framework for conversational modelling.

Decoding stops when end of statement token is generated.

C. Decoder with Beam-Search

Greedy decoder suffers from local maxima problem, hence, giving less accurate answers. There is a better way of performing decoding, called Beam Search. Instead of only predicting the token with the best score, we keep track of k hypotheses (k is beam width). At each new time step we have V new possible tokens, resulting in k*V new hypothesis. We keep k best ones and repeat.

IV. EXPERIMENTS

A. Data Set

The experiments are performed on Amazon product data [26]. The dataset contains questions regarding different product along with one or more answers given as response and relevancy score of answer. This data is converted into question and answer pairs for further use. All sentences are shuffled, duplicates removed, 40k words with highest frequency are chosen while all remaining are replaced with <UNK>token indicating that the word is out of vocabulary.

B. Model Setup

Perplexity is used as measure for analyzing model performance, which is the average per-word log-probability on holdout dataset$e^{\frac{-1}{N}\sum_i \ln p_{w_i}}$. We compute perplexity by summing over all the words including the end of sentence token.

Cornell movie transcript dataset with vocabulary size of 40000 without any pre-processing and cleaning. The sentences are shuffled and are given as input to model, start and end of statement is indicated with <SOS>and <EOS>tokens. For training maximum word length is set to 50.

C. Training Procedure

Using the predicted token as input to then next step during training increases errors as errors would accumulate over time-steps taken to generate output. This makes training slow if not impossible. To speed-up training as well as increase accuracy of model trained the actual output sequence is fed to the decoder LSTM while training while using the generated token for next step of decoding for inference.

The model is trained till convergence with ADAM optimizer using learning rate of 0.001 with learning rate decay of 0.999 and decay step size of 1000. Batch size of 32 is used while using 2 bidirectional LSTM layers with residual connections while using network size of 512. For decoding beam width is set to 30.

V. RESULTS

The proposed model is trained on aforementioned datasets and is evaluated using test perplexity and training loss. During training process the learning of model can be validated through performance metrics like loss and perplexity. Performance metrics for proposed work are illustrated in figure 3 and 4.

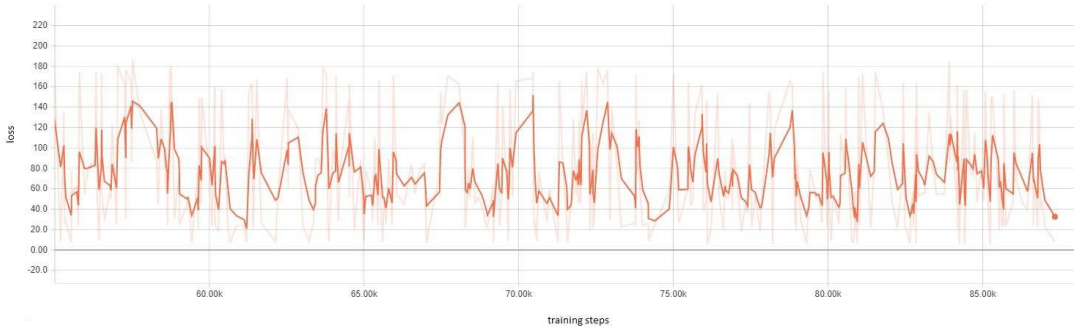

Figure 3: Training loss

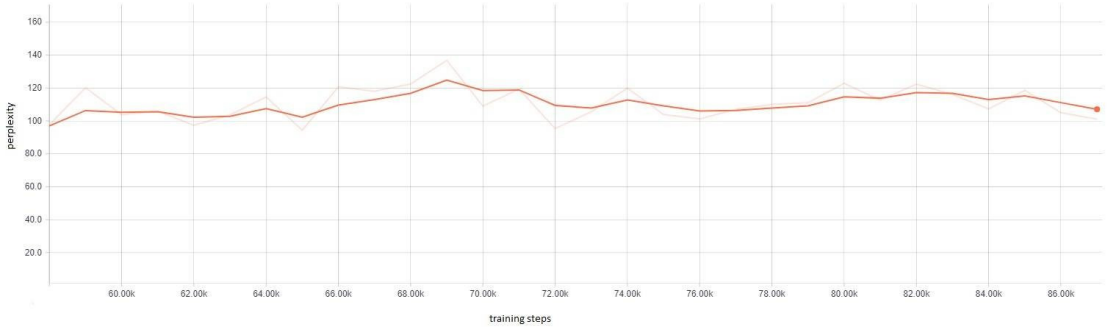

Figure 4: Test perplexity

Perplexity on test dataset after 90k steps of training is between 90-115 and BLEU score is between 0.7 to 0.9 at same training steps. Though, the perplexity score is little higher than the baseline models for translation proposed model is able to generate acceptable responses at with same checkpoint.

VI. CONCLUSION

Proposed work uses ability of Recurrent Neural Networks to build model for conversation modeling by mapping inputs to expected outputs while training. This model is able to understand the relation between words and is able to generate meaningful sentences in response to inputs. Since there are no rules, in some cases the model does not map given input correctly and generates ambiguous response or out of context response.

REFERENCES

[1] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks", in Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, IEEE, 2013, pp. 6645–6649.

[2] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention", in International Conference on Machine Learning, 2015, pp. 2048–2057.

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in Advances in neural information processing systems, 2012, pp. 1097–1105.

[5] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification.", in AAAI, vol. 333, 2015, pp. 2267–2273.

[6] M. E. Ruiz and P. Srinivasan, "Hierarchical text categorization using neural networks", Information Retrieval, vol. 5, no. 1, pp. 87–118, 2002.

[7] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial Training Methods for Semi-Supervised Text Classification", ArXiv e-prints, May 2016. arXiv: 1605.07725[stat.ML].

[8] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification", C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 649–657, 2015. [Online]. Available: http://papers.nips.

cc/paper/5782-character-level-convolutional- networks-for-text-classification.pdf.
[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", in Advances in neural information processing systems, 2014, pp. 3104–3112.
[10] O. Vinyals and Q. Le, "A Neural Conversational Model", ArXiv e-prints, Jun. 2015. arXiv: 1506.05869[cs.CL].
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[12] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", ArXiv e-prints, Jun. 2014. arXiv: 1406.1078[cs.CL].
[13] K. Yao, G. Zweig, and B. Peng, "Attention with Intention for a Neural Network Conversation Model", ArXiv e-prints, Oct. 2015. arXiv: 1510.08565.
[14] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization", ArXiv e-prints, Sep. 2015. arXiv: 1509.00685[cs.CL].
[15] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation", CoRR, vol. abs/1508.04025, 2015.
[16] S. Wiseman and A. M. Rush, "Sequence-to-Sequence Learning as Beam-Search Optimization", ArXiv e-prints, Jun.2016. arXiv: 1606.02960[cs.CL].
[17] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra, "Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models", ArXiv e-prints, Oct. 2016. arXiv: 1610.02424[cs.AI].
[18] S. Sukhbaatar, J. Weston, R. Fergus, et al., "End-to-end memory networks", in Advances in neural information processing systems, 2015, pp.2440–2448.
[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", ArXiv e-prints, Sep. 2014. arXiv: 1409.0473[cs.CL].
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", pp. 448–456, 2015.
[21] J. Chung, K. Cho, and Y. Bengio, "A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation", ArXiv e- prints, Mar. 2016. arXiv: 1603.06147[cs.CL].
[22] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models.", in AAAI, 2016, pp. 2741– 2749.
[23] J. Lee, K. Cho, and T. Hofmann, "Fully Character-Level Neural Machine Translation without Explicit Segmentation", ArXiv e-prints, Oct. 2016. arXiv: 1610.03017[cs.CL].
[24] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual LSTM (CLSTM) models for Large scale NLP tasks", ArXiv e- prints, Feb. 2016. arXiv: 1602.06291[cs.CL].
[25] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical Multiscale Recurrent Neural Networks", ArXiv e-prints, Sep. 2016. arXiv: 1609.01704[cs.LG].
[26] J. McAuley, Amazon product data, http://jmcauley.ucsd.edu/data/amazon/, [Online; accessed 30-Nov-2017], 2017.