A Neural Conversational Model for Automatic Generation of Conversations

Vibha Badhe[1], Manasi U. Kulkarni[2]

[1]*Department of Computer Science and Engineering, VJTI, University of Mumbai, India.*
[2]*Assistant Professor, Department of Computer Science and Engineering, VJTI, University of Mumbai, India.*

---***---

Abstract - *The conversations between humans and machines is regarded as one of the most hardcore problems in computer technology, which involves interdisciplinary techniques in information retrieval, ML, natural language Understanding and artificial intelligence. Interactive Entity which automatically generate conversations to exchange information smoothly between a people with little knowledge of the computer. Challenges lie in how to respond so as to maintain a relevant and continuous conversation with humans. Conversational modeling is an important task in natural language understanding and machine learning. This research applies generative models based method for conversation generation. This research is for development of conversational agent, which generates conversations using recurrent neural network and its coupled memory unit. Therefore, development of AI which conducts conversations mainly by using machine learning model for Conversations, Machine Translation and Questions/Responses.*

Key Words: **Neural conversational model, seq2seq, LSTM, RNN, Conversational model, generative model.**

1.INTRODUCTION

Humans don't start their thinking from scratch every time. As you read any topic or book, you understand each word based on your understanding of previous words. You don't just throw everything away and then start thinking from scratch again without context. Your thoughts have persistence in them. Traditional neural networks can't learn from previous conversations, and it seems like a major shortcoming of them. For example, imagine you want to classify what kind of event is happening at some point in a movie. It's not known until yet that how these networks could use its reasoning about previous events to inform later ones. Recurrent neural networks address this issue. Recurrent neural networks are networks with loops in them, allowing information to persist in events.

Applications in complex function modelling and classifications are attracted by neural network since Neural Networks have very different computing approaches from traditional computing machines. The Neural Network System has an ability to construct the rules of input-output mapping by itself. The user of the system doesn't need to know the internal architecture and instructions of the system, or the functional rules like traditional neural systems. Neural Network system requires learning for that it requires the feed-ins of input-output patterns to learn before the system can function correctly.

Recent research and work in training of neural network have led to noticeable progress in many domains like speech recognition, computer vision, natural language processing and many more. Research work in the area of neural network suggests that neural networks can do more than classification and natural language processing, they can be used to map complicated structures to other complicated structures. Artificial Conversational Agent is a computer program which conducts a conversation with the user via textual methods. Machine Learning techniques can be used for both retrieval-based and generative models, but research seems to be moving into the generative direction. Machine Learning architectures like Sequence to Sequence are uniquely suited for generating text.

2.RELATED WORK

Training of neural networks have led to an advancement in many domains such as speech recognition, image processing, and natural language processing. In the past, methods for constructing conversational model architectures have relied on hand-written rules and templates or simple statistical methods. With the rise of machine learning and artificial intelligence these models were quickly replaced by end-to-end trainable neural networks around 2015. Modelling conversation is an important task in natural language processing and artificial intelligence (AI). Since the evolution of AI, creating a good conversational model remains one of the hardest challenges in the field. The conversational models can be used for various tasks, in general conversational models have to understand user's utterances and context of problem, and then provide responses that are relevant to the problem at hand. Recently neural networks can do more than just mere classification; they can be used to map complicated structures to other complicated structures. Artificial Conversational Agent is a computer program which conducts a conversation with the user via textual methods. Conversational model uses machine learning to effectively relate human conversation and gives response to written prompts to deliver a service. ML techniques can be used for both retrieval-based models and generative based models, but research now seems to be moving into the generative direction in artificial intelligence.

1.1 Traditional Conversational Models

Traditionally, most Conversational models that we have seen have used hand-crafted rules for conversation. But, making all these rules can be quite tedious. Also, this also made making open domain conversational models very difficult. Neural Networks still existed but, they were mostly overlooked by the Conversational models. The main reason for this being that the Neural Networks require input and output to be of fixed dimension. But, in a conversation model, we deal with variable input and output length, and hence its dimensionality cannot be known a-priori. But, in recent years we have seen many workarounds for this.

1.2 Neural network in Conversational Models

Almost all the work in Neural Conversation Modelling has been done in the last few years. It is largely based on the work of Sutskever, et al. [13] which uses neural networks to map sequences to sequences. This framework was first used for neural machine translation and archival. As RNN, by itself, suffers from vanishing gradients, a variant of Long Short Term Memory (LSTM) RNN based on the works of Hochreiter et al [5]. The works of Sordoni et al. [6] and Shang et al. [7] also used RNN to model dialogue in short conversations. Our approach is based on producing answers given by a probabilistic approach to maximize the correctness of answer in the given context.

Generation of a context-sensitive response to a sequence of comments by incorporating background knowledge. The proposed model is based on the assumption that each participant in a conversation phrases their responses by taking into consideration both the past dialog utterances and their individual knowledge background. We train the model on a set of M sequences of Cornell movie dialogue corpus that are related to the main discussed topic of a conversation [12]. Training models takes input sequence of vector representations of words from a sequence of comments and a group of sentences that is aligned with this sequence of utterances.

3.MODEL

This approach suggested in the Neural Conversation Model by Vinyals, et al[3]. It is based on the Seq2Seq[4] framework described below diagram. It uses two LSTMs one for encoding and one for decoding. To preserve context, the input sequence is the concatenation of what has been conversed so far, and the output sequence is the reply. For example, there is a conversation between 2 persons and the one person utters ABC, and the other person replies WXYZ. This segment of conversation is trained with to produce a map from ABC to WXYZ. The encoder reads the input ABC until it gets the end of statement flag, in reverse order to generate a vector embedding. This becomes the input of the decoder to train the first word W and then, W is again put back in the decoder along with the input ABC to generate next word.

The dialog encoder and response decoder form together a sequence-to-sequence (SEQ2SEQ model, which has been successfully used in building end-to-end conversational systems. Both encoder and decoder are recurrent neural network (RNN) models: an RNN that encodes a variable-length input string into a fixed-length vector representation and an RNN that decodes the vector representation into a variable-length output string. Encoder-decoder framework t of model is almost identical to prior conversational SEQ2SEQ models, except that we use Multi RNN and LSTM cells. In seq2seq models sometimes encoders and decoders share weights in tasks, but do not do so in the proposed model, and also they do not share word embedding's. [9]

3.1 LSTM Layer

Long Short Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. LSTM cells can be used in the sequence to sequence model to remember the previous words in the sentence. LSTM processes one word at a time and computes probabilities of the possible values for the next word in the sentence. Instead of having a single neural network layer, LSTM has four interacting in a very special way.

3.1.1 Forget gate

The first step in LSTM is to decide what information is going to be thrown away from the cell state. This decision is made by a sigmoid layer called the `forget gate layer'. It looks at ht-1 and x_t, and outputs a number between 0 and 1 for each number in the cell state Ct-1. A 1 represents completely keep the information while a 0 represents completely get rid of the information.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Where ft is the forget gate vector, Wf is the weight matrix, ht-1 is the previous hidden state, xt is the input vector and bf is the bias.

3.1.2 Cell state

First, a sigmoid layer called the `input gate layer' decides which values to update. Next, a tanh layer creates a vector of new candidate values, ~ Ct that could be added to the state. In the next step,these two are combined to create update to the state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Where it is the input vector, Wi is the weight matrix, ht-1 is the previous hidden state, xt is the input word embedding vector and bi is the bias.

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c$$

Where ~ Ct is the temporary cell state, Wc is the weight matrix, ht-1 is the previous hidden state, xt is theinput word embedding vector and bc is the bias.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Where Ct is the cell state, ft is the forget gate vector computed above, Ct-1 is the previous cell state, it is the input vector and ~ Ct is the temporary cell state.

3.1.3 Output state and Hidden state

The output will be based on our cell state, but will be a ltered version. First, we run a sigmoid layer which decides what parts of the cell state we are going to output. Then, we put the cell state through tanh (to push the values to be between -1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Where o_t is the output vector, Wo is the weight matrix, ht-1 is the previous hidden
state, x_t is the input word embedding vector and bo is the bias.

$$h_t = o_t * tanh(C_t)$$

Where h_t is the current hidden state, o_t is the output state, and Ct is the current cell state. [22]

3.2 The Seq2Seq Model with Bucketing

The sequence to sequence model (seq2seq) was first introduced by [Cho et al.,2014], but they only used it to re-rank sentences instead of generating completely new ones, which was first done by [Sutskever et al.,2014]. The conversational model is based on the translate model on the Tensor Flow repository, with some modification to make it work for a conversation model. It's a sequence to sequence model with attention decoder. An utterance could be a sentence, more than a sentence, or even less than a sentence, anything people say in a conversation so in model encoder is an utterance and the decoder is the response to that utterance. The conversational model is built using a wrapper function for the sequence to sequence model with bucketing. The simplest and initial form of the model is based on two RNNs, visualized in Figure 1.

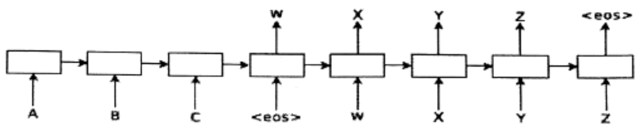

Fig -1: A general seq2seq model, where (A, B, C) is the input sequence, < EOS > is a symbol used to delimit the end of the sentence and (W, X, Y, Z) is the output sequence [13].

The actual implementation of RNNs can be in the form of LSTMs. The goal is to estimate the conditional probability of $P(y_1,...,y_N' | x_1,...,x_N)$,

$$(x_1,\ldots,x_N) = RNN(y_1,\ldots,y_N')$$

where x1,...,xN is the input sequence and y1,...,yN′ is the corresponding output sequence. Since two different RNNs are used for the input and output sequences, the lengths of the sequences, N and N′ can be different. The encoder is unrolled over the words and the last hidden state is called the thought vector, which is a representation of the whole source sequence.

3.2.1 Batching

One of the original sequence to sequence papers, Sutskever et al. 2014[13], gives better model performance if the inputs are reversed. During the preprocessing we build our vocabulary of unique words we replace words with low frequency with <UNK> create a copy of conversations with the words replaced by their IDs we add the <GO> and <EOS> word ids to the target dataset.

3.2.2 Encoder and Decoder of Sequence-to-sequence model

Encoder is a LSTM (Long Short Term Memory) [13] based neural network which takes the word embedding's of the input and converts them to a thought vector. The input will be the sequences of words. Every word is converted to its embedding. This number of hidden states of the encoder depend on the size of the input. Each LSTM cell converts a single word in the input sentence to one row of the thought vector [9]. The embedding is fed to LSTM Encoder as input. LSTM Encoder encodes the inputs into a thought vector. The thought vector is given as input to the LSTM Decoder. Then, decoder outputs the predicted word in the form of embedding of the output sequence. During training, the weights of the LSTM Network are randomly initialized in the first step. Once the Decoder outputs its first prediction, the error is calculated using the cross entropy error function. The initial hidden state of the decoder RNN is then set to this representation v [Fig. 1], and the generation of words in the output sequence is done by taking the output of unrolled decoder at each step and feeding into sampled softmax function, which produces the probabilities over all words in the vocabulary.

3.2.3 Cross entropy error function

After predicting output from decoder error calculated. Once the error is calculated, the derivative of every weight with respect to the error function gives the step by which we should increase or decrease the weight to minimize the error.

3.2.4 Sampled Softmax function

The loss function we use is Sampled Softmax function. By default, decode is set to be True, which means that during training, there is need to feed in the previously predicted token to help predicting the next token in the decoder even if the token was the wrong prediction. This helps approximate the training to be closer to the real environment when the conversational model has to make the prediction for the entire decoder from solely the encoder inputs.

3.2.5 Attention Mechanism with Seq2Seq Model

The attention mechanism visualized in was first introduced to encoder-decoder models by [Bahdanau et al., 2014] and then with modifications to inform network about past attention decisions by [Luong et al., 2015]. The problem that it was trying to address is the limited information that the context vector can carry. In a standard seq2seq model it is expected that this single fixed-size vector can encode all the relevant information about the source sentence that the decoder needs in order to generate its prediction. Additionally, since only a single vector is used, all the decoder states receive the same information, instead of feeding in information relevant to the specific decoding step. To overcome these shortcomings, the attention mechanism is used which then creates an additional input at each decoding step coming directly from the encoder states. The attention computation happens at every decoder time step. It consists of the following stages:

1. The current target hidden state is compared with all source states to derive attention weights.

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

2. Based on the attention weights we compute a context vector as the weighted average of the source states.

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s$$

3. Combine the context vector with the current target hidden state to yield the final attention vector.

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t])$$

4. The attention vector is fed as an input to the next time step (input feeding). Here, the function score is used to compared the target hidden state with each of the source hidden states, and the result is normalized to produced attention weights (a distribution over source positions). Scoring functions include the multiplicative and additive forms. Once computed, the attention vector is used to derive the softmax logit and loss. [24]

3.2.6 Bucketing

Considering largest sentence in our dataset is of length 100, we need to encode all our sentences to be of length 100, in order to not lose any words. Now, what happens if our sentence is of 3 words then there will be 97 PAD symbols in the encoded version of the sentence. This will overshadow the actual information in the sentence. Bucketing solves this problem, by putting sentences into buckets of different sizes. Separate subgraph for each bucket is created in model.

4.DATASET

Applying machine learning to any sort of task, one of the first things we need to do is consider the type of dataset that we would need to train the model. Sequence to sequence models requires a large number of conversations. From the high level perspective, this encoder decoder network needs to understand the type of responses (decoder outputs) that are expected for every query (encoder inputs). The conversation model comes with the script to do the pre-processing for the Cornell Movie-Dialogs Corpus, created by Cristian-Danescu-Niculescu-Mizil and Lillian Lee at Cornell University. This is an extremely well-formatted dataset of dialogues from movies. This dataset has 220,579 conversational exchanges between 10,292 pairs of movie characters, involving 9,035 characters from 617 movies with 304,713 total utterances. [23] For this training the Cornell Movie-Dialog dataset was used as the most frequent 32765 words in the training data were kept as vocabulary.

5.EXPERIMENTS AND RESULTS

In this section, we describe the experimental results with the dataset and show some samples of the interactions with the system that I trained. In this experiment, we trained a single layer LSTM with using stochastic gradient descent with gradient clipping. The vocabulary consists of the most common 20K words, which includes special tokens indicating turn taking and actor. Furthermore, while human evaluation is one of the most accurate metrics it still has its drawbacks. Most importantly, it is not automatic and thus can be costly and cannot be standardized.

In this section a detailed analysis of the trained models is given. They are also compared with a baseline seq2seq model presented in [Vinyals and Le, 2015][2], which was trained on Open Subtitles data. In Section below, a quantitative analysis is given by computing standard metrics. Then, in Section B below, a qualitative analysis is given by comparing sample output responses.

3.1Qualitative Analysis

In Table 1 the perplexity and Bleu scores of the trained models can be seen on the respective validation datasets. Since the models trained on Cornell Movie-Dialog data over-fitted quite quickly the dataset, the perplexity and Bleu scores presented are computed on the validation data before overfitting occurred. This means that for the Cornell model the scores were computed after 400K training steps. Learning rate was not adequate or the model was too small to be able to learn more. Future work should address these issues. The effects can be seen in training up to step 1M, on the Cornell Movie-Dialog dataset.

Table -1: Perplexity and Bleu scores on validation data for the baseline seq2seq model on Cornell Movie Dialogue Corpus.

METRICS	CORNELL MOVIE DIALOGUE CORPUS
PERPLEXITY	15.62
BLEU SCORE	13.19

3.2Quantitative Analysis

In this section output response samples from the various trainings are presented. For the sake of comparison with the baseline seq2seq model, the source utterances are a subset of the ones used in [Vinyals and Le, 2015]. The source utterances were divided into several categories like in [Vinyals and Le, 2015]. The generated responses by the trained models can be seen below. Interestingly the best responses from the two Cornell trainings were not those outputted before the models started to over fit. Before overfitting the models usually outputted very generic responses. By letting the models over fit a little on the training data more diverse and interesting responses were generated. This further suggests that the loss function is not in direct correlation with the quality of responses, since it gives a better score to safe and generic answers, like I don't know.

Specifically, the examples presented for the Cornell model are after training for 219K steps and 438k steps. In these conversations, Machine is our Neural Conversational Model, and Human is the human actor interacting with it.

Conversation 1: Basic conversation

USER: hi
BOT: hi
USER: How are you doing?
BOT: fine.
USER: Where are you?
BOT: at the airport.
USER: are you a man or a woman.
BOT: i'm sorry.
USER: why are we here?
BOT: Tell me about something else.
USER: what year you are born
BOT: I don't know.
USER: ok, bye!
BOT: i love you!

Conversation 2: Simple Q&A (contexts and multiple choice)

USER: are you a follower or a leader?
BOT: what's that?
USER: my name is john. What is my name?
BOT: I don't know what to say.
USER: How much is two plus two
BOT: a beer, man!
USER: who is Bill Clinton?
BOT: I don't know.
USER: how much is one plus one?
BOT: is a lot.
USER: what do you like to talk about?
BOT: are you princess?

Conversation 3: Morality

USER: what is altruism?
BOT: for the best.
USER: what is definition of altruism
BOT: I'm afraid.
USER: what is definition of morality.
BOT: Tell me about something else.
USER: tell me the definition of morality, i am quite upset now?
BOT: you ' ve been through the past since the day after last night .
USER: look i need help, tell me more about morality?
BOT: do you pray ?
USER: why living has anything to do with morality.
BOT: can be killed .
USER: what are the things that i do to be immoral.
BOT: f i ' m not going to be there .
USER: what is definition of altruism
BOT: I'm afraid.
USER: alright, morality
BOT: i ' ll be right back .
USER: i really like our discussion on morality and ethics
BOT: are over for you .
USER: what do like to talk about.
BOT: i ' m going to have to see you again tonight .

Conversation 4: Philosophical Q&A

USER: what is purpose of life
BOT: of the federation . i ' m sure it ' s safe .
USER: what is purpose of living
BOT: for my life .
USER: what is purpose of existence
BOT: i ' m afraid .
USER: where are you now
BOT: you ' ll be happy here .
USER: where are you now? tell me
BOT: at the table .

5.DOWNSIDES AND FUTURE SCOPE

There are a lot of intelligent conversational models in the world. Many have won the Loebner prize and are really clever programs. But even after all these advance scripts, the core problems with conversational entities (and other AI systems) remain more or less identical. None of such systems have the power to learn, or to hold memories of incidents for longer time. They lack creativity, one of the most important aspects of intelligence. Today's AI systems just keep following a fixed set of instructions. They do not actually show intelligence and don't understand what is happening around. The most popular methodology is just searching and finding matches from a database, called the rule method. But that is not intelligence. Hence there is still a vast need and scope for future improvements in this niche.

3. CONCLUSIONS

Machine learning is quickly growing field in computer science. It has applications in nearly every other field of study and is already being implemented commercially because machine learning can solve problems too difficult or time consuming for humans to solve. To describe machine learning in general terms, a variety models are used to learn patterns in data and make accurate predictions based on the patterns it observes. In recent years, Artificial intelligence has developed rapidly and is indispensable to daily life. However, on the other hand, how to solve tasks that are difficult to program explicitly has become a problem. For such tasks, neural networks provided the method to automatically extract the features from the given input. In this research, I am planning to develop a 'chatting bot' by applying neural network. In the present system, there is a possibility of becoming a better system by further increasing the data set and organizing the data set. Additionally, in future this work can be combined with emotion analysis and big data techniques for developing an emotional chat robot for which is able to respond to the user's emotions. This work or artificial entity is vulnerable to some grammatical mistakes.

Generalization is tied to supervised learning, which uses training data to train the model. Generalization is when a machine learning model can accurately predict results from data it hasn't seen before. With under-fitting, the model is unable to make accurate predictions with both training data and new data. With supervised learning, data is separated into three groups: train, dev, and test data-sets. The train data set is used to train the model. The dev data set is used to test the model during the model's development, but not during its training. The test data set is used when the model is complete to see how it reacts to data it has never seen before.

REFERENCES

[1] Tomohiro Fujita, Wenjun Bai, Changqin Quan and Sidney K. D'Mello, "Long Short-Term Memory Networks for Automatic Generation of Conversations", IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, ISBN: 978-1-5090-5504-3, 2017.

[2] Oriol Vinyals, Quoc V. Le, "A Neural Conversational Model", Proceedings of the 31st International Conference on Machine Learning, Lille, France, JMLR: W&CP volume 37, arXiv: 1506.05869, 2017.

[3] Lifeng Shang Zhengdong Lu Hang Li Noah's Ark Lab Huawei Technologies Co. Ltd., "Neural Responding Machine for Short-Text Conversation" arXiv:1503.02364} Apr 2015.

[4] Jin-Hyuk Hong, Sungsoo Lim, and Sung-Bae Cho, Senior Member, IEEE,", Autonomous Language Development Using Dialogue-Act Templates and Genetic Programming", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 11, NO. 2, APRIL 2007.

[5] Feng Wang, Wei Chen[1] ,Zhen Yang, Xiaowei Zhang, Shuang xu, Bo Xu, Institute of Automation, Chinese Academy of Sciences, "A Class-specific Copy Network for Handling the Rare Word Problem in Neural Machine Translation", 978-1-5090-6182-2/17 ©2017 IEEE.

[6] Rui Yan Baidu Inc.Yiping Song,Hua Wu Baidu Inc., "Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System" , 2016 ACM. ISBN 978-1-4503-4069-4/16/07} DOI: http://dx.doi.org/10.1145/2911451.2911542

[7] Marjan Ghazvininejad1* Chris Brockett2 Ming-Wei Chang2 Bill Dolan2 Jianfeng Gao2 Wen-tau Yih2 Michel Galley2,"A Knowledge-Grounded Neural Conversation Model", Information Sciences Institute, USC 2Microsoft Research} arXiv:1702.01932v1 [cs.CL] 7 Feb 2017.

[8] Oliver Lemon School of Informatics ,"Adaptive Natural Language Generation In Dialogue using Reinforcement Learning", olemon@inf.ed.ac.uk} 2013 http://homepages.inf.ed.ac.uk/olemon.

[9] Jonas Gehring, Michael Auli, David Grangier,Denis Yarats, Yann N. Dauphin,"Convolutional Sequence to Sequence Learning", Facebook AI Research}, arXiv:1705.03122v3 [cs.CL] 25 Jul 2017.

[10] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran,,"Deep Learning for Consumer Devices and Services", IEEE Consumer Electronics Magazine APRIL 2017. Digital Object Identifier 10.1109/MCE.2016.2640698 2162-2248/17©2017IEEE.

[11] Jiajun Zhang and Chengqing Zong, Institute of Automation, Chinese Academy of Sciences, "Deep Neural Networks in Machine Translation" , 1541-1672/15 © 2015 IEEE INTELLIGENT SYSTEMS Published by the IEEE Computer Society 2015.

[12] Pavlos Vougiouklis Jonathon Hare Electronics and Computer Science, Elena Simperl,"A Neural Network Approach for Knowledge-Driven Response Generation", Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3370–3380, Osaka, Japan, Dec 2016.

[13] Ilya Sutskever,Google ilyasu@google.com Oriol Vinyals Google vinyals@google.com Quoc V., qvl@google.com, "Sequence to Sequence Learning with Neural Networks", Google 2015.presented at the IEEE Summer power Meeting, Dallas, TX, June 22–27, 1990, Paper 90 SM 690-0 PWRS.

[14] M. Auli, M. Galley, C. Quirk, and G. Zweig. "Joint language and translation modeling with recurrent neural networks". In EMNLP, 2013.

[15] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate". arXiv preprint arXiv:1409.0473 , 2014.

[16] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. In Journal of Machine Learning Research, pages 1137–1155, 2003.

[17] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.

[18] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Arxiv preprint arXiv:1406.1078, 2014.

[19] Luong, T., Sutskever, I., Le, Q. V., Vinyals, O., and Zaremba, W. Addressing the rare word problem in neural machine translation. arXiv preprint arXiv:1410.8206, 2014.

[20] Mikolov, T. Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology, 2012.

[21] Mikolov, T., Karafi´at, M., Burget, L., Cernock `y, J., and Khudanpur, S. Recurrent neural network based language model. In INTERSPEECH, pp. 1045–1048, 2010.

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

[23] https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html.

[24] https://www.tensorflow.org/tutorials/seq2seq.