RNN and LSTM based Chatbot using NLP

Poonam Dhankhar[1]
[1]*Department of Computer Science and Engineering, MSIT, New Delhi, India*

Abstract- The paper discusses conversations from Cornell University's Movie Dialogue Corpus to build an interactive chatbot. Python has been mainly used for coding and Tensor Flow is used to build the model. The answers have been formulated to questions using a sequence-to-sequence model with LSTM cells, a bidirectional RNN encoder and decoder with attention.

Keywords – Computer vision, popularity prediction, support vector regression, video analysis, Natural Language processing.

I. INTRODUCTION

A chatbot is artificial intelligence software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, and mobile apps or through the telephone.

A basic sequence-to-sequence model consists of two recurrent neural networks (RNNs): an encoder that processes the input and a decoder that generates the output. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable length target sequence. Sequence-to-sequence is often used with attention-based that allows the decoder more direct access to the input. This model has been successfully used for many different natural language processing tasks, such as alignment, translation, and summarization. Conversational modeling can be phrased as a mapping between utterances and responses, and therefore can benefit from the encoder-decoder setup. In our model, the encoder processes an utterance by human, and the decoder produces the response to that utterance. We train the word embeddings as we train the model. We also use attention mechanism and experimenting with using GLoVe pre-trained word vectors to initialize our word embeddings. To make the bot speak like a certain character, we train vector embeddings for different characters with the hope that these embeddings would be able to encode information and style of speech of these characters. These character embeddings are trained together with the word embeddings. This is inspired by Google's Zero-shot multilingual translation system.

A chatbot is often claimed as one of the most advanced and promising expressions of interaction between humans and machines. However, from a technological point of view, a chatbot only represents the natural evolution of a Question Answering system leveraging Natural Language Processing (NLP). Formulating responses to questions in natural language is one of the most typical examples of Natural Language Processing applied in various enterprises' end-use applications.

II. LITERATURE SURVEY

From input sentence, it will be scored to get the similarity of sentences; the higher score obtained the more similar of reference sentences. The sentence similarity calculates using bigram which divides input sentence as two letters of input sentence [1]. The knowledge of chatbot is stored in the database. The chatbot consists of core and interface that is accessing that core in relational database management systems (RDBMS). The database has been employed as knowledge storage and interpreter has been employed as stored programs of function and procedure set for pattern-matching requirement. The interface is standalone which has been built using programming language of Pascal and Java [3].

A key challenge in designing conversational user interfaces is to make the conversation between the user and the system feel natural and human-like. In order to increase perceived humanness, many systems with conversational user interfaces (e.g., chatbots) use response delays to simulate the time it would take humans to respond to a message [2]. However, delayed responses may also negatively impact user satisfaction, particularly in situations where fast response times are expected, such as in customer service. This paper reports the findings of an online experiment in a customer service context that investigates how user perceptions differ when interacting with a chatbot that sends dynamically delayed responses compared to archabbot that sends near-instant responses[11]. The dynamic delay length was calculated based on the complexity of the response and complexity of the previous message. Our results indicate that dynamic response delays not only increase users' perception of humanness and social presence, but also lead to greater satisfaction with the overall chatbot interaction. Building on social response theory, we provide evidence that a chatbot's response time represents a social cue that triggers social responses shaped by social expectations [3]. BA Shawar, E Atwell presented two chatbot systems, ALICE and Elizabeth, illustrating the dialogue knowledge representation and pattern matching techniques of each. We discuss the problems which arise when using the Dialogue Diversity Corpus to retrain a chatbot system with human dialogue examples. A Java program to convert from dialog transcript to AIML [6] format provides a basic implementation of corpus based chatbot training [3]. We conclude that dialogue researchers should adopt clearer standards for transcription and markup format in dialogue corpora to be used in training a chatbot system more effectively.

Some author proposed a novel neural network model called RNN Encoder– Decoder that consists of two recurrent neural networks (RNN) [4]. One RNN encodes a sequence of symbols into a fixed length vector representation, and the other decodes the representation into another sequence of symbols. The encoder and decoder of the proposed model are jointly trained to maximize the conditional probability of a target sequence given a source sequence [9]. The performance of a statistical machine translation system is empirically found to improve by using the conditional probabilities of phrase pairs computed by the RNN Encoder–Decoder [2] as an additional feature in the existing log-linear model. Qualitatively, we show that the proposed model learns a semantically and syntactically meaningful representation of linguistic phrases [10].

Neural machine translation, a recently proposed approach to machine translation based purely on neural networks, has shown promising results compared to the existing approaches such as phrase based statistical machine translation [5]. Despite its recent success, neural machine translation has its limitation in handling a larger vocabulary, as training complexity as well as decoding complexity increase proportionally to the number of target words. In this paper, we propose a method based on importance sampling that allows us to not. The authors also showed that individual influence is also important for predicting the view count. So in summary the view count along with individual's activeness on Twitter can predict the popularity of a video [6].

III. DATASET

The Cornell Movie-Dialogs Corpus, created by CiristianDanescu-Niculescu-Mizil and Lillian Lee at Cornell University has been used in the paper. For the Cornell dataset, we use 20,000 pairs for testing, and the rest for training.

Train	201,617
Test	20,000

IV. MODEL

It’s a sequence-to-sequence model with attention mechanism that allows decoder more direct access to hidden state output by the encoder. For the RNNs, we used stacked LSTM cells of 2 layers. The encoder is the utterance by human, and the decoder is the response. We assume that in normal conversations, people listen to the first part and somewhat zone out to think of the answer, so we reverse the encoder so that the model can retain more information from the beginning of the utterance. Encoder takes a raw input text data. The output of encoder becomes the input data for decoder. Our model uses a start token or end token for encoders as well as decoders.

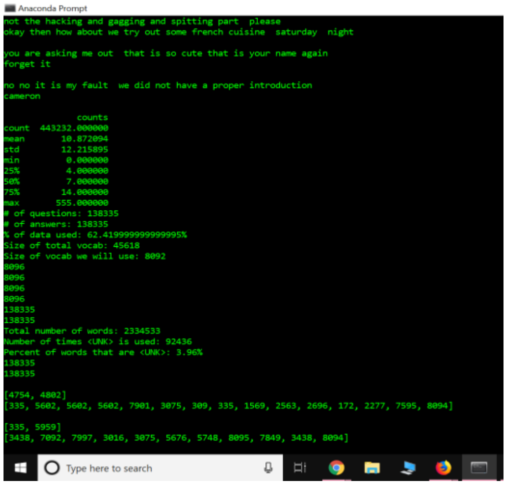
The model greedily produces the responses by using the most likely token at each decoder step.

Training: Using attention in our decoding layers reduces the loss of our model by about 20% and increases the training time by about 20%. I’d say that it’s a fair trade-off. Some notes to make:

- The model performs best when the attention states are set with zeros.
- The two attention options are bahdanau and luong. Bahdanau is less computationally expensive and better results were achieved with it.

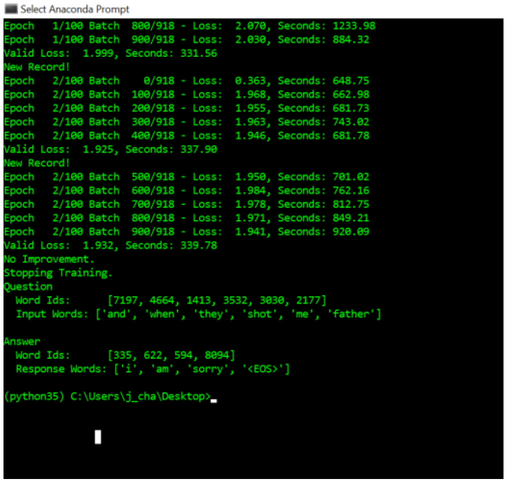
Hyperparameters: We use embedding size of 512, and the number of hidden unit in a GRU cell is 256. We use a fixed learning rate of 0.005, and clipped gradient when the norm exceeded 5.0. During training, we feed previously predicted tokens to predict the next token – this is to make the training environment similar to the testing environment.

V. RESULTS AND DISCUSSION



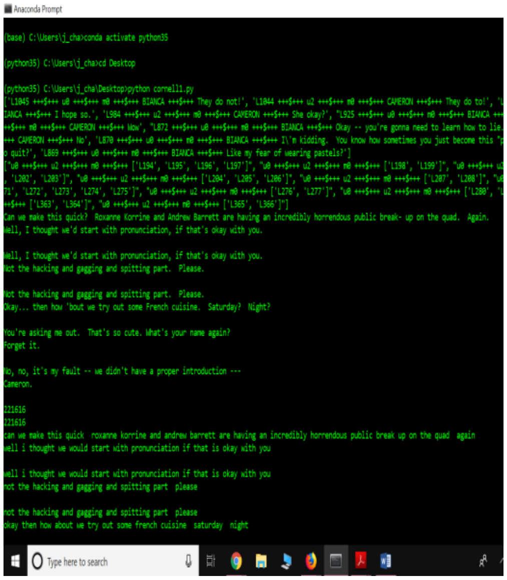
```
not the hacking and gagging and spitting part  please
okay then how about we try out some french cuisine  saturday  night

you are asking me out  that is so cute that is your name again
forget it

no no it is my fault  we did not have a proper introduction
cameron

              counts
count  443232.000000
mean       10.872094
std        12.215895
min         0.000000
25%         4.000000
50%         7.000000
75%        14.000000
max       555.000000
# of questions: 138335
# of answers: 138335
% of data used: 62.419999999999995%
Size of total vocab: 45618
Size of vocab we will use: 8092
8096
8096
8096
8096
138335
138335
Total number of words: 2334533
Number of times <UNK> is used: 92436
Percent of words that are <UNK>: 3.96%
138335
138335

[4754, 4802]
[335, 5602, 5602, 5602, 7901, 3075, 309, 335, 1569, 2563, 2696, 172, 2277, 7595, 8094]

[335, 5959]
[3438, 7092, 7997, 3016, 3075, 5676, 5748, 8095, 7849, 3438, 8094]
```

Figure 1 Output after cleaning of datasets



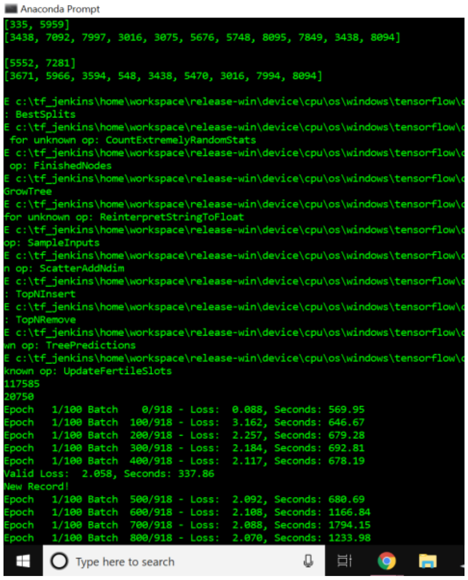
```
Epoch   1/100 Batch  800/918 - Loss:  2.070, Seconds: 1233.98
Epoch   1/100 Batch  900/918 - Loss:  2.030, Seconds: 884.32
Valid Loss:  1.999, Seconds: 331.56
New Record!
Epoch   2/100 Batch    0/918 - Loss:  0.363, Seconds: 648.75
Epoch   2/100 Batch  100/918 - Loss:  1.968, Seconds: 662.98
Epoch   2/100 Batch  200/918 - Loss:  1.955, Seconds: 681.73
Epoch   2/100 Batch  300/918 - Loss:  1.963, Seconds: 743.02
Epoch   2/100 Batch  400/918 - Loss:  1.946, Seconds: 681.78
Valid Loss:  1.925, Seconds: 337.90
New Record!
Epoch   2/100 Batch  500/918 - Loss:  1.950, Seconds: 701.02
Epoch   2/100 Batch  600/918 - Loss:  1.984, Seconds: 762.16
Epoch   2/100 Batch  700/918 - Loss:  1.978, Seconds: 812.75
Epoch   2/100 Batch  800/918 - Loss:  1.971, Seconds: 849.21
Epoch   2/100 Batch  900/918 - Loss:  1.941, Seconds: 920.09
Valid Loss:  1.932, Seconds: 339.78
No Improvement.
Stopping Training.
Question
  Word Ids:      [7197, 4664, 1413, 3532, 3030, 2177]
  Input Words: ['and', 'when', 'they', 'shot', 'me', 'father']

Answer
  Word Ids:      [335, 622, 594, 8094]
  Response Words: ['i', 'am', 'sorry', '<EOS>']

(python35) C:\Users\j_cha\Desktop>
```

Figure 2 Final output of chatbot conversation



```
(base) C:\Users\j_cha>conda activate python35

(python35) C:\Users\j_cha>cd Desktop

(python35) C:\Users\j_cha\Desktop>python cornell1.py
['L1045 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ They do not!', 'L1044 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ They do to!', 'L
IANCA +++$+++ I hope so.', 'L984 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ She okay?', "L925 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++
++$+++ m0 +++$+++ CAMERON +++$+++ Wow', "L872 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Okay -- you're gonna need to learn how to lie.
+++ CAMERON +++$+++ No', 'L870 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ I\'m kidding.  You know how sometimes you just become this "p
o quit?', 'L869 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Like my fear of wearing pastels?']
["u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L194', 'L195', 'L196', 'L197']", "u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L198', 'L199']", "u0 +++$+++ u2
, 'L202', 'L203']", "u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L204', 'L205', 'L206']", "u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L207', 'L208']", "u0
71', 'L272', 'L273', 'L274', 'L275']", "u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L276', 'L277']", "u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L280', 'L
++$+++ ['L363', 'L364']", "u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L365', 'L366']"]
Can we make this quick?  Roxanne Korrine and Andrew Barrett are having an incredibly horrendous public break- up on the quad.  Again.
Well, I thought we'd start with pronunciation, if that's okay with you.

Well, I thought we'd start with pronunciation, if that's okay with you.
Not the hacking and gagging and spitting part.  Please.

Not the hacking and gagging and spitting part.  Please.
Okay... then how 'bout we try out some French cuisine.  Saturday?  Night?

You're asking me out.  That's so cute. What's your name again?
Forget it.

No, no, it's my fault -- we didn't have a proper introduction ---
Cameron.

221616
221616
can we make this quick  roxanne korrine and andrew barrett are having an incredibly horrendous public break up on the quad  again
well i thought we would start with pronunciation if that is okay with you

well i thought we would start with pronunciation if that is okay with you
not the hacking and gagging and spitting part  please

not the hacking and gagging and spitting part  please
okay then how about we try out some french cuisine  saturday  night
```

Figure 3 Cleaning of Datasets



```
[335, 5959]
[3438, 7092, 7997, 3016, 3075, 5676, 5748, 8095, 7849, 3438, 8094]

[5552, 7281]
[3671, 5966, 3594, 548, 3438, 5470, 3016, 7994, 8094]

E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
: BestSplits
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
 for unknown op: CountExtremelyRandomStats
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
 op: FinishedNodes
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
GrowTree
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
for unknown op: ReinterpretStringToFloat
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
op: SampleInputs
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
n op: ScatterAddNdim
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
: TopNInsert
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
: TopNRemove
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
wn op: TreePredictions
E c:\tf_jenkins\home\workspace\release-win\device\cpu\os\windows\tensorflow\c
known op: UpdateFertileSlots
117585
20750
Epoch   1/100 Batch    0/918 - Loss:  0.088, Seconds: 569.95
Epoch   1/100 Batch  100/918 - Loss:  3.162, Seconds: 646.67
Epoch   1/100 Batch  200/918 - Loss:  2.257, Seconds: 679.28
Epoch   1/100 Batch  300/918 - Loss:  2.184, Seconds: 692.81
Epoch   1/100 Batch  400/918 - Loss:  2.117, Seconds: 678.19
Valid Loss:  2.058, Seconds: 337.86
New Record!
Epoch   1/100 Batch  500/918 - Loss:  2.092, Seconds: 680.69
Epoch   1/100 Batch  600/918 - Loss:  2.108, Seconds: 1166.84
Epoch   1/100 Batch  700/918 - Loss:  2.088, Seconds: 1794.15
Epoch   1/100 Batch  800/918 - Loss:  2.070, Seconds: 1233.98
```

Figure 4 Training process

VI. CONCLUSION

The trained chatbot came out to be very dramatic because of the movie lines being written dramatically. The chatbot can only respond as well as the data fed into it.

The following points were inferred in the making of this project.

1. LSTM cells typically outperform GRU cells for seq2seq tasks
2. Making the encoder bidirectional proved to be much more effective than a simple feed forward network.
3. We return only the encoder's state because it is the input for our decoding layer. Simply put, the weights of the encoding cells are what interest us.
4. Using attention in our decoding layers reduces the loss of our model by about 20% and increases the training time by about 20%.
5. The model performs best when the attention states are set with zeros.
6. The two attention options are bahdanau and luong. Bahdanau is less computationally expensive and better results were achieved with it.

7. Similar to initializing weights and biases, I find it best to initialize my embeddings as well. Rather than using a truncated normal distribution, a random uniform distribution is more appropriate.

VII. FUTURE WORK

In Future work we will use different datasets to make out chatbot more realistic. The different datasets shall be acquired from more recent TV series that are less dramatic and can hold more legitimate conversations.

VIII. REFERENCE

[1] B. Setiaji, FW Wibbowo, Chatbot using knowledge in database. 2016.

[2] Ulrich Gnewuch, Stefan Morana, Marc Adam, Alexander Maedche, Faster is Not AlwaysBetter: Understanding the Effect of Dynamic Response Delays in Human-Chatbot Interaction.2018.

[3] BA Shawar, E Atwell, Using dialogue corpora to train a chatbot. 2012.

[4] Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. 2014

[5] On Using Very Large Target Vocabulary for Neural Machine Translation. 2015

[6] A. Augello, G. Pilato, A. Machi, and S. Gaglio, “An Approach to Enhance Chatbot SemanticPower and Maintainability: Experinces within The FRASI Project,” Proc. of 2012 IEEE SixthInternational Conference on Semantic Computing, 2012, pp. 186-193,doi:10.1109/ICSC.2012.26.

[7] H. Al-Zubaide and A. A. Issa, “OntBot: Ontology Based Chatbot,” Proc. IEEE of 2011Fourth International Symposium on Innovation in Information & Communication Technology (ISIICT), 2011, pp. 7-12, doi:10.1109/ISIICT.2011.6149594.

[8] C. Erdogan, H. NusretBulus, and B. Diri, “Analyzing The Performance DifferencesBetween Pattern Matching and Compressed Pattern Matching on Texts,” Proc. IEEE of 2013International Conference on Electronics, Computer and Computation (ICECCO), 2013, pp.135-138, doi:10.1109/ICECCO.2013.6718247.

[9] J. P. McIntire, L. K. McIntire, and P. R. Havig“Methods for Chatbot Detection inDistributed Text-Based Communications,” Proc. IEEE of 2010 International Symposium onCollaborative Technologies and Systems (CTS), 2010, pp. 463-472,doi:10.1109/CTS.2010.5478478.

[10] Y. Wu, G. Wang, W. Li, and Z. Li, “Automatic Chatbot Knowledge Acquisition fromOnline Forum via Rough Set and Ensemble Learning,” Proc. IEEE of 2008 IFIP InternationalConference on Network and Parallel Computing, 2008, pp. 242-246,doi:10.1109/NPC.2008.24.

[11] S. Ghose and J. J. Barua, “Toward The Implementation of A Topic Specific DialogueBased Natural Language Chatbot As An Undergraduate Advisor,” Proc. IEEE of 2013International Conference on Informatics, Electronics & Vision (ICIEV), 2013, pp. 1-5,doi:10.1109/ICIEV.2013.6572650.

[12] G. Pilato, A. Augello, and S. Gaglio, “A Modular Architecture for Adaptive Chatbots,”Proc. IEEE of 2011 Fifth IEEE International Conference on Semantic Computing (ICSC),2011, pp. 177-180, doi:10.1109/ICSC.2011.68.