Seq2seq chatbot on small data

Alexander J. Van Roijen
Department of Computer Science
University of Washington
Seattle, WA, 98195
`avanroi1@uw.edu`

Abstract

I implement an attention boosted sequence to sequence (seq2seq) model to create a conversational AI on a small social media conversation dataset. Results show that dropout rates of 0.3 on non-user identified data of sequence length 10 produce the best results by my personal qualitative observations compared to more overfit models and similarly fit models with user identification included.

1 Background

Starting as early as the 1980s, we were using networks [1] to message each other in real time. This desire for instantaneous interaction and conversation wherever we go has driven the development and replacement of many IM platforms, such as AIM, ICQ, and many others [1]. These platforms have been replaced with various chat rooms and services that provide slightly different functionalities but all serve the ultimate purpose of providing potentially meaningful connections with others.

2 Topic

I will be focusing on chat services GroupMe, and Facebook Messenger. My main objective by the end of this project will be to use the data I have on conversations between myself and various groups (work or social) to develop an AI agent that seeks to communicate in a manner conducive the actors in the room.

(a) https://www.messenger.com/

(b) https://groupme.com/en-US/about

3 Motivation

Chatbots usually serve a purpose, such as helping organize meetings, provide answers to basic questions, or even play music. Unlike these other chat agents, such as DBpedia Chatbot and Discord's Rhythm bot[2,3], I aim to create a chatbot similar to that of Zo of GroupMe[4] that can dialogue with friends in the chat room. The main difference being to create an agent that can "roleplay", or mimic conversation in the chat. There are already smart "auto-response" prompts available on services such as LinkedIn or Facebook Messenger, but I seek to make these responses chat specific and engaging.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

4 Data and past work

I have already tinkered with Facebook messenger and GroupMe data in the past. The data for both GroupMe and Messenger come in JSON format. I have already created a parsing tool that can extract the history of user messages as well as create a naive bayes classifier on who sent a message based on its content. Initially, I was aiming to implement a chatbot on both GroupMe and Facebook Messenger data, or one agent for both datasets combined, but instead I have focused purely on the Facebook Messenger data. In figure 2 below , you can see information on the dataset containing 51798 messages over a 7 year period between 4 users with roughly 14000 unique tokens present within the corpus.

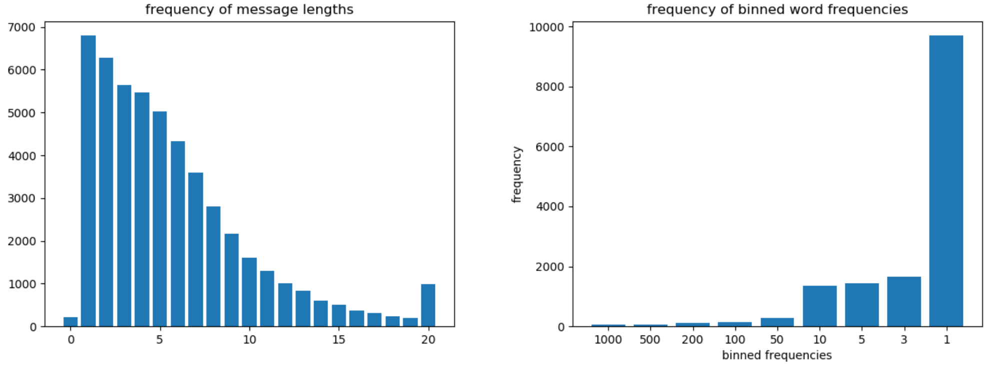

Figure 2: **Left:** Frequency of message lengths, capped at messages longer than or equal to 20.
Right: Frequency of word occurrences of bin sizes x $\geq$ binSize

5 Methodology

5.1 Architecture choices

The model follows exactly from Luong, Pham, and Manning's paper on attention based seq2seq models[5]. Unlike standard seq2seq models, this leverages a global attention layer. This means that the context of our inputs, represented by the output of our encoder, is transformed with the hidden state of our decoder to provide more focused context during translation. There is also a local attention model, but since our model isnt meant for particularly long sequences, there is no significant penalty for this implementation [6].
More specifically, the tunable parameters include the embedding dimension of our inputs, the size of our dense layers for our attention matrix, and maximum possible sequence length.

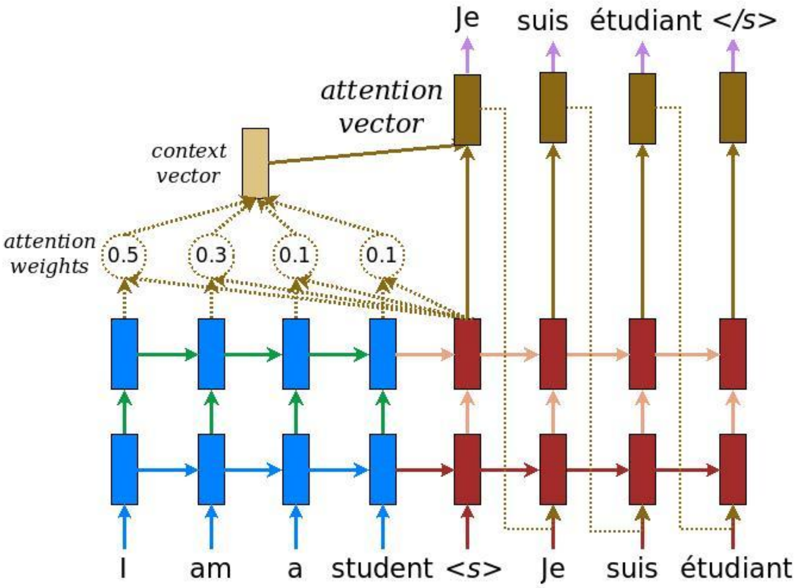

Figure 3: general format for seq2seq with global attention [5]

5.2 Data choices

Similar to Vinalys & Le, I consider the inputs and outputs to simply be sequential messages [6]. A second tunable parameter from this data is not simply the sequence length, but also the size of our vocabulary and what we do with simplified conversation data. Looking in figure 2, we can see that out of the 14000+ unique tokens in our dataset, only a few over 2000 are used more than 10 times or more. To handle this problem, two steps were used.

One, for all words that were not found in the large dictionary provided by pyspellchecker, levenshtein distance is used to determine correct spellings from the dictionary at distance of at most 2. If that word exists in our chat vocabulary, all instances of that misspelled word are changed to the former 79]. However, if the misspelling occurs more than some threshold y, the word is left alone. This allows slang terms unique to the group chat, such as "rekt", to remain.

Two, after taking care of misspellings, part of speech (POS) tags were used from NLTK for all remaining words below that threshold y and then kept in the corresponding POS dictionary [8]. During translation time, we can simply sample from the POS dictionary when its predicted by our model to replace the tag.

Lastly, we have a choice to provide user specific information to the model. Most seq2seq samples start with "<start>". However, in this setting it may be appropriate to start the sequence with "<start> <firstName>". Doing so should allow our model to learn responses to particular users, as user A may respond differently to user B for the same question asked by another user C.

5.3 Model Learning

When training, cross entropy loss was used to optimize our model. Furthermore, an ADAM updater was used to adjust learning rate over increasing epoch count. Finally, teacher forcing was used consistently during training. When the decoder would predict a word of the sequence, this token was not fed into the model for the next output unless it was inference time. Rather, the correct word of the sequence was used as input. There is a debate whether this is proper to do consistently at 100% during training, and a discussion can be found in the discussion section below.

5.4 Evaluation and final model selection

With training and architecture set, various parameters were tested and evaluated by splitting our data into a 75/25 split. Internal dimensions for the embedding layer and dense layer connections in the attention matrix range from 64 to 256 and 128 to 512 respectively. Dropout rates ranged from 0.5 to 0.40. thresholds for the message and vocabulary processing described above were between 3 and 10. Finally, sequence lengths were attempted at length 10 and length 20.

Final models were selected based on validation set loss at its most optimal point and final epoch performance. Following the lead of Vinalys & Le, I decided to spend more time evaluating clearly overfit models since our loss function doesnt fully capture our true objective [6].

6 Results

Model #	dropOut	Epochs	hasUser	seqLen	freqThresh	overFitIndex	trainLoss	valLoss	optimalValLoss
1	0.05	500	False	20	10	11	0.26	4.75	2.52
2	0.3	1000	False	10	10	12	1.048	3.53	2.46
3	0.3	1000	True	10	10	12	1.036	3.64	2.49

Table 1: hasUser = True if data included indication of what user made the statement. freqThresh represents at what threshold we replaced a word from the vocabulary as described previously. Overfit index is the epoch at which optimal validation loss was achieved.

Good Responses :

Model 1:
Human: let me know when
Machine: shoot for eight

Model 2:
Human: youre free around 5 our time mango?
Machine: prob more like 6

Model 3:
Jeremy: Why would I know who this is?
Machine: i feel like you know the broadcaster

Poor Responses :

Model 1:
Human: yoooo i saw that keyboard on reddit
Machine: yeah i was so coming into a way to explain out of a bike to the final project for

Model 2:
Human:alex when you leaving ?
Machine: were going to be able to leave
Human:okay
Machine: i got a scarf existence

Model 3:
Alex: mangos wanna do somtin today ?
Machine: kyle is kyle

Note that all responses listed above were recorded during their last epoch, not at the most optimal validation iteration.

7 Related Works

In 2015 Vinyals & Le used seq2seq to create a help desk chatbot and simulate conversation / banter [6]. Jozefowicz et al. further demonstrated how LSTMs outperform RNNs in the internal structure of seq2seq models on rare words [9]. A great summary of deep learning and its applications on NLP can be found in a paper from Young, Hazarika, Poira, and Cambria published in 2018 [10].

In a similar vein, the models above I developed by following a tutorial from tensorflow that implements a seq2seq model with an attention neural network. However, as highlighted prior, the model only learns responses with meaning when overfit to the training data, in which case it still has a limited set of responses. Looking forward, work done by Li, et al. show promising alternative objective functions that have been used to provide more human like responses [11].

8 Conclusion

Attention boosted neural networks have shown limited success on generating consistent human like responses to user queries trained on small data in the instant messaging setting. Cross entropy loss appears to provide an optimization for a related objective, but not the same objective [6]. Not shown in this paper, but a result seen by Vinalys & Le as well as by myself were underfit models that preform best on the validation set, but have very limited outputs to only a few dozen unique responses. However, we still see some promising results from slightly overfit models with modest dropout rates to generate environmentally similar responses.

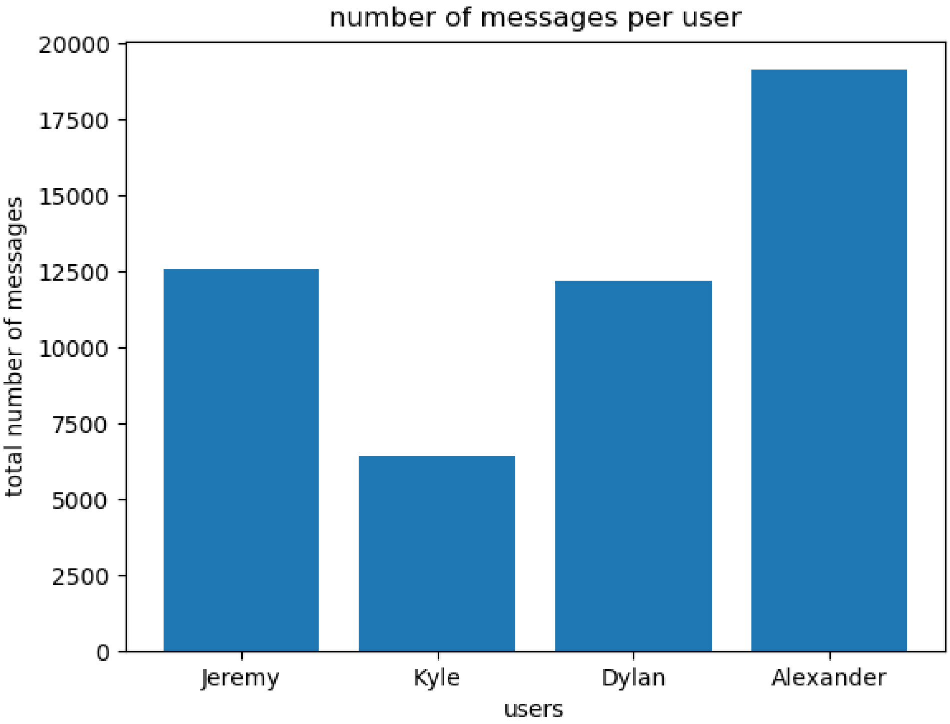

Figure 4: frequency of user messages are often highly skewed

9 Discussion

Overall I found model 2 to be the best performing based on its responses to common queries and diversity in responses. Unlike Vinalys & Le, there was no clear improvement, in my eyes, when adding the user tag to the input data [6]. This was likely due to two factors. One, by adding an additional user tag, the number of actual conversation tokens added is reduced by one for the input sequence, assuming you maintain the same sequence length size. Second, as highlighted in figure 4, there is not an even distribution of user tags, so this minority class may have much more poor results. It appears that overfitting is the only way to generate human like responses on this small dataset. Furthermore, using POS tagging delivers some poor results as the random sampling can lead to rather nonsensical answers. This is a byproduct simply of the small dataset size. To alleviate some of the overfitting concerns, I believe that by modifying teacher forcing to occur only sometimes during training may train a more adaptable model to newer sentences. Second,as seen in the related works section, there is active research being done to combat the issue of either basic or non human like responses using a different objective function [11]. My thoughts would be to use RMSE on the difference between embeddings / cosine similarity in the decoder phase before generating the vocabulary sized output vector. This would let the model use words that make sense in the context with minimal penalization. Finally, next steps would be to include multiple datasets from similar group chats, trying a one on one chat agent to try and mimic another user, and doing some data cleaning such that inputs and outputs are not just sequential messages, but blocks of messages that belong to the same user during the same time frame.

10 References

1. https://en.wikipedia.org/wiki/Instant_messaging
2. https://blog.dbpedia.org/2018/08/22/dbpedia-chatbot-2/
3. https://rythmbot.co/
4. https://help.groupme.com/hc/en-us/articles/115004382087-What-is-Zo-
5. Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025. https://arxiv.org/pdf/1506.05869.pdf
6. Vinyals, O., & Le, Q. (2015). A neural conversational model. arXiv preprint arXiv:1506.05869.
7. Barrust. (2020, February 17). pyspellchecker. Retrieved from https://github.com/barrust/pyspellchecker
8. Loper, E., & Bird, S. (2002). NLTK: the natural language toolkit. arXiv preprint cs/0205028.
9. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410.
10. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine, 13(3), 55-75.
11. Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055.

11 Appendix

Highlights were not shown for other models besides the three listed, but the other models did not utilize a dropout rate and early stopping either generated too basic responses or were too difficult to tune to determine optimal stopping points. All code used to generate the models can be found on my github