CS230

Nicebot: The Sentiment Analysis Chatbot

Yoni Lerner Stanford University
yonlern@stanford.edu

Abstract

In many applications, the goal of a chatbot is not simply to be human-like but to achieve a conversational objective. Nicebot is a chatbot that aims to produce positive responses from those that converse with it. The Nicebot architecture consists of a standard sequence-to-sequence network that is trained using sentiment analysis and a GAN-style discriminator. Nicebot has practical applications as a customer-facing chatbot and is an example of how a chatbot can be trained to accomplish conversational objectives.

1 Introduction

Chatbots represent an important challenge in natural language processing. However, many have focused on creating merely human-like chatbots, neglecting the motivation for having conversations in the first place. In practical applications chatbots usually have a conversational objective, such as helping a user navigate a website. Many companies are using chatbots in commercial applications today, and users can often become frustrated with such bots. This motivated the author to pursue a different approach to training chatbots, with maximizing user satisfaction as a driving focus. The input was defined as a snippet of text from the user. (For the sake of simplicity is it assumed the user initiates the interaction.) The output is some text response by the model, generated word-by-word.

2 Related work

The original conception of Nicebot was as a reinforcement learning agent, and the closest thing in the literature is MILABOT, a deep RL chatbot developed for the Amazon Alexa Prize competition.[7] MILABOT's core architecture has an RL agent choose a response from a set of options generated by an ensemble of existing natural-language models. This was different from the word-by-word approach of Nicebot, and meant that integrating sentiment analysis would have less effect on the responses of the bot.
In the commercial sphere, some companies, such as Dashbot, already offer sentiment analysis as a diagnostic tool for chatbot success.[3] However, to the extent that is publicly known, they do not use it to directly train their models.

3 Dataset and Features

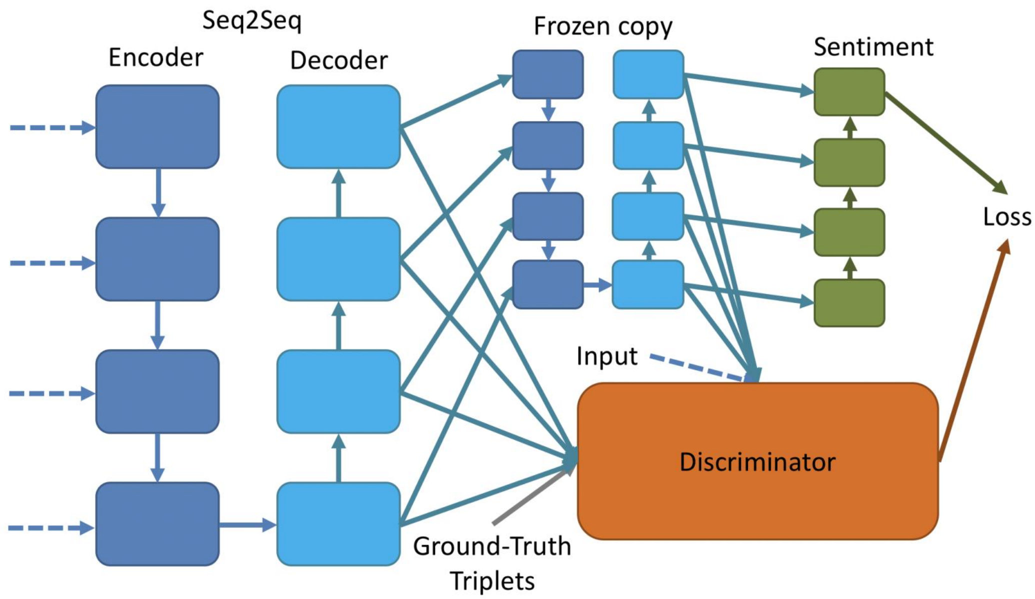
We used the Cornell Movie-Dialogs Corpus. [2] This corpus contains 220,579 two-speaker conversational exchanges. This dataset was used to generate the ground-truth triplets, as well as to sample input prompts. The data was preprocessed to tokenize and sanitize the sentences, convert words into vocabulary indices, and calculate word2vec word vectors.
The data was divided into triplets of an initial prompt, a response, and a follow-up response. Here is an example of a ground-truth triplet from the dataset:

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Initial: Oh yes, of course. There was someone with me. A lady.
Response: Looks like you're going to have no trouble at all. What was the lady's name, Mr. Cluett?
Follow-up: If you don't mind, Inspector, I'd rather not say - that is, unless it becomes absolutely essential. You see, she's married.

The sentiment analysis model was trained using the IMDB movie review dataset with similar preprocessing. [5]

4 Methods



The proposed model for Nicebot has three components - a standard Seq2Seq query/response model, a sentiment analyzer, and a GAN-style discriminator. The Seq2Seq component is given an initial prompt and generates a response. This response is then fed to a frozen copy of the same Seq2Seq network, generating a follow-up response. This creates an initial-response-followup triplet. A discriminator receives this triplet as a response, and is trained to distinguish it from round truth triplets drawn from the dataset. The Seq2Seq model is adversarially trained to fool the discriminator, ensuring that it will generate human-like conversation triplets. Finally, the followup is fed to a frozen sentiment analyzer, and the Seq2Seq model is trained to maximize the predicted sentiment of the followup. In this way, the model maintains human-like conversation while predicting the user's followup response and attempting to make it a positive one.

5 Experiments/Results/Discussion

The Nicebot model went through many iterations, starting as a reinforcement learning model. In this model, the sentiment analyzer was used to dynamically generate reward. However, the model did not converge to human-like behavior, and in the end backpropagating through the sentiment analyzer proved more effective. Unfortunately, the final architecture proved too complex to pin down in the short amount of time left. Combined with the computational requirements for training, this proved too ambitious an undertaking for a deep learning first-timer! However, the ideas behind the model are sound, and the approach of applying an adversarial discriminator to maintain human-like conversation while optimizing another objective holds merit.

6 Conclusion/Future Work

In the future, with more mathematical expertise, time, and computational resources, the full model would be optimized and evaluated to gauge its practical potential. One important area for expansion is in expanding the conversational scope of the model. Right now, Nicebot only considers the very last sentence and the very next one when composing a response. An interesting direction for exploration would be adding more context from previous segments of the conversation. Another direction would be to expand the look-ahead capability of Nicebot, possibly by predicting more than just one follow up response by looping through the frozen model several times.

7 Contributions

This was a single-person project.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[3] Dashbot. Dashbot conversational analytics. `http://www.dashbot.io/`.

[4] Hao Dong, Akara Supratak, Luo Mai, Fangde Liu, Axel Oehmichen, Simiao Yu, and Yike Guo. TensorLayer: A Versatile Library for Efficient Deep Learning Development. *ACM Multimedia*, 2017.

[5] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[6] Michael Schaarschmidt, Alexander Kuhnle, and Kai Fricke. Tensorforce: A tensorflow library for applied reinforcement learning. Web page, 2017. `https://github.com/reinforceio/tensorforce`.

[7] Iulian Vlad Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, Sai Mudumba, Alexandre de Brébisson, Jose Sotelo, Dendi Suhubdy, Vincent Michalski, Alexandre Nguyen, Joelle Pineau, and Yoshua Bengio. A deep reinforcement learning chatbot. *CoRR*, abs/1709.02349, 2017.

Libraries used: TensorFlow, [1] TensorLayer, [4] TensorForce [6]
Code: `https://github.com/c0d3rman/Nicebot2`