CSCI5622 Fall 2019 Group Project: MiMic Project Proposal and Midpoint Report

Nirvan S P Theethira
nith5605@colorado.edu

Charlie Carlson
chca0914@colorado.edu

Ketan Ramesh
kerk1228@colorado.edu

Pramod Venkatesh Kulkarni
prku8035@colorado.edu

Abstract

After careful consideration we have decided to change projects. Our original proposed project was to use the Google Quick Draw! data set to train a neural network to classify small hand-drawn sketches. We discovered that not only had many others already attempted this but they had posted their code online and used similar models to those we had hoped to try. Thus, to avoid redundancy, we are switching to a different project which we believe to be slightly more novel and much more interesting. Our new project is to train a chatbot with the personality of a TV show character. This document will serve both as a new project proposal and midpoint report where we will both review what we plan to do and report what we have already done to meet midpoint expectations.

1 Project Proposal

A chatbot is a program that provides conversational output in response to user input. They have many applications such as customer support interfaces, general question answering services, translation apps and virtual assistants. A common goal for chatbots is to simulate a human like interaction for the user. To this end many researches have investigated creating chatbots which can do more than just factually answer questions [1, 2, 3, 4, 5]. That is, they have investigated creating chatbots which have "personality" or "identity." In this work we propose attempting to train a neural network to act as a chatbot which simulates a known personality. Namely, we want to train a neural network to react as TV show personalities.

A Sequence-to-sequence model is a general purpose encoder-decoder model that can take structured text as input and output similarly structured text. For example, they can take in English sentences and output English sentences. Such models are regularly used to model translation, summarization and conversational tasks. We propose using such a model to train our TV personality chatbot as it is similar task.

For a data set, we will use transcripts from popular TV shows such as Friends and consider statement response pairs from scene dialogue. That is, for each scene in an episode we parse out the dialogue in order and pair them sequentially. Each dialogue instance includes the character speaking and the actual dialogue. We call the first dialogue instance a statement and the second a response. For each character of interest, we collect all pairs such that the character is the responder. We then train our model with the goal of it learning to react to input from any statement maker as the responder. Focusing on TV characters with strong personalities, we hope that after sufficient training, the chatbot will noticeably behave like the character of interest. For the interest of both automatic and human based metrics, we will reserve about 30% of our collected data for each character as test data.

Preprint. Under review.

Table 1: Example data point

Type	Character	Dialogue
Statement:	Charlie:	"This project is awesome!"
Response:	Ketan:	"Yes it is!"

While the task and goal for this project are clear, a major question is how to measure performance. It is believed that measuring the performance of such chatbots is difficult [6, 7, 8, 9, 10]. We propose using several metrics to measure our chatbot's final performance. While they are not usually very good indicators we will consider automatic metrics such as BLEU and METEOR. We will also look into more modern automatic metrics that have recently been introduced to measure the performance of personality based chatbots [7]. Finally, we will construct human evaluation metrics which is fairly standard for chatbot evaluation. An example idea is to ask a human to determine if a reported response does represent some notion of the character's personality. When designing theses human evaluation questions, we will have to careful consider the audience and create questions that will hopefully let us understand both how well our chatbot appears to hold a conversation and how well it matches the desired personality.

2 Midpoint Report

As of submitting this document we have collected and parsed our first dataset, the transcript for dialogue for the popular TV show Friends. We hope to collect additional data from other popular TV shows as well for our final result.

F.R.I.E.N.D.S Dataset

The chatbot training trials were started with the friends data set. The dataset was retrieved from: `https://fangj.github.io/friends/`. The text file are split up by episodes. Each text file contains dialogue scrips that indicate the character speaking. The scripts also indicate scene transitions.

```
['The One Where Monica Gets a New Roommate (The Pilot-The Uncut Version)\n',
 'Written by: Marta Kauffman & David Crane\n',
 'Transcribed by: guineapig\n',
 'Additional transcribing by: Eric Aasen\n',
 '(Note: The previously unseen parts of this episode are shown in blue text.)\n',
 '\n',
 '[Scene: Central Perk, Chandler, Joey, Phoebe, and Monica are there.]\n',
 '\n',
 "Monica: There's nothing to tell! He's just some guy I work with!\n",
 '\n',
 "Joey: C'mon, you're going out with the guy! There's gotta be something wrong with him!\n",
 '\n',
 'Chandler: All right Joey, be nice.  So does he have a hump? A hump and a hairpiece?\n',
 '\n',
 'Phoebe: Wait, does he eat chalk?\n',
 '\n',
 '(They all stare, bemused.)\n',
 '\n',
 "Phoebe: Just, 'cause, I don't want her to go through what I went through with Carl- oh!\n",
```

Figure 1: Raw Data.

For the basic implementation, the TV character 'Joey' was used. We collected all input-response pairs from every episode where Joey was the responder. The input and output were tab separated and saved in a text file. This formed 8000 input-response pairs that were used to train a sequence-to-sequence model.

```
 "oh, ju-ju-just stay calm. just be calm. for all he knows we're just hanging out together. right? just be nonchalant. (joey li
ke stands at attention with his chest forward and his hands on his sides, looking up at the ceiling with his lips pouted.) tha
t's not nonchalant!\tno idea what it means.",
 "yeah, sure... (looks outside into the corridor if joey is there)\ti don't believe this... have you guys been...",
 'you would think! joey!\tis he gone?',
 "(still can't find him) how are you doing this?\tpssst...",
 "oh no... have you thought about it how complicated this could get? what about ross?\twell, he's with charlie now.",
 "yeah, but he wants to talk to you before anything really happens with her. and as his friend, i mean, don't you think he dese
rves the same from you?\t(long pause and he twitches a bit) you're a pain in my ass, geller!",
 'get out! are they right?\tprobably, yeah... i mean, maybe we should... hold off until we talk to ross.',
 "yeah... yeah, we can wait, we don't have to do anything tonight.\tyeah, i think that'd be best... so, so i'm gonna... i'm gon
na take off...",
 '(joey walks to the door) although...\t(turns around very fast) i like although!',
 "i mean, you know... ross and i haven't dated in like... six years...\tsix years? wow... it's almost as long as highschoo
l...",
 "plus, you know, he is with charlie now.\tabsolutely! he's not thinking about you.",
```

Figure 2: Joey dialogue traning data.

Baseline Model - Seq2Seq

Sequence-to-Sequence learning deals with training models to convert sequences from one domain to another. Most common uses include Language translation but, the task can be extended to work for free-form question answering or any task involving text generation. The model takes in an input of variable length and produces a variable length output. The model contains an RNN/ LSTM that acts as an Encoder. The Encoder processes the inputs, encoding the information as internal states. These internal states act as context/conditioning in the decoding steps. The outputs of the Encoder are ignored. The Decoder, made up of an RNN/ LSTM is trained to predict the next character, given the hidden states formed by the Encoder. The output at each time step of the decoding phase is used as the input for the next time step leading to text generation.

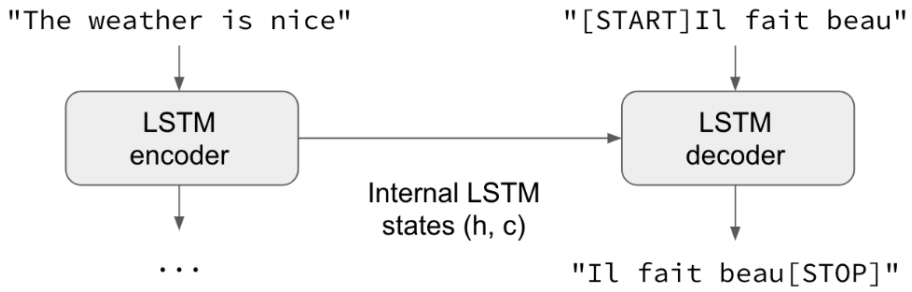

Figure 3: Baseline Seq2Seq Model Structure.

The model is trained on the data set described above. The Encoder is trained to generate hidden states for dialogues whereas the Decoder is trained to generate responses conditioned on Joey's responses.

```
-
Input sentence: (through wall) ooh... i love barbados!
Decoded sentence: yeah, i don't gonna go the gonda got and the sand and and and and and and and and and and
-
Input sentence: hey, you know, before you said that nothing could happen between us? what changed?
Decoded sentence: yeah, i don't gonna go the gonda got and the sand and and and and and and and and and and
-
Input sentence: (knocks on door) rach, you there? (joey and rachel both get up from the bed)
Decoded sentence: yeah, i don't gonna go the gonda got and the sand and and and and and and and and and and
-
Input sentence: oh, ju-ju-just stay calm. just be calm. for all he knows we're just hanging out together. right?
Decoded sentence: yeah, i don't gonna go the gonda got and the sand and and and and and and and and and and
-
Input sentence: yeah, sure... (looks outside into the corridor if joey is there)
Decoded sentence: yeah, i don't gonna go the gonda got and the sand and and and and and and and and and and
-
```

Figure 4: Sample output after 10 epochs.

Next Steps

Moving forward we have identified several key tasks that we want to complete in the next few weeks:

1. Collect additional data sets from other TV shows such as Big Bang Theory, Lost, Game of Thrones and The Simpsons.
2. Finalize and implement automatic metrics for evaluating chatbot performance.

3. Tune a sequence-to-sequence model and run on full data sets to train personality chatbot.
4. Research and implement strategies for human based evaluation metrics.

We have also considered a few reach goals that we would like to consider moving forward:

1. Research and implement other models to train same data set. One example is to use BERT [9].
2. Create a user interface for interaction with trained personality chatbots.
3. Consider using a much larger data set for general human text interaction in hopes of increasing quality of trained model's responses.

References

[1] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. *CoRR*, abs/1603.06155, 2016.

[2] Suzana Ilic, Reiichiro Nakano, and Ivo Hajnal. Designing dialogue systems: A mean, grumpy, sarcastic chatbot in the browser.

[3] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Assigning personality/identity to a chatting machine for coherent conversation generation. *CoRR*, abs/1706.02861, 2017.

[4] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. *CoRR*, abs/1704.01074, 2017.

[5] Huyen Nguyen, David Morales, and Tessera Chin. A neural chatbot with personality.

[6] Nicole M. Radziwill and Morgan C. Benton. Evaluating quality of chatbots and intelligent conversational agents. *CoRR*, abs/1704.04579, 2017.

[7] Yujie Xing and Raquel Fernández. Automatic evaluation of neural personality-based chatbots. *CoRR*, abs/1810.00472, 2018.

[8] Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *CoRR*, abs/1603.08023, 2016.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[10] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. A survey of available corpora for building data-driven dialogue systems. *CoRR*, abs/1512.05742, 2015.