Barney to Spock: A Neural Machine ChatBot

Ishaan Agrawal
Columbia University
Computer Science (B.S)
ia2341@columbia.edu

Sunand Iyer
Columbia University
Computer Science (B.S)
sri2117@columbia.edu

Viraj Rai
Columbia University
Computer Science (B.S)
vr2736@columbia.edu

Abstract

In this paper, we explore the creation of a neural chatbot from multiple different datasets with the aim of incorporating style, tone and eccentricity of a seed character to the chatbot. Aside from the tempting upside of being able to model the language of characters, such a neural chatbot would allow us to model any individual's tone and way of speaking. This could have huge implications for automated customer service and medical chatbots. We used a Seq2Seq model with an attention mechanism to train a generic chatbot before fine tuning the chatbot on movie transcripts of the desired sci-fi characters. Our model shows considerable promise in its ability to capture a character's tone, while also capturing some of the content of the movie itself. While the results are not perfect in terms of style and eccentricity, the neural responses are consistently grammatically correct, therefore, allowing the user some ability to control the flow of the conversation.

1. Purpose and Background to the Project

The ability to model a person's speaking style, grammatical structure, personality and tone can provide us with the ability to create and design a variety of communication platforms. The wide range of applications extends to electronic recorded communications for customer service firms, mental health chat bots tuned for individual patients, machine translation that preserves tone and personality etc. The aim of this investigation is to be able to model character-influenced conversation in a chat-box environment that can preserve a fictional character's individual eccentricity, personality and tone.

2. Introduction and Goal

There has been a recent rise in the effectiveness of deep-learning based chatbot. The goal of this research endeavour is to create a Seq2Seq model that models conversation between 2 fictional characters from movie scripts in a chat box environment. The model will be trained on multiple movie transcripts with dialogues between 2 characters. The chatbot should be able to speak in coherent sentences and be able to respond appropriately to dialog. Furthermore, another aim of the model will be to preserve character tone and personality in unexplored conversational domains. For example, a conversation between Darth Vader (Star Wars series) and Barney Stinson (How I Met Your Mother series) would begin with a topic question or statement and the characters will venture on a conversational back-and-forth with the same style as defined in the movie scripts. We hope to have a chatbot that can respond to input especially similar to movie dialogues. The reason behind training the model on characters from movie transcripts is threefold:

- There is abundant training data available on transcripts of movie series. For example, series such as Star Wars and Harry Potter can provide character dialogues of up to 10 hours (movie time). This gives us the ability to focus purely on the model architecture rather than worrying about data mining.

- If we can accurately model conversations between fictional characters with solely movie transcripts, then we can confidently state that one can mimic a real individual's speaking style and tone with enough conversational data.

- Given the abundance of movie transcript corpora, an amalgamation of transcripts from a number of movies and transcripts could model generic human conversations sufficiently well. Therefore, this would provide a baseline to our model, over which we can fine-tune the network weights to mimic the style of a certain character.

For the purposes of this project, we will look at the interaction between a fine-tuned chatbot and a human moderator. This provides us with the advantage of controlling the flow of a conversation (in comparison, a chatbot-to-chatbot conversation may run the risk of the conversation spiralling into redundancy). Moreover, since the vocabulary between the two models would be different, a chatbot-to-chatbot conversation may not result in a meaningful conversation many back-and-forths down the line.

3. Related Work and References

The inspiration for our initial approach to the chatbot model was the work put forth in the paper "A Neural Chatbot With Personality" [8]. In this published paper, Huyen Nguyen demonstrates the ability to train a neural chatbot solely with movie transcripts. While the conversation is not as freely flowing as one would expect, the output of the chatbot model were grammatically coherent, even though the responses were irrelevant to the input utterance [8]. In order to start with a more conversationally coherent model, we used the published article "A Neural Conversational Model" [11] as inspiration for our model architecture. We will be treating the model provided in [11] as a starting point, and will also provide a good baseline to evaluate our improvements to the model.

Human	E2C Chatbot
Red Alert.	Captain Uhura what's happening?
Engage.	Warp one sir.
Do you like me?	Bones I like you.
Shall I leave?	No Spock.
Where do you want to go?	Take me home.

Table 1. E2C Chatbot replicating Spock's style of speech in [7]

In the paper "Enterprise to Computer: Star Trek chatbot", Grishma Jena approaches the neural chatbot with a similar strategy [7]. Both [8] and [7] use the Cornell Movie Dataset [6] as the base dataset with which they train their neural models. However, [7] takes it one step further by fine tuning the generic model on Star Trek movie transcripts, thereby injecting a science-fiction personality to the chatbot. We will be using a similar approach when trying to ensure that our chatbot model that mimics a movie character does so by enforcing the character's perks, tone and eccentricity.

Previous attempts at creating a robust chatbot model used a bucketing system for inferences [8] and [7]. This led to varying results for each bucket due to lack of sufficient data for each bucket. Therefore, in order to avoid this problem, we will replace the bucketing system with a cap on the maximum length of the input sequence. This, in essence, restricts the model to one bucket that accepts input of length 0 to 10. This not only solves the problem of data sparsity, but also drastically reduces the space complexity of the model, thereby allowing us to train on larger datasets with a larger vocabulary size.

4. Dataset

For all the work performed so far, the major dataset used to train our neural network is the Cornell Movie Dataset [6] as used by [8] and [7]. This dataset contains 220,579 conversational utterances, along with a vocabulary of around 9,000 unique words that occur at least 6 times (words that appear once are treated as unknown $<UNK>$). This dataset provides a good starting point for model training. The reason for this is that the end goal of the project is to simulate conversation with a fictional character. Therefore, a generic movie transcript dataset of fictional characters will provide a good guideline for character to character interaction in a fictional setting. For example, a chatbot with Star Wars character Yoda would be difficult to mimic using a generic data corpus, as the grammatical structure of the majority of Yoda's responses are inverted to conventional English.

Therefore, the Cornell Movie Dialog Corpus provides a nice starting point to train the model.

While [6] will be used to train a generic conversational bot, we will use character specific dialog corpuses for major movie characters. Keeping the scope of the project in mind, we would like to create a model for (at most) 3 major fictional characters. Examples of these characters could include Spock (Star Trek Original Series), Barney Stinson (How I Met Your Mother TV Series) and Darth Vader (Star Wars Original Trilogy). While [6] does provide for dialogues for some of these characters, the corpus is not exhaustive. Therefore, we will also resort to "How I Met Your Mother Transcripts" [3] to collect data to fine tune the generic chatbot model on Barney Stinson, "Star Trek Transcripts" [4] to fine tune on Spock, and "Star Wars Movie Scripts" [5] to fine tune on Darth Vader. Details on data pre-processing will be provided in Section 5.

5. Current Model Architecture and Algorithm

The proposed project takes place in 3 phases.

1. Phase 1 - The first phase is to generate and train the baseline model described in [11] on the generic Cornell Movie Dialog corpus [6]. This will provide a standard baseline over which we can fine tune the model weights to mimic our selected fictional characters.

2. Phase 2 - Modify the character-specific data set to match the Cornell Corpus style. This would allow us to simultaneously train the model on both the generic as well as the specific corpus. Moreover, this gives us control over the weightage of each of these data sets while training the model.

3. Phase 3 - The third phase of this project is to train the same network architecture on the composite data set created in Phase 2. Here, we will experiment with different weightages between the two data sets, vocabulary threshold, number of epochs and other hyper-parameters to create the most "optimal" chatbot model.

Given below is a list of technical specifics used in the current model as well as improvements were made:

1. Data Preprocessing
2. Encoder - Decoder Seq2Seq Model
3. Attention Decoder
4. Word Embeddings

5.1. Data Preprocessing

In order to obtain data for the different characters, we utilized a variety of methods including webscraping, Kaggle and existing Github repositories [5] [4] [3]. In addition, we also used the Cornell Movie Dialog dataset for the generic chatbot [6].

Given below is a detailed description of how each dataset was handled:

- Cornell Dataset - We used an algorithm proposed and implemented by [1]. The algorithm takes each line from the corpus and converts it into a JSON object along with the metadata for each line. Next, for each group of lines that constitutes a conversation, the JSON objects are appended to another overarching object structure that now consists of the entire conversation. From this conversation object, different subsets of lines can be drawn to serve as input and output pairs to the model. These pairs are then converted the word embeddings before being passed to the neural network.

- Character Specific Datasets - For these datasets, data processsing was a much more involved task, as we were tasked with converting the raw dialog files to match the Cornell style corpus. As a result, specialized scripts were created for individual datasets to convert the lines to Cornell corpus style. However, the datasets were not comprehensive in delineating each conversation i.e. the data corpus provided no indication of which lines in the transcript constitute a conversation. As a result, we used a simple data heuristic to generate synthetic conversations. This heuristic ensured that a conversation consisted of only 2 lines, where the desired character responds to the other character. This ensures that the network accurately models only the character we have specified, and not any other character in the same fictional universe.

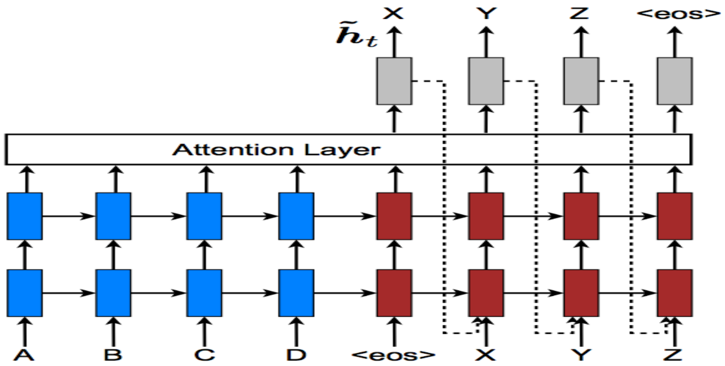

Figure 1. General Architecture of our Seq2Seq Model with an Encoder-Decoder mechanism, Embeddings layer and Attention Decoder [2]

With all of our data corpuses now in the same format as the Cornell Movie Dialog Corpus, we were able to use an already existing class to read in the data for the model [**?**].

5.2. Encoder - Decoder Seq2Seq Model

The current model is a variation on a Seq2Seq model; however, the major architecture of the model is a classic Seq2Seq model. It consists of two submodels, an encoder model and a decoder model that takes the input utterance and the desired response respectively. The encoder model will generate a vector representation of the input statement. Meanwhile the decoder will try to estimate the probability of the next token based on the encoded vector and the input of the decoder.

5.3. Attention Based Decoder

It is the opinion in the academic community that attention provides significant improvement to classical deep learning approaches to deep learning tasks. This is because the attention mechanism allows the decoder units to only focus on the desirable portions of the encoded vector generated by the encoder model. The current model also has an attention mechanism inbuilt into it. This should allow for a coherent conversational structure. The attention mechanism mimics the description suggested in “End-to-End Continuous Speech Recognition” by Bahdanau et. al [9].

5.4. Word Embeddings

One shortcoming of the baseline model proposed in [8] is the fact that it uses a hashing technique to represent individual tokens of an utterance. These tokens are then passed through an embedding layer where word embeddings are generated. Since the baseline model is trained specifically on fictional movie dialog corpuses, the generated word embeddings may not accurately reflect the semantic information stored in the token. Moreover, these embeddings are learned on the fly. However, since the dataset used to train the embeddings is on the order of 300,000 human utterances (and therefore, around 115,000 conversations), the generated embeddings may be prone to errors. Therefore, we will look to improve on the baseline by using a pre-trained word embedding. We experimented with a variety of word embeddings before arriving at the 300 dimensional word embedding, including a variety of different Stanford Glove embeddings [10]. Given below are a list of all the word embeddings that were experimented with before finazlizing the choice.

- 6B - 50d, 100d, 200d and 300d Glove Embeddings
- 42B-300d Glove Embeddings (Common Crawl)
- 27B-200d Glove Embeddings (Twitter Corpus)
- Google News Word Embeddings

While judging each pre-trained word embedding, we looked at the vocabulary quality and vocabulary size. Although the dimensionality of the Google embeddings was the largest (300 dim vectors), it resulted in a larger overall vocabulary size when combining it with the Cornell Movie Dialog corpus as well as the other character specific datasets. For example, the proper nouns “Vader”, “Yoda” and “Spock” are nonexistant in the Glove Word Embedding; however, these words do exist in the Google Word Embeddings; as a result, given the context within which we are trying to train our neural network, the 300 dimensional Google Word Embeddings seemed like the best choice. Moreover, the large dimension size may provide better information regarding the semantics of the word.

5.5. Overall Model

The overall model can be summarized as follows:

- Process Raw Data to Conversational Objects
- Seq2Seq Model - Legacy Tensorflow API
- Embedding Layer - Google Word Embeddings
- 2 LSTM Layers - Encoder Block
- Attention Decoder Block with 2 LSTM Layers

5.6. Output Inference

For the inference portion of the model, we use a Greedy approach based on the output of the neural network. The network outputs the softmax probability for each word ID at every position in the output sentence. Therefore, for each vector in the output matrix, we choose that word ID with the highest softmax probability of occuring in the sentence at the position given the input utterance. As a result, every word in the output sentence is the highest softmax output at that position. While this does not necessarily guarantee good results, the model should comprehend natural language grammatical structures i.e. the model will be conditioned on generating output based on the words surrounding it.

6. Technical Depth and Innovation

While we did build on previous models for this chatbot, we still added our own innovations. We extended a current model on the Cornell Movie Dialog Corpus to use an attention mechanism. This attention system, as previously mentioned, is implement using the Legacy Tensorflow API matching the description in [9]. Furthermore, we modified the approach of [8] which first trained on the Cornell Movie Dialog Corpus, secondly fine tuned the generic model on dialogues from a specific movie and thirdly further fine tuned on a certain character from that movie. Instead, we just fine tuned the generic chatbot on conversations of the specific character in a movie itself. This ensures that the model is not diluted by the tones and styles of other characters within the same fictional universe. Moreover, this also prevents vocabulary conflicts that may arise from training the model several times over completely different datasets. Instead we created a composite dataset comprising of the generic Cornell Movie Dialogue corpus as well as conversations involving only the desired character. Given that the variety of characters in a given fictional universe, undertaking the second step as done in [8] would only amortize the movie dataset to a more generic dataset. However, overfitting on conversations of a specific character should allow the chatbot to resemble that character by increasing the weightage of that dataset in comparison to the generic dataset. Another approach we incorporated into the training process with respect to fine tuning the generic model was repeating the conversations of a specific character to make it a high percentage of the training data. Thus the network would train multiple times over the same dialog to better capture it. What this would also enable is an enhanced vocabulary for the desired character. Therefore, fewer words utilized by the character would be characterized as $< UNK >$ (if any, depending on the vocabulary count threshold and the number of repetitions for the character-specific dataset).

7. Results and Evaluation Metrics

In this section, we will discuss the performance of each one of the fine tuned chatbots on the fictional characters and compare them with the generic chatbot. We will even discuss the evaluation metrics used to quantitatively gauge the performance of the individual chatbots.

It is interesting to note that all the chatbots (both fine tuned and generic) tend to almost always reply in grammatically correct sentences. Most of the time, the response is relevant to the input statement; however, for a few of the fine-tuned models, the response may not always be relevant. Given below is an example of the output for each one of the models given an input statement:

Human	Chatbot Response
Is Windows better than Linux?	Spock: No, doctor.
Is Windows better than Linux?	Barney: You're making sure of that
Is Windows better than Linux?	Lord Vader: Yes, of course.

Table 2: Responses of various models on the same input

Other than the difference in Spock's answer, one can clearly notice the variation of tone in these models. Moreover, the sentences are indeed gramatically correct. This is the case for the majority of neural responses for any given human input. Spock appears to be very stoic in his response (as expected from his reputation in Star Trek), while Barney employs the most informal tone. Vader's response, while varying in style, does not accurately reflect his tone as engendered in Star Wars.

This is largely due to the unfamiliar domain of the input statement to the model. The models are fine tuned on data corpuses that resemble speech and content for the given fictional universe. As a result, tone may not always be correctly reflected though unfamiliar contexts. Given below is another example of the responses of each model on a different input statement. The example below tries to explore a theme familiar to all 3 characters:

Human	Chatbot Response
I love you!	Spock: I know.
I love you!	Barney: Oh buddy, you do love me.
I love you!	Lord Vader: I love you.

Table 3: Responses of various models on a different input statement

Once again, we can tell the difference in tone amongst the responses above. Spock acts the most platonic out of all 3 characters. On the other hand, Barney acts like his jovial self. Darth Vader seems to be right in the middle of the spectrum outlined by Spock and Barney. As apparent in the prequel trilogy, Vader's underlying character is driven by love-lust and his response reflects his familiarity with those emotions.

The neural responses, however, are not always perfect. One such instance is that the model does not retain pertinent character information and facts (it only models tonal and stylistic behaviour). This becomes apparent through the following table:

Human	Chatbot Response
How old are you?	Spock: Eighteen
How old are you?	Barney: Eighteen
How old are you?	Lord Vader: Twenty eight

Table 4: Another set of neural responses

The responses above, while gramatically correct, are factually incorrect. This shows that one of the shortcomings of the proposed algorithm and neural network architecture is that it cannot effectively learn character content and information.

Given below are a couple evaluation metrics calculated while training the model. In order to create a sense of scale, the metrics were evaluated not only for the fine tuned model, but also for the initial generic model:

1. Spock Model:
    - Sampled Softmax Loss: 0.904
    - Perplexity: 2.47
2. Darth Vader Model:
    - Sampled Softmax Loss: 1.061
    - Perplexity: 2.89
3. Barney Stinson Model:
    - Sampled Softmax Loss: 0.924
    - Perplexity: 2.52
4. Generic Chatbot Model:
    - Sampled Softmax Loss: 1.089
    - Perplexity: 2.97

8. Conclusion

This investigation proves that it is indeed possible to mimic and model the speaking tone, style and eccentricity of any individual given sufficient data. The model was created solely using a Seq2Seq model without any other external rules. However, if we were to implement the knowledge graph algorithm briefly mentioned in the previous section, then we would need to design hand-crafted rules for certain inputs.

On the other hand, while the generated are not perfect, they do show some resemblance to the characters that they were trained on. While it would be preferable to have access to more data, it must be acknowledged that around 3000 conversational lines are sufficient to model any given person's identity successfully.

From the results above, it is easy to see that characters with a greater eccentricity (such as Spock and Barney) are much easier to model. This is due to the variance in their fine-tuning dataset when compared to the Cornell dialogue corpus. Therefore, modelling simplistic characters may require further research.

9. Further Work and Improvements

A potential future work would be to develop an algorithm to improve the extraction of conversations. This would allow us to create better input for the model as right now conversations of multiple lines would not be chopped into blocks. As mentioned earlier, we used a novel data heuristic to partition the dialogues of the fine-tuning data set into conversations. Either deriving a more robust heuristic, or finding an enhanced dataset would be more helpful in order to correctly model conversational dialogue.

The model could also be improved by using a Hidden Markov Model (HMM) for speech generation instead of a greedy approach. AS mentioned before, the inference stage would greedily pick a word for each part of the sentence without regards for the grammatical structure or word ordering. The current model assumes a Bayesian conditional independence between generated words, but this can leads to incoherent, unstructured sentences. We believe a HMM could significantly improve the generated text and give the chat-bot a more eccentric personality. Although the current model doesn't suffer from grammatical mistakes as much as one would anticipate from a greedy approach, the HMM approach would allow us to be more certain of the neural output.

Significant improvements could also be made if the chat-bot utilized a knowledge graph. Currently, the chat-bot learns the tone of the character and their catch phrases. This gives the chat-bot a 'personality'. But to make the chat-bot truly speak like a character, the chat-bot needs to remember the details of the character. This could be done by attaching a knowledge graph the chat-bot can develop and use to reflect the character its supposed to mimic.

References

[1] Deepqa - my tensorflow implementation of "a neural conversational model", a deep learning based chatbot. https://github.com/Conchylicultor/DeepQA.

[2] How does attention work in encoder-decoder recurrent neural networks. https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2017/08/Feeding-Hidden-State-as-Input-to-Decoder.png.

[3] How i met your mother transcripts. http://transcripts.foreverdreaming.org/viewforum.php?f=177.

[4] Star trek transcripts. http://www.chakoteya.net/StarTrek/episodes.htm.

[5] Star wars movie scripts. https://www.kaggle.com/xvivancos/star-wars-movie-scripts.

[6] C. Danescu-Niculescu-Mizil and L. 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.

[7] A. B. L. U. J. . S. Grishma Jena, Mansi Vashisht. Enterprise to computer: Star trek chatbot. https://arxiv.org/pdf/1708.00818.pdf.

[8] T. C. Huyen Nguyen, David Morales. A neural chatbot with personality. http://web.stanford.edu/class/cs224n/reports/2761115.pdf.

[9] K. C. Y. B. Jan Chorowski, Dzmitry Bahdanau. End-to-end continuous speech recognition using attention-based recurrent nn. 2014.

[10] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[11] O. Vinyals and Q. V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.