Mimic: Character Based Chatbot

Nirvan S. Theethira
nith5605@colorado.edu

Charlie Carlson
chca0914@colorado.edu

Ketan Ramesh
kerk1228@colorado.edu

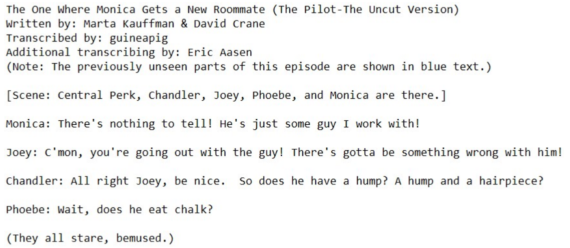
Pramod Vankatesh Kulkarni
prku8035@colorado.edu

Problem

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. That is, they have investigated creating chatbots which have "personality" or "identity." The goal of this project was to train a neural network to act as a chatbot that simulates the known TV personality Joey from F.R.I.E.N.D.S.

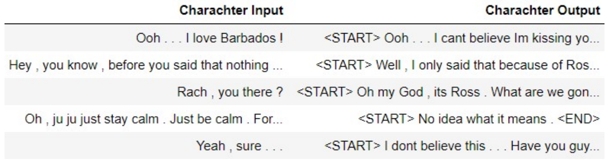
Dataset

F.R.I.E.N.D.S was chosen as the TV show for character dialogue extraction. The initial dataset consisted of the raw script of the TV show for all seasons.



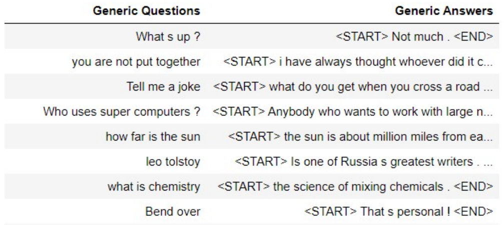
```
The One Where Monica Gets a New Roommate (The Pilot-The Uncut Version)
Written by: Marta Kauffman & David Crane
Transcribed by: guineapig
Additional transcribing by: Eric Aasen
(Note: The previously unseen parts of this episode are shown in blue text.)

[Scene: Central Perk, Chandler, Joey, Phoebe, and Monica are there.]

Monica: There's nothing to tell! He's just some guy I work with!

Joey: C'mon, you're going out with the guy! There's gotta be something wrong with him!

Chandler: All right Joey, be nice.  So does he have a hump? A hump and a hairpiece?

Phoebe: Wait, does he eat chalk?

(They all stare, bemused.)
```

The raw script was parsed to extract statement response pairs from each scene. We then collected all pairs where Joey was the responder. Joey was chosen as the character to study as he has a distinctive personality in the show. For those not familiar the show, Joey can be described as naive, sarcastic, loving, misogynistic and loud.



Charachter Input	Charachter Output
Ooh . . . I love Barbados !	<START> Ooh . . . I cant believe Im kissing yo...
Hey , you know , before you said that nothing ...	<START> Well , I only said that because of Ros...
Rach , you there ?	<START> Oh my God , its Ross . What are we gon...
Oh , ju ju just stay calm . Just be calm . For...	<START> No idea what it means . <END>
Yeah , sure . . .	<START> I dont believe this . . . Have you guy...

In additional to having Joey like personalities, we wanted our chatbot to in general respond like a human. To this end we considered pretraining our models on an additional dataset of general question and answer pairs. The goal of introducing this data set was to help the model understand generic a question answering format.



Generic Questions	Generic Answers
What s up ?	<START> Not much . <END>
you are not put together	<START> i have always thought whoever did it c...
Tell me a joke	<START> what do you get when you cross a road ...
Who uses super computers ?	<START> Anybody who wants to work with large n...
how far is the sun	<START> the sun is about million miles from ea...
leo tolstoy	<START> Is one of Russia s greatest writers . ...
what is chemistry	<START> the science of mixing chemicals . <END>
Bend over	<START> That s personal ! <END>

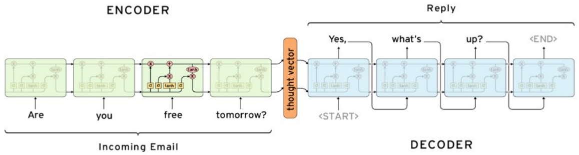
Seq2Seq

A *seq2seq model* consists of an *encoder* and a *decoder*.

Both the encoder and the decoder have an *embedding layer* that converts a word into a vector representation. Pre-trained GloVe vectors are used to set embedding layer weights. The word embeddings are fed into the encoder *long short-term memory* (LSTM) one word at a time.

The LSTM tries to capture the essence of the encoder input sequence in two state vectors that are passed to the decoder LSTM. The decoder LSTM consists of another embedding input layer and a dense *SoftMaxed* output layer. The decoder LSTM is trained using embedded output text to predict the most probable output word given two input state embeddings and an embedded word.

Once the model is trained an *inference encode decoder model* is created using the trained model weights. The inference encoder model takes in a user input question and generated two state vectors. The input state vectors are fed into the inference decoder model along with a sentence start token and all words predicted by the SoftMax layer are captured until an end token is generated.

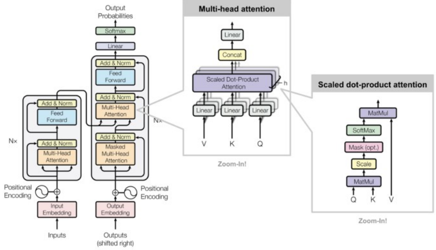


Transformer

Our second model is a *transformer model*. This model draws inspiration from the structure of the Seq2Seq model above with the additional change being the use of *transformers*. This model employs the use of *scaled dot product* attention and *multi-head attention*. The scaled attention provides a basic attention mechanism using encoder hidden states and previous hidden state of the decoder. The multi-head attention mechanism runs multiple scaled attention computations in parallel. These attentions are concatenated and linearly transformed to act as an ensembling method.

The transformer contains *encoder* and *decoder* units. The encoder generates an attention-based representation with the capability to locate a specific piece of information from a potentially infinitely large context. The encoder contains a multi-head attention mechanism followed by a fully connected feed-forward network. The decoder unit deals with retrieval from encoded representations, containing a masked multi-head attention, a multi-head attention, and a fully connected feed-forward network.

This model contains several of the above encoder-decoder units formed by transformers. A final fully connected dense layer uses decoder outputs. The loss function used is a *sparse cross entropy* loss function available in keras. The model uses the *Adam optimizer* and a learning rate decay scheduler to decrease the learning rates with respect to number of epochs.



Evaluation

The problem of evaluating a chatbot or dialogue machines is well known to be hard. The problem becomes only harder when adding the constraint that the chatbot should have some human like characteristics such as personality or memory [Liu et. al., Radziwill and Benton, Xing Fernández]. A key obstacle faced is that given any statement there could be a large number of appropriate responses. That is, there is no one ideal map from statements to responses that a chatbot would want to learn. Thus, it is hard to construct a test dataset that contains all possible valid responses for every tested statement. To efficiently evaluate, one must find a way to grade responses without being able to simply compare it against a list known answer. We made two attempts at such metrics:

Automatic Metrics: While they are known to not perform well, it is still common in the literature to attempt to use automatic metrics of some kind. Most of these metrics were originally designed to evaluate translation bots. Such bots face similar problems to those faced by chatbots; there are many appropriate translations of the same phrase for example. These metrics compare sentence structure and n-gram pattern of generated responses and reference responses to compute some overall score. We considered a few common metrics: BLEU, METEOR, ROUGE and WER.

Human Metrics: Another metric to consider is a human evaluator. This is often used as a much more informative measure of chatbot success as humans can tell if something sounds human and/or embodies a specific personality trait. For this test we asked individuals familiar with the show F.R.I.E.N.D.S. to answer a series of questions. These questions listed a statement from the TV show and asked the tester to pick which of two responses was the most Joey like. One response was generated by our model while the other was from the original script. Both responses were presented in the same format but in random order. Finally, a total percentage of selected generated responses was calculated and averaged over all testers.

Results:

Metric:	**BLEU:**	**ROUGE-1:**	**ROUGE-*l*:**	**METEOR:**	**WER Avg:**	**Human:**
Mimic (Q&A)	0.055	0.223	0.213	0.088	0.880	?
Mimic (Joey)	0.004	0.136	0.118	0.053	1.159	?
Trans (Q&A)	?	?	?	?	?	?
Trans (Joey)	?	?	?	?	?	?

Conclusion

Overall, the performance of both of the models considered was lackluster with respect to the automatic and human based metrics. As mentioned above, the poor automatic performance was expected. Namely, we understand that these metrics like to measure a similarity between some ground response and the generated response based on n-grams. It seems reasonable to expect that there would be no way to satisfy such tests consistently. We consider a few possible explanations for the poor human metric performance.

One obvious possibility is not enough training data or time spent training. Both models did seem to improve with increased training time. Hence, it seems reasonable to conjecture that both models' performance would increase somewhat with additional data and training time. Given more time it would be interesting to collect additional general conversation data what is personality neutral and use this to train both models before trying to then train to match the Joey personality. Along similar lines, one could further refine our Joey dataset to parse out less logical statement response pairs.

Another likely possibility is that neither model was sufficiently complex to capture Joey's personality. It may even be the case that the problem can't solved by any reasonable sized model. This is not completely surprising but previous success instances suggest that more complicated models may be able to have nonnegligible success. A few options for expanding the complexity of our model would be to add attention to our seq2seq model along with perhaps some additional layers trained to better predict proper English grammar.