

A Neural Chatbot with Personality

Huyenn Nguyen
Computer Science Department
Stanford University
huyenn@stanford.edu

David Morales
Computer Science Department
Stanford University
mrlsdvd@stanford.edu

Tessera Chin
Computer Science Department
Stanford University
tesserac@stanford.edu

Abstract—Conversational modeling is an important task in natural language processing as well as machine learning. Like most important tasks, it’s not easy. Previously, conversational models have been focused on specific domains, such as booking hotels or recommending restaurants. They were built using hand-crafted rules, like ChatScript [11], a popular rule-based conversational model.

In 2014, the sequence to sequence model being used for translation opened the possibility of phrasing dialogues as a translation problem: translating from an utterance to its response. The systems built using this principle, while conversing fairly fluently, aren’t very convincing because of their lack of personality and inconsistent persona [10] [5].

In this paper, we experiment building open-domain response generator with personality and identity. We built chatbots that imitate characters in popular TV shows: Barney from *How I Met Your Mother*, Sheldon from *The Big Bang Theory*, Michael from *The Office*, and Joey from *Friends*. A successful model of this kind can have a lot of applications, such as allowing people to speak with their favorite celebrities, creating more life-like AI assistants, or creating virtual alter-egos of ourselves.

The model was trained end-to-end without any hand-crafted rules. The bots talk reasonably fluently, have distinct personalities, and seem to have learned certain aspects of their identity. The results of standard automated translation model evaluations yielded very low scores. However, we designed an evaluation metric with a human judgment element, for which the chatbots performed well. We are able to show that for a bot’s response, a human is more than 50% likely to believe that the response actually came from the real character.

Keywords—Seq2seq, attentional mechanism, chatbot, dialogue system.

I. Introduction

Since the sequence-to-sequence (seq2seq) model is taught in this class CS224N, we assume that readers are already familiar with this model. For a brief introduction, a basic sequence-to-sequence model, as introduced in Cho et al., 2014 [2], consists of two recurrent neural networks (RNNs): an encoder that

processes the input and a decoder that generates the output. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence. Sequence-to-sequence is often used with attention-based that allows the decoder more direct access to the input. This model has been successfully used for many different natural language processing tasks, such as alignment, translation [1], and summarization [9].

Conversational modeling can be phrased as a mapping between utterances and responses, and therefore can benefit from the encoder-decoder setup. In our model, the encoder processes an utterance by human, and the decoder produces the response to that utterance. We train the word embeddings as we train the model. We also use attentional mechanism and experimenting with using GLoVe pre-trained word vectors to initialize our word embeddings.

To make the bot speak like a certain character, we train vector embeddings for different characters with the hope that these embeddings would be able to encode information and style of speech of these characters. These character embeddings are trained together with the word embeddings. This is inspired by Google’s Zero-shot multilingual translation system [4]. For more information, see the Method section. We write our code in TensorFlow v0.12.

A. Evaluation

To test our models, we use both automatic metrics and human judgment. For automatic metrics, we use the BLEU [7] and ROUGE [6] metrics for our model, as these are popular metrics commonly used for translation models. The BLEU metric uses a modified n-gram precision score that attempts to model a human judgment of how accurate a translation is. The ROUGE-S metric we use measures the overlap of skip-bigrams between a candidate response and a reference response, measuring how similar the model’s response is to the expected, script response.

We find that our bots can make reasonable responses

and sometimes hold decent conversations. For more details, see Results section.

B. Related Work

Building chatbots with personality or persona is pretty much still an open challenge. There have been several attempts without much success. In 2016, Li et al. at Stanford University published a paper that uses distributed embeddings to capture personas of chatbots [5]. They proposed Speaker-Addressee model to predict how speaker i would respond to a message produced by speaker j . This achieves interesting results. We deal with a different problem, the bot doesn't know who it is talking to.

II. Dataset

Initially, we used the Cornell Movie-Dialogs Corpus, created by Cristian Danescu-Niculescu-Mizil and Lillian Lee at Cornell University. This is a well-formatted dataset of dialogues from movies. It includes 220,579 conversational exchanges between 10,292 pairs of movie characters, involving 9,035 characters from 617 movies with 304,713 total utterances. We used 20,000 pairs for testing, and the rest for training.

While it is well formatted, this dataset has a huge problem which is the conversations are very dramatic and unrealistic. They deal a lot with love, violence, and murder. A sample conversation from the dataset.

```
The people at this school are so incredibly
foul.

But imagine the things he'd say during sex.

I realize that the men of this fine institution
are severely lacking, but killing yourself
so you can be with William Shakespeare is
beyond the scope of normal teenage
obsessions. You're venturing far past
daytime talk show fodder and entering
the world of those who need very expensive
therapy.
```

After training a baseline chatbot on the Cornell Movie-Dialog Corpus, we determined that the movie dialogues created a chatbot that seemed fake and over-dramatic. A sample conversation from that baseline chatbot.

```
HUMAN ++++ what do you like to do in your
free time?
BOT ++++ and what about the gun ?
HUMAN ++++ do you have a gun?
BOT ++++ yes .
HUMAN ++++ you're dangerous!
BOT ++++ i ' m not .
HUMAN ++++ do you go hunting?
```

```
BOT ++++ i can do .
HUMAN ++++ what do you hunt?
BOT ++++ virgin .
```

Moreover, erratic changes in movie situations created inconsistencies in the chatbot. So, we decided to use dialogue from TV shows whose content more closely modeled reality and whose characters spoke more like average people and less like dramatic heroes and villains. We used transcripts from *The Big Bang Theory*, *Friends*, *The Office*, and *How I Met Your Mother*.

Considering that each scene has its own topic, context, and speakers and addressees, we decided to split the dialog scripts into conversations of contiguous utterances that maintained the same aspects mentioned above. This way, utterances from a completely new conversation won't be connected as responses to the previous scene. To separate the dialog into these conversations, we introduced a separator each time the setting or scene changed, using different heuristics based on the format of the transcripts.

The transcript for *The Big Bang Theory*, uses a "Scene" character whose utterance is a description of the setting for the current scene. Such lines appear when scenes change, so we used the occurrence of the "Scene" speaker as a heuristic to split into a different conversation.

```
Penny: Well imagine how I'm feeling.
Sheldon: Hungry? Tired? I'm sorry this really
        isn't my strong suit.
Scene: The living room.
Leonard: You told her I lied, why would you
        tell her I lied?
Sheldon: To help you.
```

For *Friends*, we used a similar heuristic. Each new scene in the transcript is preceded by a description of the setting and scene inside square brackets. For that reason, we split conversations at the appearance of these square brackets.

```
Phoebe: Hey! Ooh, how was your first day
        working at the restaurant?
Joey: (checks his watch) Damn!
[Scene: Allesandros, Monica is cooking.]
Joey: (entering from the dining room) Hey.
Monica: Hey.
```

The *How I Met Your Mother* transcripts had a lot more information than we needed, so splitting conversations turned out to be more difficult. Considering this transcript's complexity, we felt it was more important to use a heuristic that might split a conversation that shouldn't have been split, instead of not splitting a

conversation that should have been split. For this reason, we considered each line in the transcript that didn't match the "Speaker: utterance" pattern as a conversation separator.

```

162,1,1,1,Marshall:...she drinks scotch?
163,1,1,1,[Flashback to Date]
164,1,1,1,Robin: I love a scotch that's old
        enough to order its own scotch.
165,1,1,1,[Flashback over.]
166,1,1,1,"Marshall: Can quote obscure lines
        from ""Ghostbusters""?"
167,1,1,1,[Flashback to Date]
168,1,1,1,"Robin: Ray, when someone asks you
        if you're a god you say, ""Yes!""
169,1,1,1,[Flashback over.]

```

The online transcript for *The Office* was already split into conversation-like blocks, so we made sure to note and maintain the same separations when scraping the transcript.

As a result of obtaining our data from public sources available online, many of our transcripts suffer from inconsistencies in formatting and accuracy, most likely due to the transcripts being created manually and possibly crowdsourced. These inconsistencies include missing punctuation, grammatical errors, and spelling errors, all of which affect our model's learning. Another problem with these datasets scraped from the Internet is that they are small. Each TV shows has at most 50,000 pairs of utterance-response, which is not enough information to train a neural net chatbot on.

For the Cornell dataset, we use 20,000 pairs for testing, and the rest for training. For the TV show datasets, due to their small size, we only use 4% of the samples for testing.

	Cornell	BBT	Friends	HIMYM	Office
Train	201,617	42,585	48,255	23,201	43,590
Test	20,000	1,545	2,056	1,055	1,893

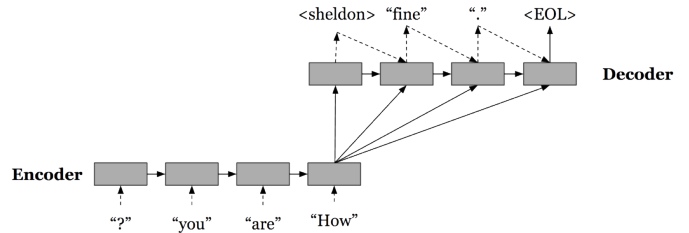
III. Method

A. The model

We use a model very closely based on Google's Neural Machine Translation model published in 2016 [12]. It's a sequence-to-sequence model with attentional mechanism that allows decoder more direct access to hidden state output by the encoder. For the RNNs, we used stacked GRU cells of 3 layers.

The encoder is the utterance by human, and the decoder is the response. We assume that in normal conversations, people listen to the first part and somewhat zone out to think of the answer, so we

reverse the encoder so that the model can retain more information from the beginning of the utterance.



Unlike other seq2seq models, our model doesn't use a start token or end token for encoders, and use only end token for decoders. We assume that the character token id is sufficient to signal the beginning of a response. We process 23 total characters in 4 TV shows, and whichever speaker not among these 24 characters is assigned the name of either 1_rando or 2_rando. Below is the list of characters:

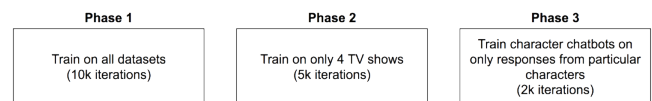
Show	Characters
BBT	1_sheldon, 1_leonard, 1_penny, 1_howard, 1_raj, 1_bernadette, 1_amy
Friends	4_monica, 4_joey, 4_chandler, 4_phoebe, 4_ross, 4_rachel
HIMYM	2_rando, 2_marshall, 2_ted, 2_barney, 2_lily, 2_robin
The Office	4_michael, 4_pam, 4_jim, 4_dwight, 4_andy

We also experiment with initializing embeddings for encoder vocabulary and decoder vocabulary using GloVe 300d Common Crawl [8]. To ensure that our model was implemented correctly, we trained it on a subset of data (3,000 pairs) and saw that the loss converged to 0, which means the model learns the dataset perfectly well.

The model greedily produces the responses by using the most likely token at each decoder step.

B. Training

Due to small amount of utterances for each character that we train on, we first train these bots on more general datasets. Our training involved 3 phases, each involving a different set of data and a different number of training iterations, where each iteration consists of 64 batch examples.



Phase 1: We train on all five of our datasets, the four TV show transcripts and the Cornell Movie-Dialogs Corpus, for 10k iterations. This initial phase is meant

for capturing basic dialog patterns between speakers and addressees.

Phase 2: We limit the training to only the four TV show transcripts for 5k iterations. By training only on the TV shows transcripts, we tune our model to be more realistic, capturing the more conversational dialog patterns found in TV shows.

Phase 3: We finish by training our character chatbots on only the utterances made by the characters they are trying to emulate, for 2k iterations. This final phase fine tunes the model to be more specific to a particular character.

C. Vocabulary and Sampled Softmax

We combine the entire vocabulary of each show. For the Cornell dataset, we only use the tokens that appear at least twice. In the end, the size of the vocabulary for the encoder is 53,589 and the size of the vocabulary for the decoder is 53,692.

These vocabulary sizes are too large to use full softmax, so we use sampled softmax, an approximation of softmax [3]. Using sampled softmax, each iteration takes about 0.2 seconds on Azure server.

D. Hyperparameters

To save unnecessary computations and to group pairs of similar encoder length and decoder length together, we use three different buckets of size BUCKETS = [(15, 15), (25, 25), (40, 40)]. This means that the first bucket takes all pairs whose encoder length is no more than 15 and whose decoder length is no more than 15 and so on.

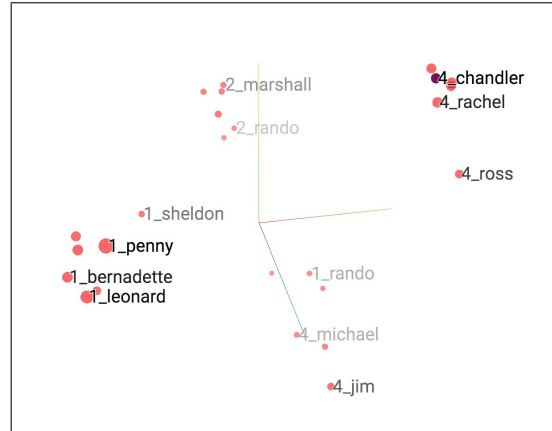
We use embedding size of 300, and the number of hidden unit in a GRU cell is 256. We use a fixed learning rate of 0.5, and clipped gradient when the norm exceeded 5.0. During training, we feed previously predicted tokens to predict the next token – this is to make the training environment similar to the testing environment.

IV. Results

To our surprise, our baseline model doesn't do very well with GLoVe pre-trained word vectors. The responses generated are full of repetitive, common words. We suspect that this is due to the pre-trained word vectors we use are trained on general vocabulary (common crawl), which is very different from the vocabulary the characters use. Therefore, we only focus on the model without pretrained word vectors.

It's clear that the system learns the embedding of characters very well. By the end of the phase 2 training, which is before we start training the bots separately, the character embeddings are already very

distinct. Characters of the same TV shows are close to each other.



The characters closest to Sheldon are most characters from the same show *The Big Bang Theory*, with the exception of Penny. This makes sense since Penny is supposed to have very different educational background from Sheldon. Fun fact, the character not in the show but talks the most like Sheldon is Ted Mosby from *How I met your mother*. This is quite interesting given that this information is not included anywhere in the training data. We only have character token id for encoders, not decoders, so the responder doesn't know who s/he is talking to.

Search by

neighbors

distance COSINE EUCLIDIAN

Nearest points in the original space:

1_amy	0.279
1_howard	0.335
1_raj	0.397
1_bernadette	0.431
2_ted	0.465
2_rando	0.525
1_leonard	0.589
2_barney	0.602
2_lily	0.665
2_robin	0.684
4_dwight	0.702

The characters also seem to learn their identities. If we ask "who are you?", each character answers this question in a distinct way.

```

Sheldon:
  > who are you?
  1_sheldon i ' m sheldon .
Barney:
  > who are you?
  2_barney i am the master of the most barney .
Michael:
  > who are you?
  4_michael i ' m just a friend .
  > what's your name?
  4_michael michael scott .
Joey:
  > who are you?
  4_joey oh , joey ' s joey ! joey ?
  
```

They also tend to respond differently in regular conversational dialogue.

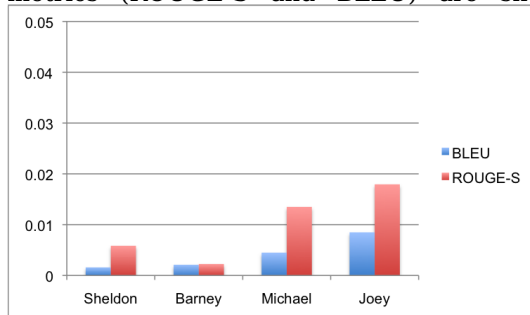
```

Sheldon:
  > let's go.
  1_sheldon what are you doing?
Barney:
  > let's go.
  2_barney what?
Michael:
  > let's go.
  4_michael ok.
Joey:
  > let's go.
  4_joey okay.
  
```

This is quite remarkable since none of these exact utterance-response pairs are in the training sets. In the training set, Michael says "I am Michael Scott" several times, but never "Michael Scott". Barney says "I am the master of impossible." but never "I am the master of the most barney." The bots made the responses up themselves! For more sample conversations, see the Appendix section.

A. Evaluation Metrics

Results for both the automated evaluation metrics (ROUGE-S and BLEU) are shown below:



This automated evaluation of our chatbots produce very low scores. We suspect this could be for several reasons. First, these metrics assume that the reference translation is in some sense the "right" answer, and penalize hypothesis translations for being different. However, the main purpose of our chatbots is to hold a conversation with a particular character. Especially in the sense of carrying a dialogue, there are often infinitely many responses that would be appropriate, even within the constraints of one personality. Therefore, we feel that these metrics, though fast and automated, do not accurately represent the success of our chatbots.

In light of these findings, we devised a custom set of tests that allowed our bots' responses to be presented to human judges. The test was created as follows: given some conversation utterance, we present the judge with a candidate response. The judge then indicates whether they believe the response was from a chatbot, or from the actual TV show's script.

We created a test for four of our personality chatbots: Sheldon (*BBT*), Joey (*Friends*), Michael (*The Office*), and Barney (*How I Met Your Mother*). For each character, we picked 20 random test utterances from the corpus, then labelled 10 with the gold response from the corpus and 10 with the chatbot's output.

```

Sheldon
Input: What for?
Candidate: I have to go to the bathroom.
{Truth: bot}
Input: Hey, what's the matter?
Candidate: My equations, someone's tampered with my equations.
{Truth: real script}
Barney
Input: Just shut up and eat.
Candidate: All right.
{Truth: bot}
Input: That's the stupidest thing I've ever heard. That's not real.
Candidate: You're right, Ted. I'm just making that up.
{Truth: real script}
  
```

We were able to administer this custom test to n=12 subjects, results shown below.

Michael, *The Office*

		Human judgment	
		Bot	Real
True class	Bot	56	61
	Real	47	66

Joey, *Friends*

		Human judgment	
		Bot	Real
True class	Bot	53	62
	Real	52	60

Sheldon, *Big Bang Theory*

		Human judgment	
		Bot	Real
True class	Bot	63	70
	Real	43	52

Barney, *How I Met Your Mother*

		Human judgment	
		Bot	Real
True class	Bot	44	66
	Real	61	56

From these confusion tables, we gain insight into how our bot is performing. False positives, or examples where the response is gold, but the human judge marks it as a bot, don't say much about our bot itself. Rather, it is more likely that these discrepancies are caused by lack of conversational context.

The false negatives represent the test examples where the response was actually from the chatbot, but human judges thought it was from the script. These are the instances where our chatbot performs well, when it provides a conversational response that sounds as if it were coming from the real character. Our chatbots for Michael, Joey, Sheldon, and Barney received false positive rates of 52%, 53.9%, 52.6%, and 60%, respectively. This shows that for all of the bots, human judges were more than 50% likely to believe that the bot's response actually came from the real character.

In addition to quantitative results, the human test provided insight into qualitative aspects of our chatbots. As subjects were deciding whether the candidate response was from the chatbot or not, many expressed feelings of ambivalence and difficulty in making a judgment. We also observed that the chatbot often fails at generating longer-length responses. Often, it will create grammatical errors or begin to loop and repeat itself. Conversely, longer responses in the script are much more coherent, and more commonly found.

Overall, these are encouraging results, since it means that for the bot's responses, on average, there were no dead giveaways, and humans could do no better than guessing at whether the response was real or from the chatbot.

V. Conclusions

This model demonstrates that it's possible to encode certain aspects of the personality and identity of a character in the chatbot. The model is trained entirely end-to-end with no handcrafted rules. The model is

scalable. Using the generic datasets (Cornell and all TV shows) to train phase 1 and phase 2, we can use train phase 3 on a much smaller dataset to make the bot talk like a certain character.

While the bots can speak reasonably fluently, they fail to take into account information given in the past, since we only predict response from one single utterance. The next step would be to use previous exchanges as part of the encoder. This will be more like a summarization problem: the encoder would be much longer than the decoder.





The TV show dialogues scraped from Internet are of poor quality, and we had to use heuristics to separate conversations, which affected the performance of the bots. The lack of reliable evaluation metrics also made it hard for us to optimize. We couldn't tell if the chatbot was getting better or not. Future work in this direction may include designing a more scalable human-test approach, or modifying existing evaluation metrics to be more tailored to general dialogue systems.

References

- [1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
- [2] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).
- [3] Jean, SAlbastien, et al. "On Using Very Large Target Vocabulary for Neural Machine Translation." arXiv preprint arXiv:1412.2007 (2015).
- [4] Johnson, Melvin, et al. "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation." arXiv preprint arXiv:1611.04558 (2016).
- [5] Li, Jiwei, et al. "A persona-based neural conversation model." arXiv preprint arXiv:1603.06155 (2016).
- [6] Lin, Chin-Yew. "ROUGE: A package for automatic evaluation of summaries." Text summarization branches out: Proceedings of the ACL-04 workshop. Vol. 8. 2004.
- [7] Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.
- [8] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.
- [9] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).
- [10] Vinyals, Oriol, and Quoc Le. "A neural conversational model." arXiv preprint arXiv:1506.05869 (2015).
- [11] Wilcox, Bruce. ChatScript. GitHub Repository. (2017). <https://github.com/bwilcox-1234/ChatScript>
- [12] Wu, Yonghui, et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." arXiv preprint arXiv:1609.08144 (2016).

VI. Appendix

Sample conversations:

Michael (The Office) Rude, abrupt Thinks he's the best boss 	Barney (HIMYM) Very bro-y Likes to hit on women Drinks a lot 	Sheldon (BBT) Socially awkward Thinks other people are stupid 	Joey (Friends) Not very smart but kind A player 
<p>> hi 4_michael have a seat . > who are you? 4_michael i ' m just a friend . > what's your name? 4_michael michael scott. > do you have a girlfriend? 4_michael i have a lot . > tell me a joke 4_michael no , i ' m not gonna tell you . > you are boring. 4_michael you are . > where do you work? 4_michael right here . > tell me michael. why do you work here? 4_michael i thought i had to do something better . > bye bye michael 4_michael bye .</p>	<p>> who are you? 2_barney i am the master of the most barney . > hi barney! 2_barney hi , have a drink . > oh thanks. what drinks do you have? 2_barney whatever what do you do ? > i'm a student. 2_barney a . . . buddy . > yeah? 2_barney sleep with her mom ? > lol no 2_barney because she ' s not happy . > ha ha okay. how's work? 2_barney okay . > just okay? 2_barney oh , okay . > what do you think of that girl over there? 2_barney do not even try to try to that girl . > noooooooooo 2_barney legendary !</p>	<p>> hi 1_sheldon hello . > who are you? 1_sheldon i ' m sheldon . > what's up? 1_sheldon i see how i could do . > what can you do? 1_sheldon i have to tell you a secret . > tell me. 1_sheldon no . > what do you like to do for fun? 1_sheldon i want to leave . > to go home? 1_sheldon no . > how are you doing? 1_sheldon i just wanted to sleep . > sleep! 1_sheldon penny , i am getting too much for you .</p>	<p>> hey what's up 4_joey hey ! > how are you? 4_joey oh , oh , i ' m fine ! > what's your name 4_joey joey , joey . > who are you? 4_joey oh , joey ' s joey ! joey ? > joey joey joey 4_joey what ? > tell me a joke. 4_joey a woman . > a woman is not a joke! 4_joey oh , i am very sorry .</p>