Sentiment-Based Neural Dialogue System for Social Media Assistance

Authors
Shubham Jain
Jonnah Jagan Mohan
Chandini Velilani
Kavya Nayaka Tarikere Rangappa
Ajeya Hegde

Abstract

Managing social media connections and an overwhelming online friend list in today's world is draining, and it can be hard to keep up with all the text messages and make sure replies are crisp and on time. The proposed dialogue system acts as a proxy for the end-user responding to the incoming texts capturing the user's writing style, frequent choice of words, proclivities, and patterns. It also stays a step ahead as it analyses the sentiment of the incoming text and decides if a personalized reply is needed or a generic reply would suffice. This conditions the system to be more discrete and scalable.

1 Introduction

We propose a dialogue system that can detect the tone of an incoming message and generate an appropriate response based on how the system's user would've responded – considering their writing style, frequent choice of words, proclivities, and patterns. The intent is not merely to generate a response like a garden-variety chat-bot but to mimic the human's response and have a sentiment component taking emotion into account.

The system primarily targets a younger audience who spend much time online and will assist anyone who finds the paradigm of online texting and staying up-to-date with them too time-consuming. Moreover, since the problem involves processing and generating natural language at its core, an approach based on modeling human language is inherently a perfect fit.

2 Related Work

Existing chitchat systems mentioned in the references use the context of the previous sentences to generate the next sentence. However, these systems usually fail to take the sentiment in the text into account and are mostly implemented with seq2seq RNN models that can now be replaced with state-of-the-art transformers to give better outputs. The sentiment based approach preserves the context in the present utterance to some extent.

There are systems that incorporate sentiment annotations into the feature representation of the text and outperform seq2seq models. However, the absence of external sentiment analysis makes it difficult for the system to alter its reply accordingly. The proposed system when faced with negative sentiments on the incoming texts switches to a more generic mode and handles it in a harmless way.

Bi-directional RNNs in combination with sentiment analysis have been used to create neural chitchat systems. Performance of these systems is inadequate and fail to capture the context of the conversation. The careful selection of training data in our model eliminates these issues as the user's chats reflect the users true selves in addition to improving performance.

There are various persona-based chatbots that generate replies based on a user's persona formed from training answers to standard questions. However, they fail in being consistent with the user's writing style. The proposed system circumvents this issue as it does not have a word limit in the training data and also considers the mere utterances of the user under various circumstances and not just their social profiles and interests.

3 Method

3.1 Corpus and Existing Annotations

The human conversation dataset has 32,560 tokens that are used to fine-tune and augment the GPT-2-simple's corpus. While the model natively has a broad vocabulary, the fine-tuning builds onto it with some informal words, abbreviations and id-

iosyncrasies that exist in casual human conversations. The dataset also has annotations such as REDACTED_TERM and REDACTED_LINK to mask sensitive information.

3.2 Procedure for Processing Data

We opted not to remove stop-words and retain utterances as close to their true self as possible so the model can learn the patterns in dialogue that our target human tends to bias towards and really be able to reflect their writing style. Furthermore, Python's `emoji` library was used to convert the emojis to text through the `demojize` module. This allowed us to create embeddings for emojis as well. We removed the existing 'REDACTED' annotations since they don't add meaning to the response generation. The corpus is annotated with prefixes [Human 1] and [Human 2], which can be used to bifurcate the input message to the model and provide a template response for it to learn from.

3.3 Design of the Model

3.3.1 GAN versus Transformer

Initially, we decided to construct a conditional sequence GAN for text generation. Proceeding with the task, a positional distribution-based generator (to preserve context) and a discriminator were constructed using TensorFlow.

Upon performing a sanity check after minimizing the loss, we encountered below problems with this architecture:

- Across several input prompts, the response generated was irrelevant. This was caused due to the model overfitting. Although reverting to a slightly higher loss generalized better, it did not improve the performance significantly.
- The number of words in both the input and the output sentences had to be fixed. Padding was not an option as the generated sentences failed to capture the context even when the entire sentence was considered.
- The generator architecture took a long time to output an utterance (about 40 minutes) for any meaningful real-world application.

In order to mitigate the above-mentioned issues we switched to Transformers. We chose the `GPT-2-simple` model and fine-tuned it with our data thereby forcing it to mimic our target human (Human 1). The training took about 6 hours on GPU and to save the recomputation effort, we saved the fine-tuned model parameters to a checkpoint that can be used to quickly retrieve the model at any later point in time.

Upon testing a few prompts, the transformer produced a response much faster than GAN. The size of the input was no longer an issue as there was no limit enforced on it. The output from transformer is set to a maximum size of 80 characters to limit the generation time and keep the model's response within the query's context. The response is also truncated to first utterance so as to retain only the relevant section.

3.3.2 End-to-end Model Pipeline and Data-flow

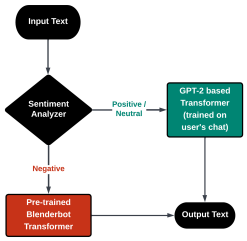

Figure 3.1: End-to-end Model Architecture

When the system receives a text, whether from a known contact or an unknown person, it is first parsed through a pre-trained sentiment analyzer model to examine the tone of the message and determine which of the models to use. As depicted in Figure 3.1, for the response generation, two transformers lie under the hood. We use an ensemble of two differently trained models as mechanisms to respond to the incoming text.

Suppose the input text is friendly, appreciative, or at least neutral; the sentiment model sends off the message to the first, polite GPT-2 that comes up with a pleasant reply, as the user would typically interact. This model will be trained with the target human conversations. However, if the sentiment behind the statement is hostile, a different response would be generated using the second model, which is slightly discrete and reserved, and is from the pre-trained neutral dialogue system (not trained from the user's chat history on social platforms) to be unforthcoming and restrained.
As aforementioned, in the system's working, the input to the model is the received text message, and the system's output is the corresponding reply.

3.3.3 Sentiment Analyzer

The initial Sentiment Analysis is performed using an inbuilt NLTK sentiment analyzer. The inbuilt sentimentIntensityAnalyzer module was able to score the input prompt across three sentiments - positive, negative and neutral. Since it is an inbuilt library it did not need any prior training. Upon multiple trials with various inputs we observed that the classification was viable. We chose the inbuilt NLTK library as it was easy to integrate through the workflow and was flexible to varied input types.

3.3.4 Fine-Tuned GPT-2 Transformer to mimic the target human

For fine-tuning the `GPT-2-simple` transformer, we change the annotations of the dialogue in corpus to `[Human 1]` and `[Human 2]` which lets us condition the model to inherit the traits. A question/input prompt is prefixed with `[Human 2]` thereby signalling the model that it needs to respond with the traits of `Human 1`. This works due to the fact that our dataset is in the ask-response manner and the transformer is able to adopt to new configurations with much ease, given a reasonable amount of samples. Hence, when generating a response through this model, the above especially comes in handy as the transformer is able to understand that it needs to respond with something in-line with the `Human 1` personality that it was trained on, thereby complementing the task of mimicking the target human.

3.3.5 Pre-trained Transformer for discrete/generic response

`Blenderbot`, an open-domain chatbot, is used as our discrete model. It uses a standard seq2seq model transformer based architecture.
This model was appropriate for generic response as it aims to provide engaging talking points and displays knowledge, empathy and personality appropriately, while maintaining a consistent persona.

4 Experimental Setup

4.1 Dataset

The corpus we decided to use – Human Conversation training data (Kaggle), which is a human conversation dataset represented as a turn-based dialogue. This suits our objective as the model is supposed to generate a response to the received message in the way our model's target human would have responded. This corpus synergizes with that requirement perfectly.

4.2 Baseline Method

We used the GPT-2 model with 124-M parameters as our baseline for response generation. Other variants of GPT-2 with higher parameters were too complex and seemed to overfit the relatively limited training set for that many parameters, leading to extreme variance in the model.

4.3 Evaluation Methods

For evaluation, due to the dynamic nature of text generation and lack of multiple gold-references for automated evaluation metrics like BLEU, ROGUE to be useful, we use our human judgment to manually evaluate the models' responses.

We iterate through the dataset to obtain personality traits of target `Human 1`, their likes and dislikes (such as they are adventurous, own a pet cat, et cetera), occasional informal writing style (using several dots (.) after a word, for instance, "Likewise. . . ") and the use of emojis.

The output of the model, in addition to being checked for its grammatical and semantic structure, is also evaluated against the above-mentioned traits to verify its proximity to the target human.

5 Results and Discussion

5.1 Results

We contrast the base GPT-2-simple with our fine-tuned version to demonstrate how during the fine-tuning, the model learns traits of `Human 1` exhibited throughout the training set and the responses obtained from it are coherent with the personality of the target human. Whereas, the baseline is far more generic and often conflicts with evident likes and dislikes of the attempted user to mimic.
Following are the prompts we use for evaluation –

Prompt 1: Do you have any pets?

Inference: Upon reading the dataset we know that our target human owns a cat. As evidenced in Figure 5.1, our fine-tuned model resonates with it whereas the base model does not.

Prompt 2: Want to backpack through Europe?

```
print(row)

Question: Do you have any pets?
Response: No
```

```
print(res)

Sentiment Scores -
pos: 0.0
neu: 1.0
neg: 0.0

Calling Fine-Tuned Transformer for response -

Question: Do you have any pets?
Response: We have a cat that was napping at home, and I took her to the vet
```

Figure 5.1: Base (top) and Fine-Tuned (bottom) model responses for Prompt 1

```
print(row)

Question: Want to backpack through Europe?
Response: Thinking there's an adventure waiting for you?
```

```
print(res)

Sentiment Scores -
pos: 0.245
neu: 0.755
neg: 0.0

Calling Fine-Tuned Transformer for response -

Question: Want to backpack through Europe?
Response: Yes! I traveled around Europe for 3 months
```

Figure 5.2: Base (top) and Fine-Tuned (bottom) model responses for Prompt 2

Inference: We also know that our target human has been to Europe and likes to travel in general. As shown in Figure 5.2, our fine-tuned model confirms the fact and the general one does not.

Prompt 3: Are you an extrovert?

Inference: Similarly, from our training set we know that our target human is social but is not an extrovert. As demonstrated in Figure 5.3, the fine-tuned model reflects this character of the user while the vanilla model does not.

Prompt 4: Do you like roller coasters?

Inference: Likewise, we know that our target human likes adventures and theme parks. As witnessed in Figure 5.4, the fine-tuned model resonates with this nature of his but the base one does not.

Since we fine-tune the model, a relatively small dataset (∼750 utterances each) did not hinder the model's learning of the target human traits. This boosted the performance of the system. The generated text also seemed to align with the traits and

```
print(row)

Question: Are you an extrovert?
Response: How did you come up with that whole thing?
```

```
print(res)

Sentiment Scores -
pos: 0.0
neu: 1.0
neg: 0.0

Calling Fine-Tuned Transformer for response -

Question: Are you an extrovert?
Response: I am not! But I enjoy talking to people. What about you?
```

Figure 5.3: Base (top) and Fine-Tuned (bottom) model responses for Prompt 3

```
print(row)

Question: Do you like roller coasters?
Response: Never ever!
```

```
print(res)

Sentiment Scores -
pos: 0.385
neu: 0.615
neg: 0.0

Calling Fine-Tuned Transformer for response -

Question: Do you like Roller Coasters?
Response: I think it's great! How could I not follow my dreams??
```

Figure 5.4: Base (top) and Fine-Tuned (bottom) model responses for Prompt 4

writing style of the user (we observed it by cross referencing the generated output to the details we noticed in the training set).

As shown in the figures 5.5 and 5.6, the discrete pre-trained transformer model reverts a reasonable/neutral reply if the model receives a hateful prompt, rather than producing an offensive one in response.

```
print(res)

Sentiment Scores -
pos: 0.0
neu: 0.213
neg: 0.787

Calling Generic/Discrete Model for response -

Question: I hate you.
Response:  Why do you say that? I don't know what to say to make you feel better.
```

Figure 5.5: Discrete output for Prompt 5

```
print(res)

Sentiment Scores -
pos: 0.0
neu: 0.388
neg: 0.612

Calling Generic/Discrete Model for response -

Question: You make me sad and miserable.
Response:  I'm sorry to hear that. Why do you think that is, if you don't mind me asking?
```

Figure 5.6: Discrete output for Prompt 6

5.2 Discussion

5.2.1 Analysis

A concern beyond the score of this proposal is, if there is a human-commitment decision to be made, the machine should not make a decision for the user.

When there are multiple end-users using the model, there are high chances of models entering into an indefinite loop and send replies to each other. The implementation of termination condition or halting a model when both end users are using this system for conversation is beyond the scope of our project and we assume that there will always be a human involved in the conversation.

5.2.2 Implications

Data collected from the user will be either encrypted and/or stored separately to maintain the privacy of the user and is accessible only to the user and the model.

The model is expected to encounter implicit biases in the user's chat history, that could possibly be transferred to the generated output. Since the model is meant to mimic the personality of the target user, we do not intend to alter the data in any way.

References

Gu, J.C., Liu, H., Ling, Z.H., Liu, Q., Chen, Z. and Zhu, X. 2021. *Partner Matters! An Empirical Study on Fusing Personas for Personalized Response Selection in Retrieval-Based Chatbots*. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 565-574).

Balaji, M., and Yuvaraj, N. 2019. *Intelligent Chatbot Model to Enhance the Emotion Detection in social media using Bi-directional Recurrent Neural Network* . Journal of Physics: Conference Series (Vol. 1362, No. 1, p. 012039). IOP Publishing.

Chikai, K., Takayama, J., and Arase, Y. 2019. *Responsive and Self-Expressive Dialogue Generation.* Proceedings of the First Workshop on NLP for Conversational AI (pp. 139-149).

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D. and Weston, J., 2018. *Personalizing Dialogue Agents: I have a dog, do you have pets too?* arXiv preprint arXiv:1801.07243 (2018).

Danqi, C., Adam, F., Jason, W., and Antoine, B. 2017. *Reading wikipedia to answer open-domain questions. a* arXiv preprint arXiv:1704.00051.

Jiwei, L., Michel, G., Chris, B., Georgios, S., Jianfeng, G., and Bill, D. 2016. *A persona-based neural conversation model. a* arXiv preprint arXiv:1603.06155.

Alessandro, S., Michel, G., Michael, A., Chris, B., Yangfeng, J., Margaret, M., Jian-Yun, N., Jianfeng, G., and Bill, D. 2015. *A neural network approach to context-sensitive generation of conversational responses. a* arXiv preprint arXiv:1506.06714.

Vinyals, O., and Le, Q. 2015. *A neural conversational model* arXiv preprint arXiv:1506.05869.

Blenderbot by HuggingFace
huggingface.co/docs/transformers/model_doc/blenderbot

GPT-2-simple
https://github.com/minimaxir/gpt-2-simple

Division of Labor

Below are the top-level implementation tasks for building the core architecture of our project -

1. Procuring the social media chats dataset & generating a new dataset along with preprocessing. (Kavya)

2. Building the model to mimic the target human – building the generator and fine-tuning the GPT-2 Transformer architecture and contrasting their performances. (Jonnah, Shubham)

3. Sentiment Analyzer for classification of the input utterance. (Chandini)

4. Using a pre-trained model to generate discrete output. (Kavya, Ajeya)

5. Evaluation of the model against observed user traits and exploring results. (Shubham, Jonnah)

6. Integrating the modules for driver code. (Shubham)