Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Xinman Zhang April 13, 2021

Automatic Generation of Emotion-based Responses to User Profiles

by

Xinman Zhang

Jinho Choi
Adviser

Department of Computer Science

Jinho Choi

Adviser

Manuela Manetta

Committee Member

Davide Fossati

Committee Member

2021

Automatic Generation of Emotion-based Responses to User Profiles

By

Xinman Zhang

Jinho Choi

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Computer Science

2021

Abstract

Automatic Generation of Emotion-based Responses to User Profiles
By Xinman Zhang

While existing conversational agents are able to generate grammar-correct responses, the responses are often lack of personal expressions when users start to talk about themselves. This thesis aims to improve the engagement and naturalness of responses when responding to user profiles, such as personality traits, preferences, and experiences. Our main idea is to embed the emotion factors into the response generation task given user profiles. We first developed a profile classifier to identify whether a sentence is a profile or not. We also developed an emotion classifier with 32 emotion labels, which allows us to detect the emotion of any sentence. Then, we posted profiles extracted by the profile classifier and collected corresponding responses in two parts including exclamation and follow up. Using this dataset, we presented two models that concatenated the response emotion predicted by the emotion classifier either before or after the given utterance. The result shows that concatenating the desired emotion before the given utterance generates better results for the exclamation prediction task, and concatenating the desired emotion after the given utterance generates better results for the follow up prediction task. Overall, the generated responses are encouraging. Since the profile classifier allows us to extract profiles from a large dataset, it will be easy to generate more input data to the model and thus improve the quality of responses in the future.

Automatic Generation of Emotion-based Responses to User Profiles

By

Xinman Zhang

Jinho Choi

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Computer Science

2021

Acknowledgements

First, I would like to express my thanks to my advisor Dr. Jinho Choi. I came to Dr. Choi in my junior year. Dr. Choi admitted me to the Emory NLP lab, guided me on conducting research, and inspired me on the meaning of learning. Having Dr. Choi as my advisor has greatly shaped me in the way of learning Computer Science.

Also, I want to express my thanks to my committee members, Dr. Manuela Manetta and Dr. Davide Fossati. They not only gave great suggestions for my honors thesis but also showed consistent support during my undergraduate study at Emory University.

Further, I want to give my thanks to Sarah Finch, a third-year Computer Science Ph.D. student. Starting from January 2021, I have been working closely with Sarah on the profile detection task and the emotion-based response generation task. Sarah advised me on different approaches and helped me with problems that came up during my research.

Last but not least, I want to give my thanks to my parents and my friends who have shown me great support during this special time of COVID-19.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Response generation in dialogue systems has achieved remarkable progress in recent years. Many conversational agents are specialized for a specific domain such as sports [1]. However, when the user starts to talk about themselves, which is a much more open domain, the responses generated by the conversational agent are often generic, non-captivating, and lack of personal expressions. They learn from collections of responses and generate one based on a given utterance without considering the context of the dialogue and the inner motivation of a response.

The main purpose of this paper is to improve the engagement and naturalness of responses when responding to a user profile, which refers to the backstory of a user, including elements like personality traits, preferences, and experiences. Our main idea is to endow automated dialogue agents with the ability to perceive and express emotions. More specifically, we want the

automated agents to be conscious of the user's emotional state change and also respond with an appropriate emotion. For example, when the user says 'I play jazz piano in a band', we want the agents to know that the user is saying this proudly and to respond with an appropriate emotion like 'impressed', such as 'That's impressive! How long have you been playing for?'

Our main goal is to automatically generate emotion-based responses to user profiles. We tackle this problem in three steps. Firstly, we developed a BERT-based profile classifier using a customized dataset to identify whether a sentence is a profile or not. This was used for us to extract profile statements as the input utterances in our response generation task. Secondly, we developed a RoBERTa-based emotion classifier using a dataset containing 32 emotion labels. It allows us to predict the emotion of any given sentence. Then, we collected a crowd-sourced profile-response dataset from Amazon Mechanical Turk and performed several data analyses including categorization and similarity. Using this novel dataset, we developed two models for the response generation task using RNN and Transformer, respectively, and embedded the desired emotion as another token into the input profile sequence.

In this paper, Chapter 2 addresses related literature review and introduces all Natural Language Processing models that we used. Chapter 3 develops the

BERT-based profile classifier, and Chapter 4 develops the RoBERTa-based emotion classifier. Chapter 5 focuses on the emotion-embedded response generation tasks. Chapter 6 discusses potential future improvement of this work. Chapter 7 concludes the paper.

Chapter 2

Background

2.1 Related Work

Vinyals and Le [12] presented the Sequence-to-Sequence approach used for automatic response generation tasks. Trained with a large conversational dataset, the model is able to predict the response when given a sentence. This model serves as a baseline for automatic response generation tasks using the Seq2Seq model. Based on this, many works are devoted to improving the quality of the response. For instance, Zhang [14] endowed the automated agents with personas to engage the user with personal topics. Wu and Wei[13] proposed a prototype-then-edit model for response generation, which first generates a prototype response and then edits it to apply to the current context. Huang [4] concatenated the desired emotion of an utterance into the encoder and the decoder to express emotion in the response.

2.2 Models

2.2.1 Seq2Seq Model With RNN

Recurrent Neural Networks (RNN) [3] is a commonly used model for sequential data. The RNN model takes the input data one by one at a time in a sequence. At each step, it performs calculations on the input element and produces the output which is known as the hidden state. This hidden state is then combined with the next input element in the sequence to produce the second hidden state. The process continues until it reaches the last input element. In this way, the final result is dependent on all the previous inputs. In other words, the model has the ability to remember the inputs. Due to its internal state memory, RNN is perfectly suited for tasks related to sequential data like text.

A Sequence to Sequence network [10], usually referred to as Seq2Seq, is a model consisting of two RNNs called the encoder and the decoder. The encoder reads an input sequence and outputs a special token. The decoder then reads that special token and outputs a sequence. The idea of having two RNNs together rather than a single RNN frees us from sequence order. Therefore, this structure is commonly used for tasks related to two sequences of text.

2.2.2 Transformer

The essence of the Transformer [11] model is Attention. The Attention mechanism looks at an input sequence and decides at each step which other parts of the sequence are important. The idea is natural to human beings. For instance, when you are reading a sentence, you not only focus on each word you read but also are aware of the context of the whole sentence.

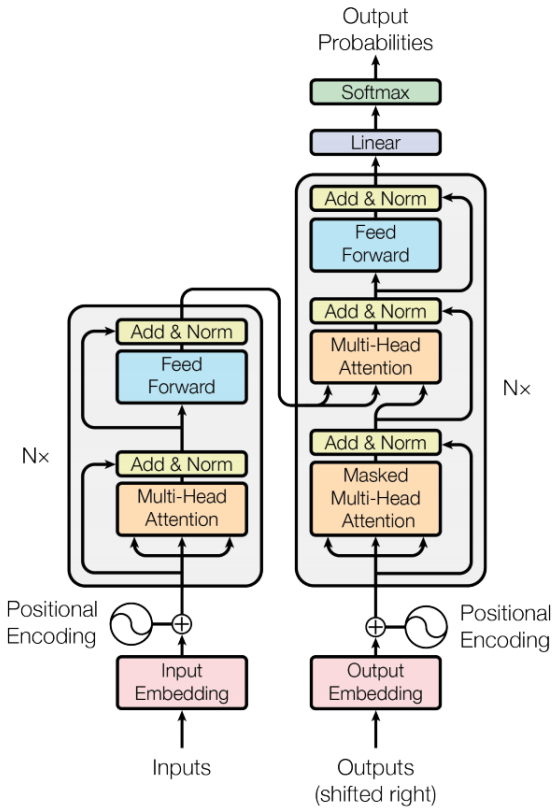

Figure 2.1: Transformer architecture [11]

As we could see from Figure 2.1 [11], the Transformer model also consists of the encoder and the decoder. The modules inside the encoder and decoder structure can both be stacked on top of each other multiple times. The encoder consists of a Self-Attention layer and a Feed-Forward layer. The decoder also has another layer of Encoder-Decoder Attention layer which helps the decoder to focus on appropriate parts of the input. Another unique thing about Transformer is positional encoding. The aim of this is to save the position of each word in the sequence when it is fed into the model.

Overall, the Transformer model has the ability to remember the hidden information of each element in the input using the Attention mechanism and achieves fast processing of the input sequence using multiple attention heads in parallel.

BERT (Bidirectional Encoder Representations from Transformers) [2] is a variant of the Transformer model introduced by Google. It applies the bidirectional training of Transformer on a great amount of unlabeled text data, which allows the model to have a deeper sense of understanding of language context and flow. It outperforms most currently-existing models on many NLP tasks. Therefore, the BERT-based Transformer has become the model of choice for text-related tasks.

RoBERTa [5], released by Facebook, further improves the training methodology and optimizes the approach of BERT. It is shown to have a better performance than BERT.

Chapter 3

Profile Detection

To be able to extract profile statements from a large amount of data, we first need to develop a profile classifier to identify whether a given statement is a profile or not. This chapter introduces a BERT-based profile classifier.

3.1 Dataset

To train the binary profile classification model, we need a dataset containing statements and the corresponding profile-like tag, indicating whether or not the statement is a profile. In other words, we need a dataset containing both profile statements as positive samples and non-profile statements as negative samples. Since there is no existing dataset that well satisfies our needs, we created our own customized dataset using Persona Chat[14] and Reddit Comments.

3.1.1 Persona Chat

The Persona Chat[14] dataset is released by Facebook. The dataset contains 10,907 multi-turn dialogues conditioned on personas. Each dialogue is performed by two crowd-sourced workers assuming some personas, which are described by 3 to 5 profile sentences, such as "I am very athletic", "I wear contacts", and "I hate carrots". In total, there are 5,578 unique profile statements (4,710 for training, 450 for validation, and 418 for testing), and we took all of these profile statements as positive samples for our customized dataset.

3.1.2 Reddit Comments

We already have the positive samples extracted from the Persona Chat dataset, and we need to generate a set of negative samples for our customized dataset, which are non-profile statements.

The Reddit Comments dataset is a pre-existing dataset containing comments posted by users on Reddit. The dataset contains 217,423 comments. Given the fact that the positive samples from Persona Chat are normally sentences with at least 4 words, we also selected sentences with at least 4 words and at most 20 words from the Reddit Comments dataset to improve

the consistency of our customized dataset. After this step, we filtered out 84,316 noisy comments.

As a way to roughly distinguish profile-like Reddit comments and non-profile-like Reddit comments and to extract non-profile-like comments for our customized dataset, we borrowed the following four rules introduced in the paper 'Training Millions of Personalized Dialogue Agents' [6].

- Each profile-like comment should contain at least 4 words or punctuation marks.
- Each profile-like comment should contain 'i' or 'my'.
- Each profile-like comment should contain one verb.
- Each profile-like comment should contain at least one noun, pronoun, or adjective.

After applying the four rules, we obtained 42,701 profile-like comments and 90,406 non-profile-like comments. Those non-profile-like comments are candidates for negative samples in our customized dataset.

3.1.3 My Customized Dataset

In total, we have 5,578 positive samples (4,710 for training, 450 for validation, and 418 for testing) from the Persona Chat dataset and 90,406 negative samples from the Reddit Comments Dataset. As we notice, the size of negative samples largely exceeds the size of positive samples. Since the balance of negative and positive samples is crucial for a binary classification task, we randomly selected the same number of negative samples and appended them with the positive samples. We then attached a label of 1 with positive samples and 0 with negative samples. In this way, we obtained our customized dataset. The size is shown in Table 3.1.

Size	Training Set	Validation Set	Testing Set
Profile statements	4,710	450	418
Non-profile statements	4,710	450	418
Total	9,420	900	836

Table 3.1: Size of our customized dataset for the profile classifier

The customized dataset is ready to use for our profile detection task. Table 3.2 is a sample of our customized dataset.

Label	Sentence
1	I like to go hunting.
1	My favorite holiday is halloween.
1	I have four sisters.
1	I work as a stand up comedian.
1	I come from a small town.
0	He's also very appearance driven.
0	It's something you should be avoiding though.
0	You make me sad sir.
0	It took me a minute to get this.
0	The fight s already over.

Table 3.2: Sample of our customized dataset for profile classifier

3.2 Approach

We chose to use the BERT-based Transformer model introduced in Section 2.2.2 because of its outstanding performance on classification tasks. The input sentences are all cleaned and preprocessed. The label is encoded into 1 or 0.

3.3 Experiments and Evaluation

We fine-tuned the model by changing the hyper-parameters. The accuracy with different max sequence lengths and batch sizes is shown in Table 3.3 and Table 3.4. The accuracy of label 1 and label 0 is also shown. The combination of hyper-parameters with the best accuracy is shown in bold.

Max Seq Length	Accuracy of 1	Accuracy of 0	Overall Accuracy
25	98.57%	99.28%	98.93%
50	**98.58%**	**99.76%**	**99.17%**
75	99.05%	99.28 %	99.16%

Table 3.3: Accuracy of the profile classifier with different max sequence lengths

Batch Size	Accuracy of 1	Accuracy of 0	Overall Accuracy
32	97.66%	100.00%	98.83%
64	98.58%	99.52%	99.05%
128	**98.58%**	**99.76%**	**99.17%**
256	97.42%	99.51%	98.47%

Table 3.4: Accuracy of the profile classifier with different batch sizes

3.4 Results and Analysis

As we could see from the evaluation results, the accuracies on the testing set in all settings are above 98%. It indicates that the model is doing extremely well on identifying whether a statement is a profile or not. Because of this high accuracy, we would want to perform some further evaluation to make sure it does reflect the true performance of the classifier. Therefore, we decided to run the model on the 42,701 profile-like comments obtained from the Reddit Comments dataset using the four basic rules in Section 3.1.2. After we applied the profile classifier on this set, we obtained 12,117 predicted profile statements and 30,584 predicted non-profile statements. The distribution of each prediction is shown in Table 3.5.

Prediction	Number	Percentage
True	12,117	28.38%
False	30,584	71.62%
Total	42,701	100%

Table 3.5: Profile classifier prediction on profile-like comments from Reddit Comments

The sample prediction result is shown in Table 3.6.

	Sentence	Pred	Manual
1	I see what you did there	False	False
2	I want only the best for you	False	False
3	I'm here to comfort you	False	False
4	I can care less if they stick with it	False	False
5	I've been watching anime for a long while now	True	True
6	I'm gonna stay home and play some games	True	True
7	I did the same thing last year	True	False
8	I've already heard it all	True	False
9	I rooted for the giants in the superbowl	False	True
10	I guess it s because I just can't fathom thoughts	False	True

Table 3.6: Model prediction on profile-like comments from Reddit Comments

The 'pred' column is the predicted result of the model. The 'manual' column is the manual evaluation of whether the sentence is a profile or not. As we could see from lines 1-6, the model is predicting correctly on profile sentences and non-profile sentences. One important thing to notice is that this set of data comes from the Reddit comments dataset that satisfies the 4 rules. However, the training data of the model consists of Persona Chat as positive samples and Reddit comments that do not satisfy the 4 rules as

negative samples. Therefore, this set of data, to some extent, is more similar to the positive samples in the training dataset. If the model is not learning any pattern as we suspected before, it will simply classify this set of data to be 'True'. However, since the model is actually distinguishing between sentences that are all similar to the positive samples, we conclude that this profile-classifier has the ability to classify profiles.

3.5 Conclusion

Overall, we conclude that this profile classifier has the ability to distinguish between profile statements and non-profile statements and thus has the ability to extract profile elements from a large dataset to be used for our later work. From the Reddit Comments dataset, we obtained 12,117 predicted profile statements in total.

Chapter 4

Emotion Detection

To embed the emotion into the response generation task, the very first step needed is to develop an emotion classifier that allows us to detect the emotion of any given sentence. This chapter introduces a RoBERTa-based emotion classifier.

4.1 Dataset

To train the emotion classifier, we used the Empathetic Dialogues dataset [8]. Empathetic Dialogues is a novel dataset released by Facebook, containing 22,908 conversations (17,623 for training, 2,747 for validation, and 2,538 for testing). Each dialogue consists of an emotion label, a situation description, and a multi-turn dialogues between two parties based on the situation. Figure 4.1 [8] shows an example of a dialogue in Empathetic Dialogues.

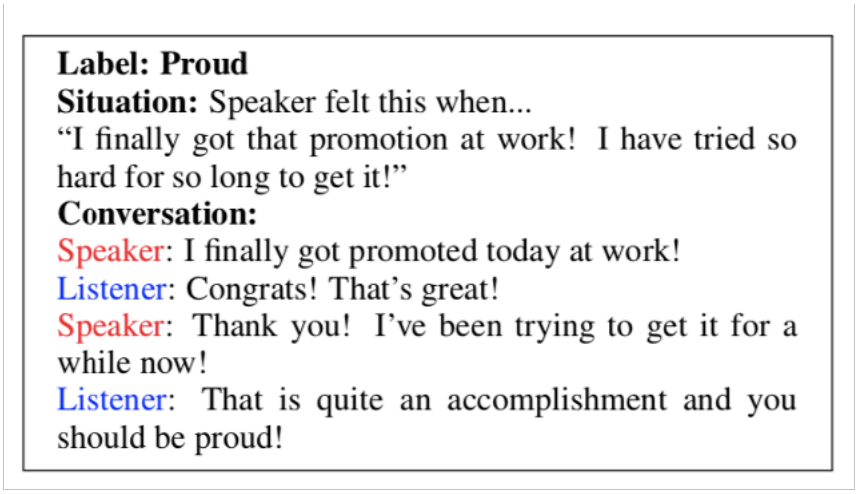

Figure 4.1: Sample of Empathetic Dialogues [8]

There are 32 emotion labels in total, as shown in Table 4.1. For training purposes, all emotion labels of string types were converted to integers according to the alphabetical order of the emotions.

1	afraid	9	confident	17	furious	25	nostalgic
2	angry	10	content	18	grateful	26	prepared
3	annoyed	11	devastated	19	guilty	27	proud
4	anticipating	12	disappointed	20	hopeful	28	sad
5	anxious	13	disgusted	21	impressed	29	sentimental
6	apprehensive	14	embarrassed	22	jealous	30	surprised
7	ashamed	15	excited	23	joyful	31	terrified
8	caring	16	faithful	24	lonely	32	trusting

Table 4.1: 32 emotion lables in Empathetic Dialogues

The reasons why we chose to use this dataset are

1. The dataset is constructed in a dialogue manner so that we could

perform dialogue level analysis, which will be useful for our response generation task later.

2. The dataset contains 32 emotion labels, which is a relatively large set. It allows more varieties in emotions rather than the classical categorization of neutral, positive, and negative.

3. The dataset is relatively balanced among all emotions (see Figure 4.2), which is crucial for a classification task.

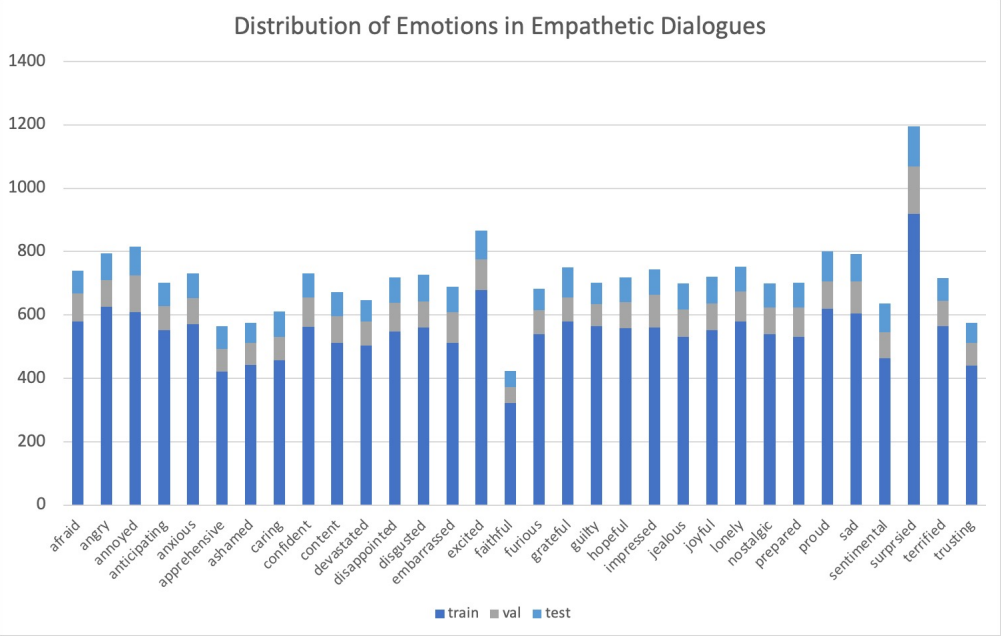

Figure 4.2: Distribution of emotions in Empathetic Dialogues

4.2 Approach

To train the emotion classifier, we used the Transformer model introduced in Section 2.2.2. We performed experiments with both BERT and RoBERTa to compare the performance.

4.2.1 Train on situation

Given that the situation of each dialogue best summarizes the main topic of the whole dialogue with the emotion expressed, we first extracted the situation of each dialogue along with the corresponding emotion label. The aim of this combination is to train the emotion classifier so that it could detect the emotion associated with a given situation. The model trained on this combination performs as a normal emotion classifier.

4.2.2 Test with utterance

We also noticed that a unique feature of the Empathetic Dialogues dataset is that it includes conversational dialogue exchanges, which might be useful for our future work in embedding emotions into dialogue settings. Therefore, it is important to analyze the utterances separately instead of interpreting the dialogue as a whole. More specifically, we tested the model on each utterance.

4.3 Experiments and Evaluation

For purpose of comparison, we kept all the hyper-parameters the same for the two different models. We used the batch size of 32. The accuracy on the evaluation dataset using different models is shown in Table 4.2.

Model	**Accuracy**
bert	57.95%
roberta	60.61%

Table 4.2: Accuracy of emotion classifier using 32 labels on evaluation set

We then used the RoBERTa model to make predictions on the testing set. After applying the model, there were 3 labels in total:

- given emotion label
- emotion label predicted by the situation
- emotion label predicted by the utterance

The accuracy of at least one match between these 3 labels in each dialogue is shown in Table 4.3.

Combo	At least 1 match
given label VS label by situation	59.64%
given label VS label by utterance	71.09%
label by situation VS label by utterance	77.85%

Table 4.3: Accuracy of emotion classifier using 32 labels on the testing set

4.4 Results and Discussion

In Table 4.3, the accuracy between the given emotion label and the emotion label predicted by the situation is simply the accuracy of the emotion classifier on the testing set. It is listed there for comparison purposes. The accuracy between the label predicted by the single utterance and the two other labels is much higher than the simple accuracy, which implies that some utterances inside the dialogue can better represent the emotion of the whole conversation rather than the given summary. However, we still chose to move forward with the given summary since it includes comprehensive information of the dialogue and is easily accessible.

The accuracy of the model trained on the summary is about 60%. Since the dataset contains a large number of emotion labels, it is possible that one emotion is too close to another for the classifier to detect. Therefore, we decided to plot the confusion matrix to further inspect the results.

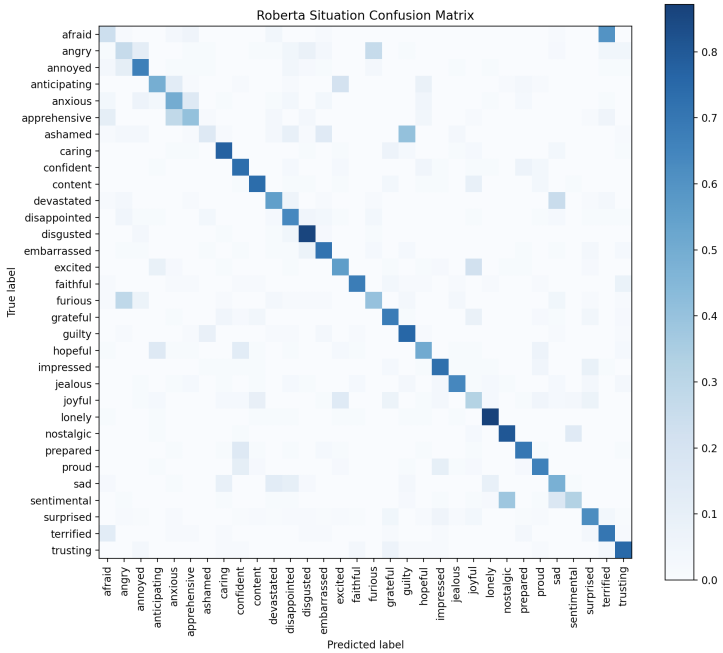

Figure 4.3: Confusion matrix of emotion classifier using 32 emotion labels

As we observed from the confusion matrix in Figure 4.3, there is a clear dark-colored diagonal. This indicates that the model overall has the ability to distinguish between emotions. However, we also noticed some dark squares off the diagonal. For instance, the model tends to predict those who are actually 'afraid' to be 'terrified', predict those who are actually 'ashamed' to be 'guilty', and predict those who are actually 'sentimental' to be 'nostalgic'. It also confuses between 'angry' and 'furious', and 'anxious' and 'apprehensive'. These patterns in the confusion matrix conform with what we have speculated

about the model confounding between similar emotions.

As a confirmatory experiment, we merged the 32 labels into 18 labels based on the confusion matrix. Similar and related emotions were grouped into one class. The actual grouping is shown in Table 4.4.

0	afraid, terrified
1	angry, annoyed, furious
2	anticipating, excited, content, joyful, hopeful
3	anxious, apprehensive
4	ashamed, embarrassed, guilty
5	caring
6	confident, prepared
7	devastated, sad, lonely
8	disappointed
9	disgusted
10	faithful
11	grateful
12	impressed
13	jealous
14	nostalgic, sentimental
15	proud
16	surprised
17	trusting

Table 4.4: Merge 32 emotion labels to 18 emotion labels

Then we ran the same model based on the merged set of 18 emotion labels. The accuracy is shown in Table 4.5.

Encoder	32 labels	18 labels
BERT	57.95%	73.13%
RoBERTa	60.61%	74.95%

Table 4.5: Accuracy of emotion classifier using 18 labels on evaluation set

We observed that the accuracy of the merged class has vastly improved by 26.19% and 23.67%. We also plotted the confusion matrix to see whether the model continues to confuse between similar emotions or not.

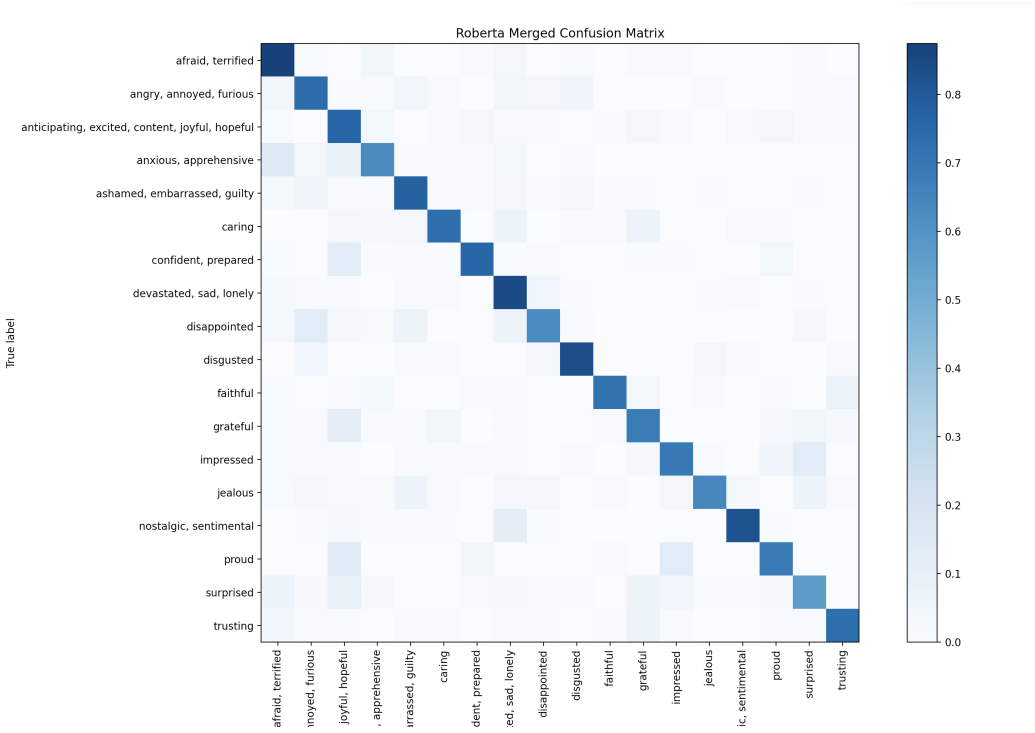

Figure 4.4: Confusion matrix of emotion classifier using 18 emotion labels

We could see from the confusion matrix in Figure 4.4 that there is nearly no obvious dark dot off the diagonal. This implies that the wrong predictions

of the previous model mostly come from the confusion of similar emotions. After we merged related emotions, the performance of the classifier largely improved.

4.5 Conclusion

Overall, it is clear that the emotion classifier has the ability to distinguish between emotions. When similar emotions are grouped together, the performance also improves a lot. However, we decided to move forward with the emotion model using 32 labels instead of 18 labels for two reasons. Firstly, the grouping of the emotion labels from 32 labels to 18 labels is based on the confusion matrix. In other words, the way of grouping is based on the result we got from the first model. The aim of merging is to confirm our conjecture that most wrong predictions of the previous model can be fixed by decreasing the emotion labels set. Therefore, simply merging similar emotions is not changing anything other than improving the accuracy. The nature of the model stays the same. Secondly, even if some of the emotions are similar, there still exist discrepancies between them. Empathetic Dialogues is a high-quality dataset for the emotion detection task. Therefore, we still want to keep using the original dataset to maintain the variety of emotions.

Chapter 5

Emotion-based Response Generation

With the emotion classifier in hand, we are able to detect any emotion given a sentence. We then focused on embedding emotional information into the response generation task given user profiles to further improve the engagement of the responses.

5.1 Dataset

To tackle the response generation task, we need a dataset containing the profiles and the corresponding responses. We also want the responses to contain some sorts of emotions. While there exist many dialogue datasets, none of them satisfy our two requirements. Therefore, we decided to create our own dataset.

5.1.1 Data Collection From Mturk

We acquired crowd-sourced data via Amazon Mechanical Turk (MTurk). Amazon Mechanical Turk is an online crowd-sourcing marketplace that individuals and businesses can outsource jobs to a distributed workforce who can perform these tasks virtually. A time-consuming project is usually broken down into smaller and more manageable tasks to be completed by distributed workers.

The input sentences of our response generation task were limited to profiles. In total, we had 5,578 personas from the Persona Chat dataset and 12,117 profiles filtered by the profile classifier in Chapter 3. Since the total number of 17,695 profile statements was relatively large, it would have cost a large amount of money to post all of them on MTurk. Therefore, we decided to first use the 5,578 personas from the Persona Chat dataset to complete our task.

We crowd-sourced the set of 5,578 personas from the Persona Chat dataset, each requiring 3 responses from 3 different workers. Each worker was limited to complete the task for each persona only once. We asked the workers to imagine that they were having a conversation with another person, and their task was to provide a single response to the given sentence. The response is

composed of 2 parts.

1. **Exclamation**: a short phrase used as an interjection to express your emotional reaction to the sentence with examples like "oh no", "wow", "no way", "thats cool".

2. **Follow up**: a sentence that directs the subsequent conversation in an engaging and relevant way.

The responses were required to be given in this way because we would like workers to express emotions in the response in the way of a short emotional phrase (exclamation) and then give the actual answers (follow up).

We also provided 3 good examples and 3 bad examples for them to better understand the task in Table 5.1.

Good Examples		
Given sentence	**Exclamation**	**Follow up**
I bought a new house.	Congratulations!	Where are you moving to?
	That's exciting.	Is this your first house?
	Wow, I'm jealous.	My house is so small.
		I'd love to move.
Bad Examples		
Given sentence	**Exclamation**	**Follow up**
I run track.	Really! What distance?	100m and 200m.
	I see.	Moving is a lot of work.
	Yeah.	Houses are cool.

Table 5.1: Examples given in MTurk instructions

The problem of bad example 1 is that the MTurk worker gives a multi-turn dialogue instead of a single response. The problem of bad example 2 is that the exclamation does not express any emotion. The problem of bad example 3 is that the follow up is not giving any useful information to lead the conversation.

The task is only limited to workers in several English-speaking countries including the United States, the United Kingdom, Australia, and Canada. The exclamation should have a minimum length of 3, and the follow up should have a minimum length of 6.

Before posting all of the 5,575 personas to the market, we performed 3 mini rounds of 50 personas to check whether our explanation is clear enough for workers to give responses in the way we want. We did the manual check of each mini round, and the percentage of good results is shown in Table 5.2.

Mini round 1	Mini round 2	Mini round 3
76.67%	84.67%	92%

Table 5.2: Result of mini rounds for MTurk task

Between each mini round, we modified our instructions to eliminate ambiguity. After the mini round 3, we were confident that the instructions were clear enough for workers to generate responses of good quality.

The final user interface of our task is shown in Figure 5.1. We got 3 responses for 5,578 personas, which are 16,734 sentence-response pairs.

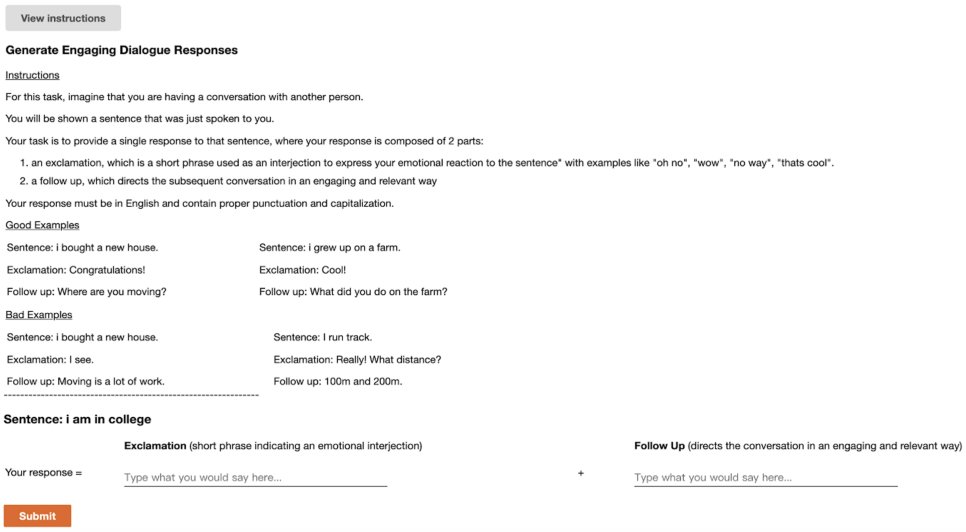

Figure 5.1: User Interface of Mturk Task

5.1.2 Apply Emotion Classifier

After we collected this novel dataset from Mturk, we used the emotion classifier developed in Chapter 4 and predicted the emotion of personas, exclamations, and follow ups, respectively. After sampling from the predictions, we found out that the predicted emotions of follow ups were not always useful, since the nature of the follow up defines it to be an information-oriented question that usually doesn't convey emotions. The predicted emotions of exclamations are the most useful since those exclamations are intended to be emotional

reactions. The sample result is shown in Table 5.3.

Type	Sentence	Predicted Emotion
Persona	I am very athletic.	proud
Exclamation	Awesome!	impressed
Follow up	What do you do to stay in shape?	prepared
Persona	I play jazz piano in a band.	confident
Exclamation	Sounds fun!	excited
Follow up	Do you ever perform in concerts?	confident
Persona	I am currently unemployed.	sad
Exclamation	Oh dear!	sad
Follow up	Do you need help finding a job?	caring

Table 5.3: MTurk dataset with predicted emotions

5.2 Data Analysis

After we collected this novel dataset from MTurk, we thought of several potential uses in dialogue systems. The first analysis is categorization. We thought that the categorized groupings might be able to feed into graph representation in dialogue systems to further improve the generated responses. We might also be able to generate the knowledge-based implication rules using the grouping analyses.

5.2.1 Categorize Personas

One possible goal of categorization is to investigate how people respond to a specific category of sentences. We achieved this by grouping the given

sentences (personas) into 9 different categories as shown in Table 5.4.

Category	**Sample personas**
Activity	I had a gig at local theater last night
	I exercise everyday
	I eat large meals
Characteristic	I am very athletic
	I have brown hair
	I am not afraid of what others think
Family	My father was born in australia
	My boyfriend works for nasa
	Both my parents were teachers
Hobby	I like to go hunting
	I love iced tea
	Halloween is my favorite holiday
Identity	My birthday is in june
	My name is omar
Occupation	I am a guitar player
	I work as a stand up comedian
	I have my own salon
Statement	Pudding makes me gassy
	I can only see 200 feet in front of me
Statement with attitude	I believe that mermaids are real
	I cannot wait to start my new life
	I hope it to become a doctor one day
Status	I have two cats growing up
	I am stuck in a wheel chair
	I am pregnant with my first child

Table 5.4: 9 categorization of personas

5.2.2 Categorize Follow Ups

Another possible goal of categorization is to investigate usual ways of responding to users. We achieved this by grouping the follow ups into 18 different

categories by topic.

The distribution of the categorization is shown in Figure 5.2.

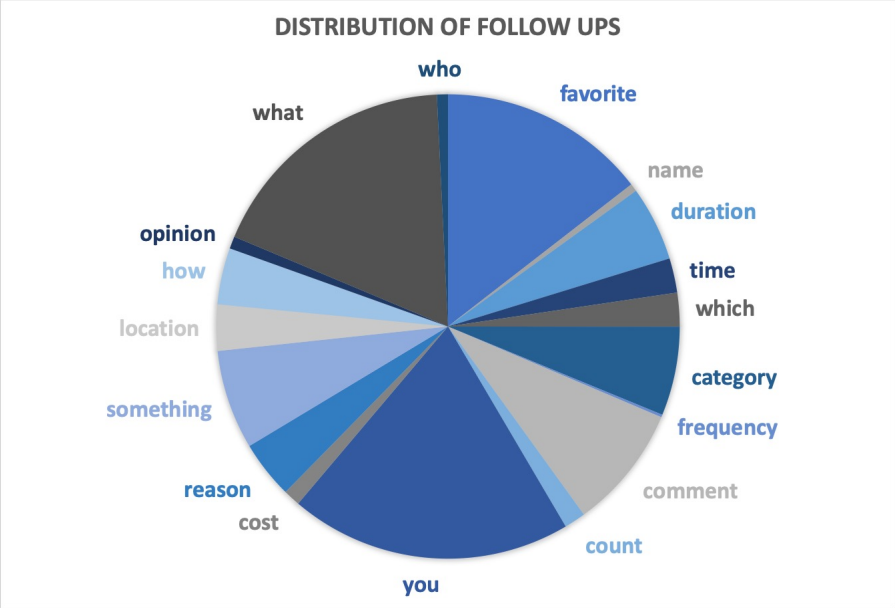

Figure 5.2: Distribution of 18 follow up categorizations

The categories and sample follow ups are shown in Table 5.5.

Categories	**Sample follow ups**
favorite	do you prefer contacts or glasses?
	what's your favorite team?
	what kind of concerts do you like to watch?
opinion	what did you think of the city?
	do you think you would've passed if you hadn't flirted?
reason	why you feel so tired easily?
	how come you haven't spoken to them?
category	what type of homemade meals do you cook?
	what kind of sports do you played in high school?
duration	how long were you dating before you got married?
	how much time do you spend shopping?
time	when did you first start flying?
	what time do you usually get up?
frequency	how often do you cut your hair?
	how many times a week do you do yoga?
count	how many stories have you written so far?
	how many meals do you eat a day?
location	where did you go to college?
name	what is the name of your band?
cost	how much extra money are you able to bring in doing that?
	how much did it cost?
how	how did you get to be so talented?
	exactly how are you pursuing that?
who	who is your girlfriend?
	who takes care of them?
which	which instruments do you play?
	which event did you win it in?
what	what interesting news will come every night?
	what did you do in the military?
comments	i really admire creative people like you.
	you're probably going to be able to get a great job!
something	is that a common side effect?
	don't the cats disturb the birds?
you	are you working full time?
	did you ever meet any movie stars?
	do you still sing anyway?
	have you tried hiring a vocal coach?
	were you involved in any active military operations?

Table 5.5: 18 categorization of follow ups

5.2.3 Similarity Between Responses

Another unique thing about the dataset is that we collected 3 different responses for each given sentence. This gives us the possibility to analyze the similarity between each response. More specifically, we are interested in what kinds of sentences people tend to respond to in similar ways. To achieve this, we used the SentenceBert model [9] to find the similarity score between 3 responses for a given sentence.

Most Similar Exclamations	
Persona Exclamations	I got married last year. Congratulations! Congratulations! Congratulations!
Persona Exclamations	I did not graduate high school. Oh no! Oh no! Oh no!
Most Dissimilar Exclamations	
Persona Exclamations	I fantasize about taking over the world. No way! That's funny! That's ambitious!
Persona Exclamations	I grew up as an orphan. That sounds rough! I am sorry. How interesting!

Table 5.6: Similarity samples of exclamations

The samples of the most similar exclamations and the most dissimilar

exclamations are shown in Table 5.6.

The samples of the most similar follow ups and the most dissimilar follow ups are shown in Table 5.7.

Most Similar Follow Ups	
Persona	I had two cats growing up.
Follow ups	What were their names?
	What were their names?
	What were their names?
Persona	I play many instruments.
Follow ups	What instruments do you play?
	What instruments do you play?
	What are some of the instruments you play ?
Most Dissimilar Follow Ups	
Persona	I walk dogs for a living.
Follow ups	How did you get started?
	Is it great fun or a little crazy?
	What breed of dog you have?
Persona	I fake a British accent to seem more attractive.
Follow ups	Where did you learn the accent?
	Why don't just be your self?
	Do you also post heavily-filtered selfies?

Table 5.7: Similarity samples of follow ups

5.3 Approach

5.3.1 Embed Emotions

The idea of embedding emotion as a token into the response generation task was inspired by the paper "Automatic Dialogue Generation with Expressed

Emotions" [4]. We chose both the RNN model and the Transformer model described in Chapter 3 since they are both suitable for Seq2Seq tasks. For this response generation task, these models take the input (persona, exclamation, follow up), which can be represented as (X, Y, Z). The aim of the model is to minimize the cross-entropy loss $L = \log p(Y|X)$ and $\log p(Z|X)$ and generate the output response as exclamation + follow up, which is $Y + Z$. In our case, each input tuple (X, Y, Z) has a corresponding desired emotion pair (e_y, e_z) predicted by the emotion classifier in Section 5.1.2. Our goal is then to minimize $\log p(Y|X, e_y)$ and $\log p(Z|X, e_z)$.

To embed the desired emotion pairs (e_y, e_z), we chose to concatenate the emotion as a token into the input, either before X which is $(e + X)$, or after X which is $(X + e)$. We performed this separately to each emotion to see which is the best position for each.

5.3.2 Evaluation Methods

Since the output we generated is response text, it is hard to evaluate the performance of this model simply using accuracy. We proposed 5 different evaluation methods.

- **All matches**: The number of exact matches between the predicted

response and the given response. It ranges from 0%-100%.

- **Match/persona**: For each persona, the number of at least 1 exact match between the predicted response and all 3 given responses. It ranges from 0%-100%.

- **Match/emotion**: For each persona, the number of at least 1 exact match between the predicted response and all given responses with the same emotion. It ranges from 0%-100%.

- **Similarity score**: Similarity score between two sentences calculated by the SentenceBert model. It ranges from 0-1. 1 normally means two texts are identical.

- **BiLingual Evaluation Understudy (BLEU) score** [7]: It is usually used to evaluate the quality of translations. Here we took the average of BLEU-1, BLEU-2, BLEU-3, and BLEU-4. It ranges from 0-1. 1 normally means two texts are identical.

5.4 Experiments and Evaluation

We first embedded the emotion of exclamation (e_y) into the input sentence X and tried to predict the exclamation. That is to minimize $\log p(Y|X, e_y)$. The

evaluation result is shown in Table 5.8 and Table 5.9. The best performance for each column is in bold.

Type	All matches	Match/persona	Match/emotion
RNN-without	13.42%	33.45%	14.64%
RNN-before	35.96%	71.74%	39.36%
RNN-after	36.43%	72.27%	39.58%
Trans-without	18.31%	42.04%	18.75%
Trans-before	**37.27%**	**72.45%**	**40.55%**
Trans-after	36.67%	71.56%	39.88%

Table 5.8: Accuracy of exclamations using Seq2Seq models

Type	Similarity score	Average BLEU score
RNN-without	0.5265	0.6589
RNN-before	0.6784	0.7181
RNN-after	0.6789	**0.7247**
Trans-without	0.5647	0.6396
Trans-before	**0.6877**	0.7157
Trans-after	0.6764	0.7084

Table 5.9: Evaluation scores of exclamations using Seq2Seq models

The predicted exclamation samples is shown in Table 6.10.

Persona	I can play the piano.
Given exclamation	That's cool!
RNN-without	Cool!
RNN-before	That's great!
RNN-after	That's cool!
Trans-without	Cool!
Trans-before	That's awesome!
Trans-after	That's great!

Table 5.10: Sample predicted exclamations using Seq2Seq models

We then embedded the emotion of follow up (e_z) into the input sentence X and tried to predict the follow up. That is to minimize $\log p(Z|X, e_z)$. The evaluation result is shown in Table 5.11 and Table 5.12. The best performance for each column is in bold.

Type	All matches	Match/persona	Match/emotion
RNN-without	0.12%	0.36%	0.07%
RNN-before	0.18%	0.54%	0.21%
RNN-after	0.24%	0.72%	0.29%
Trans-without	2.21%	5.90%	2.36%
Trans-before	2.21%	5.55%	2.29%
Trans-after	**2.74%**	**7.51%**	**3.08%**

Table 5.11: Accuracy of follow ups using Seq2Seq models

Type	Similarity score	Average BLEU score
RNN-without	0.2582	0.0832
RNN-before	0.3130	0.0897
RNN-after	0.2973	0.0855
Trans-without	0.4071	0.1302
Trans-before	0.4167	0.1405
Trans-after	**0.4180**	**0.1434**

Table 5.12: Evaluation scores of follow ups using Seq2Seq models

Persona	I can play the piano.
Given follow up	How long have you been playing?
RNN-without	Is that your favorite?
RNN-before	How long have you been playing?
RNN-after	How's your like to have?
Trans-without	How long have you been playing?
Trans-before	How long have you been playing piano?
Trans-after	Do you know how impressive it is to learn something like that?

Table 5.13: Sample predicted follow ups using Seq2Seq models

5.5 Results and Analysis

There are a lot of patterns we could observe and interpret from these results. We performed the analyses from three levels:

- **Model-level**: performance comparison between RNN and Transformer
- **Task-level**: performance comparison between exclamation and follow up
- **Approach-level**: performance comparison between different embedded locations of emotion tokens

5.5.1 Model-Level Analysis

As we could see in Table 5.8 and Table 5.9 for exclamation prediction tasks, RNN and Transformer have similar performance in each task. The accuracy and the evaluation scores are all pretty close. The Transformer model overall has a slightly better performance than RNN, while RNN has a better average BLEU score than Transformer.

When we moved on to Table 5.10 and Table 5.11 for follow up prediction tasks, we observed a great discrepancy between these two models. The Transformer model overall has a much better performance than RNN in all evaluation methods.

The reason behind this phenomenon is that the RNN model constructs its own word embedding using the training data, while the Transformer model has the pre-trained BERT embedding. When the target output is short and easy like 'oh no!', the RNN model has the ability to recognize the pattern even if there exists some word that the model doesn't know. However, when the target output is long and complicated like 'Do you have anyone close to you that you can spend time with?', the RNN model lacks the ability to perform the task as expected compared to the Transformer model.

The Transformer model overall has a better performance, which is not

surprising. As explained in Section 2.2.2, the BERT-based Transformer model has become the state-of-the-art model for many NLP tasks because of its innovation in architecture. It outperforms many other models.

5.5.2 Task-Level Analysis

When we look at the prediction results for exclamation and follow up, we observed that the exclamation task has much better performance. This makes sense since follow ups are normally longer and contain more open information for the model to learn the pattern. Also, the fact that the accuracy and evaluation scores are low only means that the predicted responses are different than the given responses. However, the generated responses might still make sense to some extent.

5.5.3 Approach-Level Analysis

As we could see in each table, the accuracy and evaluation scores without the emotion embedded are remarkably worse than those with the emotion embedded. This indicates that our main idea of embedding emotion into the response generation task does improve the overall quality of the generated responses.

In Table 5.8, the highest accuracy is found in *Trans-before*. In Table

5.9, it shows that *Trans-before* and *RNN-after* both have good performances. Therefore, we think that *Trans-before*, which embeds the emotion before the persona using Transformer, is the best combination to tackle the exclamation generation task.

In Table 5.11, the highest accuracy is found in *Trans-after*. In Table 5.12, it shows that *Trans-after* also has the best performance. Therefore, we think that *Trans-after*, which embeds the emotion after the persona using Transformer, is the best combination to tackle the follow up generation task.

5.6 Conclusion

Based on the accuracy and evaluation analysis, we would use the Transformer model to embed the exclamation emotion before the persona to train the exclamation model and to embed the follow up emotion after the persona to train the follow up model. Overall, embedding the emotion into the response generation model does improve the quality of the generated responses.

Chapter 6

Future Work

While overall we have achieved what we expected to do at the beginning of this task, there is still something that could be accomplished in the future to further improve this project. This chapter discusses some of the future work that could be done.

6.1 MTurk More Profiles

As we have mentioned in Section 5.1.1, we only posted 5,578 personas from the Persona Chat dataset to avoid high costs at the first stage. Since the response generation task turned out to work well, we will post the other 17,695 profile statements which are filtered by the profile classifier in Chapter 3. Since the Seq2Seq model often requires a large amount of data to learn the pattern and generate good results, we are confident that largely increasing the size of our dataset will further improve our emotion-based response generation

task.

6.2 Desired-Emotion Predictor

In Chapter 5, we conclude that we embed the exclamation emotion before the persona to train the exclamation model and embed the follow up emotion after the persona to train the follow up model. However, in an actual situation, when we are given a profile statement, we do not have the desired exclamation emotion and the desired follow up emotion. Therefore, we need to develop a desired emotion predictor. We will use the persona, exclamation emotion, and follow up emotion in our MTurk dataset to train a Transformer-based desired emotion predictor. Using this predictor, we will first find the desired response emotion given a profile statement and then input the combination as we experimented to our Seq2Seq model to generate emotion-based responses.

Chapter 7

Conclusion

As we have mentioned, Seq2Seq models generally have a better result with a huge amount of training data. Since we have successfully developed a BERT-based profile classifier that allows us to extract profiles, we will be able to generate as many profile statements as possible to feed into our emotion-based response generation model. In this way, we are confident that the model will respond to the profile statements from users in a specific, natural, and engaging way.

Further, we have successfully developed a RoBERTa-based emotion classifier based on 32 labels. It allows us to detect any emotion throughout the conversation. In addition to embedding the emotion as a token with the profile, there might be other different ways to inject the emotions.

We have also collected a novel dataset containing personas, exclamations, and follow ups via MTurk. By embedding the exclamation emotion before

the persona and embedding the follow up emotion after the persona, we are able to generate satisfied responses using the Transformer model.

Overall, the emotion-based response generation task that we have accomplished in this paper does satisfy our expectations at the beginning. The model successfully generates human-like responses to user profiles with the emotion expressed. It also shows a lot of potentials to promote this task generally in the future.

Bibliography

[1] Ali Ahmadvand, Harshita Sahijwani, and Eugene Agichtein. Would you like to talk about sports now? towards contextual topic suggestion for open-domain conversational agents, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] Alex Graves. Generating sequences with recurrent neural networks, 2014.

[4] Chenyang Huang, Osmar Zaïane, Amine Trabelsi, and Nouha Dziri. Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 49–54, New Orleans, Louisiana, June 2018.

Association for Computational Linguistics. doi: 10.18653/v1/N18-2008. URL https://www.aclweb.org/anthology/N18-2008.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[6] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. Training millions of personalized dialogue agents, 2018.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

[8] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: a new benchmark and dataset, 2019.

[9] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[12] Oriol Vinyals and Quoc Le. A neural conversational model, 2015.

[13] Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. Response generation by context-aware prototype editing, 2018.

[14] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too?, 2018.