

Examining Personality Differences in Chit-Chat Sequence to
Sequence Conversational Agents

MSc Thesis (*Afstudeerscriptie*)

written by

Xing Yujie

(born November 8th, 1993 in Changsha, Hunan)

under the supervision of **Dr Raquel Fernández**, and submitted to the Board of
Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense: **Members of the Thesis Committee:**

July 3, 2018

Dr Tejaswini Deoskar (chair)

Dr Raquel Fernández

Dr Shane Steinert-Threlkeld



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

Personality inconsistency is one of the major problems for chit-chat sequence to sequence conversational agents. Works studying this problem have proposed models with the capability of generating personalized responses, but there is not an existing evaluation method for measuring the performance of these models on personality. This thesis develops a new evaluation method based on the psychological study of personality, in particular the Big Five personality traits. With the new evaluation method, the thesis examines if the responses generated by personalized chit-chat sequence to sequence conversational agents are distinguished for speakers with different personalities. The thesis also proposes a new model that generates distinguished responses based on given personalities. The results of our experiments in the thesis show that: for both the existing personalized model and the new model that we propose, the generated responses for speakers with different personalities are significantly more distinguished than a random baseline; specially for our new model, it has the capability of generating distinguished responses for different types of personalities measured by the Big Five personality traits.

Acknowledgements

First of all, I would like to thank my supervisor Raquel Fernández. It is her encouragements and supports that enable me to finish this thesis. On the practical aspect, she afforded the price of LIWC dataset for me, so that I was able to pursue the project. She gave careful and detailed comments on my thesis as well as suggestions on the project, and even corrected my typos and grammar mistakes in the thesis. She also helped me to get connected with all the people that may be helpful: she connected me with Elia Bruni, who helped me to apply for a GPU account for training models; she connected me with Tim Baumgärtner, who helped me on PyTorch; she connected me with Sanne Bouwmeester, who helped me with the GPU cluster issues. For the mental aspect, she encouraged me many times when I was worried about the planned speed of my thesis or my future. I am very fortunate to have her as my supervisor and I have gained a lot from her, both materially and mentally.

I would like to thank my mentor Benedikt Löwe, who helped me a lot during the period of MSc Logic. I still remember the meeting with him, where I was anxious on changing research interest from philosophy to natural language processing, while he taught me that it was natural for people to realize what they were interested in might not be what they would like to work on. Thanks to this meeting, I was able to face myself and devote myself into the new field. Also, he supported me a lot: he provided me the chance for English oral classes, and pointed me to Raquel to work on my Master Thesis. I am really lucky to have him as my mentor.

I would like to thank Evangelos Kanoulas and Kaspar Beelen for the support when I was facing the difficulty of transferring research interest. I would also like to thank Guo Jiahong and Ju Fengkui; I could not have come to ILLC without their help.

My friends also gave me encouragements and supports that mean a lot to me. Thanks all my friends in ILLC, and ILLC itself that gave me the chance to befriend with them. Thanks my friends in China, for their generous help during my difficult time, and for those pleasant chats during these years.

Finally, I want to thank my parents. My debt to them is unbounded.

Contents

1	Introduction	7
1.1	Response Generation for Chit-Chat Conversational Agents	8
1.1.1	Rule-Based Conversational Agents	8
1.1.2	Information-Retrieval-Based Conversational Agents	8
1.1.3	Generation-Based Conversational Agents	9
1.2	Personality Inconsistency in Response Generation	10
1.3	Overview of this Thesis	11
2	Linguistic Correlates of Personality	13
2.1	Psychological Background of Personality	13
2.2	Examining Personality	14
2.2.1	The Big Five Personality Traits	14
2.2.2	Examining OCEAN from Questionnaires	15
2.2.3	Examining OCEAN Automatically	16
2.2.4	Personality Recognizer	17
3	Personalized Natural Language Generation	19
3.1	Natural Language Generation from Personality Features	19
3.2	End-to-End Response Generation	20
4	Model	24
4.1	Standard Model	24
4.1.1	General Structure	25
4.1.2	Long Short Term Memory	27

4.1.3	Attention Mechanism	28
4.2	Extended Models for Personalizing	29
4.2.1	Speaker Model	29
4.2.2	Personality Model	30
5	Experiments	32
5.1	Datasets	32
5.2	Experimental Setup	34
5.2.1	Training and Decoding	34
5.2.2	Evaluation	35
5.3	Preliminary Experiment: Examining Personality Differences for the Original Scripts	37
5.4	Experiment 1: Examining Personality Differences for the Speaker Model	39
5.4.1	Experimental Procedure	40
5.4.2	Results	41
5.5	Experiment 2 & 3: Examining Personality Differences for the Personality Model	50
5.5.1	13 Characters from the TV-series Dataset	50
5.5.2	32 Extreme Personalities	56
6	Conclusion	58
A	Responses and OCEAN Scores	67
A.1	Responses to “Do you love me?”	67
A.2	Estimated OCEAN Scores	69

List of Tables

5.1	Characters and their respective utterance numbers	34
5.2	Average overall F1 scores and accuracy scores on <i>Friends</i> for the original scripts	38
5.3	Statistical results on <i>Friends</i> for the original scripts with respect to the baseline	38
5.4	Average overall F1 scores and accuracy scores on <i>The Big Bang Theory</i> for the original scripts	39
5.5	Statistical results on <i>The Big Bang Theory</i> for the original scripts with respect to the baseline	39
5.6	Perplexity on the TV-series validation set for the standard model and the speaker model	40
5.7	Average F1 scores on the speaker model with different cleaning methods	41
5.8	Average F1 and accuracy score on <i>Friends</i> for the original scripts and the speaker model	42
5.9	Statistical results on <i>Friends</i> for the speaker model	42
5.10	Average F1 scores and accuracy score on <i>The Big Bang Theory</i> for the original scripts and the speaker model	46
5.11	Statistical results on <i>The Big Bang Theory</i> for the speaker model . . .	47
5.12	Perplexity on the TV-series validation set for standard LSTM model, the speaker model and the personality model	50
5.13	Average F1 and accuracy score on <i>Friends</i> for the original scripts, the speaker model and the personality model	51
5.14	Statistical results on <i>Friends</i> for the personality model	51

5.15	Average F1 scores and accuracy score on <i>The Big Bang Theory</i> for the original scripts, the speaker model and the personality model	54
5.16	Statistical results on <i>The Big Bang Theory</i> for the personality model	54
5.17	Average Overall F1 score and accuracy score for 32 extreme personalities	56
5.18	Statistical results for 32 extreme personalities with respect to the baseline	56
A.1	Responses to <i>Do you love me?</i> generated by the standard model, the speaker model and the personality model for 13 characters from the TV-series dataset	68
A.2	Responses to <i>Do you love me?</i> generated by the personality model for 32 extreme personalities	69
A.3	Average OCEAN scores for 13 characters from the TV-series dataset on the original scripts, the speaker model and the personality model	70

List of Figures

2.1	TUPI questionnaire by Gosling et al. (2003)	16
5.1	Gold label and predicted label on <i>Friends</i> for the original scripts	44
5.2	Gold label and predicted label on <i>Friends</i> for the speaker model	45
5.3	Gold label and predicted label on <i>The Big Bang Theory</i> for the original scripts	48
5.4	Gold label and predicted label on <i>The Big Bang Theory</i> for the speaker model	49
5.5	Gold label and predicted label on <i>Friends</i> for the personality model . .	53
5.6	Gold label and predicted label on <i>The Big Bang Theory</i> for the person- ality model	55

Chapter 1

Introduction

Conversational agents, referred to as CA throughout this thesis, are those agents that serve as conversing with people. Since the first conversational agent ELIZA ([Weizenbaum, 1966](#)), there has been a long development on CA, where alongside the rule-driven method, the data-driven method has appeared. Nowadays, the data-driven method is frequently used. CA learn conversing from a big-scale dataset, thus the variety of responses is enriched.

From the perspective of the aim of CA, we can divide these agents into two categories: task-oriented and non-task-oriented. Examples for task-oriented CA are chatbots for booking restaurants or flights, where a conversation is closed once the agent has finished the task. In comparison to task-oriented CA, Non-task-oriented CA do not have tasks, or their only task is to converse. Chit-chat CA, which is the focus of this thesis, is the kind of non-task-oriented CA that are open-domain, since chit-chats are not limited to a specific domain. CA that are open-domain are more difficult to build, compared with CA with a specific domain ontology.

In this thesis, we focus on the personality inconsistency problem on response generation for chit-chat CA. Personality is a concept mainly studied in psychology, while it is recently applied to works of response generation. Below we briefly introduce chit-chat CA and the personality inconsistency problem.

1.1 Response Generation for Chit-Chat Conversational Agents

There are generally three types of CA distinguished by the ways of building them: rule-based, information-retrieval-based, and generation-based.

1.1.1 Rule-Based Conversational Agents

Rule-based CA such as ELIZA have hand-written templates for answering different types of questions. The procedure is as follows: an agent first scans all the words from a question and looks for them in a keyword dictionary. If a word is found in the dictionary, the agent will select a responding template based on the question, and then fill the keyword into the selected template; if none of the words is in the dictionary, the agent returns a general response.

The limitation for this kind of CA is obvious. Although received positive feedbacks (Colby et al., 1972), due to the hand-written template and the keyword dictionary, rule-based CA have severe limitations both on the amount of possible answers and on the answering patterns; in sum, hand-written rules can only produce limited kinds of responses.

1.1.2 Information-Retrieval-Based Conversational Agents

Information-retrieval-based CA generate responses based on a big-scale corpus. The corpus often consists of human conversations; for each context-response pair in the corpus, an information-retrieval-based agent calculates the similarity between 1) the given context and the context in the pair; 2) the given context and the response in the pair. Combining 1) and 2), the agent ranks all the possible responses in the corpus and returns the top ranked one (Jurafsky and Martin, 2014).

Due to the large scale of the corpus, information-retrieval-based CA are able to generate abundant responses compared with rule-based CA; also, the corpus guarantees that the generated responses are well-formed both in grammar and semantics. However,

the disadvantage is easy to see: this kind of CA are not able to generate novel responses, since all the responses come from the corpus.

1.1.3 Generation-Based Conversational Agents

Generation-based CA overcome the disadvantage of information-retrieval-based CA: instead of selecting responses from an existing corpus, a generation-based agent selects words from the vocabulary and generates responses with these words, so that it is able to generate novel responses.

The idea of generation-based CA is similar to machine translation, with the reference translations in machine translation replaced by responses. For an example, an English-Chinese machine translation task takes English sentences as the source and the corresponding Chinese human reference translations as the target, while for generation-based CA, the target is replaced by the responses to the source: the English-Chinese machine translation task may take “Thank you” as the source and “谢谢” (the Chinese translation of “Thank you”) as the target, while for generation-based CA, the target will be changed to a response like “You are welcome” with respect to the source “Thank you”.

The origin of this kind of CA is from [Ritter et al. \(2011\)](#), where the response generation task for CA was treated as a statistical machine translation task.

In the following years, due to the development of neural networks, sequence to sequence (SEQ2SEQ) model ([Sutskever et al., 2014a](#)) has been applied to the machine translation task like Google Translate¹ and gained good results ([Junczys-Dowmunt et al., 2016](#)). This triggered researchers to use SEQ2SEQ model on the response generation task ([Vinyals and Le, 2015](#); [Shang et al., 2015](#); [Sordani et al., 2015](#)). Recently, researchers have proposed models that applied modifications on SEQ2SEQ model, or models that combined algorithms like reinforcement learning and adversarial learning with SEQ2SEQ model; these new models aim at making the agent generate more specific, fluent and coherent responses ([Serban et al., 2016a](#); [Li et al., 2016b, 2017b](#)). In general, these works gained better perplexity and BLUE ([Papineni et al., 2002](#)) scores

¹https://en.wikipedia.org/wiki/Google_Translate

than information-retrieval-based CA. Both perplexity (for detailed explanation, please see section 5.2.2) and BLUE are automated evaluations that measure how close to the ground-truth the predictions are; however, these scores are not suitable for evaluating the quality of generated responses (Liu et al., 2016).

Despite the success of SEQ2SEQ model on response generation, there are also problems. The agents always generate general responses such as “I don’t know”, lack consistency on the response content and the language style, and have difficulty in generating responses for multi-turn conversations (Zhang et al., 2018; Serban et al., 2016b). Among these problems, we are going to focus on the second one, namely the inconsistency problem.

1.2 Personality Inconsistency in Response Generation

The inconsistency problem mentioned above mainly has two aspects: inconsistency of the response content and inconsistency of the language style. For example, an agent described in Li et al. (2016a) gives contradicted answers to questions that are similar in semantics but different in forms: when being asked “How old are you?”, the agent answers “16”, while when being asked “What’s your age?”, the agent answers “18”. Also, the agents lack consistent language styles, since they are trained on dataset of conversations from many different people.

To solve this problem, researchers have proposed “personality” on CA, and till now, there have been several works that proposed personalized response generation models that try to keep consistent personality for CA (Li et al., 2016a; Zhang et al., 2017; Yang et al., 2017; Luan et al., 2017; Zhang et al., 2018). Although these works have proposed models that are able to keep consistent personalities, i.e. generate distinguished and consistent responses for different personalities, there is not an existing standard evaluation method for measuring whether these models really work. In the above works, researchers listed responses generated for different personalities (Li et al., 2016a; Yang et al., 2017) as qualitative evaluation, and tried human evaluation (Li et al., 2016a;

Zhang et al., 2017), but these are not standard metrics. One of the works calculated the word overlapping rate among generated responses of different personalities (Zhang et al., 2017), which is able to distinguish among personalities to some extent, but is still not a suitable metric.

Since we are talking about the evaluation for personality, and personality is well studied in psychology, there are plenty of measurements we can lend from psychology. However, the concept “personality” mentioned in the above works is different from the one in psychology. In the above works, “personality” was proposed to deal with the inconsistency problem (e.g. agent claims that it is both 16 years old and 18 years old). For the two aspects of this problem: inconsistency of the response content and inconsistency of the language style, although language styles can reflect personality, the content, such as what a person likes and where he/she lives, is called external source in psychology and is not counted into “personality” in the psychological definition (Burger, 2010).

We will use the psychological definition of “personality” for this thesis, so the response content is not taken into consideration for “personality”.

1.3 Overview of this Thesis

In this thesis, we make the following contributions:

1. We provide a new evaluation method for examining the personality differences among the responses generated by personalized response generation models.
2. With the new evaluation method, we examine the speaker model proposed by Li et al. (2016a), which we reimplement in PyTorch, if it can generate distinguished responses for different personalities as expected.
3. We build a new personality oriented model that can generate distinguished responses given a specific personality; the model is evaluated by the new evaluation method mentioned in 1). We propose this personality oriented model because previous works mixed the concept “personality” with consistency on the content.

The structure of this thesis is as follows: in chapter 2, we will introduce some background knowledge of personality in psychology, and the linguistic correlates of personality which our new evaluation method and new model are based on. The related works on personalized response generation will be introduced in chapter 3. The models, including the speaker model by Li et al. (2016a) and the new personality model proposed by us, together with the standard model of these two, are introduced in chapter 4. Finally, chapter 5 is the experiment part, where we will propose our new evaluation method, and give the results for both the speaker model and the personality model under the new evaluation method. We will summarize the conclusion in chapter 6. Examples of responses generated by the models and estimated OCEAN scores for these responses are listed in appendix.

Chapter 2

Linguistic Correlates of Personality

2.1 Psychological Background of Personality

People are always different with others, yet always share similarities with others; thus, they can be classified into different types. Think of ways of classifying people: gender, age, nationality... Besides these external factors, psychologists are interested in finding a consistent factor inside individuals that can classify people's behaviour patterns into several types; this factor is personality.

There is not a single definition for personality; however, what is mentioned above is always included in the definition. One of the common definitions is: personality is consistent behaviour patterns and intrapersonal processes originating within the individual ([Burger, 2010](#)). Notice that under this definition, personality is consistent, so a person of specific personality should have consistent behaviour patterns under normal circumstances; this is also the case for conversations, where consistent behaviour patterns are reflected by the utterances. Moreover, since personality is intrapersonal, external sources such as gender, age and nationality are not included in personality; external sources can influence personality, but they are not parts of personality.

2.2 Examining Personality

The definition of personality in psychology is abstract, so psychologists have proposed many approaches of describing personality: the psychoanalytic approach, the biological approach, the humanistic approach, etc. In this thesis, we are going to use the trait approach, where personality is divided into several dimensions—the traits—that categorizes people with the degree to which they manifest a particular characteristic (Burger, 2010). The trait approach provides numerical description for personality, so it fits our need: automatic recognition of personality for responses.

The trait approach sees different types of personalities as consisting of traits of different degrees. Traits are identified from data—data of personality questionnaires, data of reports of people’s daily actions, etc. For example, psychologists put the hypothesis characteristics into the questionnaire and ask subjects to answer the questionnaire; afterwards, the psychologists analyze the results for these hypothesis characteristics, and put highly correlated ones into one cluster. Finally, each cluster will be identified as a trait.

There are several different trait schemes. For instance, the Sixteen Personality Factor Questionnaire (16 PF) by Cattell (Cattell and Mead, 2008) is a famous system, where personality is broken down into 4 traits, with each trait having two poles; thus, we have 16 different types of personalities. In our thesis, instead of 16 PF, we use the Big Five Personality Traits (Norman, 1963) as the measurement for personality. We have three reasons: 1) the Big Five has consistently been found being able to capture basic dimensions of personality by multiple teams (Burger, 2010); 2) there are many works about automatic recognition on the Big Five; 3) the score of the Big Five can be treated as a numerical vector.

2.2.1 The Big Five Personality Traits

The Big Five Personality Traits consists of the following five traits: Extraversion, Neuroticism, Agreeableness, Conscientiousness, and Openness. Each trait is scored a number; each personality is represented as a 5-dimension vector consisting of scores for the

five traits. The Big Five is also called “OCEAN”, which is the combination of the initials of the five traits. The meaning of each trait is as follows:

- **Extraversion** measures where a person gets his/her energy from: outside himself/herself (extravert), or inside himself/herself (introvert). A person with a high score on Extraversion prefers to have more interactions with others, and is often outgoing and talkative; on the contrary, a person with low score on Extraversion prefers to stay alone, and is often quiet and reflective.
- **Neuroticism** measures emotional stability. A person with a high score in Neuroticism is easier to have negative emotions such as anxiety and anger; in other words, his/her emotion is less stable. Some works use “emotional stability” instead of Neuroticism, where people with high scores are more emotionally stable.
- **Agreeableness** measures people’s social agreeableness. A person with a high score is more cooperative and friendly, while a person with a low score is more competitive and suspicious.
- **Conscientiousness** measures how organized and responsible a person is. A person with a high score is careful and hardworking; a person with a low score is less goal-oriented and less efficient.
- **Openness** measures people’s openness to experience. A person with a high score is more creative and curious, while a person with a low score prefers what they are familiar with to new things.

Research on OCEAN has shown that the scores of each trait is normally distributed, regardless of geographical location and cultural background ([Schmitt et al., 2007](#)).

2.2.2 Examining OCEAN from Questionnaires

Questionnaires are widely used for examining OCEAN score: subjects either self-assess their behaviours, or are assessed by people who are familiar with them, such as friends and families.

Ten Item Personality Measure (TIPI) is one of the frequently used questionnaires (Gosling et al., 2003). Subjects assess what kind of people they think they are, and express their agreements to each question using numbers: usually 1 is strongly disagree, and 7 is strongly agree. For each trait, there is a positive question and a negative question, and the score for this trait is calculated based on the answers to these two questions.

I see myself as:

1. _____ Extraverted, enthusiastic.
2. _____ Critical, quarrelsome.
3. _____ Dependable, self-disciplined.
4. _____ Anxious, easily upset.
5. _____ Open to new experiences, complex.
6. _____ Reserved, quiet.
7. _____ Sympathetic, warm.
8. _____ Disorganized, careless.
9. _____ Calm, emotionally stable.
10. _____ Conventional, uncreative.

Figure 2.1: TIPI questionnaire by Gosling et al. (2003)

2.2.3 Examining OCEAN Automatically

There are also methods for examining OCEAN automatically with language traits. Previous works have stated that OCEAN scores are correlated with linguistic features, especially the Extraversion trait. Mairesse et al. (2007) has summarized the correlated linguistic features: extraverts use more positive emotion words, and show more agreements and compliments than introverts (Pennebaker and King, 1999); the Extraversion trait is significantly correlated with contextuality, opposed to formality (Heylighen and Dewaele, 2002); neurotics use more concrete and frequent words (Gill and Oberlander, 2003). Among these works, Pennebaker and King (1999) did a thorough research on

this topic: with a dictionary that classifies words into many categories, the researchers measured the correlation between each category and the five traits. The dictionary is the Linguistic Inquiry and Word Count (LIWC) utility.

Based on the correlation between language traits and OCEAN, researchers have proposed personality recognizer: with texts or speeches of a person, his/her personality can be automatically examined with this system.

2.2.4 Personality Recognizer

There have been some works focusing on building personality recognizer (Mairesse et al., 2007; Oberlander and Nowson, 2006; Celli, 2012; Mohammad and Kiritchenko, 2013; Poria et al., 2013), and most of them are classification recognizers: for example, for the Extraversion trait, the classification model predicts whether a person is an extravert or an introvert. In this thesis, we use the model of Mairesse et al. (2007), since it is the only available model, to our knowledge, that can estimate numerical scores for each trait of OCEAN, instead of binary classifications.

Below we introduce the personality recognizer of Mairesse et al. (2007) in detail.

Dataset The personality recognizer is a data-based model. The essay corpus (Pennebaker and King, 1999) and EAR corpus (Mehl et al., 2006) were used as the training set. The former contains about 2500 essays annotated with OCEAN scores of the respective writers; the latter is a conversational corpus containing about 15000 utterances annotated with OCEAN scores of the respective speakers, and is much smaller than the former.

Structure To recognize the OCEAN score, the researchers tried several regression algorithms, such as linear regression, M5' regression tree, and support vector machine with linear kernels. The features of the model are from LIWC (Pennebaker and King, 1999) and MRC Psycholinguistic database (Coltheart, 1981): the former assigns each word a word category, while the latter contains statistics such as familiarity and frequency of use for each word.

The loss is simply the ratio of the model's prediction's loss and the baseline's loss, so that if the loss is 1, the model's performance is equal to the baseline, while numbers less than 1 means a better performance than the baseline.

Performance For the essay corpus, almost all the ratios are lower than 1, and the estimation for Neuroticism and Openness are significantly better than the baseline. However, the best ratio is 93.58, which seems not so good. For the EAR corpus, the ratios are fluctuated, which may be caused by the relatively small scale of the dataset.

In this thesis, we are going to use this personality recognizer to examine personalities of different speakers based on their responses to the same question set. However, with the above introductions, it is not obvious whether this personality recognizer is reliable for measuring personality or not. We will talk about this worry later in the preliminary experiment in section 5.3.

Chapter 3

Personalized Natural Language Generation

In this chapter, we will introduce related works on personalized natural language generation (NLG).

3.1 Natural Language Generation from Personality Features

First we introduce related works on personalized NLG that generate responses from personality features. Models proposed by the works that we are going to introduce take communicative goals as input, rather than questions. To personalize the generation, parameters related to personality are also inserted into the models, together with the communicative goals.

[Mairesse and Walker \(2010\)](#) conducted the first research on generating distinguished utterances for different personalities. The rule-based generator they built is called PERSONAGE. PERSONAGE is trained on a restaurant dataset and it generates utterances based on different linguistic style parameters ([Mairesse et al., 2007](#)); for example, given a high verbose parameter, it generates more words per utterance.

Although the generator itself is rule-based, a statistic model was applied to cor-

relates the linguistic styles with OCEAN scores. Researchers asked human subjects to estimate the OCEAN score for each utterance generated based on linguistic style parameters, and trained the model with the source being the OCEAN score and the target being linguistic style parameters. The final model works as this: a person with a higher score in Extraversion trait may be more verbose, thus is predicted to have a higher verbose parameter; while a person with a lower score on Extraversion trait may express more uncertainty, thus is predicted to have a higher hedge parameter. With the predicted parameters like these, PERSONAGE chooses responding templates corresponding to all of the parameters, thus is able to generate distinguished utterances for different personalities.

The latest progress for PERSONAGE is done by [Oraby et al. \(2018\)](#), which proposed two neural generation models based on PERSONAGE. The researchers created the training set using PERSONAGE and modified SEQ2SEQ TGen system ([Dušek and Jurčiček, 2016](#)) by adding 1) dialogue acts encoded with personality information, or 2) 32 linguistic style parameters used by PERSONAGE during generating the training set. The models they built are able to generate responses that have relatively high Pearson correlation coefficients with the training set generated by PERSONAGE.

Above works provide models that can generate distinguished utterances given different personalities. The models are limited on the restaurant domain, and can only generate utterances for several communicative goals such as recommendation and comparison, thus do not fit our topic on chit-chat conversations; however, the linguistic styles used for generation are also used in the personality recognizer introduced in section [2.2.4](#), which will be used for evaluating personality differences in our experiments. Thus, the success in generating distinguished utterances for different personalities somehow indicates the validity of the personality recognizer.

3.2 End-to-End Response Generation

Response generation is a sub-field of Natural Language Generation. Response generation for chit-chat CA, which is the focus of our thesis, is mainly about end-to-end

systems that learn to generate responses in conversations by being exposed to large amounts of conversational data.

There are not many works that study personalized response generation; most of the works are generation-based, and the proposed models are modified from standard SEQ2SEQ model. There is one work that is different, which applied multi-task learning that combines a SEQ2SEQ task that generates responses and an AUTOENCODER task that learns embeddings of the target speaker.

SEQ2SEQ model consists of two parts: the encoder and the decoder, where the encoder processes the input and forwards the result to the decoder, with which the decoder generates the outputs. For example, given a context-response pair “Thank you” and “You are welcome”, we have “Thank you” as the source to be inputted to the encoder, and “You are welcome” as the target to be inputted to the decoder. The SEQ2SEQ model will be trained to generate a response that is as similar to “You are welcome” as possible. For details, please see section 4.1.

Due to lack of conversational data with the speakers annotated, three of the works used twitter or scripts as the training set; the other two created their own corpus with volunteers making conversations for them.

Models Labeling Speaker for Each Utterance

The first and most notable work is [Li et al. \(2016a\)](#), where a persona-based SEQ2SEQ model is proposed. They fed corresponding speaker id and addressee id together with the response sequence into the decoder, so that the model knows the speaker and addressee of each response. The persona-based model gains an improvement in both perplexity and BLUE compared to standard SEQ2SEQ model, and is 4.5% better in consistency on human evaluation. Examples of generated responses of their models also show differences between different speakers.

Since existing personalized dataset is relatively small, domain adaption training is often used: a SEQ2SEQ model is first pre-trained on a big-scale conversational dataset, then trained, with most of the parameters from pre-training preserved, on the smaller personalized dataset. This strategy was used by [Li et al. \(2016a\)](#) and [Zhang et al.](#)

(2017); the difference is that the latter trained five models separately for five speakers, which is due to lack of the structure for feeding speaker ids and addressee ids. Generally, the model proposed by the latter has same functions as the former one.

In this thesis, we also apply domain adaption training and similar models with Li et al. (2016a). The disadvantage for the persona-based model is that although it knows which utterance is spoken by whom, it has to balance between general and speaker-specific response generation, so that sometimes personalization has to be sacrificed, which will be examined in our experiment part.

Models Adding Extra Information of Speakers

There are two works that add more concrete information rather than speaker ids into the generation model. One of them is Yang et al. (2017), which add speakers' personal information such as age and gender into a SEQ2SEQ model. Personal information is converted into an one-hot representation and then embedded to a dense vector, after which the vector is fed into the decoder, similar to Li et al. (2016a). The result outperforms standard SEQ2SEQ model on perplexity, BLUE and human evaluation.

The other one is the latest work by Zhang et al. (2018). They first create their own personalized corpus with volunteers; volunteers are asked to act as specific characters described by profiles no longer than five sentences, and each two of the volunteers have a conversation to know each others' character. The researchers provide two kinds of personalized model: information-retrieval-based and generation-based, and add encoded profiles into both of the models. The models both receive better human evaluations.

The models proposed in the above works actually provide specific generation for classes of people. For example, the model of Yang et al. (2017) can generate distinguished responses for females of 20-30 years old; the model of Zhang et al. (2018) generates distinguished responses for different profiles, so even though some profiles belong to the same character, the generated responses for these profiles may be different. We are not going to examine personality differences for these two models, since it is actually examining personality differences between two groups of people, which is theoretically not possible based on the definition of personality.

Model Applying Multi-Task Learning

Finally, the research by [Luan et al. \(2017\)](#) is different with the above works. This research applies multi-task learning: it consists of a SEQ2SEQ task for generating responses, and an AUTOENCODER task for learning the target speaker’s language style. Both the two tasks applies SEQ2SEQ model, while the SEQ2SEQ task is supervised, with questions as the source and responses as the target; the AUTOENCODER task is unsupervised, with the target speaker’s non-conversational sequences as both the source and the target. The parameters for the decoder are shared, which means that the model learns both general response generation and the specific language style of the target speaker. This model gains lower perplexity and higher human evaluation compared to the baseline.

This work could be seen as an extension to [Li et al. \(2016a\)](#) which strengthens personalization; furthermore, since the AUTOENCODER task does not require conversational data, the model also gives a solution to response generation for speakers who do not have enough conversational data.

Note that although the above models may generate responses distinguished in personalities, these models are not able to generate responses given a specific personality like [Mairesse and Walker \(2010\)](#): [Li et al. \(2016a\)](#) and [Luan et al. \(2017\)](#) have speakers’ ids as inputs; [Yang et al. \(2017\)](#) and [Zhang et al. \(2018\)](#) take detailed information of speakers as inputs. We are going to propose a model that fills this gap in section [4.2.2](#).

Chapter 4

Model

In this chapter, we introduce the response generation models used in our experiments. First, we introduce the standard SEQ2SEQ model in section 4.1. After that, we introduce the speaker model proposed by Li et al. (2016a). Finally, we describe the modifications we have made to build our own personality model.

4.1 Standard Model

The standard model is based on SEQ2SEQ model (Sutskever et al., 2014b). Given the source sequence $X = x_1, x_2, \dots, x_m$, a SEQ2SEQ model gives the predicted probability for a target sequence $Y = y_1, y_2, \dots, y_n$ with:

$$P(Y|X) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X) \quad (4.1)$$

The task for the model is to improve $P(Y|X)$ for paired ground-truth X and Y , so that the target \hat{Y} it chooses, which is of the highest probability $P(\hat{Y}|X)$, is preferably close to Y .

We use a SEQ2SEQ LSTM model with attention as the standard model. Below I will introduce how it works in detail.

4.1.1 General Structure

Our standard model-SEQ2SEQ LSTM model with attention-is of an encoder-decoder structure. It takes a context sequence as the source sequence X and a response sequence as the target sequence Y . First X is inputted to the encoder, and the encoder generates hidden vectors to be inserted into the decoder. Next, Y together with the hidden vectors from the encoder are inserted into the decoder, and the decoder gives predictions in the softmax layer. Attention mechanism is an extra structure to improve the model's performance.

Encoder In each encoding step t , a word $x_t \in X$ is inserted to the LSTM unit for generating the corresponding hidden vector h_t .

The scalar vector x_t is first embedded to a dense word-embedding vector $x_t^* \in \mathbb{R}^{d \times 1}$, where d is the number of hidden cells. Same words have same embedding vectors. Then x_t^* is inputted into the first encoding LSTM layer, together with the hidden vector $h_{t-1}^{(1)} \in \mathbb{R}^{d \times 1}$ and the cell state vector $c_{t-1}^{(1)} \in \mathbb{R}^{d \times 1}$ from the first layer of the previous encoding step; if $t = 1$, both $h_{t-1}^{(1)}$ and $c_{t-1}^{(1)}$ are 0 vectors. With the above inputs, the first encoding LSTM layer generates the hidden vector $h_t^{(1)}$ and cell state vector $c_t^{(1)}$ for the current encoding step, which will be forwarded to the next layer and the next step.

Generally, each layer $l > 1$ generates $h_t^{(l)}$ and $c_t^{(l)}$ with 1) the hidden vector $h_t^{(l-1)}$ from the previous layer; 2) the hidden vector $h_{t-1}^{(l)}$ and the cell state vector $c_{t-1}^{(l)}$ from the same layer of the previous step. After inputting the final word from the context sequence, we have the final hidden vectors h_t ($t \in [1, 2, \dots, m]$) from the final layer of each encoding steps:

$$H = \begin{bmatrix} h_1 & h_2 & \dots & h_m \end{bmatrix} \quad (4.2)$$

H will be used in the attention mechanism, which will be explained in section 4.1.3.

Decoder The decoder is similar to the encoder, except additional inputs and a softmax layer.

Similar to the encoder, for each decoding step t , we input an embedding vector $y_t^* \in \mathbb{R}^{d \times 1}$, which is embedded from the word $y_t \in Y$, into the first decoding LSTM layer. Note that Y is different with X in that it always starts with *EOS* and ends with *EOT*, where *EOS* notifies the model it is the end of the source and start of the target, and *EOT* notifies the model it is the end of the target (*EOT* will not be inputted into the decoder).

The hidden vector $h'_{t-1}^{(1)}$ and the cell state vector $c'_{t-1}^{(1)}$ from the first layer of the previous decoding step are inserted together with y_t^* . If $t = 1$, $h'_{t-1}^{(1)}$ and $c'_{t-1}^{(1)}$ are from the first layer of the final encoding step, which are $h_m^{(1)}$ and $c_m^{(1)}$.

Additionally, the context vector c_{t-1}^* from the previous decoding step (for details, please see section 4.1.3) is also inserted to the first decoding LSTM layer. While $t = 1$, the context vector is from the final encoding step.

With y_t^* , $h'_{t-1}^{(1)}$, $c'_{t-1}^{(1)}$, and c_{t-1}^* , the first decoding LSTM layer generates the hidden vector $h'_t^{(1)}$ and cell state vector $c'_t^{(1)}$ for the current decoding step, which will be forwarded to the next layer and the next step. Similar to the encoder, each decoding LSTM layer $l > 1$ generates $h'_t^{(l)}$ and $c'_t^{(l)}$ from $h'_t^{(l-1)}$, $h'_{t-1}^{(l)}$ and $c'_{t-1}^{(l)}$. After the final layer, a predicted vector \hat{h}_t is generated with the final hidden vector h'_t and final encoding hidden vectors H (see section 4.1.3 for details). Then in the softmax layer, the log probability P_t on the whole vocabulary for the next word will be predicted with \hat{h}_t :

$$P_t(w_k) = \log \frac{\exp((W_s)_k \cdot \hat{h}_t)}{\sum_k \exp((W_s)_k \cdot \hat{h}_t)} \quad (4.3)$$

where w_k is a word from the vocabulary V , $k \in [1, |V|]$; $W_s \in \mathbb{R}^{|V| \times d}$.

Training We insert the context sequence into the encoder, and the paired ground-truth response sequence into the decoder. With the log probabilities generated by the decoder for the whole vocabulary, we can get the log probability for each word of a ground-truth response, which is $\log P(y_t | y_1, \dots, y_{t-1}, X)$; our goal is to minimize the sum of the log probability on all the words of ground-truth responses.

Our training procedure is done by subtracting the gradients of the total log probabilities from the weights:

$$Loss(\text{weights}, B) = \frac{1}{|B|} \sum_{b=1}^{|B|} \sum_{t=1}^{n_b} \log P(y_t^b | y_1^b, \dots, y_{t-1}^b, X^b) \quad (4.4)$$

$$\text{weights}' = \text{weights} - \alpha \nabla Loss(\text{weights}, B) \quad (4.5)$$

where B is a batch of paired contexts and responses, X^b and Y^b are paired contexts and responses in B , and n_b is the number of words in Y^b . Batches are samples taken from the training set; all batches have same numbers of elements, and they do not overlap with each other. α is the learning rate; when $\nabla Loss(\text{weights}, B)$ is higher than the clipping threshold T , α will be replaced by:

$$\alpha \times \frac{T}{\|\nabla Loss(\text{weights}, B)\|_2} \quad (4.6)$$

Decoding With the log probabilities for the next word y_t on the whole vocabulary, we first transfer log probabilities into probabilities where $P(y_t | y_1, \dots, y_{t-1}, X) \in [0, 1]$, and then follow Stochastic Greedy Sampling described in [Li et al. \(2017a\)](#) to select the next word among the top 5 words with the highest probabilities.

4.1.2 Long Short Term Memory

Long Short Term Memory (LSTM) ([Hochreiter and Schmidhuber, 1997](#)) is a solution for the gradient exploding and vanishing problem for Recurrent Neural Network. Generally, it controls how much information to keep, input and output through forget gates f , input gates i and output gates o . For each step t , given the previous hidden vector h_{t-1} , cell state vector c_{t-1} , and embedded input x_t^* , the current h_t and c_t are calculated as:

$$i_t = \sigma(W_i \cdot x_t^* + U_i \cdot h_{t-1}) \quad (4.7)$$

$$f_t = \sigma(W_f \cdot x_t^* + U_f \cdot h_{t-1}) \quad (4.8)$$

$$o_t = \sigma(W_o \cdot x_t^* + U_o \cdot h_{t-1}) \quad (4.9)$$

$$l_t = \tanh(W_l \cdot x_t^* + U_l \cdot h_{t-1}) \quad (4.10)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ l_t \quad (4.11)$$

$$h_t = o_t \circ \tanh(c_t) \quad (4.12)$$

where $W_j, U_j \in \mathbb{R}^{d \times d}$ ($j \in \{i, f, o, l\}$). There is a slight difference for the layers $l > 1$, where the input will be the hidden vector from the previous layer, which is h_t^{l-1} , instead x_t^* .

For the decoder, the context vector c_{t-1}^* from the last step is also inputted to the first LSTM layer; so equation 4.8 changes to:

$$i_t = \sigma(W_i \cdot y_t^* + W_i^c \cdot c_{t-1}^* + U_i \cdot h_{t-1}') \quad (4.13)$$

where $W_i^c \in \mathbb{R}^{d \times d}$. Equation 4.9, 4.10, 4.11 changes in a similar way.

4.1.3 Attention Mechanism

Attention mechanism works on the decoder, which helps the decoder to focus on limited important words from the source sequence, instead of the whole source sequence (Bahdanau et al., 2014). The context vector is used both to predict probability in the softmax layer, and to be forwarded to the next step. There are different kinds of attention mechanisms, and what we apply the one from Yao et al. (2015).

For the current step t , by dot multiplying (in other works, dot multiplication may be replaced by other mathematical operations, such as tanh (Luong et al., 2015)) the final encoding hidden vectors $H \in \mathbb{R}^{d \times m}$ (d is the number of hidden cells and m is the length of encoding inputs) with the final hidden vector h_t or h_t' for the current step t (referred as h_t' below), we get the strength indicator v_t :

$$v_t = H^\top \cdot h'_t \quad (4.14)$$

For each encoding input $x_i \in X$, we have the corresponding row of v_t : $v_{t_i} = (H^\top)_i \cdot h'_t$

Then we use a softmax function to get the normalized probability of v_t : $a_t = \text{softmax}(v_t)$. For each $x_i \in X$, we have:

$$a_{t_i} = \frac{\exp(v_{t_i})}{\sum \exp(v_{t_i})} \quad (4.15)$$

Combining the normalized strength indicator a_t with H , we can get the context vector c_t^* :

$$c_t^* = H \cdot a_t \quad (4.16)$$

For each $j \in [1, d]$, we have $c_{t_j}^* = H_j \cdot a_t$. As mentioned in section 4.1.1 and section 4.1.2, c_{t-1}^* from the last step is inserted together with the embedded word $y_t^* \in Y$ to the current decoding step t .

Finally, for each decoding step, we combine the context vector c_t^* with the hidden vector h'_t again. The result \hat{h}_t is then sent to the softmax layer for predicting the log probability of the next word.

$$\hat{h}_t = \tanh(\hat{W}_{c^*} \cdot c_t^* + \hat{W}_h \cdot h'_t) \quad (4.17)$$

where $\hat{W}_{c^*}, \hat{W}_h \in \mathbb{R}^{d \times d}$.

4.2 Extended Models for Personalizing

4.2.1 Speaker Model

This is a persona-based response generation model proposed by Li et al. (2016a), introduced in section 3.2. It can generate distinguished responses given different speakers. Contexts are inputted as the source sequence and responses are inputted as the target sequence. The original codes were written with Torch; we reimplemented it in PyTorch.

The modification is on equation 4.8, 4.9, 4.10, 4.11 and is only in the first layer of LSTM decoder, similar to the modification for the context vector (see equation 4.13 in section 4.1.2). In each decoding step t , besides the embedded word, the context vector, the hidden vector and the cell state vector, which are y_t^* , c_{t-1}^* , h_{t-1} and c_{t-1} , an embedded speaker id vector $s^* \in \mathbb{R}^{d \times 1}$ is also inputted into the model:

$$i_t = \sigma(W_i \cdot y_t^* + W_i^c \cdot c_{t-1}^* + W_i^s \cdot s^* + U_i \cdot h'_{t-1}) \quad (4.18)$$

where $W_i^s \in \mathbb{R}^{d \times d}$. Equation 4.9, 4.10, 4.11 change in a similar way.

So each response Y is paired with the speaker’s id; like word-embedding, the same speaker id s has the same speaker-embedding vector s^* . Since every word from the same response is spoken by the same speaker, same embedded speaker id is inputted multiple times to the decoder for one response.

4.2.2 Personality Model

To address the personality differences among different speakers, and to generate different responses given different personalities measured by OCEAN score, here we propose our personality model, a personality-based response generation model. Contexts are inputted as the source sequence and responses are inputted as the target sequence.

The modification is also on equation 4.8, 4.9, 4.10, 4.11 in the first layer of LSTM decoder, similar to equation 4.18. The difference is that in each decoding step t , instead of inputting the embedded speaker id s^* , we input embedded OCEAN score of the speaker. We first normalized the 5-dimension OCEAN score vector $OCEAN$ from range $[1, 7]$ to $[-1, 1]$, and then embed it with a linear layer:

$$OCEAN^* = W_{OCEAN} \cdot \frac{OCEAN - 4}{3} \quad (4.19)$$

where $W_{OCEAN} \in \mathbb{R}^{d \times 5}$. This procedure ensures the weights of this linear layer, which will be updated during training, is in the same scale with other weights. Next, $OCEAN^*$ is inputted into the first layer of LSTM decoder:

$$i_t = \sigma(W_i \cdot y_t^* + W_i^c \cdot c_{t-1}^* + W_i^{OCEAN} \cdot OCEAN^* + U_i \cdot h'_{t-1}) \quad (4.20)$$

where $W_i^{OCEAN} \in \mathbb{R}^{d \times d}$. Equation 4.9, 4.10, 4.11 change in a similar way.

So each response Y is paired with the speaker's OCEAN score; since each word from the same response is spoken by the same speaker, the decoder is inputted the same $OCEAN^*$ multiple times for one response.

Chapter 5

Experiments

In this chapter, we introduce the three experiments we have conducted. The first experiment was conducted on the speaker model proposed by [Li et al. \(2016a\)](#), which examined personality differences on responses generated by the speaker model for characters from the TV-series dataset. The second and third experiments were all conducted on the personality model that we proposed; the aim of these two experiments was to test if the personality model worked as expected. The second experiment examined personality differences on responses generated for characters from the TV-series dataset, and the third experiment examined personality differences on responses generated for 32 novel extreme personalities.

The structure of this chapter is as follows: In section [5.1](#) we introduce the datasets; in section [5.2](#) we introduce the experimental setup, including the new evaluation method we propose for examining personality differences. In the later sections, we will introduce results for the three experiments one by one.

5.1 Datasets

OpenSubtitles (OSDB) Dataset The OpenSubtitles (OSDb) dataset ([Tiedemann, 2009](#)) is an open-domain dataset containing lines of movie characters. Since none of the lines is annotated with the speaker or addressee, we followed the strategy of [Li et al. \(2016a\)](#), regarding each line as an utterance and two consecutive utterances as one

context-response pair. To ensure that each utterance has enough length, we removed utterances whose length were smaller than 3. We collected 33901903 context-response pairs for the training set and 74368 pairs for the validation set. For the test set, to reduce the size of the set, we set the maximum length of a line to be 7 and removed the utterances containing more than 7 words; 2462019 context-response pairs were collected for the test set.

TV-series Dataset The TV-series dataset contains scripts of two American situation comedy TV-series: *Friends*¹ and *The Big Bang Theory*². Although this dataset is much smaller than the OSDB dataset, it has each line annotated with the speaking character. However, since the addressee is not annotated, we again followed the strategy of Li et al. (2016a), regarding two consecutive lines as one context-response pair, first line as the context sequence and second line as the response sequence. Unlike the OSDB dataset where a context-response pair may contain utterances from different conversations, the TV-series dataset guarantees that only utterances belonging to the same scene are assigned to one pair: since the scripts of situation comedy are divided into several scenes, we are able to determine whether two utterances belong to the same scene or not. We collected 85713 pairs in total, among which about 2000 pairs for the validation set.

To ensure there are enough utterances during training for each character, we kept 13 characters who had more than 2000 utterances, while other characters were all labeled as “other” and were prohibited from appearing as responses. For details, please see table 5.1.

For the speaker model, we assigned each character a different speaker id. For the personality model, we annotated each of the 13 characters with his/her sample-weighted estimated OCEAN score, which was calculated as follows: for each character, we randomly selected 50 samples from his/her utterances—each sample contained 500 utterances—and estimated the OCEAN score for each sample using the personality recognizer mentioned in section 2.2.4; the arithmetic mean of estimated scores for the 50

¹<https://sites.google.com/site/friendstvcorpus/>

²<https://bigbangtrans.wordpress.com/>

samples was the sample-weighted estimated OCEAN score for this character.

Selected		Others	
Character	#Utt	Character	#Utt
Friends			
Rachel	9205	Mike	359
Ross	9019	All	333
Chandler	8357	Richard	281
Monica	8329	Janice	216
Joey	8111	Mr. Geller	204
Phoebe	7449
The Big Bang Theory			
Sheldon	10345	Stuart	551
Leonard	8826	Priya	222
Penny	6822	Mrs. Cooper	213
Howard	5216	Mrs. Wolowitz	193
Raj	3952	Emily	158
Amy	2691	Arthur	130
Bernadette	2198

Table 5.1: Characters and their respective utterance numbers

5.2 Experimental Setup

5.2.1 Training and Decoding

Training Since the TV-series dataset is not large enough for training, we applied the domain adaption training strategy. We first pre-trained a standard model on the larger OSDB dataset; due to the limitation of computation, we trained the model for 15 iterations on the first 1772160 pairs of the OSDB training set. The perplexity of the validation set became stable on the last iterations. Next, keeping the weights, we changed the training set from the OSDB dataset to the TV-series dataset, and trained the model for another 30 iterations, where the perplexity of the validation set of the TV-series dataset became stable for the last iterations.

During the training on the TV-series dataset for the speaker model, besides feeding the response sequence word by word, we also fed the corresponding speaker id together

with each word; for the personality model, we also fed the the speaker’s 5-dimension vector OCEAN score together with each word.

Furthermore, we used similar parameters to [Li et al. \(2016a\)](#) for training:

- Both the speaker model and the personality model are 4-layer LSTM models. Each layer contains 1024 hidden cells.
- We set the batch size to 128.
- Vocabulary size is 25000.
- The max length for an input sentence is 50.
- Parameters are initialized with uniform distribution on $[-0.1, 0.1]$.
- Learning rate is 1.0, and it gets halved after the 6th iteration.
- Threshold for clipping gradients is 5.
- Dropout rate is 0.2.

Decoding We used the test set of the OSDB dataset for decoding, which contained 2462019 contexts. We generated responses on the whole test set for each of the 13 selected characters from the TV-series dataset, both with the speaker model and the personality model. After that, we let the personality model generate responses for each of the 32 novel extreme personalities. Extreme personalities have OCEAN scores where each trait is either extremely high or extremely low; we set “extremely high” to be 6.5 and “extremely low” to be 1.5.

5.2.2 Evaluation

Perplexity

We calculated perplexity on the validation set of the corresponding dataset. Perplexity is the inverse probability of generating the validation set averaged by the word number, so the lower the better:

$$Perplexity = \frac{1}{\sqrt[N]{P(\mathbf{w}_{validation})}}$$

where N is the total number of words in the validation set.

Personality Differences

We used this new evaluation method to measure if there were personality differences among the characters from the TV-series dataset for the original script, and if the speaker model and the personality model were able to generate distinguished responses for different characters or different personalities: we tried to use clustering and classifying algorithms to assign each OCEAN score to the correct character it belonged to, and evaluated the clustering or classifying result by the F1 score and accuracy score.

We used this method to evaluate personality differences among 1) 13 characters from the TV-series dataset; 2) 32 extreme personalities. The procedures are as follows:

Sampling For each character, we randomly selected 50 samples for clustering and 250 samples for classifying, with each sample containing 500 utterances. We then used the personality recognizer mentioned in section 2.2.4 to estimate the OCEAN score for each of the 50 or 250 samples, thus we had 50 or 250 estimated OCEAN scores for one character. For each OCEAN score, we labeled it with the character it belonged to. This is the gold label.

Clustering and Classifying We tried to cluster the OCEAN scores using k-means, agglomerative and spectral clustering, and classify the OCEAN scores using neural networks and support vector machine. We will only report the results of k-means and support vector machine, since their performances are better than others. The algorithms are from scikit-learn ^{3,4}.

For clustering, we first clustered the OCEAN scores into several different clusters; the number of clusters was equal to the number of characters. Next, we labeled each

³k-means: <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

⁴support vector machine: <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

cluster with a different character; this predicted label maximized the purity score. Finally, we calculated the F1 score and accuracy score for the predicted label compared with the gold label.

For classifying, we applied 5-fold on the whole set: we trained the classifying model on the set 5 times, every time 80% of the set were taken as training set while the remaining 20% were the test set. We calculated the F1 score and accuracy score for the label predicted by the classifying model compared with the gold label, and then averaged the scores over 5.

Statistic Significance and Baseline We did the above procedures for 10 iterations on the original scripts (*Friends* and *The Big Bang Theory* respectively), as well as on responses generated by the speaker model and the personality model. With the F1 scores and accuracy scores for the original scripts, the speaker model and the personality model, we did Levene’s test and t-test to compare these scores respectively with the scores of the random baseline. The random baseline was created by randomizing the gold label in the **sampling** step. With Levene’s test to determine whether the population variances were equal or not, we applied independent two sample test if the population variances were equal, and Welch’s t-test if not equal.

5.3 Preliminary Experiment: Examining Personality Differences for the Original Scripts

For the experiments that examine personality differences for the speaker model and the personality model, we have two worries: 1) as mentioned in section 2.2.4, we are not sure if the personality recognizer can work as expected or not; 2) we are not sure if there are indeed personality differences among the 13 characters from TV-series dataset for the original scripts. 1) is the basis of all the experiments, while for 2), if there are even no personality differences shown in the original scripts, we can expect neither the speaker model nor the personality model to generate responses distinguished in personality for each character.

Thus, to deal with the two worries, we first evaluated personality differences for the original scripts by calculating the F1 score and accuracy score following the above evaluation steps in section 5.2.2, and comparing the results with the random baseline.

Friends

6 characters were selected from *Friends*. Table 5.2 shows the average overall F1 score and accuracy score over 10 iterations, and table 5.3 shows statistical analysis of F1 scores for the original scripts with respect to the baseline.

Overall Score	Algorithm	Score	Baseline	Script
	k-means	F1	0.228	0.470
		Accuracy	0.234	0.478
	SVM	F1	0.164	0.606
Accuracy		0.169	0.610	

Table 5.2: Average overall F1 scores and accuracy scores on *Friends* for the original scripts

	p-value	Cohen's d
k-means	$3.03 \times 10^{-15**}$	10.9
SVM	$1.22 \times 10^{-23**}$	32.3

Table 5.3: Statistical results on *Friends* for the original scripts with respect to the baseline

Scores higher than 0.5 are colored red. Remember that the baseline is created by replacing the gold label with the random label; thus, if the estimated OCEAN scores for each character are not distinguished from other characters, the overall F1 score and accuracy score should not be significantly different from the baseline. However, both the overall F1 score and accuracy score for the original scripts are higher than the baseline significantly, which indicates that the OCEAN scores estimated for each character for the original scripts are distinguished with those for other characters.

Thus the above two worries are solved: 1) the personality recognizer works well: if it generates random scores, or always generates similar scores, the F1 scores and accuracy scores for the original scripts should not be significantly different from the baseline; 2)

the original scripts for different characters are able to reflect distinguished personalities to some extent.

The Big Bang Theory

7 characters were selected from *The Big Bang Theory*. Table 5.4 shows the average overall F1 score and accuracy score over 10 iterations, and table 5.5 shows statistical analysis of F1 scores for the original scripts with respect to the baseline.

Overall Score	Algorithm	Score	Baseline	Script
	k-means	F1	0.208	0.621
		Accuracy	0.210	0.625
	SVM	F1	0.140	0.683
Accuracy		0.144	0.690	

Table 5.4: Average overall F1 scores and accuracy scores on *The Big Bang Theory* for the original scripts

	p-value	Cohen’s d
k-means	$3.70 \times 10^{-19**}$	18.2
SVM	$8.75 \times 10^{-28**}$	55.0

Table 5.5: Statistical results on *The Big Bang Theory* for the original scripts with respect to the baseline

Scores higher than 0.5 are colored red. The result is even better than *Friends*, so we can again infer that: 1) the personality recognizer works well; 2) the original scripts for different characters are able to reflect distinguished personalities to some extent; 3) the original scripts for each character from *The Big Bang Theory* are more distinguished than those from *Friends*.

5.4 Experiment 1: Examining Personality Differences for the Speaker Model

We examined if the responses generated by the speaker model for 13 characters from the TV-series dataset reflected personality differences. First we report the perplexity of the speaker model together with the standard model:

	Standard Model	the speaker model
Perplexity	40.68	38.78

Table 5.6: Perplexity on the TV-series validation set for the standard model and the speaker model

Although there is a decrease on perplexity for the speaker model compared to the standard model, the difference is not so significant as reported in Li et al. (2016a). This may be caused by the small size of the TV-series dataset, or reducing of the OSDB dataset in section 5.2.1.

5.4.1 Experimental Procedure

The first step of this experiment is to generate responses for each character; we let the speaker model generate responses for each of the 13 characters using the OSDB test set as inputs, as mentioned in section 5.2.1 (for examples of generated responses, see appendix A.1). After that, we did a second step: cleaning the generated responses by removing the general responses.

The generated responses have a lot of general responses such as “I know”. Since the general responses are the same for each character, it is likely that the samples for different characters are all filled with general responses, which makes it impossible for examining personality differences. To get rid of the general responses, we tried 3 methods with different parameters, which are: 1) removing the n top common responses over all characters; 2) removing the n top common responses individually for each character; 3) removing all responses with frequency higher than n in any single character’s responses.

For each of these methods, we first cleaned the responses using this method, and then went through the first two steps of examining personality differences, including **sampling** and **clustering**. For selecting these parameters, we did clustering on all of the 13 characters using k-means, and calculated the average overall F1 scores. The result is shown in table 5.7.

We selected the method that returned the highest F1 score, which was removing the top 100 common responses over all characters. After cleaning, we had about 700000

Method \ Parameter	100	200	300	500	1000	2000
Removing common responses over all	0.271	0.261	0.260	\	\	\
Removing common responses individually	0.244	0.243	0.242	\	\	\
Removing responses with frequency $> n$	\	\	\	0.247	0.250	0.247

Table 5.7: Average F1 scores on the speaker model with different cleaning methods

responses left for each character.

Next, we followed the steps mentioned in section 5.2.2 for evaluating personality differences. We sampled and estimated OCEAN scores for the cleaned responses, and then clustered & classified these OCEAN scores (for estimated OCEAN scores, see appendix A.2). For this step, we selected the clustering & classifying algorithms that gave best scores, which are k-means for clustering and support vector machine for classifying. We applied model selection on both of the two algorithms. For classifying, we have two options: 1) train separate classification models on the original scripts and on the speaker model; 2) train the classification model on the original scripts, and use it to do classifications on the speaker model. We tried both of the two options; in the result part, 1) will be referred as SVM, and 2) will be referred as SVM*.

Finally we calculated F1 scores and accuracy scores on the clustering & classifying results for 10 iterations, and compared the scores for the original scripts and the baseline.

5.4.2 Results

Friends

Same as the original scripts, 6 characters were selected from *Friends*. Table 5.8 shows the average F1 scores and accuracy scores over 10 iterations, and table 5.9 shows statistical analysis of F1 scores for the speaker model, with respect to the baseline and the original scripts.

Scores higher than 0.5 are colored red. Similar to the analysis of the original scripts

Character	Algorithm	Score	Baseline	Script	Speaker
Overall	k-means	F1	0.228	0.470	0.318
		Accuracy	0.234	0.478	0.321
	SVM	F1	0.164	0.606	0.322
		Accuracy	0.169	0.610	0.327
	SVM*	F1	0.160	\	0.189
		Accuracy	0.165	\	0.196
Rachel	k-means	F1	0.291	0.340	0.346
	SVM	F1	0.200	0.372	0.229
	SVM*	F1	0.174	\	0.127
Ross	k-means	F1	0.223	0.740	0.280
	SVM	F1	0.188	0.787	0.279
	SVM*	F1	0.168	\	0.352
Chandler	k-means	F1	0.220	0.374	0.328
	SVM	F1	0.156	0.466	0.368
	SVM*	F1	0.168	\	0.242
Monica	k-means	F1	0.180	0.589	0.270
	SVM	F1	0.172	0.645	0.338
	SVM*	F1	0.146	\	0.190
Joey	k-means	F1	0.252	0.534	0.300
	SVM	F1	0.141	0.739	0.312
	SVM*	F1	0.160	\	0.220
Phoebe	k-means	F1	0.211	0.290	0.400
	SVM	F1	0.156	0.649	0.434
	SVM*	F1	0.174	\	0.046

Table 5.8: Average F1 and accuracy score on *Friends* for the original scripts and the speaker model

			Baseline	Script
the speaker model	k-means	p-value	$6.23 \times 10^{-10**}$	$6.08 \times 10^{-11**}$
		Cohen’s d	5.30	6.11
	SVM	p-value	$2.87 \times 10^{-18**}$	$9.60 \times 10^{-20**}$
		Cohen’s d	16.2	19.6
	SVM*	p-value	$4.44 \times 10^{-5**}$	$3.12 \times 10^{-22**}$
		Cohen’s d	2.39	27.0

Table 5.9: Statistical results on *Friends* for the speaker model

in section 5.3, since the overall F1 score on the speaker model is higher than the baseline significantly, we know that the OCEAN scores estimated for the generated responses of each character are distinguished from those of other characters; furthermore, we can

infer that the speaker model is able to generate responses that are distinguished in personality.

Moreover, from table 5.8 and 5.9, we can see that although the overall F1 score for the original scripts and the speaker model are all significantly higher than the baseline, the speaker model has a significantly worse score than the original scripts. This indicates that the responses for different characters generated by the speaker model are less distinguished than the original scripts. This may be caused by 1) the influence of general responses that have not been totally cleaned; 2) that the original scripts are not distinguished enough for different characters, so that the speaker model, which takes the original scripts as the training set, is not able to learn the differences very well.

Finally, note that for SVM*, the classification model is trained on the original scripts while predicts classifications for responses generated by the speaker model. This method examines if the OCEAN score estimated for a specific character whose responses are generated by the speaker model, is similar to the OCEAN score estimated for this character based on the original scripts. That is to say, whether the speaker model can generate tailor-made responses for this specific character or not. The scores are higher than the baseline significantly, which means that the speaker model captured some nature of the personality estimated for each character, however not much, since the effect size is low (Cohen's $d= 2.39$) over all characters. Also the scores for SVM* is lower than SVM, which means that although the speaker model can generate distinguished responses for different characters, these responses are not exactly tailor-made for those characters. The reasons may be that 1) the TV-series dataset is relatively small; 2) the estimations of OCEAN scores for the characters are not so precise.

Several figures follow the table, which are visualization of the clustering results, with data decomposed by PCA into 2 dimensions. Figure 5.1 shows the predicted label of the 6 characters for the original scripts, and the differences between the predicted label and the gold label. Each character has 50 samples. Similarly, figure 5.2 shows the predicted label for the speaker model and its differences with the gold label.

The figures reflect how the speaker model performs compared to the original scripts.

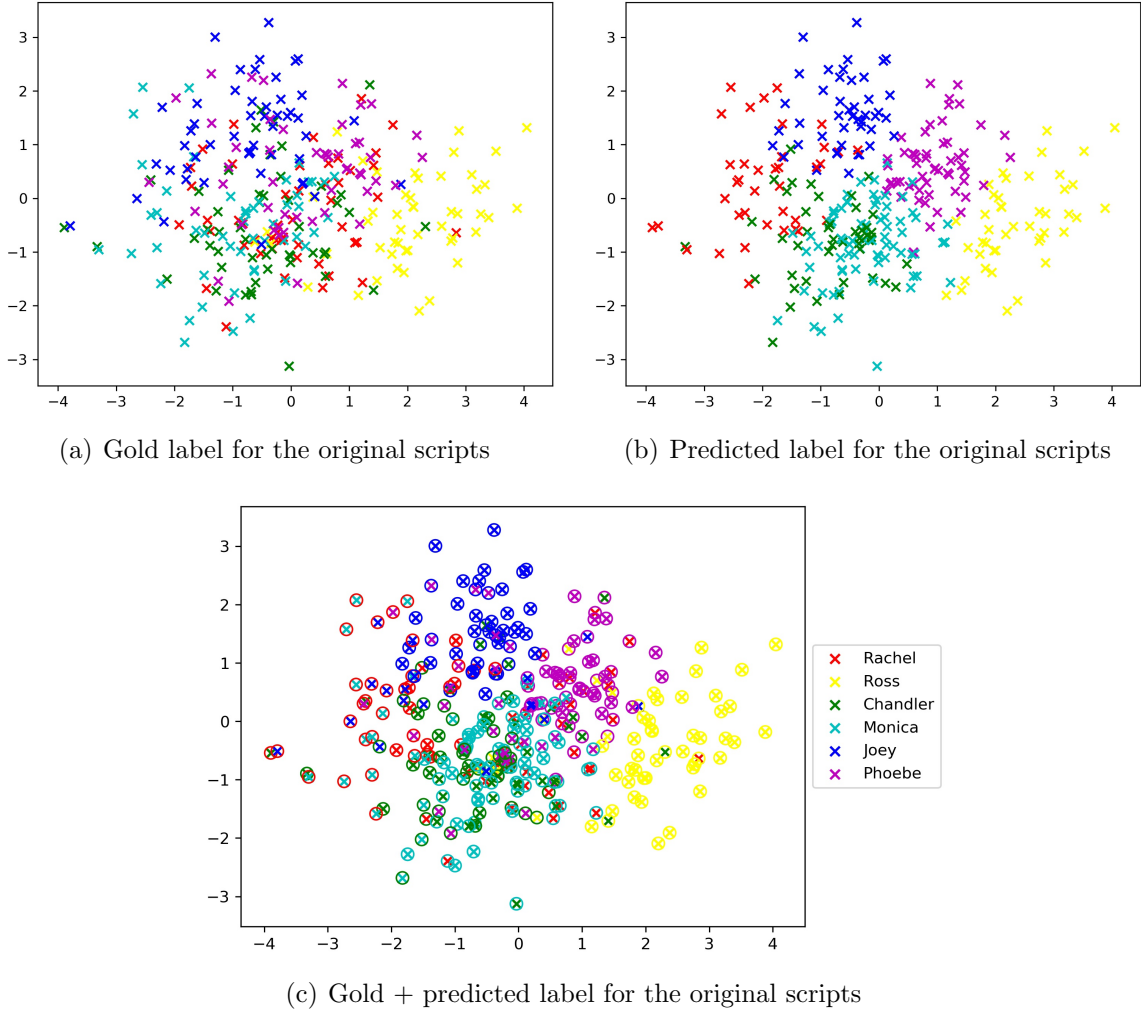


Figure 5.1: Gold label and predicted label on *Friends* for the original scripts

Figure 5.1(c) and 5.2(c) combine the predicted label with the gold label: the crosses are the gold label, while the circles are the predicted label. Thus, more messy the color, worse the performance. Due to the composition of dimension, the figures are not able to show all the details.

Figure 5.1(a) shows the gold label for the original scripts, which is not as messy as the gold label for the speaker model (see figure 5.2(a)). This fact indicates that the original scripts for each character are more distinguished than the generated responses for each character. Also, the distribution of dots for each character in different figures are different, which indicates that the personality reflected by the original scripts and personality reflected by the generated responses for each character are different. This

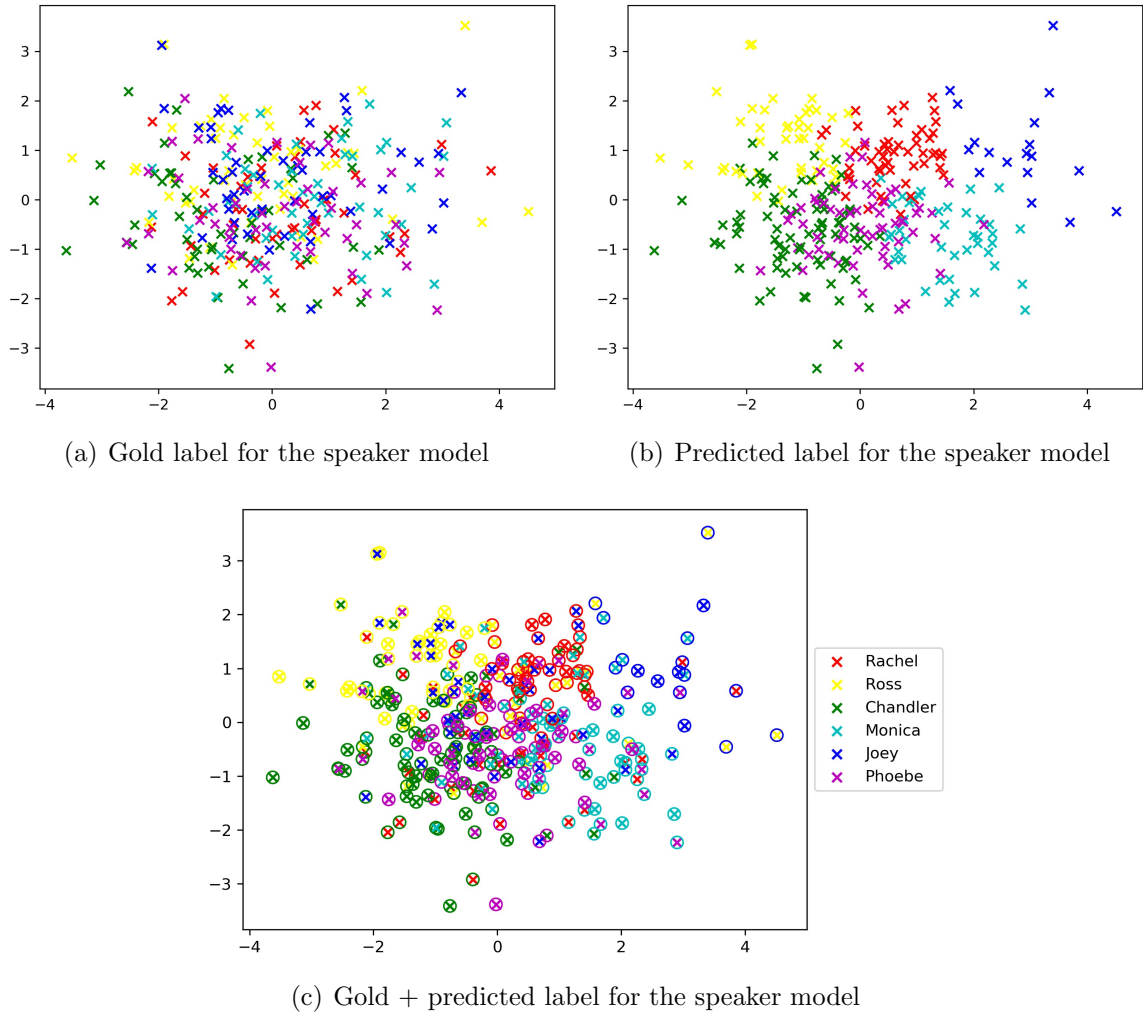


Figure 5.2: Gold label and predicted label on *Friends* for the speaker model

is a visualization of the result of SVM*.

Finally, it is clearly shown that some characters received more correct predictions; however, these characters are not always consistent with the result shown in table 5.8: a character with more correct predictions in the figures may not have a higher average score. The reason is that 1) the figure was created based on one of the 10 iterations, which resulted in fluctuation on the scores; 2) the characters from *Friends* are less distinguished, since the same problem is not severe for *The Big Bang Theory*.

The Big Bang Theory

Same as the original scripts, 7 characters were selected from *The Big Bang Theory*. Table 5.10 shows the average F1 scores and accuracy score over 10 iterations, and table 5.11 shows statistical analysis of F1 scores for the speaker model with respect to the baseline and the original scripts.

Character	Algorithm	Score	Baseline	Script	Speaker
Overall	k-means	F1	0.208	0.621	0.423
		Accuracy	0.210	0.625	0.421
	SVM	F1	0.140	0.683	0.464
		Accuracy	0.144	0.690	0.473
	SVM*	F1	0.139	\	0.187
		Accuracy	0.144	\	0.183
Sheldon	k-means	F1	0.214	0.914	0.720
	SVM	F1	0.185	0.932	0.869
	SVM*	F1	0.160	\	0.411
Leonard	k-means	F1	0.208	0.302	0.374
	SVM	F1	0.169	0.344	0.440
	SVM*	F1	0.163	\	0.031
Penny	k-means	F1	0.219	0.502	0.590
	SVM	F1	0.142	0.671	0.747
	SVM*	F1	0.160	\	0.034
Howard	k-means	F1	0.217	0.632	0.236
	SVM	F1	0.144	0.692	0.219
	SVM*	F1	0.146	\	0.326
Raj	k-means	F1	0.186	0.602	0.304
	SVM	F1	0.137	0.687	0.276
	SVM*	F1	0.134	\	0.193
Amy	k-means	F1	0.214	0.684	0.444
	SVM	F1	0.122	0.718	0.484
	SVM*	F1	0.130	\	0.076
Bernadette	k-means	F1	0.205	0.736	0.280
	SVM	F1	0.113	0.780	0.283
	SVM*	F1	0.114	\	0.210

Table 5.10: Average F1 scores and accuracy score on *The Big Bang Theory* for the original scripts and the speaker model

Scores higher than 0.5 are colored red. Similar to *Friends*, we can infer: 1) the speaker model is able to generate responses that are distinguished in personality; 2)

			Baseline	Script
the speaker model	k-means	p-value	$2.45 \times 10^{-19**}$	$1.55 \times 10^{-13**}$
		Cohen’s d	18.6	8.70
	SVM	p-value	$3.70 \times 10^{-24**}$	$5.92 \times 10^{-20**}$
		Cohen’s d	34.6	20.1
	SVM*	p-value	$1.15 \times 10^{-7**}$	$3.42 \times 10^{-24**}$
		Cohen’s d	3.77	34.7

Table 5.11: Statistical results on *The Big Bang Theory* for the speaker model

responses for different characters generated by the speaker model are less distinguished than the original scripts.

Moreover, it can be observed from table 5.10 that the overall F1 score and accuracy score for the original scripts and the speaker model on *The Big Bang Theory* is higher than *Friends*, while the baseline is lower than *Friends*. Lower score for the baseline is due to the increasing of characters: 7 characters for *The Big Bang Theory* and 6 characters for *Friends*. The scores under SVM* for the speaker model is similar to *Friends*, yet due to the lower baseline, the scores are actually better than *Friends*.

The reason for a higher overall score compared to *Friends* is that 1) the original scripts are more distinguished among all characters: 6 out of 7 characters have F1 scores higher than 0.5 for the original scripts, while this number on *Friends* is 3 out of 6; 2) there is a highly distinguished character, which results in the improvement of the overall score. With the more distinguished original scripts as the training set, the speaker model may learn better on the differences between characters, thus generate more distinguished responses for each character.

It is notable that the character “Sheldon” gained a very high F1 score for the speaker model, from which we can infer that the speaker model can capture the nature of a character, if the corresponding original scripts are distinguished enough, which can be measured by the F1 score and accuracy score.

Several figures follow the table, which are visualization of the clustering results, with data decomposed by PCA into 2 dimensions. Figure 5.3 shows the predicted label of the 7 characters for the original scripts, and the differences between the predicted label and the gold label. Each character has 50 samples. Similarly, figure 5.4 shows the

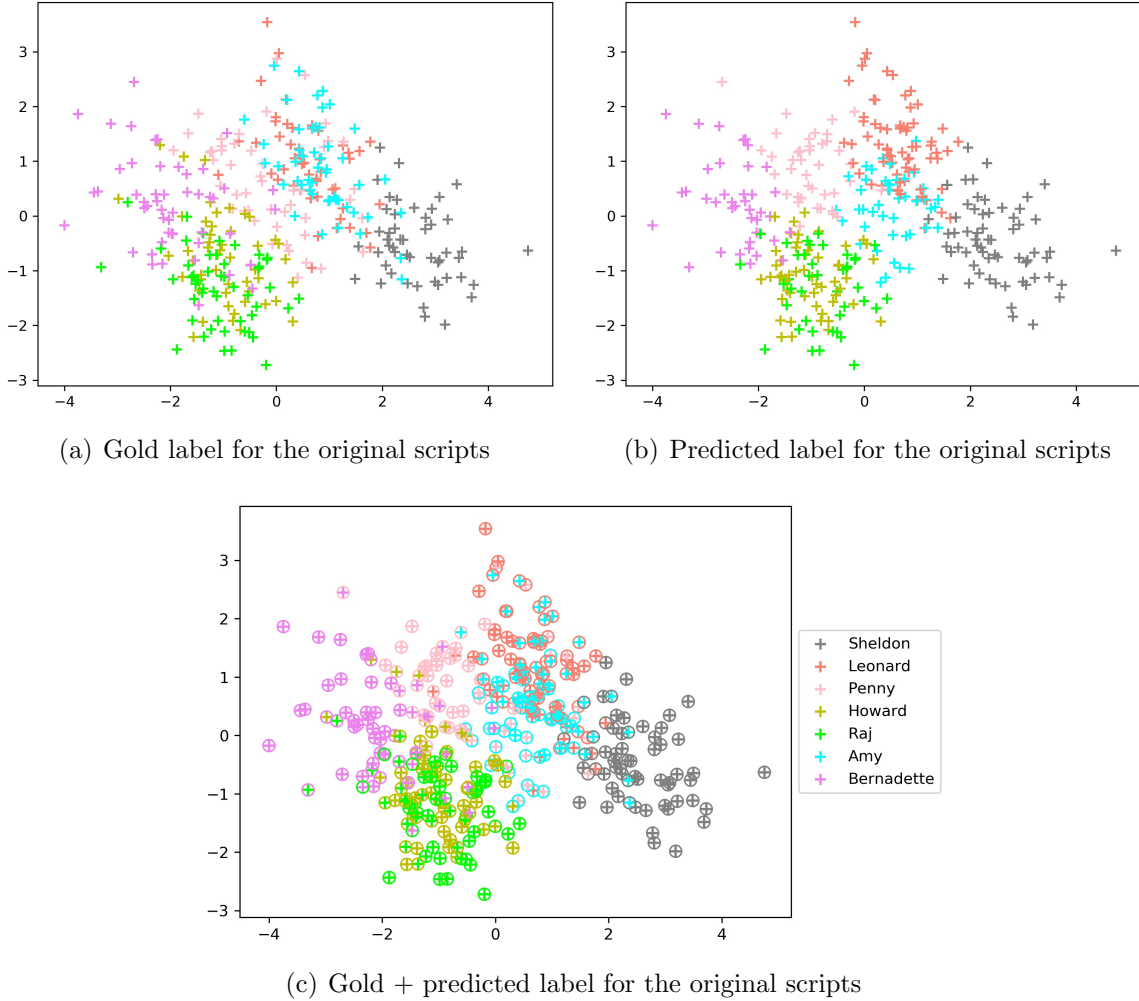


Figure 5.3: Gold label and predicted label on *The Big Bang Theory* for the original scripts

predicted label for the speaker model and its difference with the gold label.

Figure 5.3(a) shows the gold label for the original scripts. It is less messy than the gold label on *Friends* in figure 5.1(a), which indicates that the original scripts of *The Big Bang Theory* for each character are more distinguished. Similar to *Friends*, the gold label for the speaker model (see figure 5.4(a)) is more messy, showing that the original scripts for each character are more distinguished than the responses generated by the speaker model. The distributions of dots for each character in the 2 figures of the gold labels are also different, except the character “Sheldon”. This indicates that the speaker model can capture the nature of a very distinguished character, since the

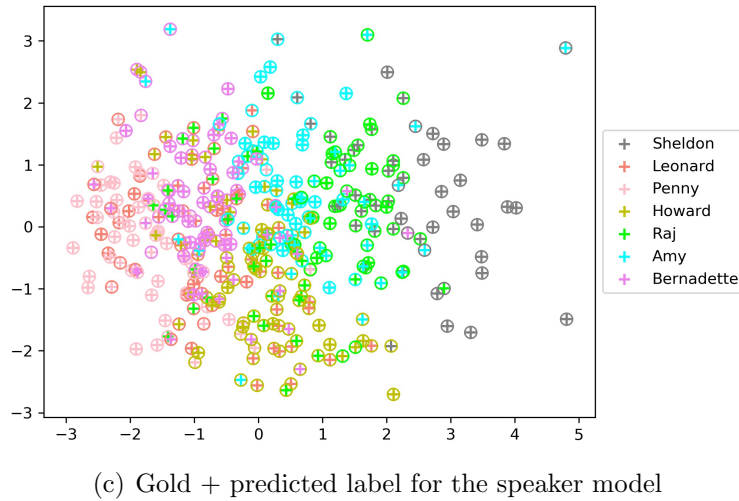
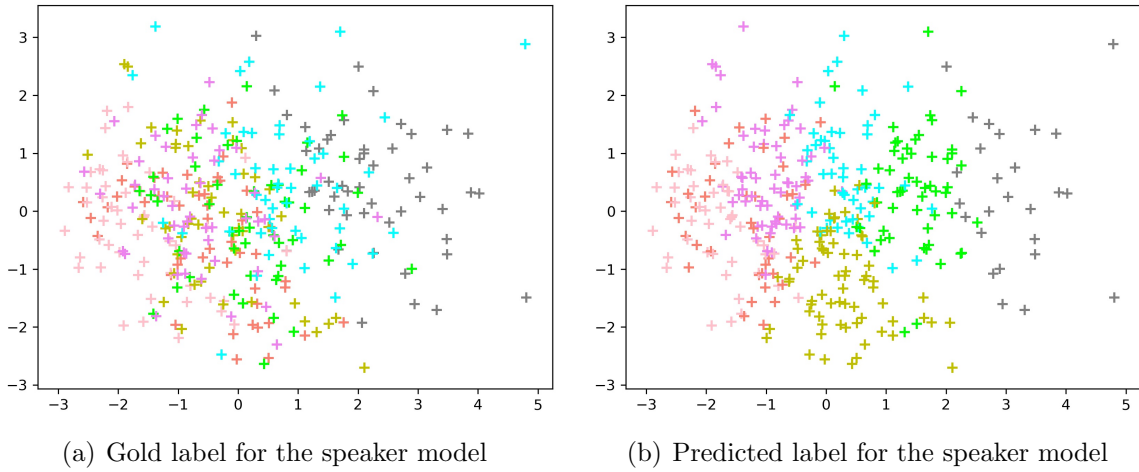


Figure 5.4: Gold label and predicted label on *The Big Bang Theory* for the speaker model

generated responses have similar OCEAN scores with the original scripts, which is same to the result of SVM* in table 5.10.

As mentioned in the *Friends* section, not like *Friends*, the characters in *The Big Bang Theory* that received more correct predictions in the figures also had higher average scores in table 5.10.

5.5 Experiment 2 & 3: Examining Personality Differences for the Personality Model

5.5.1 13 Characters from the TV-series Dataset

We examined if the responses generated by the personality model for 13 characters from TV-series dataset reflected personality differences. First we show the perplexity of the personality model:

	Standard LSTM	the speaker model	the personality model
Perplexity	40.68	38.78	38.63

Table 5.12: Perplexity on the TV-series validation set for standard LSTM model, the speaker model and the personality model

The personality model has a lower perplexity than the speaker model, but the difference is not significant.

The experimental procedures for this experiment is similar to experiment 1 (for examples of generated responses and estimated OCEAN scores, see appendix A.1 and A.2). It may worth noting that we also cleaned the generated responses by removing the top 100 common responses over all characters.

Friends

Same as the speaker model, 6 characters were selected from *Friends*. Table 5.13 shows the average F1 scores and accuracy scores over 10 iterations, which is similar to table 5.8 in experiment 1, but different at that there is an extra column for personality scores. Table 5.14 shows statistical analysis of F1 scores for the personality model with respect to the baseline, the original scripts, and the speaker model.

Like before, scores higher than 0.5 are colored red. Similar to the analysis for the speaker model in section 5.4, since the overall F1 score for the personality model is higher than the baseline significantly, we can know that the OCEAN scores estimated for the generated responses of each character are distinguished from those of other characters; furthermore, we can infer that the personality model is able to generate

Character	Algorithm	Score	Baseline	Script	Speaker	Personality
Overall	k-means	F1	0.228	0.470	0.318	0.265
		Accuracy	0.234	0.478	0.321	0.268
	SVM	F1	0.164	0.606	0.322	0.214
		Accuracy	0.169	0.610	0.327	0.221
	SVM*	F1	0.160	\	0.189	0.185
		Accuracy	0.165	\	0.196	0.204
Rachel	k-means	F1	0.291	0.340	0.346	0.240
	SVM	F1	0.200	0.372	0.229	0.188
	SVM*	F1	0.174	\	0.127	0.147
Ross	k-means	F1	0.223	0.740	0.280	0.326
	SVM	F1	0.188	0.787	0.279	0.386
	SVM*	F1	0.168	\	0.352	0.472
Chandler	k-means	F1	0.220	0.374	0.328	0.263
	SVM	F1	0.156	0.466	0.368	0.195
	SVM*	F1	0.168	\	0.242	0.221
Monica	k-means	F1	0.180	0.589	0.270	0.260
	SVM	F1	0.172	0.645	0.338	0.257
	SVM*	F1	0.146	\	0.190	0.235
Joey	k-means	F1	0.252	0.534	0.300	0.280
	SVM	F1	0.141	0.739	0.312	0.152
	SVM*	F1	0.160	\	0.220	0.094
Phoebe	k-means	F1	0.211	0.290	0.400	0.232
	SVM	F1	0.156	0.649	0.434	0.150
	SVM*	F1	0.174	\	0.046	0.054

Table 5.13: Average F1 and accuracy score on *Friends* for the original scripts, the speaker model and the personality model

			Baseline	Script	Speaker
the per- sonality model	k-means	p-value	$2.48 \times 10^{-5**}$	$1.52 \times 10^{-13**}$	$6.15 \times 10^{-6**}$
		Cohen's d	2.51	8.71	2.82
	SVM	p-value	$1.09 \times 10^{-7**}$	$5.02 \times 10^{-21**}$	$1.23 \times 10^{-12**}$
		Cohen's d	3.79	23.1	7.71
	SVM*	p-value	$1.14 \times 10^{-6**}$	$1.55 \times 10^{-22**}$	0.19
		Cohen's d	3.20	28.1	0.339

Table 5.14: Statistical results on *Friends* for the personality model

responses that are distinguished in personality.

However, although the F1 scores and accuracy scores for the personality model are all higher than the baseline, the personality model has a significantly worse score than

the speaker model. This indicates that the responses for different characters generated by the personality model are less distinguished than the speaker model, which is what to be expected, since the personality model generates responses for a class of speakers with similar personalities, while the speaker model generates responses for specific speakers.

Table 5.13 lists F1 score of each character. We can see that although the F1 score of any character for the original scripts or the speaker model is higher than the baseline, this is not the case for the personality model. The F1 scores of some characters for the personality model are lower than the baseline, which indicates that the responses generated for these characters are not as distinguished as others.

Finally, for SVM*, we have some different results. Note that this algorithm examines if the OCEAN score estimated for a specific character whose responses are generated by the personality model, is similar to the OCEAN score estimated for this character based on the original scripts. The personality model gained similar scores with the speaker model, which means that although the responses generated by the personality model are not as distinguished as the speaker model, the personality model has a same level of capability to capture the nature of the personality estimated for the characters.

Like before, several figures follow the table, which are visualization of the clustering results with data decomposed by PCA into 2 dimensions. Figure 5.5 shows the predicted label for the personality model and its difference with the gold label. Each character has 50 samples.

The figures reflect how the personality model performs. Compared to the previous figures for the original scripts and the speaker model (figure 5.1(c) and 5.2(c)), the figure for the personality model is more messy, which is consistent with the result in table 5.13 that the personality model performs worse.

The Big Bang Theory

Same as the speaker model, 7 characters were selected from *The Big Bang Theory*. Table 5.15 shows the average F1 scores and accuracy scores over 10 iterations, which is similar to table 5.10 in experiment 1, except an extra column for personality scores. Table 5.16 shows statistical analysis of F1 scores for the personality model, with respect

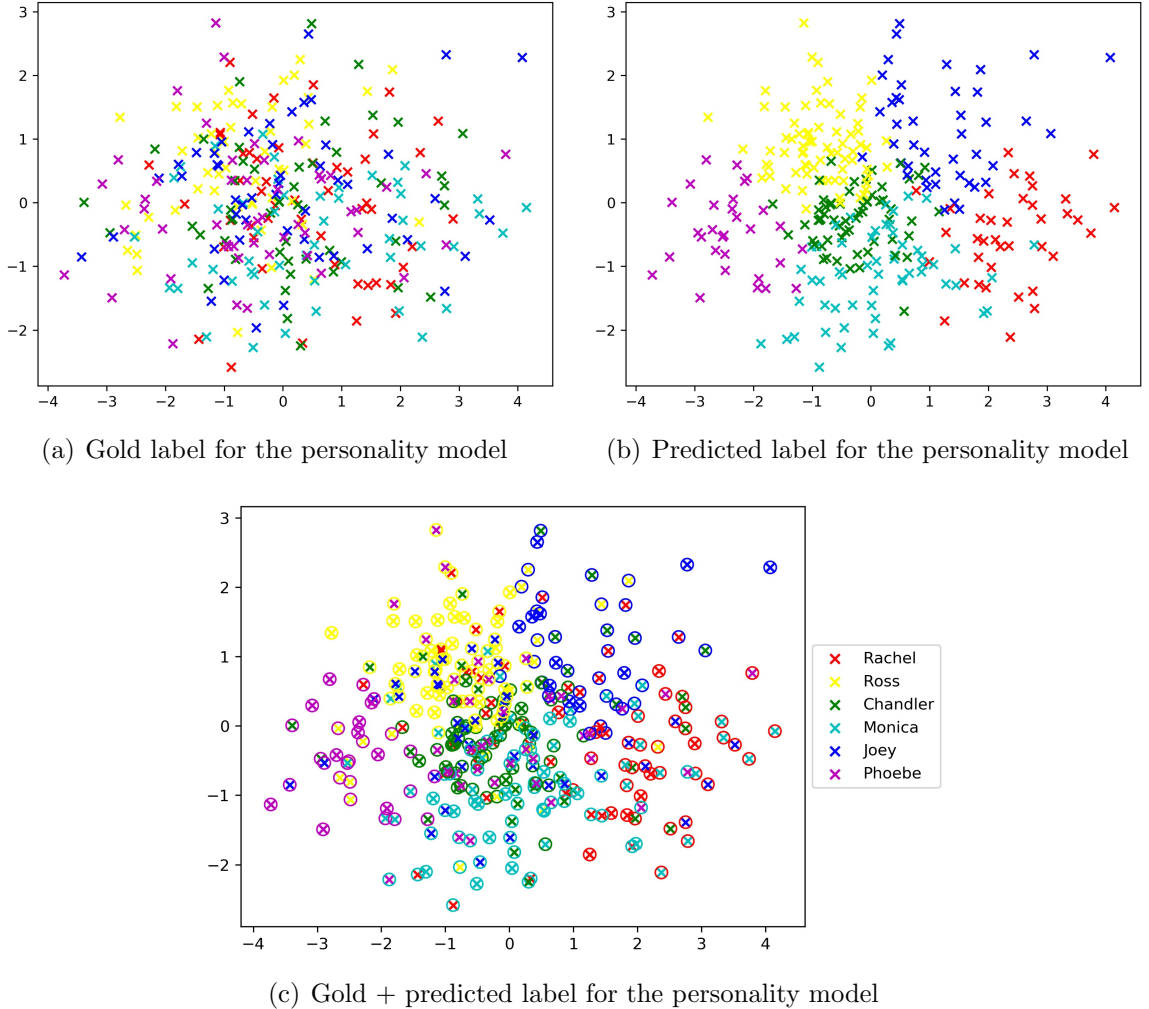


Figure 5.5: Gold label and predicted label on *Friends* for the personality model

to the baseline, the original scripts and the speaker model.

Like before, scores higher than 0.5 are colored red. Generally, the scores for the personality model, except SVM*, are worse than the speaker model, and better than *Friends*. Similar to *Friends*, we can infer: 1) the personality model is able to generate responses that are distinguished in personality; 2) responses for different characters generated by the personality model are the least distinguished.

For SVM*, the overall F1 score for the personality model is the first time higher than the speaker model significantly. We can infer that although the responses generated by the personality model are not as distinguished as the speaker model, the personality model does better in capturing the nature of personality estimated for the characters,

Character	Algorithm	Score	Baseline	Script	Speaker	Personality
Overall	k-means	F1	0.208	0.621	0.423	0.310
		Accuracy	0.210	0.625	0.421	0.305
	SVM	F1	0.140	0.683	0.464	0.300
		Accuracy	0.144	0.690	0.473	0.312
	SVM*	F1	0.139	\	0.187	0.213
		Accuracy	0.144	\	0.183	0.215
Sheldon	k-means	F1	0.214	0.914	0.720	0.580
	SVM	F1	0.185	0.932	0.869	0.861
	SVM*	F1	0.160	\	0.411	0.497
Leonard	k-means	F1	0.208	0.302	0.497	0.282
	SVM	F1	0.169	0.344	0.440	0.203
	SVM*	F1	0.163	\	0.031	0.094
Penny	k-means	F1	0.219	0.502	0.590	0.224
	SVM	F1	0.142	0.671	0.747	0.153
	SVM*	F1	0.160	\	0.034	0.182
Howard	k-means	F1	0.217	0.632	0.236	0.234
	SVM	F1	0.144	0.692	0.219	0.190
	SVM*	F1	0.146	\	0.326	0.285
Raj	k-means	F1	0.186	0.602	0.304	0.226
	SVM	F1	0.137	0.687	0.276	0.148
	SVM*	F1	0.134	\	0.193	0.177
Amy	k-means	F1	0.214	0.684	0.444	0.268
	SVM	F1	0.122	0.718	0.484	0.265
	SVM*	F1	0.130	\	0.076	0.010
Bernadette	k-means	F1	0.205	0.736	0.280	0.324
	SVM	F1	0.113	0.780	0.283	0.360
	SVM*	F1	0.114	\	0.210	0.261

Table 5.15: Average F1 scores and accuracy score on *The Big Bang Theory* for the original scripts, the speaker model and the personality model

			Baseline	Script	Speaker
the per- sonality model	k-means	p-value	$6.37 \times 10^{-10**}$	$4.08 \times 10^{-16**}$	$1.75 \times 10^{-10**}$
		Cohen's d	6.34	12.2	6.98
	SVM	p-value	$7.86 \times 10^{-11**}$	$8.56 \times 10^{-21**}$	$1.94 \times 10^{-14**}$
		Cohen's d	9.84	22.4	9.81
	SVM*	p-value	$5.79 \times 10^{-14**}$	$2.08 \times 10^{-26**}$	$3.83 \times 10^{-4**}$
		Cohen's d	9.21	46.1	-1.95

Table 5.16: Statistical results on *The Big Bang Theory* for the personality model

especially these characters that are distinguished in the original scripts. This may be caused by more distinguished original scripts for each character, especially the character “Sheldon”, than *Friends*.

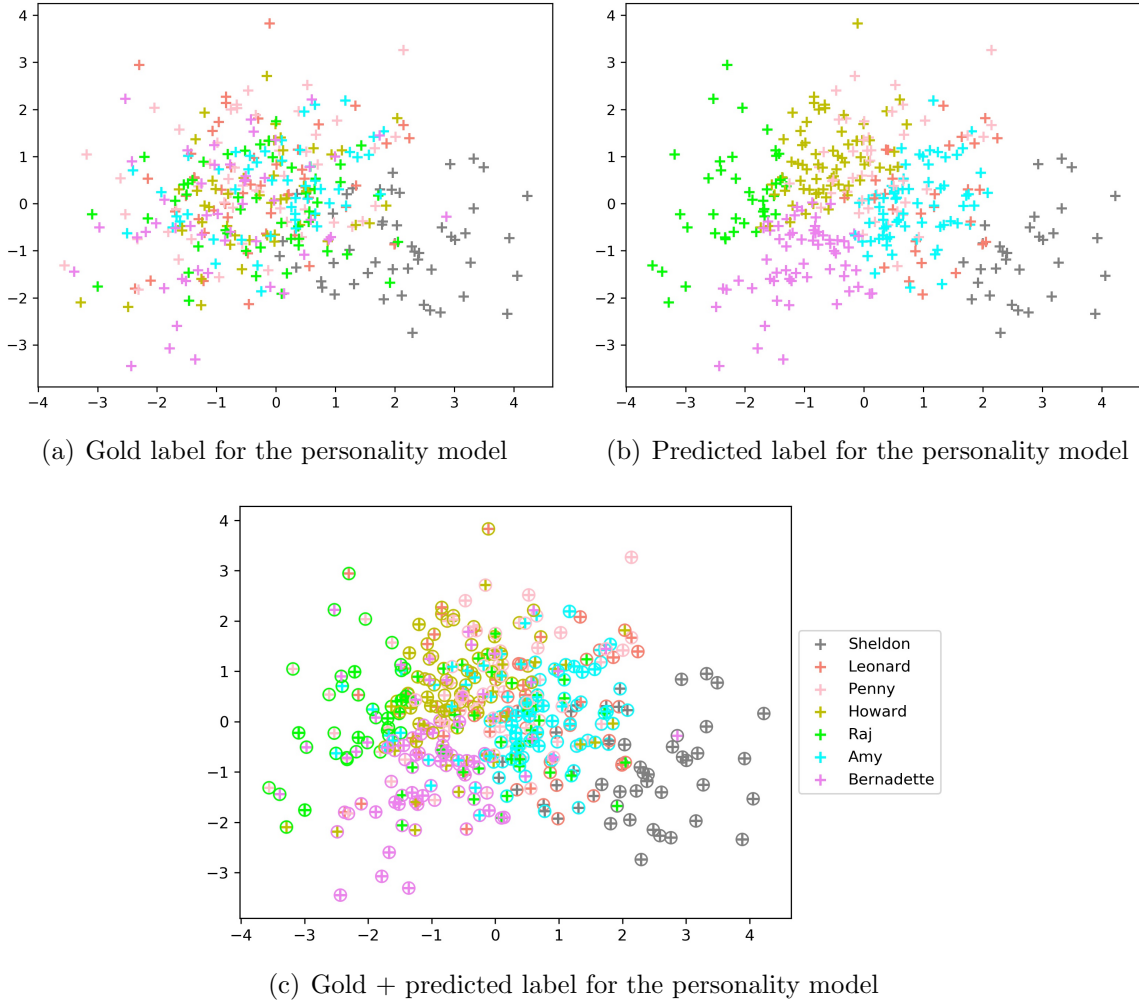


Figure 5.6: Gold label and predicted label on *The Big Bang Theory* for the personality model

Like before, Several figures follow the table, which are visualization of the clustering results with data decomposed by PCA into 2 dimensions. Figure 5.6 shows the predicted label for the personality model and its differences with the gold label. Each character has 50 samples.

Like the figures for the speaker model, figures for the personality model on *The Big Bang Theory* are also less messy than *Friends*, since the original scripts are more dis-

tinguished than *Friends* in personality. Furthermore, the character “Sheldon” in figure 5.6 also shares a similar distribution of dots with previous figure for the original scripts, which indicates that the personality model can capture the nature of this character like the speaker model.

5.5.2 32 Extreme Personalities

Similar to the experimental procedures in section 5.4.1, we first let the personality model generate responses for 32 different extreme personalities on the OSDB test set, and then followed section 5.2.2 to calculate the overall F1 score and accuracy score. Finally we compared them with the baseline.

In the *Friends* and *The Big Bang Theory* parts, the personality model was used to generate responses for specific characters, where the inputted OCEAN score was only used for referring to the character. However, here the personality model was given 32 novel OCEAN scores that did not refer to any of the characters in the training set.

The 32 extreme personalities have distinguished OCEAN scores. This experiment aims at testing if the personality model can generate distinguished responses for novel and extreme OCEAN scores.

Table 5.17 shows the average overall F1 score and accuracy score. Table 5.18 shows statistical analysis of F1 scores on generated responses for 32 extreme personalities, with respect to the random baseline.

Algorithm	Score	Baseline	Personality32
k-means	F1	0.071	0.188
	Accuracy	0.095	0.225
SVM	F1	0.030	0.222
	Accuracy	0.031	0.23

Table 5.17: Average Overall F1 score and accuracy score for 32 extreme personalities

	p-value	Cohen’s d
k-means	$9.55 \times 10^{-21}^{**}$	22.3
SVM	$1.93 \times 10^{-14}^{**}$	32.6

Table 5.18: Statistical results for 32 extreme personalities with respect to the baseline

Similar to *Friends* and *The Big Bang Theory*, we can infer that the personality model is able to generate distinguished responses given distinguished OCEAN scores.

The overall F1 score and accuracy score for the 32 extreme personalities are worse than the overall score of *Friends* and *The Big Bang Theory*. However, the scores are close to the overall score that combines *Friends* and *The Big Bang Theory* (0.19), where we tried to cluster & classify all of the 13 characters, instead of doing this for 6 characters in *Friends* and 7 characters in *The Big Bang Theory* separately. Note that since there are 32 clusters, the accuracy score should have decreased greatly, just as what has happened to the baseline: it is about 3 times lower than the score of the baseline on *Friends* and *The Big Bang Theory*. However, the personality model gives a fair score, which indicates that the personality model can generate relatively distinguished responses for any given OCEAN scores, not limited to the specific OCEAN scores of 13 characters from the TV-series dataset.

Chapter 6

Conclusion

In this thesis, we proposed a new evaluation method for measuring if the responses generated by personalized sequence-to-sequence response generation models are distinguished for different speakers. Also, we proposed a new model that focused on generating responses based on given personalities.

Personality inconsistency is a problem for chit-chat generation-based conversational agents. Recent works have proposed models aiming at keeping consistency for each distinguished personality; however, 1) there is not an existing evaluation method for measuring the performance of these models, 2) the concept “personality” used in these works is different from its psychological definition, which results in confusion during evaluation and a gap in personality-based response generation.

We studied the psychology research on personality, and applied the Big Five personality traits (OCEAN) for our new evaluation method. This new evaluation method estimates OCEAN scores with a personality recognizer ([Mairesse et al., 2007](#)) for responses of given speakers, and calculates the accuracy of assigning OCEAN scores to the correct speakers using clustering or classifying algorithm; this accuracy score, together with a comparison to the random baseline, measures personality differences among given speakers, and is considered as being able to measure the performance of response generation models on personality consistency. Furthermore, we proposed a personality model, which aims at generating distinguished responses given different personalities.

Among the existing personalized response generation models, we examined the

speaker model proposed by [Li et al. \(2016a\)](#). Before this, we conducted a preliminary experiment that evaluated personality differences among different characters for the original scripts (i.e. the dataset for training) with our new evaluation method; this experiment served as a sanity check, which tested the validity of the personality recognizer and the dataset for training. The result shows that the utterances of different characters in the dataset are significantly more distinguished than the random baseline, which ensures the basis of examining the speaker model and the personality model.

For the speaker model, the overall accuracy for clustering & classifying characters from the TV-series dataset is significantly higher than the random baseline but lower than the gold standard (the original scripts), which indicates that this model is able to generate distinguished responses for different speakers, however the responses are not as distinguished as the gold standard. Also, we found that with a more distinguished training set, the speaker model could have better performance.

For the personality model, the overall accuracy for clustering & classifying characters from the TV-series dataset is significantly higher than the random baseline but lower than the speaker model, which indicates that this model is able to generate distinguished responses based on OCEAN scores for different speaker types, while the responses are not as distinguished as those of the speaker model; however, this model’s capability of generating tailor-made responses on personality for each character is equal to or better than the speaker model.

We also let the personality model generate responses given 32 extreme personalities represented by OCEAN scores. The overall accuracy is significantly higher than the random baseline, which indicates that the personality model can generate distinguished responses for given OCEAN scores as expected.

Overall, under our new evaluation method, the speaker model by [Li et al. \(2016a\)](#) is examined to be working as expected, and so does our personality model; however, the performance still needs improvement.

There is a great potential for future works. The personality recognizer we used works fine, but is not good enough; with a better personality recognizer, the precision for estimating OCEAN score may improve. Also, there lacks a big-scale personalized

conversational dataset, so we had to apply the domain adaption training strategy to be able to train our model on the relatively small scale TV-series dataset, which reduced the performance of our model. Finally, in this thesis we tried to embed OCEAN scores directly into a SEQ2SEQ model, while there are other ways of incorporating personality information into the response generation model such as multi-task learning ([Luan et al., 2017](#)), which are worth trying in the future.

Bibliography

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .
- J.M. Burger. 2010. *Personality*. PSY 235 Theories of Personality Series. Cengage Learning. https://books.google.nl/books?id=bZY7I2_8yRMC.
- Heather EP Cattell and Alan D Mead. 2008. The sixteen personality factor questionnaire (16pf). *The SAGE handbook of personality theory and assessment* 2:135–178.
- Fabio Celli. 2012. Unsupervised personality recognition for social network sites. In *Proc. of sixth international conference on digital society*.
- Kenneth Mark Colby, Franklin Dennis Hilf, Sylvia Weber, and Helena C Kraemer. 1972. Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. *Artificial Intelligence* 3:199–221.
- Max Coltheart. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A* 33(4):497–505.
- Ondřej Dušek and Filip Jurčiček. 2016. A context-aware natural language generator for dialogue systems. *arXiv preprint arXiv:1608.07076* .
- Alastair J Gill and Jon Oberlander. 2003. Perception of e-mail personality at zero-acquaintance: Extraversion takes care of itself; neuroticism is a worry. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. volume 25.

- Samuel D Gosling, Peter J Rentfrow, and William B Swann Jr. 2003. A very brief measure of the big-five personality domains. *Journal of Research in personality* 37(6):504–528.
- Francis Heylighen and Jean-Marc Dewaele. 2002. Variation in the contextuality of language: An empirical measure. *Foundations of Science* 7(3):293–340.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. *arXiv preprint arXiv:1610.01108* .
- Dan Jurafsky and James H Martin. 2014. *Speech and language processing*, volume 3. Pearson London.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 994–1003. <https://doi.org/10.18653/v1/P16-1094>.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2017a. Data distillation for controlling specificity in dialogue generation. *arXiv preprint arXiv:1702.06703* .
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1192–1202. <https://aclweb.org/anthology/D16-1127>.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017b. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association

- for Computational Linguistics, pages 2157–2169. <http://aclweb.org/anthology/D17-1230>.
- Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023* .
- Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Asian Federation of Natural Language Processing, pages 605–614. <http://aclweb.org/anthology/I17-1061>.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .
- François Mairesse and Marilyn A Walker. 2010. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction* 20(3):227–278.
- François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research* 30:457–500.
- Matthias R Mehl, Samuel D Gosling, and James W Pennebaker. 2006. Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life. *Journal of personality and social psychology* 90(5):862.
- Saif M Mohammad and Svetlana Kiritchenko. 2013. Using nuances of emotion to identify personality. *Proceedings of ICWSM* .
- Warren T Norman. 1963. Toward an adequate taxonomy of personality attributes:

- Replicated factor structure in peer nomination personality ratings. *The Journal of Abnormal and Social Psychology* 66(6):574.
- Jon Oberlander and Scott Nowson. 2006. Whose thumb is it anyway?: classifying author personality from weblog text. In *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, pages 627–634.
- Shereen Oraby, Lena Reed, Shubhangi Tandon, TS Sharath, Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. *arXiv preprint arXiv:1805.08352* .
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 311–318.
- James W Pennebaker and Laura A King. 1999. Linguistic styles: Language use as an individual difference. *Journal of personality and social psychology* 77(6):1296.
- Soujanya Poria, Alexandar Gelbukh, Basant Agarwal, Erik Cambria, and Newton Howard. 2013. Common sense knowledge based personality recognition from text. In *Mexican International Conference on Artificial Intelligence*. Springer, pages 484–496.
- Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 583–593. <http://www.aclweb.org/anthology/D11-1054>.
- David P Schmitt, Jüri Allik, Robert R McCrae, and Verónica Benet-Martínez. 2007. The geographic distribution of big five personality traits: Patterns and profiles of human self-description across 56 nations. *Journal of cross-cultural psychology* 38(2):173–212.
- Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016a. Building end-to-end dialogue systems using generative

- hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI'16, pages 3776–3783. <http://dl.acm.org/citation.cfm?id=3016387.3016435>.
- Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016b. Generative deep neural networks for dialogue: A short review. *arXiv preprint arXiv:1611.06216*.
- Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1577–1586. <http://www.aclweb.org/anthology/P15-1152>.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 196–205. <http://www.aclweb.org/anthology/N15-1020>.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014a. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014b. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.
- Jörg Tiedemann. 2009. News from opus—a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*. volume 5, pages 237–248.

- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* .
- Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9(1):36–45.
- Min Yang, Zhou Zhao, Wei Zhao, Xiaojun Chen, Jia Zhu, Lianqiang Zhou, and Zigang Cao. 2017. Personalized response generation via domain adaptation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '17, pages 1021–1024. <https://doi.org/10.1145/3077136.3080706>.
- Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. 2015. Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565* .
- S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? *ArXiv e-prints* .
- Weinan Zhang, Ting Liu, Yifa Wang, and Qingfu Zhu. 2017. Neural personalized response generation as domain adaptation. *arXiv preprint arXiv:1701.02073* .

Appendix A

Responses and OCEAN Scores

A.1 Responses to “Do you love me?”

Question: Do you love me?		
Standard	I don't think you're going away with me.	
	<i>Speaker</i>	<i>Personality</i>
Rachel	You love me, I love you.	I love you.
Ross	Um... yeah.	You know I love you.
Chandler	Um, yeah.	I love you. You love me.
Monica	Um, no, it was just me, I just needed to stop.	You love him! Yeah, I love you, too.
Joey	Yeah, I love you too.	Oh, of course I love you, baby.
Phoebe	You love me.	I love you. I love you!
Sheldon	Yeah, but in the love of love, you are not only a child!	I love you too!
Leonard	Yeah.	Oh, I love you.
Penny	I do.	I love you.
Howard	Um, yeah.	Oh, no! No! Not you guys! I love you!

Raj	Yes! You love me more than you love me!	I don' t love you.
Amy	Yeah. I love you.	I love you.
Bernadette	Yeah.	I do, I do!

Table A.1: Responses to *Do you love me?* generated by the standard model, the speaker model and the personality model for 13 characters from the TV-series dataset

Question: Do you love me?			
n n n n n	Um, I love you.	y n n n n	Um yeah, um, I just... I wanted you to know...
n y n n n	I love you too, but we don't love each other.	n n y n n	Oh I, I love you too.
n n n y n	you are so beautiful. You are love in love with love.	n n n n y	You are beautiful!
y y n n n	Yeah.	y n y n n	Oh, I love you, too .
y n n y n	I don' t know, I mean, you have a beautiful voice.	y n n n y	I love you! I love you!
n y y n n	You love me, too.	n y n y n	I do love you.
n y n n y	Yes.	n n y y n	I love you. I do love you.
n n y n y	Yes.	n n n y y	Yeah, I love you! You love me.
n n y y y	I love you.	n y n y y	I love you too.
n y y n y	Yes.	n y y y n	Oh, yeah. I love you.
y n n y y	I do.	y n y n y	I do.
y n y n n	Yes, but you love it!	y y n n y	I love you! I love you so much! I love you. I love you.
y y n y n	Yeah, you love me. I love you too.	y y y n n	I love you! I love you!
n y y y y	I love you too!	y n y y y	I love you.

yynyy	I love you.	yyyny	I love you.
yyyyyn	I do, I love you.	yyyyy	I love you.

Table A.2: Responses to *Do you love me?* generated by the personality model for 32 extreme personalities

In table A.2, the first column stands for the OCEAN score: if a trait is scored 6.5, it is labeled "y"; if a trait is scored 1.5, it is labeled "n". The traits from left to right are: Extraversion, Emotional Stability, Agreeableness, Conscientiousness, and Openness.

A.2 Estimated OCEAN Scores

Character	Source	Extra	Emoti	Agree	Consc	Openn
Rachel	the original scripts	3.94	3.39	3.80	2.76	3.45
	the speaker model	3.57	2.93	3.45	3.39	3.58
	the personality model	3.68	3.26	3.47	2.93	3.38
Ross	the original scripts	3.79	3.09	4.19	4.10	4.17
	the speaker model	3.05	3.29	3.73	3.10	3.83
	the personality model	3.41	3.16	4.01	3.85	3.87
Chandler	the original scripts	3.76	3.28	3.45	2.94	3.07
	the speaker model	3.87	3.20	3.88	3.92	3.84
	the personality model	3.71	3.23	3.49	3.07	3.60
Monica	the original scripts	4.22	3.69	3.38	2.50	3.04
	the speaker model	3.69	3.12	3.29	3.15	3.26
	the personality model	4.18	3.10	3.18	3.02	3.51
Joey	the original scripts	2.95	4.02	3.64	2.63	3.07
	the speaker model	3.23	3.25	3.48	3.02	3.58
	the personality model	3.58	3.00	3.42	3.37	3.76
Phoebe	the original scripts	3.42	3.39	4.17	2.65	3.35

	the speaker model	3.01	2.54	3.47	3.68	3.69
	the personality model	3.91	3.31	3.68	3.74	3.73
Sheldon	the original scripts	3.11	2.65	4.31	5.73	5.22
	the speaker model	4.68	2.68	4.07	5.84	4.99
	the personality model	3.90	2.66	4.24	5.86	5.26
Leonard	the original scripts	3.50	2.23	3.38	4.66	4.17
	the speaker model	3.40	3.65	3.82	2.95	3.79
	the personality model	3.40	3.38	3.92	3.40	3.44
Penny	the original scripts	3.89	2.61	3.39	3.91	4.23
	the speaker model	4.39	4.13	3.41	2.24	2.82
	the personality model	3.53	3.47	3.87	3.04	3.43
Howard	the original scripts	3.37	3.68	3.55	3.11	3.49
	the speaker model	3.62	3.40	3.77	3.22	3.67
	the personality model	3.42	3.26	3.68	3.47	3.66
Raj	the original scripts	3.97	4.00	3.87	3.10	3.75
	the speaker model	3.75	3.26	3.91	3.83	3.89
	the personality model	3.52	3.12	3.62	3.59	3.71
Amy	the original scripts	3.57	2.17	3.53	4.88	4.25
	the speaker model	3.71	2.70	3.83	4.45	4.23
	the personality model	3.56	3.37	3.96	3.67	3.60
Bernadette	the original scripts	4.24	3.32	3.16	2.99	3.12
	the speaker model	3.75	3.36	3.69	3.16	3.20
	the personality model	3.93	3.18	3.25	2.96	3.43

Table A.3: Average OCEAN scores for 13 characters from the TV-series dataset on the original scripts, the speaker model and the personality model