A thesis submitted in partial satisfaction of the requirements
for the degree of Master of Computer Science and Engineering
in the Graduate School of the University of Aizu

Building an Open Domain Dialogue Agent Using Twitter Conversations

to Advance Knowledge for Humanity
THE UNIVERSITY OF AIZU
· 1993 ·

by

Hiroshi Yamaguchi

March 2019

The thesis titled

Building an Open Domain Dialogue Agent Using Twitter Conversations

by

Hiroshi Yamaguchi

is reviewed and approved by:

Main referee

Associate Professor

Maxim Mozgovoy

Senior Associate Professor

Evgeny Pyshkin

Professor

Ian Wilson

THE UNIVERSITY OF AIZU

March 2019

Contents

List of Figures

List of Tables

Acknowledgment

I'm extremely grateful for advices and assistance for experiment to my supervisor, Professor Maxim Mozgovoy. Thanks to my Lab's members and other people who participated our experiments.

Abstract

A chat dialogue agent or a conversational agent is a computer program designed to hold a conversation using natural language. Such computer programs have been getting popularity in recent years as the demand for the kind of applications have increased in many different areas. In addition, recent advances in other related language technologies such as speech recognition and natural language understanding give us the chance to communicate with the device to get information easily. However, there are still a lot of difficulties on designing a chat dialogue agent, especially open domain dialogue agent.

In this work, we presents implementing a dialogue agent using Twitter conversation to communicate with the human naturally and make them enjoyable. As a result, we have implemented an open domain dialogue agent using the large amount of Twitter conversation data.

Chapter 1

Introduction

1.1 Background

A dialogue agent is a computer program designed to imitate a conversation with users [1]. In recent years, a dialogue agent have been expected be applied in various fields, but many challenges still exist in developping them. In general, dialogue agent can be categorized into two types: *task-oriented* dialogue agent which are used to help the user to complete various tasks. Typical examples of *task-oriented* dialogue agent are used booking transportation or accommodation services, question and answer for guidance system, etc [2–4]. On the other hand, *open-domain* dialogue agent which aim to perform a natural conversation with the user [5]. To realize *task-oriented* dialogue agent, it is important role to perform open domain conversation with the user not only entertainment but also for efficient task accomplishment [6]. In addition, Takeuchi at el. reported that the user sometimes made an utterance unrelated to the task accomplishment in the case of *task-oriented* conversation [4]. Since in order to achieve a task smoothly and realize a user-friendly *task-oriented* dialogue agent, it needs to handle such open domain conversation appropriately.

Both *task-oriented* and *open-domain* dialogue agent have been studied for a long time. There are mainly two problems as the way of response and how to collect data. Conventional approaches to build a dialogue agent have used hand crafted rules more than tens of thousands called rule based model [7]. However, it requires a lot of manual efforts to construct and leading to expensive maintenance costs. Another problem with the rule based model is the low coverage of topics. To handle these problems, recent studies have used social media websites. There are numerous samples of conversation data on social media websites such as Twitter, Facebook and Reddit. Twitter is one of the rich source frequently used in the field of dialogue agents [8, 9]. Twitter offers many APIs to retrieve individual tweets and tweet streams. Twitter dialogues come relatively close to informal daily conversations. So we can get a large amount of conversation data easily and solves the coverage problem.

Recently, the approach to build a dialogue agent have used statistical based model which use statistical processing with a large amount of conversation data on Twitter [9, 10]. It is not necessary a lot of manual efforts to construct a dialogue agent and leading to low maintenance costs. However, since Twitter conversation data contains noise such as named entity so that this approach has potential to output semantically incorrect sentences. To tackle this problem, Inaba et al. proposed utterance acquisition method from Twitter [10]. But this method use only utterance containing a topic word.

In this paper, we propose a retrieval based model using social data on Twitter. Our model use statistical processing with a large amount of conversation data. We aim at building a dialogue agent which is close to daily conversation so that we do not adopt the utterance acquisition method by Inaba. Because they only acquire the sentence containing topic word.

1.2 Objective

Previously, we implemented a dialogue agent both rule based model [11] and retrieval based model [12]. However, the experimental results show that we could not get the results as we expected. In this research, we continue to implement the retrieval based model using Twitter conversation. The goal of this research is implementing a dialogue agent which is able to talk naturally and make the user enjoyable.

Chapter 2

Related Work

In this chapter, we briefly provide existing well known dialogue agent model as rule based [7, 11, 13, 14] and statistical based [8, 9, 15] model.

2.1 Rule based model

A large number of dialogue agents respond to a user's utterance based on response rules. This is called rule based model which is used by ELIZA [13], PARRY [14], A.L.I.C.E. [7] and our previous [11]. One of the most popular mechanisms of representing rules is AIML (Artificial Intelligence Markup Language) [16]. AIML is a simple XML based markup language that gained popularity after being used in a successful dialogue agent A.L.I.C.E that won the Loebner Prize three times. The Loebner Prize is an annual competition in Artificial Intelligence to find the dialogue agent considered by the judges to be the most human-like. The format of the competition is based on the Turing test. The main drawback of AIML-based model lies in the large number of rules required to imitate a natural conversations, especially in case of open domain models. Therefore, AIML-based dialogue agent require a lot of manual efforts to describe its knowledgebase, leading to expensive and error-prone development process.

Previously, we tried generating AIML rules automatically using conversation data acquired from Twitter [11]. However, we found that there are lot of difficulties causing with the lack of approximate matching. Moreover, if the number of rules exceeds a certain threshold hindering the improvement of perfomance of AIML-based dialogue agent [17]. It suggested that there is a limit to the performance of AIML-based dialogue agent. So we shifted rule based model to retrieval based model as we mention next section.

2.2 Statistical based model

The recent explosion of public conversations on social media websites have promoted the development of approaches statistical based model. There are two major approaches in the Statistical based model as example based model also called retrieval based model [15, 18, 19] and machine translation (MT) model also called generative models [8, 9].

Retrieval based model searches a large database of predefined responses for given user input selecting appropriate response. Web data especially Twitter data are used to this approach [10, 15, 20]. Although Twitter data can deal with a wide variety of topics to user input due to the diversity, Twitter data contain noise. Since Inaba et al. constructed a dialogue agent based on their proposed method to suppress this noise and acquire an appropriate utterance from the candidate responses which are collected only extracted the sentences containing topic words in a large amount of Twitter data. They assessed their dialogue agent using human subject conversation to evaluate. For comparison, they constructed a dialogue agent using a chat API developed by NTT Docomo and a Wizard of Oz (WOZ) method based agent. They confirmed that their dialogue agent is superior to the chat API. In addition, this approach indicates that performance is powerful since dialogue agent based on this approach include Jabberwacky [21] which won the Loebner Prize. Previously, we built a dilaogue agent retrieval based model using Twitter conversaiton data [12]. In this research, we try to do another approach.

Generative model don't rely on predefined responses which generate new response from scratch. Ritter et al. proposed Machine translation method for response generation. They prepared tweet reply pairs of status post and reply regarding them as the source language and the target language respectively. In other words, instead of the source language to the target language, this method translate an user input to an output. Sordoni et al. extended the MT method, they collected Twitter conversation A-B-A triples as context, message and response. They exploited generating responses that are sensitive to the context of the conversation which performs better than Ritter et al. proposed model.

Deep Learning can be used for both retrieval based model or generative model [22]. Although generative model is an active area of research, but it doesn't work well in practice at current time. Because generative model tend to make grammatical mistakes compared to retrieval based model. Retrieval based model output a response from potential response which is written by human. However, generative model output a response from scratch so that produce irrelevant response easily, if the generative model could not optimize well. Since we adopt retrieval based model to build a dialogue agent.

Chapter 3

Method

In this chapter, we introduce how to collect conversation data on Twitter and building a dilaogue agent using collected data.

3.1 Twitter

We crawled Twitter conversations using Twitter Streaming API and Rest API to collect a large amount of conversation data on Twitter. The collected data corresponds to the tweets posted between October 2016 to November 2018. Each conversation consists of 3 tweets as follows.

1. Original tweet by user A

2. Reply tweet by user B

3. One more reply tweet by user A

We use Tweepy [23] which is an easy to use Python library for accessing the Twitter API. Individual tweets are tagged with a number of attributes, including the tweet language, timestamp, and in-reply-to fileld. We fetched Japanese tweets only, and extracted the content of each attribute. As a result, we gathered a corpus of 2,653,088 dialogues consisting of three tweets like above.

3.2 MeCab and TF-IDF

This section provides the necessary prior knowledge to talk about the following sections.

3.2.1 MeCab and Neologd

It is necessary to do morphological analysis to tokenize the text in Natural Language Processing. Mecab is the one of the tool of morphological analyzer for text written in Japanese [24]. Japanese is written continuous string unlike text written in English which is separating each words with single space. It is easy to get single token in the case of English due to each words separated with single space. Since we use MeCab to get single token from continuous string written in Japanese. MeCab is tokenize the text based on IPA dictionary. The problem of this method is that doesn't assure accurate analysis and update the IPA dictionary. Since the dictionary has not been updated, there are many new words unrecorded in it such as iphone and AKB48. To solve this issue, we found customized system dictionary for MeCab called mecab-ipadic-NEologd (Neologism dictionary) [25].

Neologd includes many neologisms, which are extracted from many language resources on the Web. Neologd can tokenize Japanese that can not be recognized using IPA dictionary such as iphone and AKB48. But Neologd also have weak points as the named entity. Neologd cannot classify well the named entity and not named entity word is recorded as named entity. For example, suppose the source sentence is as ”AKB 総選挙か、、、ナゴヤドーム付近すんげぇー混みそうやな (笑)”. The result of morphological analysis is like below.

1. （IPA） ’AKB’, ’ 総’, ’ 選挙’, ’ か’, ’、 ’, ’、 ’, ’、 ’, ’ ナゴヤ’, ’ ドーム’, ’ 付近’, ’ すん’, ’ げ’, ’ぇ’, ’ ー’, ’ 混み’, ’ そう’, ’ や’, ’ な’, ’(’, ’ 笑’, ’)’

2. （Neologd） ’AKB 総選挙’, ’ か’, ’、 ’, ’、 ’, ’、 ’, ’ ナゴヤドーム’, ’ 付近’, ’ すん’, ’ げ’, ’ぇ’, ’ ー’, ’ 混み’, ’ そ う’, ’ やな’, ’(’, ’ 笑’, ’)’

’AKB 総選挙’ is neologisms so that IPA dictionary could not analyze appropriately. In addition, although IPA get morphemes ’ ナゴヤ’ and ’ ドーム’, Neologd get an appropriate morpheme as ’ ナゴヤドーム’. It is a stadium in Nagoya.

Both IPA and Neologd dictionary have weak points. But the developer recommended us to use both dictionary so that we use together through compiling the system dictionary.

3.2.2 TF-IDF

Term frequency inverse document frequency (TF-IDF) [26] is a statistic that intends to capture how important a given word is to some document. This is a technique often used in document classification and information retrieval. *Term frequency* term is simply counts the number of occurrences of word w appeared in a given document d, the exact ordering of the words in a document is ignored but the number of occurrences of each word is material, while the *inverse document frequency* term puts a penalty on how often the word appears elsewhere in the corpus. The final score is calculated as the product these two terms, and has the form:

$$tfidf(w, d, C) = f(w, d) \times \log \frac{|N|}{|d \in C : w \in d|}$$

where C is the set of all sentences in the preprocessed collection, $f(w, d)$ indicates the number of times word w appeared in the document d, N is the total number of dialogues, and the denominator represents the number of dialogues in which the word w appears. To implement this, we rely on the default similarity measure implemented in the scikit-learn library [27], which is an TF-IDF weighted Vector-Space similarity.

3.3 Preprocessing

Our dialogue agent response to the user based on the database having a large amount of conversation data on Twitter. Twitter conversation data are noisy so that we need to do preprocessing before creating the database. There has been proposed the filtering method to eliminate noisy data [8, 10, 20]. We performed the filtering method are similar to those used by Inaba et al. and Higashinaka et al.. The most different things between their approach and ours are topic word. We use sentences that do not have topic word. To suppress noise, we perform the preprocessing as follows.

1. Twitter expression - There are many irrelevant expression to our informal dairy conversation in tweets. These tweets typically consist of hyperlinks, hashtags, retweeted content and user names. We remove such tweets from the source collection and replace user names with empty string. In addition, there are the others typical of Japanese keyword as ”フォロー”, ”フォロワー”, ”フォロバ”, ”リプ”, ”ff 外”, ”検索から”, ”DM”, ”垢”, ”bot”.

We also remove tweets containing such keyword. (e.g. 無言フォロー失礼しました (; __ ;) タグに反応ありがとうございます もし宜しければ此方に呼び方お願いしたいです！ 3 日以内にフォロバない場合リムらせて頂きますので御承知ください)

2. Time expression - We remove tweets containing time expression words such as time, date and relative dates. (e.g. 1 時から進路指導みたいなのがあるから今日も学校行ってる ( )！ 2 日連続学校行けるか不安だったけどもう電車乗れた !!! 電車も割と最近はもう乗り換えとか降りて休憩なしで乗れるようになった !!)

3. Personal pronouns - There are many personal pronouns in Japanese unlike English such as ”わたし”, ”おれ”, ”ぼく”. We normalized the first person pronoun to ”わたし”.

4. Long sentence - If the sentence is too long (more than 50 characters) are removed because they may not be appropriate as colloquial sentence. (e.g. ハロー、夜にこっそりロケットリーグやったがあれは実に面白い…また機会があったらやろうっ。 今日も天気良好な所がおおいのではないだろうか、わたしはバーチャルゲームセンターにお出かけしてくるぞ。 久しぶりだっ)

5. Proper name - Collected tweets may contain name of the other person in the conversation. Such tweets to be removed because proper name is only known to both the speaker. (e.g. 琴詠さんのお腹にひっついて飛ばなきゃいけない (使命感))

6. Frequency - We remove one frequent bigram that appeared only once in the source collection.

Table 3.1 show the result of preprocessing. We use 369,671 valid triples to build a dialogue agent. We convert them into AIML format for response generation as we detailed in the next section.

corpus	2,653,088
URL Tag RT	1,044,932
twitter expression	117,466
time expression	220,339
long sentence	326,996
proper name	108,195
Frequency	465,489
valid triples	369,671

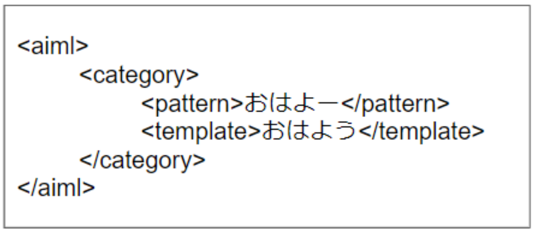
Table 3.1: Preprocessing Result

3.4 AIML

To build a dialogue agent, we use AIML for response generation. AIML is a derivative of Extensible Mark-up Language (XML) [16]. It was developed by the Alicebot free software community during 1995-2000 to enable people to input dialogue pattern knowledge into chatbots based on the ALICE free software technology. In general, its system belong to rule based model. However, we use it retrieval based model.

3.4.1 AIML Basic Categories

AIML is based on XML, and thus consists of hierarchically organised elements. Individual ”units of knowledge” are known as *categories* in AIML. Category is basic unit and should define at least two compulsory elements: a *pattern* that contains a sample input and a *template* that contains the corresponding response of the dialogue agent. In the Figure 3.1, if the user inputs おはよー, the agent should reply おはよう.

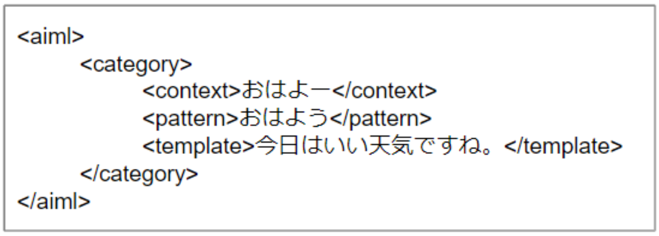


```
<aiml>
        <category>
                <pattern>おはよー</pattern>
                <template>おはよう</template>
        </category>
</aiml>
```

Figure 3.1: AIML Category

3.4.2 AIML context

We add *context* tag into basic categories like the Figure 3.2. The merits of adding context tag is allows specifying the context where the given rule is applicable and thus keeping dialogues coherent. We rely on this capability when converting Twitter dialogues into AIML rules. The resulting system uses the rules including all three elements like the Figure 3.2. Here, the dialogue agent will reply 今日はいい天気ですね。 only if the two preceding dilaogue lines were おはよー and おはよう.



```
<aiml>
        <category>
                <context>おはよー</context>
                <pattern>おはよう</pattern>
                <template>今日はいい天気ですね。</template>
        </category>
</aiml>
```

Figure 3.2: AIML Context

3.5 Response Model

We use Twitter conversation data for response to the user. After preprocessing, we have triples consisting of three consecutive utterances. Our system attempts to find the top 100 categories are retrieved on the basis of the similarity between the system context and the context in the categories. We adopt the cosine similarity of TF-IDF weighted word vectors. Here, we have 100 categories so that we have 100 patterns. To get response from the agent, we narrow down the candidates categories to ten. For a given user input as a query and find the top ten categories are retrieved on the basis of the similarity between the query and the pattern in the top 100 categories. Then, one of the retrieved category is randomly selected for response to the user.

Chapter 4

System

In this chapter, we describe the system implementation, architecture, and development of the dialogue agent. In our system, between the user and the dialogue agent hold a conversation through input text message, not speech recognition.

4.1 Implementation

We implement the dialogue agent by converting twitter dialogues into AIML rules and retrieve *template* as response to a given user input.

The process of converting the raw tweet corpus into a set of AIML rules consists of the following steps. First, we preprocessing raw tweet to eliminate unsuitable data for conversation as we mentioned previous chapter and retrieve a large amount of preprocessed tweets. Second, each element in our corpus contains three consecutive dialogue lines that are to be mapped to the AIML tags <context>, <pattern> and <template>. Our system attempts to retrieve the similar context and pattern for the current situation using TF-IDF approach [26]. This process requires tokenization into individual morphemes which is done with the help of Japanese morphological analyzer MeCab splits the text into individual part-of-speech tagged morphemes. Third, we generate AIML rules from three consecutive dialogues and storing them as AIML format files. Each triples are tokenized into individual morphemes which is transformed into an individual AIML rules. Triple elements are mapped to the AIML tags <context>, <pattern> and <template>. As a result, we generated 369,671 rules for the system.

4.2 Architecture

Conversation flow between the user and the dialogue agent as follows.

1. Loading AIML files and converting the documents into TF-IDF weighted word vectors.

2. The dialogue agent starts a dialogue with a line こんにちは.

3. User turn: input text message and send it as an user utterance to Response model.

4. Agent turn: provides a similar context/pattern pair and randomly select a template from them as an agent utterance to the user.

5. Back to 3.

Figure 4.1 presents conversation flow between the user and the dialogue agent. Response model is receive the user utterance, then it search similar context to the previous response of the agent and retrieve 100 context/pattern pairs. To get the response from the agent, Response model narrow down context/pattern pairs to the top ten similar patterns to the user input based on cosine similarity. After calculating similarity, Response model randomly select template as the agent utterance and send it to the user.

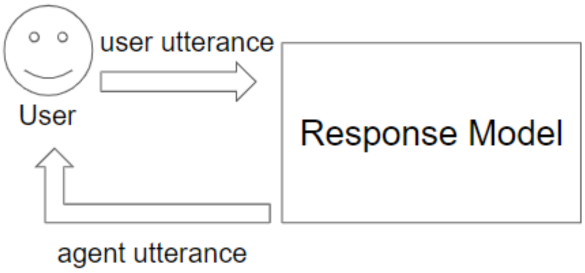


Figure 4.1: System architecture

4.3 Development

The system is designed as a client-server application with a web interface and a python backend. The user enter the conversation page from Login page in Figure 4.2. The frontend processes user message and sends it to the server side for further processing in Figure4.3. The agent's decision making is supported by AIML rules containing conversations, stored in AIML format files.

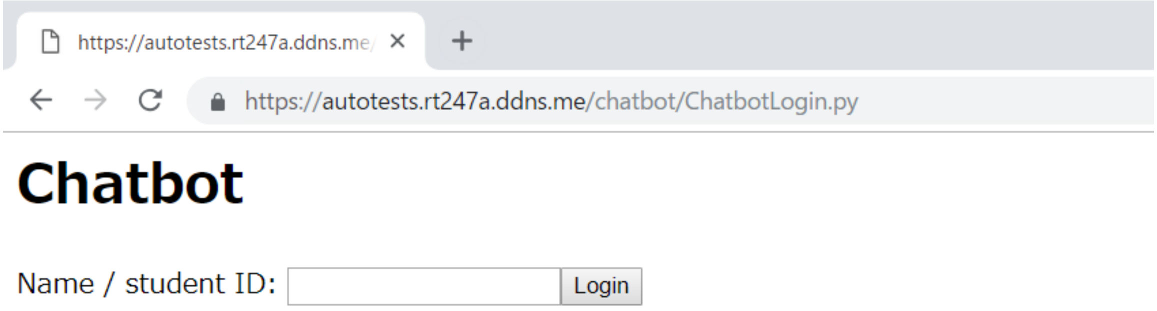
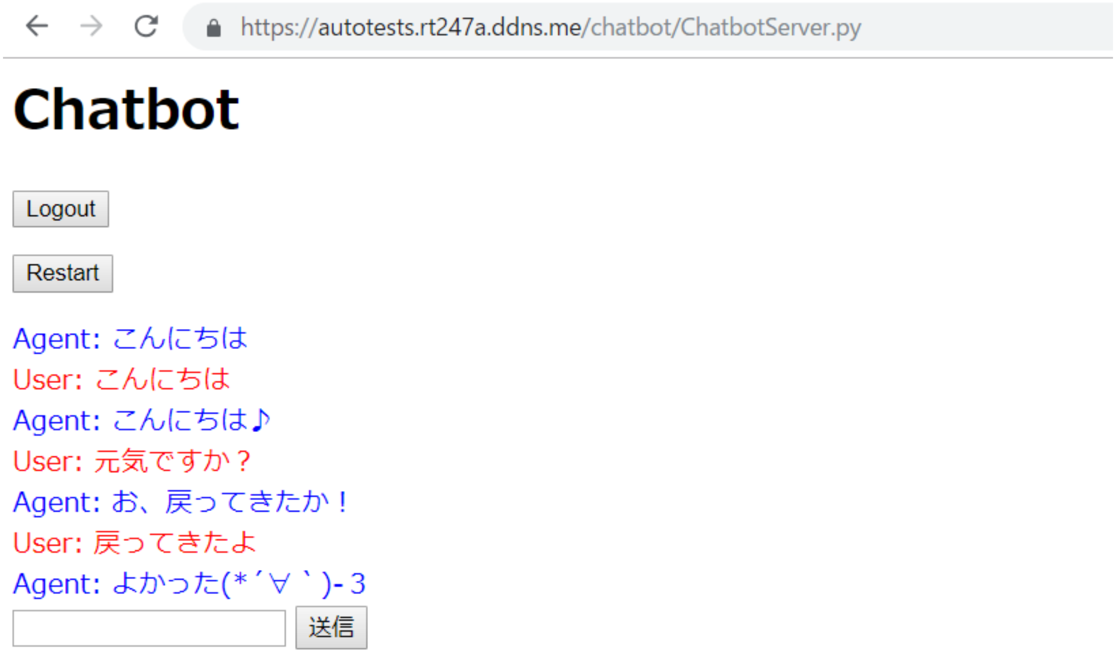

https://autotests.rt247a.ddns.me/
https://autotests.rt247a.ddns.me/chatbot/ChatbotLogin.py
Chatbot
Name / student ID:
Login

Figure 4.2: Login page



https://autotests.rt247a.ddns.me/chatbot/ChatbotServer.py

Chatbot

Agent: こんにちは
User: こんにちは
Agent: こんにちは♪
User: 元気ですか？
Agent: お、戻ってきたか！
User: 戻ってきたよ
Agent: よかった(*´∀｀)-3

Figure 4.3: Conversation page

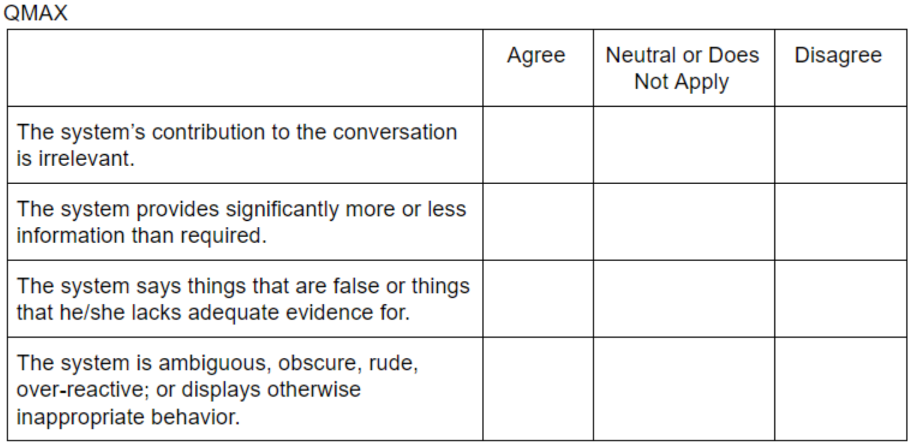
Chapter 5

Experimental Evaluation

In this chapter, we describe how we conducted the experiment to evaluate our system. We did two experimental evaluation [28] and [29].

5.1 Pragmatic analysis

We conducted experiment on May 2018 [12], involving ten respondents (5 female and 5 male), all speakers of Japanese (6 undergraduate students and 4 older adults of 29-58 age). Each person made 3 attempts at chatting with the agent through a web interface, resulting in total 30 chats (23 of which were 10 lines long on average; and 7 ranging from 17 to 41 lines). The evaluation questions on a 3-point Likert scale were adopted from [28] and answered by each respondent, following questionnaires in Figure 5.1 and Figure 5.2. We analyzed the experimental result by pragmatic analysis.



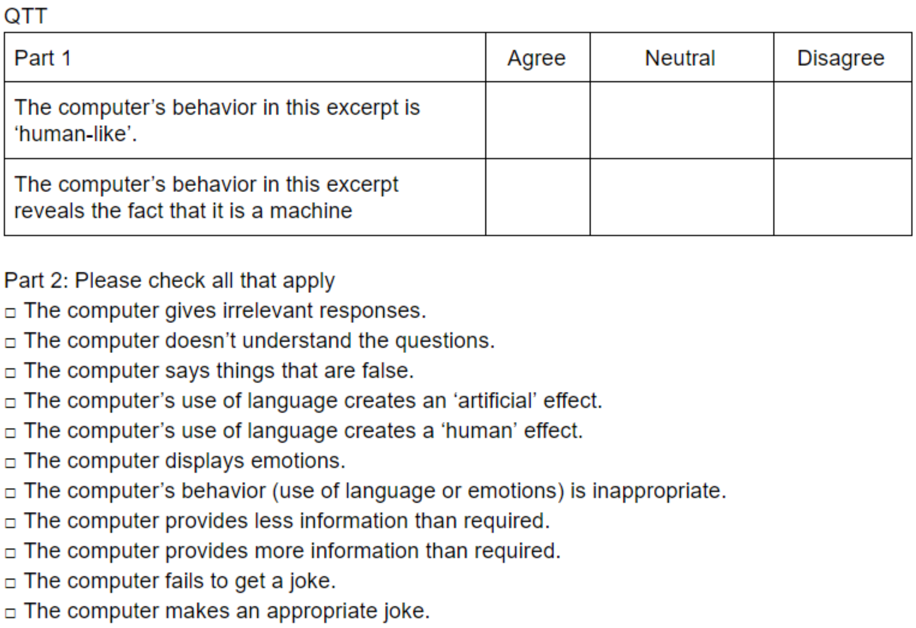
QMAX

	Agree	Neutral or Does Not Apply	Disagree
The system's contribution to the conversation is irrelevant.			
The system provides significantly more or less information than required.			
The system says things that are false or things that he/she lacks adequate evidence for.			
The system is ambiguous, obscure, rude, over-reactive; or displays otherwise inappropriate behavior.			

Figure 5.1: Questionnaire 1

5.1.1 Discussion

Based on the pragmatic analysis of the 30 chats in our sample dataset, certain tendencies regarding “conversational behaviour” of both the dialogue agent (A) that of the users (U) were observed. It can be noted that the dialogue agent tends to successfully complete adherent speech acts in opening sequences, such as greetings (“Hello” — “Hello”); small talk questions about general well-being (U: “How are you?” — A: “I’m well”, reinforced at times by emoticons) or questions or remarks about the weather (“The weather is hot”).



QTT

Part 1	Agree	Neutral	Disagree
The computer’s behavior in this excerpt is ‘human-like’.			
The computer’s behavior in this excerpt reveals the fact that it is a machine			

Part 2: Please check all that apply

□ The computer gives irrelevant responses.
□ The computer doesn’t understand the questions.
□ The computer says things that are false.
□ The computer’s use of language creates an ‘artificial’ effect.
□ The computer’s use of language creates a ‘human’ effect.
□ The computer displays emotions.
□ The computer’s behavior (use of language or emotions) is inappropriate.
□ The computer provides less information than required.
□ The computer provides more information than required.
□ The computer fails to get a joke.
□ The computer makes an appropriate joke.

Figure 5.2: Questionnaire 2

Generally, it tackles question-answer sequences in a satisfactory way by providing a generic answer (U: “Will you go shopping?” / ” Where will you go?” — A: “I don’t know”); by answering a yes/no question (U: “Are you hungry?” A: “Yes, I am”; U: “Do you like music?” — A: “Yes, I like it [note emoji]”) or by giving a more detail answer (U: “Where do you work?” — A: “I work in a factory [thumbs up emoji]”). Depending on the domain and question complexity, an attempt at a more elaborate answer may stretch over a number of lines, if not “interrupted” by the user’s impatience and an abrupt change of subject (U: “Do you enjoy painting?” — A: (···) “I’m not good at sketching in five minutes [emoticons]. But I’ll get experience [emoticons]. I want to improve my skills. I will try”). In most cases, however, the users, not aiming at exploration of a given topic, fail to ask further questions, (e.g., “What kind of music do you like? Do you like jazz?”), and change the topic abruptly.

In addition, the conversational agent simulates emotive reactions (with reference to senses) on a number of occasions, reinforced by punctuation marks, emoticons and/or emoji (“Lean on me” — “I want to pat you on the head” — “Oh··· you’re fluffy” — all in one conversational sequence). Unfortunately, the users tend not to follow up on such “emotional vibes”, resorting to dispreferred options (causing a mismatch in speech acts) in their responses, as illustrated by this example:

|A: I love you

|U: No, thank you.

Here the user exhibits outright violation of maxim of relevance, as well as that of politeness. The dialogue agent makes rather successful attempts at simulating emotions — frequently reinforced by emoticons/emoji (“Oh, I’m embarrassing”, “Envy”), It also “expresses” concern, but, again, such attempts are not pursued further by the users, reluctant to explore the topic of emotions:

|A: My heart is frozen by your reply

|U: My muscles are aching!.

Emotional content is thus commonly met with inappropriate answers, repetition or change of subject.

5.2 Quantitative analysis

We conducted the experiment in reference to [29] on January 2019. This experiment involving 5 male respondents, all speakers of Japanese and undergraduate students. We compared the response method with our system calling as *IR-context* to *IR-status* and *IR-response* [9]. Both *IR-status* and *IR-response* are approach to response generation in information retrieval. Given a novel status s and twitter corpus of status/response pairs corresponding to pattern/template pairs respectively in AIML formats, two retrieval strategies can be used to return a best response r_i: *IR-status* is retrieve the response r' whose associated pattern p_i is most similar to the user's input s $[r_{argmax_i,similarity(s,p_i)}]$. On the other hand, *IR-response* is retrieve the response r_i whose associated template t_i is most similar to the user's input s $[r_{argmax_i,similarity(s,t_i)}]$. We proposed the response method *IR-context* as we mentioned in chapter §3. Each person made an attempt at chatting with each response method through a web interface, resulting in total 15 chats which were at least 10 lines. The evaluation questions on a 5-point Likert scale were adapted from [29] and answered by each respondent, following evaluation criteria in Table 5.1. We analyzed the experimental result based on Table 5.1.

Table 5.1: Evaluation Criteria

Criteria	Content
(1) Correctness	User can understand the agent utterance as Japanese.
(2) Naturalness	Agent utterance are appropriate in each turn.
(3) Diversity	Agent utterances are diverse.
(4) Enjoyability	User have a fun conversing with the agent.
(5) Engagement	User want to engage in dialogue again.

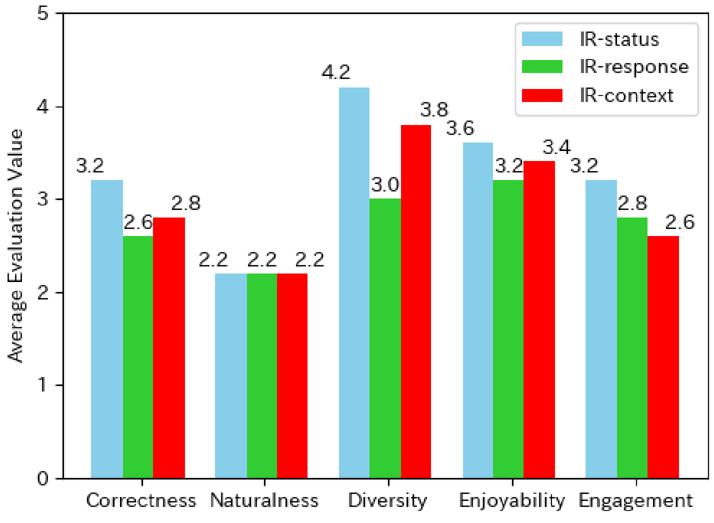
5.2.1 Discussion

Results of the experiment in each approach elicited average evaluation values for correctness, naturalness, diversity, enjoyability and engagement as shown in Figure 5.3. According to the result shown in Figure 5.3, results confirmed that *IR-status* is the best in the order of *IR-context* and *IR-response* at a glance. Evaluation value of correctness is lower score than as we expected. Because we used Twitter conversations posted by human so that there should be few grammatical mistakes. Evaluation value of diversity is the best score in this experiment. It indicates the result of taking advantage of Twitter conversations is talked about a wide range of topics. Participants judged relatively enjoyable through talking to the agent. Although they felt enjoy talking, they do not want to engage in dialogue with the agent again. This is because evaluation value of naturalness is the worst score. Participants tend to feel funny when they got an unexpected and inappropriate response from the agent. For example, one of the participant assign naturalness as 1 and enjoyable as 3. The content of conversation in Table 5.2. The agent did not retrieve the appropriate response and the participant got unexpected response in this conversation. Since, they evaluated the agent enjoyable but not engage in dialogue again.

Table 5.2: Example of Conversations

Speaker	Utterance
Agent	Hello
Human	Hello
Agent	Hello!!
Human	How are you?
Agent	It was by the end of today.
Human	Do you play video game?
Agent	Thank you!!! I will wait for you!!!
...	...

Figure 5.3: Results of quantitative evaluation



Chapter 6

Conclusion

This paper presented building a dialogue agent based on retrieval based model using a large amount of Twitter conversations. In addition we tried doing preprocessing and implementing response method *IR-context*.

We use Twitter conversations to build a dialogue agent. The merit of Twitter conversations are used many people all over the world so that we can collect many conversations posted by wide range of generations. Since, we can get a source with a wide variety of topics. Experimental results showed that Twitter conversations can serve as a source to coverage of diversity topics. So we think that Twitter conversations will be able to incorporate into dialogue agent applications used in various people.

We tried doing preprocessing to remove noisy data. The preprocessing methods are similar to those proposed by Inaba et al. [10] with some extensions. The most different point between Inaba and our approach are a topic word. Although Inaba used a source only containing topic word, we used a source without a topic word. Because in our daily conversation, we make a response with a topic word but also without it such as agreement. However, we confirmed that our preprocessing method is insufficient. Our agent sometimes retrieved a response to remove in the conversation between the user and agent of the experiments. Those caused low evaluation value of Naturalness. To solve this problem, revise preprocessing methods are required.

In this research, we demonstrated that our dialogue agent can make the user enjoyable and handle the wide range of topics using Twitter conversations. For future work, we plan to revise preprocessing methods and upgrade response method. In addition, we need to set up the experiment involving wide range of generation and same gender ratio to evaluate the our agent with more precision. We think that the dialogue agent using Twitter conversations will lead to application of creation for artificial intelligence which is used by people in the future.

References

[1] B. AbuShawar and E. Atwell, “Alice chatbot: trials and outputs,” *Computación y Sistemas*, vol. 19, no. 4, pp. 625–632, 2015.

[2] S. Seneff and J. Polifroni, “Dialogue management in the mercury flight reservation system,” in *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*. Association for Computational Linguistics, 2000, pp. 11–16.

[3] S. Busemann, T. Declerck, A. K. Diagne, L. Dini, J. Klein, and S. Schmeier, “Natural language dialogue service for appointment scheduling agents,” in *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics, 1997, pp. 25–32.

[4] S. Takeuchi, T. Cincarek, H. Kawanami, H. Saruwatari, and K. Shikano, “Construction and optimization of a question and answer database for a real-environment speech-oriented guidance system,” 2007.

[5] X. Wu, K. Ito, K. Iida, K. Tsuboi, and M. Klyen, “りんな: 女子高生人工知能,” 言語処理学会第 *22* 回年次大会発表論文集, pp. 306–309, 2016.

[6] T. Bickmore and J. Cassell, “Relational agents: a model and implementation of building user trust,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001, pp. 396–403.

[7] R. S. Wallace, “The anatomy of alice,” in *Parsing the Turing Test*. Springer, 2009, pp. 181–210.

[8] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, “A neural network approach to context-sensitive generation of conversational responses,” *arXiv preprint arXiv:1506.06714*, 2015.

[9] A. Ritter, C. Cherry, and W. B. Dolan, “Data-driven response generation in social media,” in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 583–593.

[10] M. Koshinda, M. Inaba, and K. Takahashi, “Machine-learned ranking based non-task-oriented dialogue agent using twitter data,” in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2015, pp. 5–8.

[11] H. Yamaguchi and M. Mozgovoy, “Generating aiml rules from twitter conversations.”

[12] H. Yamaguchi, M. Mozgovoy, and A. Danielewicz-Betz, “A chatbot based on aiml rules extracted from twitter dialogues,” 09 2018, pp. 37–42.

[13] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[14] A. Paranoia, “A computer simulation of paranoid processes,” 1974.

[15] M. Shibata, T. Nishiguchi, and Y. Tomiura, “Dialog system for open-ended conversation using web documents,” *Informatica*, vol. 33, no. 3, 2009.

[16] R. Wallace, “The elements of aiml style,” *Alice AI Foundation*, 2003.

[17] R. Higashinaka, T. Meguro, H. Sugiyama, T. Makino, and Y. Matsuo, “On the difficulty of improving hand-crafted rules in chat-oriented dialogue systems,” in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 1014–1018.

[18] R. E. Banchs and H. Li, “Iris: a chat-oriented dialogue system based on the vector space model,” in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012, pp. 37–42.

[19] R. T. Lowe, N. Pow, I. V. Serban, L. Charlin, C.-W. Liu, and J. Pineau, “Training end-to-end dialogue systems with the ubuntu dialogue corpus,” *Dialogue & Discourse*, vol. 8, no. 1, pp. 31–65, 2017.

[20] R. Higashinaka, N. Kobayashi, T. Hirano, C. Miyazaki, T. Meguro, T. Makino, and Y. Matsuo, “Syntactic filtering and content-based retrieval of twitter sentences for the generation of system utterances in dialogue systems,” in *Situated Dialog in Speech-Based Human-Computer Interaction*. Springer, 2016, pp. 15–26.

[21] A. De Angeli and R. Carpenter, “Stupid computer! abuse and social identities,” in *Proc. INTERACT 2005 workshop Abuse: The darker side of Human-Computer Interaction*, 2005, pp. 19–25.

[22] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[23] “Tweepy,” *http://www.tweepy.org*.

[24] T. Kudo, “Mecab: Yet another part-of-speech and morphological analyzer,” *http://mecab. sourceforge. jp*, 2006.

[25] S. Toshinori, “Neologism dictionary based on the language resources on the web for mecab,” 2015. [Online]. Available: https://github.com/neologd/mecab-ipadic-neologd

[26] J. Ramos *et al.*, “Using tf-idf to determine word relevance in document queries,” in *Proceedings of the first instructional conference on machine learning*, vol. 242, 2003, pp. 133–142.

[27] “scikit-learn,” *https://scikit-learn.org/stable/*.

[28] A. P. Saygin and I. Cicekli, “Pragmatics in human-computer conversations,” *Journal of Pragmatics*, vol. 34, no. 3, pp. 227–258, 2002.

[29] 杉山弘晃, 目黒豊美, 東中竜一郎, and 南泰浩, “任意の話題を持つユーザ発話に対する係り受けと用例を利用した応答文の生成,” 人工知能学会論文誌, vol. 30, no. 1, pp. 183–194, 2015.