Financial Chatbot Based on LSTM

Liangde LI, Yaxin Zhang, Linfeng Zhu, Yuqiao Xie, Qi Liu
Department of Mathematics, School of Science,
The Hong Kong University of Science and Technology
{llidd, yzhangjo, lzhuau, yxieba, qliubh}@connect.ust.hk

Abstract

This article introduces a professional chat robot: Doraemon, which can answer questions related to finance. We create a financial corpus and train Doraemon to make it a professional financial professional coach. Through Python software, using the seq2seq method, a 3-layer long-short-term recurrent neural network is selected as the encoder and decoder, and the framework chooses Keras. In addition, we use another daily conversational robot built with pytorch as a comparison. In fact, because we limit some of Doraemon's answers, the model results are more stable than daily conversational robots. In this report, we first introduce our timeline and reflect on two of these articles; then, introduce the theory of the project; besides, will guide you through the steps and results of building a financial chat robot; what's more, suggestions for comprehensive and further research; and finally, personal contributions and references are listed. Code has been made available at https://github.com/zlf-zju-hkust/chatbot1, and presentation video has been uploaded to https://youtu.be/eeCuLXH9_HI.

1 Introduction

Chatbots are a hot topic in artificial intelligence research. Without many people noticing, chatbots have great potential and wide applications in modern society, including customer service APIs and internet helpdesks. The traditional chatbots are usually based on retrieval-based models, where the response is generated in specific forms. This limitation restricted the usage of chatbot to small domains. As time went by, we need more robust chatbot to deal with more general situations. We want to teach a machine to deliver a wonderful human-like expert in a specific field so that it can give you a clear explanation when you are confused with a definition, which becomes reachable as people gain deeper understanding of deep learning.

The reason that we choose finance as our topic is that it is closely related with our major, and we wish more people could have a novel and accessible way to know about finance through the chatbot we created. We aim to give users a concise and precise answer through a chat form but not that tedious and unfocused reply through search engines. Providing a way for people to get financial definitions in a easy and comfortable way is our ultimate goal.

2 Progress and Learning from Group Project

Date	Progress and feelings
March	We took two online courses related to chatbot and got the basic algorithms by reading aggressively.
April	We had several discussions and finalized our topic and decided to build a chatbot that introduces Quant, including job hunting, school ranking, career introduction. By adopting NPL and deep learning techniques, the chatbot is self-learning. At this point, the information source includes both websites and pdf.

	We submitted the proposal and divided the work accordingly 1. Scrap information from website and build language database 2. Preprocess user query and extract key words 3. Search language database and match Q-A pairs 4. Rank searched results and select the best one. 1. 5. Link with API and polish output based on ranking.
May	We got inspired by some articles, and learnt several new frameworks to build chatbot, including a persona-based neural conversational model and an end-to-end trainable task-oriented dialogue system.
	We successfully built a crawler to get information from website.
	We went through a tutorial to train chatbot using movie scripts. Besides, we learnt three new sentence functions that can be used to control the generation of response to a Chinese sentence. These three functions are interrogative, imperative and declarative. More importantly, the model can generate informational responses, which well suits the need of our project to build a chatbot that teaches people knowledge in Quant.
	Debug & Report Writing

3 Reflection on AI articles

In this section, we review two articles on AI, which are closely related to our project. Reflection on these articles inspired us a lot and provide us deeper understanding on the project.

3.1 Article I: A Person-Based Neural Conversation Model

Purpose: to address the challenge of consistency and how to endow data-driven systems with the coherent "persona" needed to model human-like behavior, whether as personal assistants, personalized avatar-like agents, or game characters.

Method: The article introduces two persona-based models:

- **Speaker Model**

  The Speaker Model is a typical SEQ2SEQ model. The innovation of this model is that it encodes personas in distributed embeddings that capture individual characteristics such as background information and speaking style and the embeddings are learned by back propagating word prediction errors to each neural component during training.

- **Speaker-Addressee Model**

  This Model is in line with the idea of Speaker Model but it considers a more detailed situation, that is speaking style, register and content does not vary only with the identity of the speaker, but also with that of the addresses.

Datasets: Training data was extracted from the Twitter FireHose, speaker ID information and scripts from the American television comedies Friends and The Big Bang Theory.

Result: Twitter Persona dataset and TV Series dataset are both observed a performance boost by nearly 20%.

Reflection: Response inconsistency is a critical issue in multiple rounds of dialogue. If the problem of consistency is not solved well, the performance of chatbot will be poor. The article presents some new framework to build chatbot, including a persona-based neural conversational model and an end-to-end trainable task-oriented dialogue system. Persona-based models can be used for handling the issue in neural response generation, which is considering user identity (such as background, user portrait, age, etc.) into the model and constructing personalized models for different users.

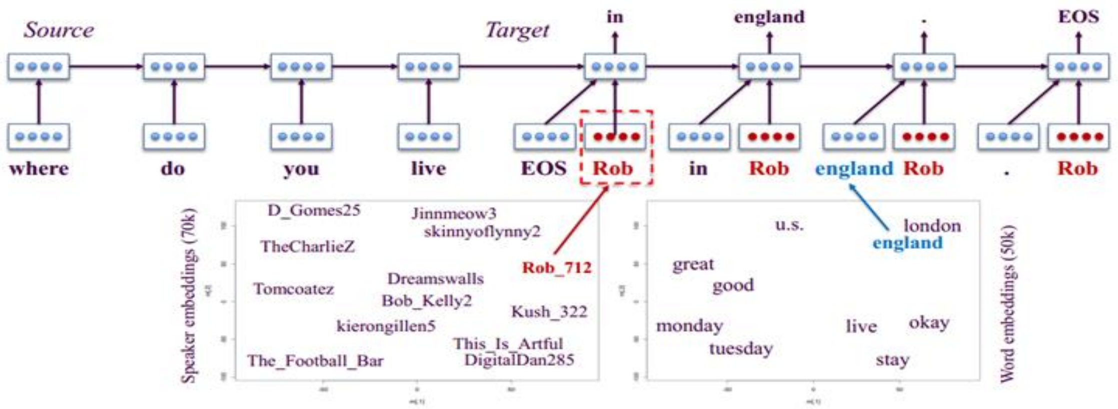


Figure 1: Illustrative example of the Speaker Model introduced in this work.

3.2 Article II: Deep Reinforcement Learning for Dialogue Generation

Purpose: Recent neural models of dialogue generation tend to be short-sighted, predicting utterances one at a time while ignoring their influence on future out-comes. The article shows how to integrate the goal of generating coherent and interesting dialogues, applying deep reinforcement learning to model future reward in chatbot dialogue.

Method: The article introduces a neural reinforcement learning (RL) generation method. The model uses the encoder-decoder architecture as its backbone and simulates conversation between two virtual agents to explore the space of possible actions while learning to maximize expected reward.

The learning system consists of two agents. A dialogue can be represented as an alternating sequence of sentences generated by the two agents: p1, q1, p2, q2, ..., pi, qi. The generated sentences are viewed as actions that are taken according to a policy defined by an encoder-decoder recurrent neural net-work language model. The parameters of the network are optimized to maximize the expected future reward using policy search, as described in the Figure 2.

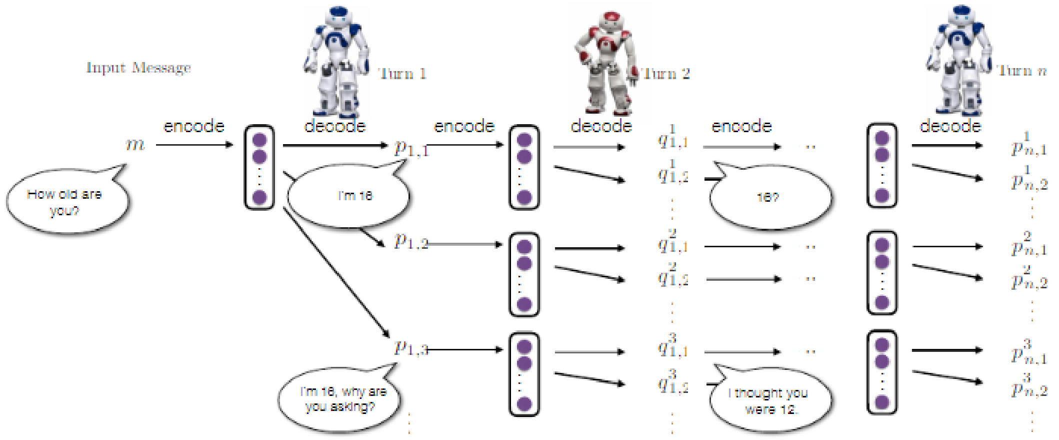


Figure 2: Dialogue simulation between the two agents.

The article using policy gradient methods to reward sequences that display three useful conversational properties: informativity, coherence, and ease of answering (related to forward-looking function). A reward is observed after the agent reaches the end of each sentence.

Dataset: The article takes a subset of 10 million messages from the OpenSubtitles dataset and extract 0.8 million sequences with the lowest likelihood of generating the response “I don’t know what you are taking about” to ensure initial inputs are easy to respond to.

Result: The article uses three metrics to evaluate dialogue systems: length of the dialogue, diversity and human evaluation. The results show that the RL based agent indeed generates more interactive responses than the other baselines.

Reflection: Modeling the future direction of a dialogue is crucial to generating coherent, interesting dialogues. The author uses the method of deep reinforcement learning to improve the effect of multiple rounds of dialogue, and puts forward three ways to define reward, which can be regarded as a good example of the combination of DRL and NLP. The article provides us a innovative method to evaluate the dialogue system accurately.

4 Model Theory

In this Python project with source code, we are going to build a chatbot using deep learning techniques. The chatbot will be trained on the dataset which contains categories (topics), pattern and responses. We use a special recurrent neural network (LSTM) to classify which category the user’s message belongs to and then we will give a random response from the list of responses.

The structure:

The chart below is the backbone of our chatbot. The starting point is the input of users, which is then processed by our engine to extract key word and analyze grammar. Then we search language database, which is built by web crawler in our case, and generate multiple sentences that matches the keyword from questions. Then we rank all potential answers and pick the one with highest consistence. The last step is to combine the pieces and output a complete answer.

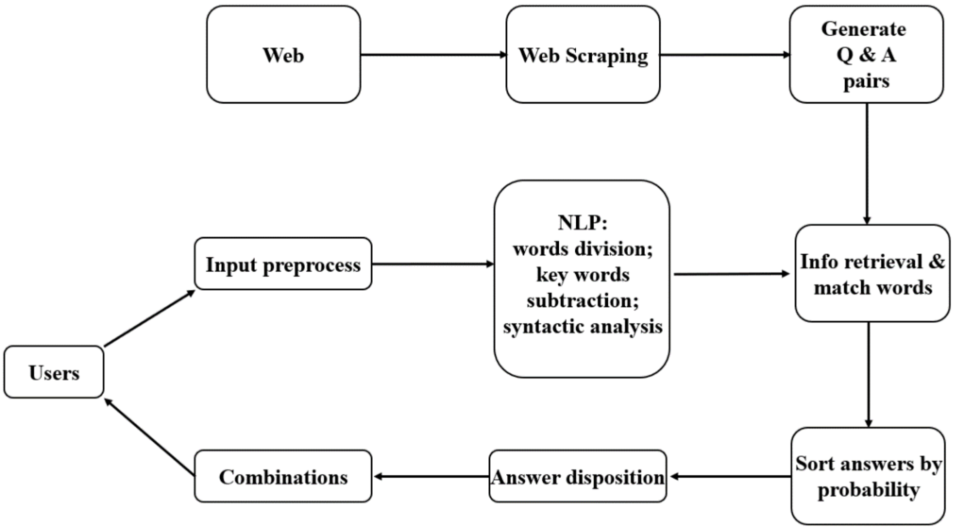

Web
Web Scraping
Generate Q & A pairs
Input preprocess
NLP: words division; key words subtraction; syntactic analysis
Info retrieval & match words
Users
Combinations
Answer disposition
Sort answers by probability

Figure 3: Backbone of the Chatbot

About LSTM

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) and were refined and popularized by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

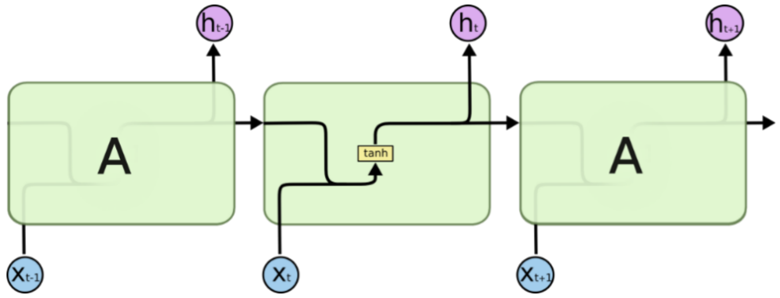


Figure 4: The repeating module in a standard RNN contains a single layer

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

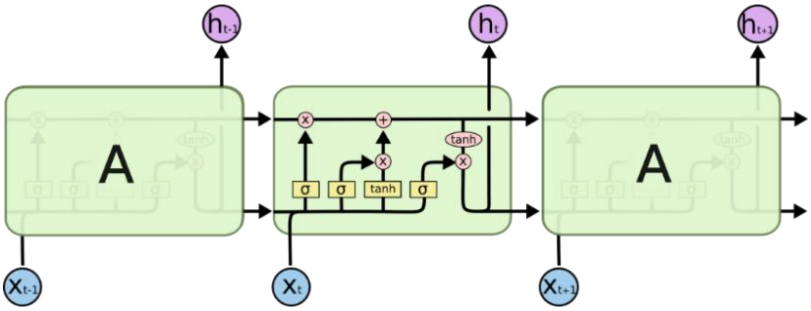


Figure 5: The repeating module in an LSTM contains four interacting layers

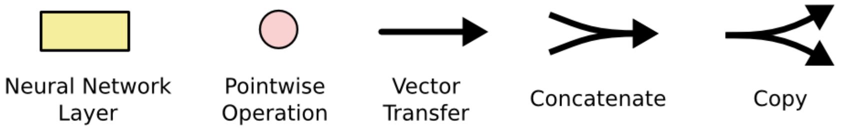


Figure 6: Symbols

About Seq2Seq

Define Seq2Seq Model We need to define the seq2seq model, which use two independent recurrent neural nets together. One RNN is called the encoder while the other RNN is the decoder. The core part of encoder is a multi-layered Gated Recurrent Unit and it iterates through the input sentence one word at each processing unit. The Decoder generates the response answers in a token-by-token way. We use LSTM for both RNNs.

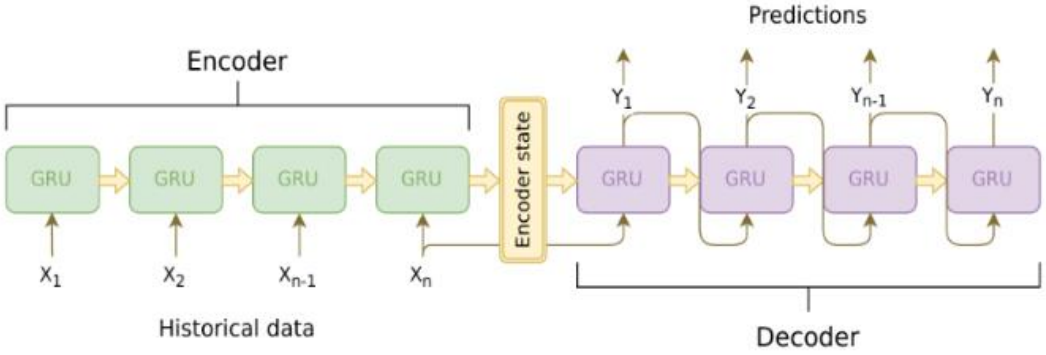


Figure 7: Seq2Seq Model

5 Model Procedures

Using Python software and Keras as a framework, the model process can be divided into 8 steps: **Corpus Creation, Data Cleaning, Using LSTM for Encoder and Decoder, SGD Compilation and Model Fitting,**

Query Processing and Prediction, Graphical User Interface, Parameters Adjustment, Model Comparison.

5.1 Corpus Creation

We hope to get a financial corpus that can answer simple definition questions. Therefore, we searched and crawled all content about finance and its expansion from Wikipedia, a total of 300 entries, and use this to create question-answer question and answer pairs.

On the one hand, we hope that it can answer some specific questions, such as "hello", "who are you", "what can you do", "thank you", "goodbye". So we added these limits and added the robot's name "Doraemon" to the answer to make it more vivid. On the other hand, when there are too many topic categories, there will be poor training or even failure-because they are all financial categories, questions and answers are inevitably similar, so we deleted some uncommon topics.

Finally, our corpus has saved a total of 63 topics, these topics include "finance", "bond", etc., and saved in Json format.

5.2 Data Cleaning

Firstly, load the necessary packages and import the corpus.json file; Secondly, we use the NLTK tokenizer to classify and tokenize files: divided into "words", "topic", and "documents", where "words" are words, "topics" are all topics, and each question and answer pairs are "documents"; Then, delete useless punctuation marks, change uppercase letters to lowercase letters, reduce the word form, etc., and encapsulate in pickle files; Finally, turn the text into numbers: according to whether the question belongs to this topic to carry out a binary classification. And the data is divided into dependent variable and independent variable, independent variable is one of the problems, and dependent variable is the topic to which the input problem belongs.

5.3 Using LSTM for Encoder and Decoder

We use the Keras sequential API to build a long-short term deep neural network with 3 layers, which is one of the Seq2Seq models. The first layer and second layer are: 128 and 64 neurons, activation = 'relu', dropout = 0.5; the number of neurons in the third layer is equal to the number of topics, activation = 'softmax'.

5.4 SGD Compilation and Model Fitting

Select the Stochastic gradient descent method to compile the model, the loss function is 'categorical crossentropy', and the learning rate is selected as 0.001. After training for 180 epochs, the accuracy of the model did not decrease significantly, so the number of trainings was chosen to be 180. When the model has no errors, it will print "Model is established", and the model was saved as 'model.h5'.

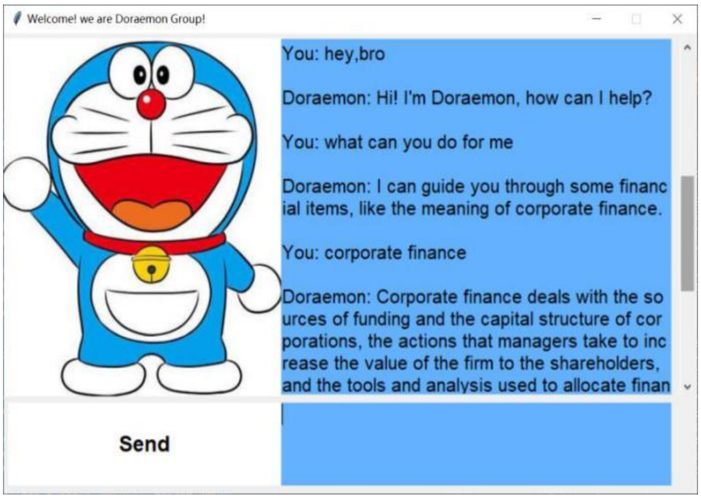
5.5 Query Processing and Prediction

When a new query appears, first, we preprocess the text so that it provides the same independent variables as during training. That is, to cut words and tokenize them: clear the word form, remove useless symbols, and change uppercase words to lowercase words. Second, we need to predict the topic corresponding to the query, and then randomly return an answer from the topic.

5.6 Graphical User Interface

In order to make chatting more vivid, we chose the Tkinter library to create a chat window, called Graphical User Interface (GUI), which allows people and computers to communicate. Besides, with the image of Doraemon on the side, the theme color also chose the color related to it.

This window has three parts: one is the message input by the user; the second is a send button, after clicking, the program receives the input query and processes it in the previous step; the third is a dialog window that displays and records the customer input and the answer returned by the model. Screenshots of some chats are as follows:



You: hey,bro

Doraemon: Hi! I'm Doraemon, how can I help?

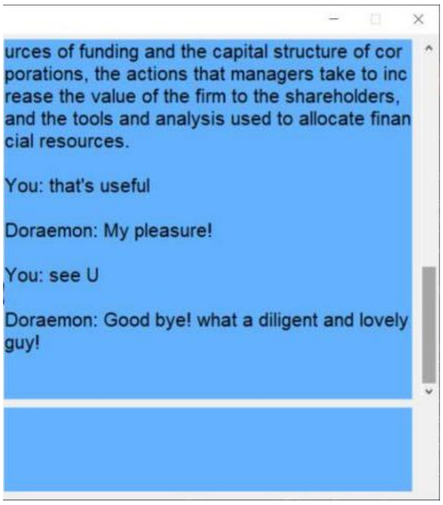
You: what can you do for me

Doraemon: I can guide you through some financial items, like the meaning of corporate finance.

You: corporate finance

Doraemon: Corporate finance deals with the sources of funding and the capital structure of corporations, the actions that managers take to increase the value of the firm to the shareholders, and the tools and analysis used to allocate finan

Figure 8: GUI screenshot



urces of funding and the capital structure of corporations, the actions that managers take to increase the value of the firm to the shareholders, and the tools and analysis used to allocate financial resources.

You: that's useful

Doraemon: My pleasure!

You: see U

Doraemon: Good bye! what a diligent and lovely guy!

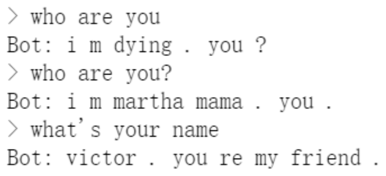
Figure 9: GUI screenshot (cont.)

5.7 Parameters Adjustment

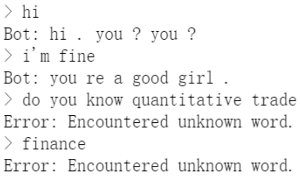
In this article, the main adjustable parameters are in the fitting part of LSTM and SGD. We can choose different learning speeds and training times. In fact, the current parameters are appropriate. For example, when the learning speed rises to 0.01, the model will have vocabulary confusion, that is, it is not sensitive to words; and a small increase in training times will have little effect on model results.

5.8 Model Comparison

We compare our financial robot with daily conversation chatbot, both of which are retrieval-based models. The latter is built with Cornell Movie-Dialogs corpus, pytorch framework, and bidirectional RNN model. Although it is a rich dataset of movie character dialogues, it contains 220,579 dialogues of 10,292 pairs of movie characters. But the result is not satisfactory, some chat screenshots are as follows:



```
> who are you
Bot: i m dying . you ?
> who are you?
Bot: i m martha mama . you .
> what's your name
Bot: victor . you re my friend .
```



```
> hi
Bot: hi . you ? you ?
> i'm fine
Bot: you re a good girl .
> do you know quantitative trade
Error: Encountered unknown word.
> finance
Error: Encountered unknown word.
```

Figure 10: daily conversation chatbot screenshot

Figure 11: daily conversation chatbot screenshot (cont.)

We can see that the daily chat robot has inappropriate answers and unstable questions. These are very fatal; however, in our model, because of the effort and energy in the establishment of the corpus, it makes its answers more appropriate. In addition, because we set up answers in the corpus in advance to encounter words that have not appeared, we can be more decent when encountering new words.

6 Synthesis and Suggestions for further study

In this project, we implement a recurrent sequence-to-sequence models in keras. Then we jointly train encoder and decoder models using mini-batches and implement greedy-search decoding module. In this way, we build a chatbot based on deep learning, applied LSTM in our encoder and decoder, scraped the text from web and processed them into Q&A pairs, we tried to use all automatic methods in our project and have checked several algorithms. We choose the one with best performance finally, the output is really surprising. The chatbot is like a real person chatting with you!

Our innovations are: 1. Use crawlers to get a financial corpus and re-process it manually 2. Create a financial chatbot that is both unique and useful to us 3. Using a GUI, the chat interface is pleasant and meets our original intention-create a Doraemon! Nevertheless, there are still rooms for improvement in the process. Here are some suggestions for future study.

Data amount & Cleaning: There are many sources of data about a specific topic, but we only use data from Wikipedia, which means our data set is not abundant enough. It can be solved through crawl more websites. Besides, data cleaning and data analysis are of vital importunate. Data from ordinary website is easy access but biased. Therefore, we need to do a tradeoff and do more data cleaning. There are many new techniques and researches in these areas that could be applied, such as how to dig out key words, clustering, emotion analysis, etc.

Evaluation Method not that objective: While using deep learning to improve the performance of multiple rounds of dialogue, it is important to evaluate the accuracy of the response. By far, how to evaluate accurately in the dialogue system has always been a major issue. In this project, we evaluate the accuracy of the chat based on our subjective judgement about the match between Q & A which may not be that objective. We can apply more quantitative evaluating methods. For open-domain dialogue system, the main evaluation methods include objective metric evaluation and simulated human scoring Simulated manual scoring. Different evaluation method can be conducted to better improve the response.

Anyway, we feel truly honored to take this course and really appreciate the teaching and chatting with both professors. We have learnt about a lot, not only AI algorithms and its applications, but also the techniques in job search which are both practical. Professors best wishes to you!

7 Individual Contribution

Name	ID	Contribution
Liangde LI	20653566	Report 1,2,3,4 & Presentation
Yaxin Zhang	20661795	Corpus preparation & Coding & Report 5 &Presentation
Linfeng Zhu	20652201	Coding, Report 1,2,3,4 & Presentation
Yuqiao Xie	20655851	Corpus preparation & Coding& Report 1,4,8 & Presentation
Qi Liu	20659974	Corpus preparation & Report 2,3,4,6 & Presentation

8 References

[1] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. arXiv preprint arXiv:1603.06155.

[2] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proc. of the International Conference on Learning Representations (ICLR).

[4] Ali Ahmadvand, Ingyu Jason Choi, Harshita Sahijwani, Justus Schmidt, Mingyang Sun, Serge Volokhin, Zihao Wang, and Eugene Agichtein. 2018. Emory irisbot: An open-domain conversational bot for personalized information access. Alexa Prize Proceedings

[5] Ian Goodfellow and Yoshua Bengio and Aaron Courville. 2016. Deep Learning. MIT Press.

[6] Dataflair team. Python Chatbot Project – Learn to build your first chatbot using NLTK & Keras. 2020

[7] Build an IT support chatbot by using IBM Watson Assistant

[8] Dan Shewan. 10 of the Most Innovative Chatbots on the Web. 2020

[9] Transformers: State-of-the-art Natural Language Processing for Pytorch and TensorFlow 2.0.

[10] DeepAI: The front page of A.I.

[11] Pytorch-CHATBOT TUTORIAL

[12] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[13] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, Advances in optimizing recurrent networks, CoRR, vol.abs/1212.0901, 2012.