A NLP-based Chatbot

Introduction

A chatbot is a computer program designed to simulate conversation with human user. It can processes natural-language input from a user and generates smart and relative responses that are then sent back to the user. The earliest version of chatbot was called Eliza, which was developed at the MIT Artificial Intelligence Laboratory in 1966. Eliza served as an impersonated psychotherapist which examined the keywords of user input and then triggered the output according to a defined set of rules. With advances in technology, various chatbots were launched in smartphone era. Siri by Apple was the first conversational virtual assistants, which is able to provide useful information including weather, stocks and locations. Soon after Google launched their Google Assistant for Android, then followed by Amazon Alex and Microsoft Cortana. In recent years, with the booming development of AI technology, many bot platforms were introduced to allow developers to publish chatbots with various services such as polls, news, games, integration, and entertainment[1]. Popular bot platforms include Telegram, Facebook Messenger, Slack and Skype.

Nowadays, chatbot is applied to a wide range of industries, including e-commerce, insurance, healthcare, education and marketing & advertising. Due to its accessibility, efficiency and availability[2], chatbot has brought great commercial profits to companies. Therefore, the growing demand for chatbot system development in different industries can be predicted in the future.

Background

Intent is a term used for programmatically identifying the intention of the user who is interacting with the chatbot[1]. Before building a chatbot system, developers should consider possible user senarios and decide what actions the chatbot will be able to perform. Intent recognization is always the first step for a chatbot to understand the user input. For example, if a user says "I want to order food" to a chatbot, it should be able to identify the intent as "order_food". Also, when a user says "what will the weather be like tomorrow?", the intent will be "get_weather". Intent classification is a typical machine learning problem, which can be solved by using a technique called text classification. A classifier is an algorithm which can classify documents into multiple classes. In a chatbot system, the user input will be classified into corresponding predifined intent for further processing. There are various types of classifiers including linear models, probablistic models, similarity-based methods and so on.

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence that enables computers to process and analyze the human language. There are many fundamental methods of NLP that can help to build chatbots. Tokenization is one of the basic concepts of NLP where we split a text into meaningful segments (tokens) for further processing. Part of speech (POS) tagging assigns each word or token their grammatical function in a sentence, such as noun, verb and adjective. These tags are used as word feattures which can help to filter information and disambiguate meaning. Word standardisation includes stemming and lemmatization. Stemming is the process of extracting the common part of the word inflections to their word stem, while lemmatization is the algorithmic process of determining the lemma of a word based on its dictionary form. Word filtering means that some useless data need to be filted out from the input document, including high-frequency words that have no discriminative power and low-frequency words that are not really representative.

Information retrievel (IR) in computer science is the process of searching for information from stored data and returning relevant documents that satisfy user's requirement. Question & answering (Q&A) is an intersection of IR and NLP, which is concerned with building systems that automatically answer user questions in a natural language.

Proposed system

In this part, we will discuss how to build a NLP-based chatbot system with python, which contains the functions of intent classification, identity management, small talks and question & answering.

1. Text Classification

First of all, we need to collect some training data relating to the three intents (identity management, small talks and Q&A) respectively. For identity management intent, the chatbot will be able to detect name statements and store the name that will be used in follow-up conversation. For small talk intent, the chatbot will be able to handle user input such as greetings, random questions or short snippets of information, which is a funny way to build connection between chatbot and users. For Q&A intent, the data should cover a wide range of topics so that the chatbot can retrieve useful information to answer users'questions.

Then we need to preprocess the training data, including tokenization, filtering (stopwords and low frequency words) and word standardisation. These data can be used to build a bag-of-word model, which is a representation of documents that can transfer the text data into a common vector space. The dimension of the vector space is the size of the vocabulary of the text data. Each document is a vector in that vector space, where the frequency of each word in that document is a dimension of the vector. The function CountVectorizer in Scikit-Learn is an effective way to do the tokenising, filtering and item counting in documents. It also allows us stem the input data before feeding it into the CountVectorizer vocabulary.

TF-IDF (Term Frequency-Inverse Document Frequency) is a good standard term weighting technique because it gives more weight to important words and less to common words.The TfidfTransformer method of Scikit-Learn lets us apply transformation functions on the simple counts of the CountVectorizer.

After representing the document in a fixed-size vector and putting appropriate weights to each term, we can feed the document vectors into machine learning algorithms. Logistic Regression algorithm is a simple and fast classification algorithm used in NLP which relies on strong statistical foundations. We can use this classifier to implement intent classification by mapping the user input to predifined catagories.

2. Information Retrieval

There are three key components to build a retrieval-based chatbot system: (1). represent documents and query in the same space; (2) Use a relevance function to evaluate the query; (3) Rank documents by decreasing relevance.

The core idea of both small talk and Q&A process is similarity matching. Firstly we need a huge data set of question-answer pairs, which then should be mapped into a common vector space with the user query, same with text classification. Then we should find the most similar question in data set to the query by computing similarity of query vectors and question vectors. A traditional NLP similarity measure is cosine similarity which compute the cosine of the angle of the document vectors. We can use Scipy to implement the cosine distance of two vectors and take the inverse of the distance to get similarity. Once the documents is ranked by decreasing similarity, we can extract the most matching question and retrieve the corresponding answer which is considered as the most relevant one to the query. To avoid being a rigid chatbot, every question is paired with several answer templates so that the retrieved answer is random.

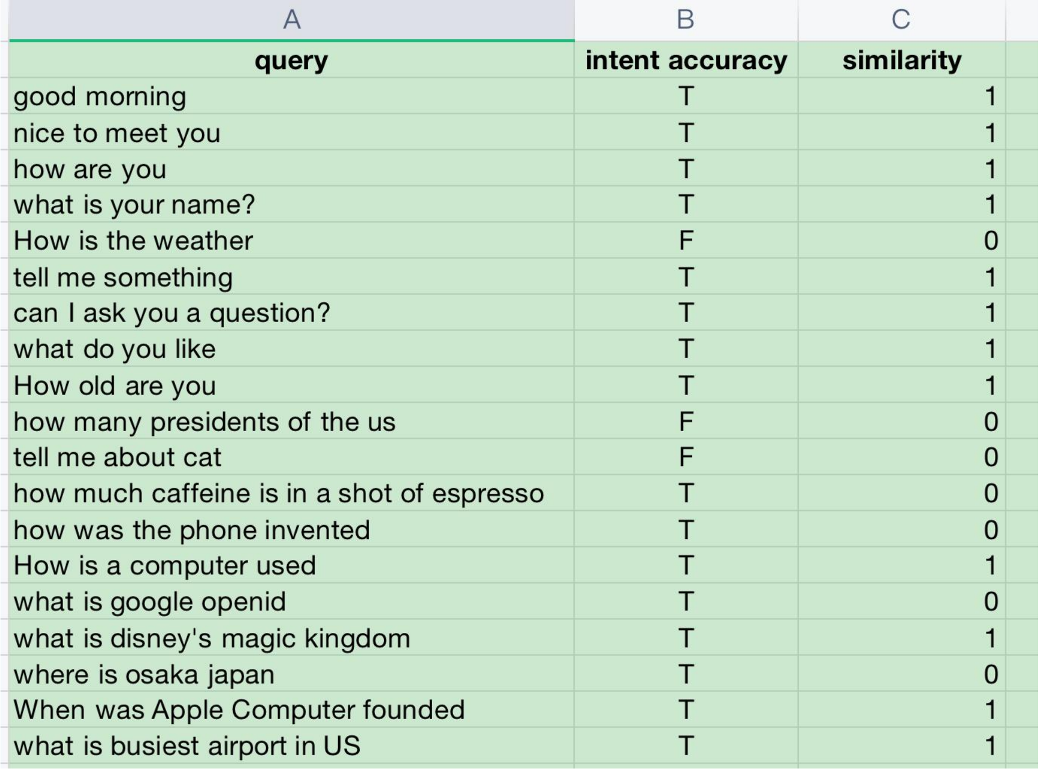
The slight difference between small talk and Q&A process is about document pre-processing. Small talk is about concise dialogue like greetings, so there is no need to remove stopwords otherwise there will be little useful information left. However, it can improve the efficiency to do the similarity matching in Q&A session by filtering high-frequency and useless words.

3. Identity Management

Identity management is about managing the name of the user, including detecting statements containing user name and storing the name, as well as using the stored name to address the user in output. We can use regular expression to detect the predifined sentence pattern and extract useful information (the user name). Besides, we can use template-based natural language generation (NLU) to generate fixed-structured output which containing the user name. Although the output of template-based system is repetitive and unimaginative to some degree, it has low error rate due to its grammatically coherent templates. Therefore, it is an appropriate way to implement the identity management.

Evaluation

To test the feasibility and effectiveness of the chatbot system, building a benchmark dataset is practibcable. The testing dataset includes 20 pairs of questions and expected answers, which then used to evaluate the accuracy of intent matching and similarity matching of the system.



A	B	C
query	**intent accuracy**	**similarity**
good morning	T	1
nice to meet you	T	1
how are you	T	1
what is your name?	T	1
How is the weather	F	0
tell me something	T	1
can I ask you a question?	T	1
what do you like	T	1
How old are you	T	1
how many presidents of the us	F	0
tell me about cat	F	0
how much caffeine is in a shot of espresso	T	0
how was the phone invented	T	0
How is a computer used	T	1
what is google openid	T	0
what is disney's magic kingdom	T	1
where is osaka japan	T	0
When was Apple Computer founded	T	1
what is busiest airport in US	T	1

Discussion

We can see from the evaluation that both intent matching and similarity matching can go wrong. Intent classification errors occur when the number of words in query is quite small. It is easy to confuse the intent between small talk and Q&A when the query is short and in a sentence pattern which is similar to both intent. For example, the query with Q&A intent starting with "who"and "how"is of great chance to be classified into small talk intent, getting an irrelevant answer which seems a little wired. Similarity matching errors usually result from various factors, including the lack of vocabulary of training dataset, different word orders and sentence patterns, etc.

There are some possible solutions may help to fix these mistakes. Firstly, balancing the amount of training data among different intents may help to reduce the error rate in intent classification. And it may be helpful to choose different methods when processing the data, such as word standardisation and weighting functions. Besides, we can try different types of classifiers like probabilistic models and similarity-based methods. As for similarity matching, we may try different similarity functions such as Jaccard index and Euclidean distance.

Small talk is a funny function of a chatbot. Actually, small talk inputs maybe the most common messages a chatbot will receive. It can output oral language when a user tests it, which makes it more like a human who is genuinely listening and responding to the user queries[4]. A smart chatbot with excellent small talk function can easily build trust with users, while a dull one will leave a bad impression on users or even lose them.

Q&A chatbots have many benefits in the real world in many aspects. It can not only bring big profit for companies by reducing repetitive human work, but also can benefit customers a lot by offering plenty of services quickly and effectively. However, sophisticated functionality means complicated coding, thus more bugs and errors which may lead to poor service and cause loss to companies..

The main bias in the chatbot system may come from the data corpus. The training dataset is the only learning resource of a chatbot. If there is not enough data or showing a lack of diversity in the dataset, it will skew the chatbot's ability to understand the user intents and motivations thus leading to a low system accuracy. Beesides, it is important to judge the quality of content for that training dataset, which may contain human-like biases. The language manner also matters because we do not want a rude chatbot. In addition, bias can hide in word embedding becasue documents are represented as vectors to compute similarity, which are used to decide semantic meaning.

Conclusion

This report has gone through a complete process about building a chatbot system with functions of identity management, small talk and question & answering. The introduction part discusses the development history, present situation as well as the general trend of chatbot systems. In background part, a primary termiology intent is introduced, along with many NLP techniques such as tokenization, word standardisation and word filtering. From the proposed system part, we can see how to use python and its libraries to implement text classification and information retrieval, which are the two core parts of building a chatbot system. Besides, we use a benchmarking single system to test the feasibility and effectiveness of the system, and have discussed the potential impact and overall fairness about this system in real world.

Reference

1. Sumit Raj. Building Chatbots with Python Using Natural Language Processing and Machine Learning; 2019

2. Rashid Khan Anik Das. Build Better Chatbots A Complete Guide to Getting Started with Chatbots; 2018

3. https://medium.com/@divalicious.priya/creating-a-chatbot-using-the-basic-ml-algorithms-part-1-70f6af52c2e3

4. https://medium.com/twyla-ai/40-small-talk-questions-your-chatbot-needs-to-know-and-why-it-matters-63caf03347f6