Hong Kong University of
Science and Technology

MSBD5014 Independent Project

Classifying Intentions in User Queries

Author:
Huang Yilun
20476669

Instructor:
Prof. LIN Fangzhen

1 Problem Description

Nowadays, there are millions of customer services whose everyday jobs are solving problems from customers of their products. However, thousands of problems in a specific product or area are very similar and even the same. That means it is a great waste for paying such a lot human resource to response users' requests. If there is a solution that the system can automatically answer the questions of users and these answers are satisfactory for customers, enterprises or government organizations can save a lot of money which will definitely push the industry forward.

This is the job of intelligent robot of customer service. In this project, we aim to solve the first step in the whole pipeline —classifying users' intentions in their queries.

Our dataset comes from a mobile operator in mainland China. It includes nearly 70,000 rows of text records of dialogues between their customers and services.

In this project, as a member of a three-people team, I am mainly responsible for data preprocessing and word embedding. So this report will mainly focus on the part of my duty.

2 Data Preprocessing

The original datasets has over 70,000 rows. We select three useful columns that respectively represents sessionID, request and response. Each row is either a request of an user or a response of a service.

In Figure. 1, it shows overall rows of an unique session ID. We can easily find that there are many redundancies that need to be cleaned up.

There are five main steps in our data preprocessing: deleting, cleaning, filtering, merging and splitting.

	sessionid	id	acceptnumber	requesttime	request	serviceid	responsetime	response
0	13410000258T16081911402229APP	63	10000000258	19AUG16:11:40:27	人工	SZ31537	31DEC99:00:00:00	-1
1	13410000258T16081911402229APP	64	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:40:40	哈喽，无论晴天雨天，萌萌达真人小和（工号SZ31537）始终在您身边，请问有什么可以为您效劳...
2	13410000258T16081911402229APP	65	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:41:19	亲，您已进入人工服务，请问有什么可以帮到您?
3	13410000258T16081911402229APP	66	10000000258	19AUG16:11:41:30	实名制了怎么还不能打电话?	SZ31537	31DEC99:00:00:00	-1
4	13410000258T16081911402229APP	67	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:42:13	请问是本机吗?
5	13410000258T16081911402229APP	68	10000000258	19AUG16:11:43:00	是	SZ31537	31DEC99:00:00:00	-1
6	13410000258T16081911402229APP	69	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:43:29	我帮您开机，请稍等哈
7	13410000258T16081911402229APP	70	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:44:15	已经帮您开通了哦
8	13410000258T16081911402229APP	71	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:45:02	亲，请问还有其他可以帮到您吗?
9	13410000258T16081911402229APP	72	10000000258	19AUG16:11:45:55	那我试一下。	SZ31537	31DEC99:00:00:00	-1
10	13410000258T16081911402229APP	73	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:46:07	好的
11	13410000258T16081911402229APP	74	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:46:17	如果没有其他业务，小和先退下了，收到10086046短信后请回复数字1，赏赐小和本人1008...
12	13410000258T16081911402229APP	75	10000000258	19AUG16:11:46:43	可以了谢谢	SZ31537	31DEC99:00:00:00	-1
13	13410000258T16081911402229APP	76	10000000258	31DEC99:00:00:00	-1	SZ31537	19AUG16:11:46:51	不客气

Figure 1: Data preview

2.1 Deleting

First, we want to delete useless sentences in the queries. It is easy to find that their are some polite formulae in both requests and responses such as saying ‘hello’ or ‘thank you’. We decide to detect these sentences based on their frequency in the whole dataset. If their frequency is higher than the threshold, then we delete the row it belongs to.

After this step, the dataset is compressed to 64241 rows.

2.2 Cleaning

Second, we want to clean the inside of each sentence. For example, there are some meaningless words and symbols in the dialogues such as ‘亲’, ‘尊敬的’, ‘<P>’ and so on. We find them by observation on dataset and our Chinese background knowledge. Then we delete them by using regular expression.

2.3 Merging

As is described above, each row is a either a request of an user or a response of a service in our dataset. We need them to be request-response pairs for our model building. So we try to merge the continuous rows with the same sessionID that are continuous responses or requests.

After this step, the dataset is compressed to 15260 rows.

2.4 Filtering

Third, we want to select the sentences that are more useful to build a pure training set for our further clustering and training in the following steps. That means we need to filter the sentences that have certain information containing specific business. We try to build up a list of keyword which describes every domain of this mobile operator. Then we select the sentence including these keywords.

After this step, the dataset is compressed to 8696 rows.

2.5 Splitting

Lastly, we need to split the sentence into words for further processing. Using python package jieba, we split the sentences into words.

After splitting, we implement a stop word removal by a stop word list. Stop words are the words or symbols that are useless or meaningless like '了、的、地' in Chinese.

	sessionid	request	response
0	13410000258T16081911402229APP	实名制了怎么还不能打电话?	请问是本机吗? 我帮您开机，请稍等哈

Figure 2: Cleaned data

Finally, we get our cleaned dataset which is suitable for following steps of modeling. For example, the session with ID 13410000258T16081911402229APP

in Figure. 1 is transformed to only one row in Figure 2. We extract the main information of this session which will benefit following model building.

The dataset includes three columns: sessionid, request and response.

3 Model Implementation

3.1 Word Embedding

There are two ways of word embedding.

The first one is using a pre-trained model for Chinese word embedding which we can get from the Internet.

The second one is using our own corpus to train a new word embedding model.

Both ways use an algorithm called word2vec.

Compared with the first method, the second one shows a more accurate embedding as the corpus used for training is more related to the context of this job.

3.2 Sentence Embedding

After word embedding, we need to embed a sentence since both requests and responses are sentences.

There also exists several methods for sentence embedding. The first and most traditional way is using the average value of all the word vectors in the sentence. It is computationally cheap and easy to implement.

We also try some other ways. One is using a recurrent neural network (LSTM) to train a model whose input is the word vectors by word2vec and output is the word vectors by one-hot encoding. This model try to discover the relation between words and sentences. After training, we can use this RNN to implement sentence embedding. We called it encoder-decoder model.

The other one is using weighted average rather than pure average. As we have built a key word list which contains some important word of the business. We can put more weight on these words.

The last one is using IDF for sentence embedding. In this way, we give more weights on the words that have larger IDF.

3.3 Label requests by clustering on responses

There is a crucial problem in this project that the training samples don' t have labels for classification. Therefore, we need to label the samples at first, which is an important foot stone for classification on users' intention.

The traditional method of labelling is manual recognition which cost a lot of time and resource, especially in today that has a huge volume of data.

Rule matching by keywords is a more advanced method than manual recognition as it can automatically label the samples that is matched by some specific rules. It is useful when the data is somehow clean and the context is somehow limited.

According to our dataset, we try to implement a new labelling method. We treat this problem from the view of responses of customer services rather than directly take requests as a start, because responses is less diversified compared to users' requests. So we try to do clustering on responses and based on the result of clustering, we give a label to its corresponding requests.

In order to evaluate the label quality, we first label some samples manually. According to the business of this mobile operator, we set up seven domains of users' intentions.

We have tried three clustering algorithms: k-means, DBSCAN, agglomerative clustering. Finally, we find that k-means have a better performance and according to elbow method, we surprisingly find that it is really rational to set the number of clustering to seven.

After clustering, we can manually give a label to each cluster according to our observation on the result of clustering.

Then we can get our training set for classification.

Finally, we can compare the clustering result of each sentence embedding method (Figure. 3).

In this figure, we show the distribution of clustering results of each model.

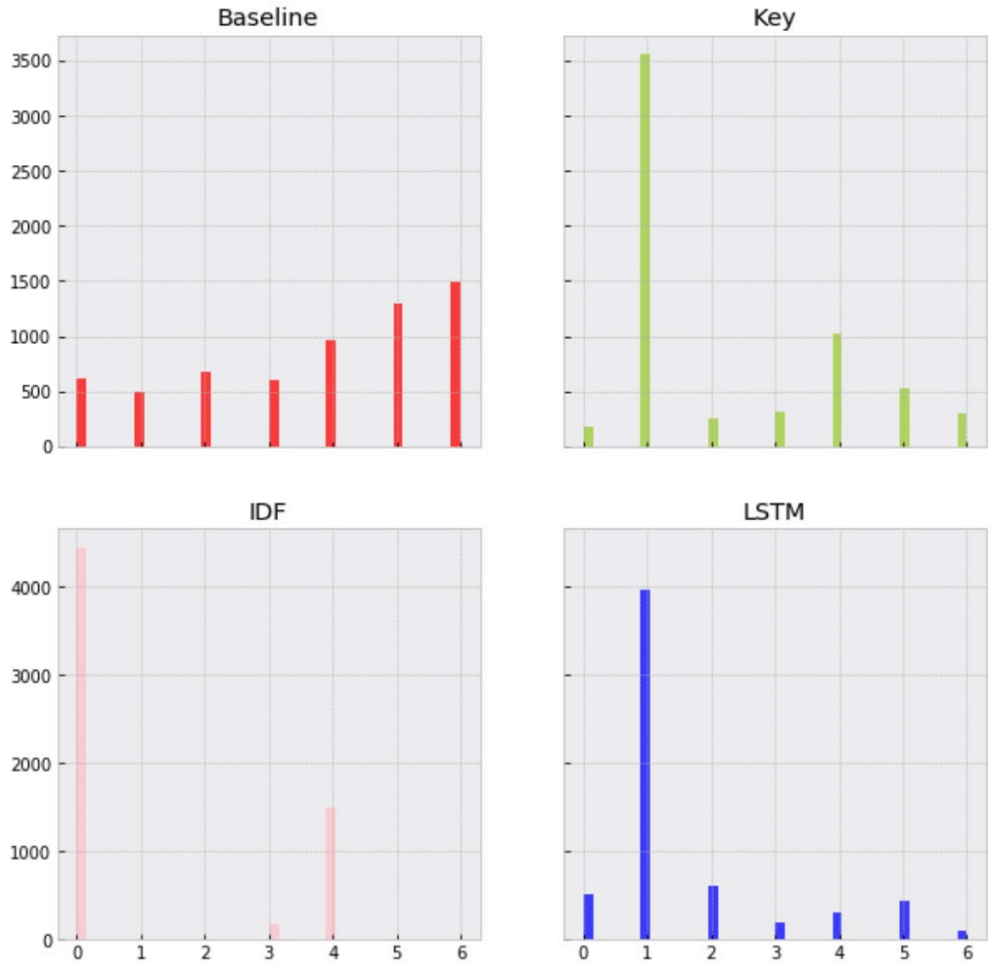

Figure 3: Distribution of Clustering

3.4 Classification

After labeling our original samples, we get a complete training set for a classification model about the column of request.

As this is a problem about natural language processing and the input is a series of words, we decide to use recurrent neural network (RNN) to build a classification model. We use a bidirectional LSTM architecture (Figure. 4). The input is a sequence of word vectors in each request. After the LSTM unit, we add a time distributed layer and two dense layers to get a softmax output. We have tried to tune the parameters of the model such as batch size, drop out, etc.

We split the dataset into training set (70%) and validation set (30%) to preview the quality of our model.

This table shows the accuracy of the models based on four kinds of sentence embedding methods.

Model	Train accuracy	Test accuracy
Baseline model (Average)	0.441	0.372
Weighted Average based on IDF	0.625	0.586
Weighted Average based on Keywords	0.674	0.647
Encoder-decoder model	0.705	0.709

4 Conclusion and Way Forward

From the above table, we find that the encoder-decoder model has the best performance for this question. And as encoder-decoder model give a somehow proper result about distribution of clustering, we can determine the encoder-decoder model is better at all in this project.

In this project, we implement a pipeline about NLP including data cleaning, word and sentence embedding, clustering for labeling and classification.

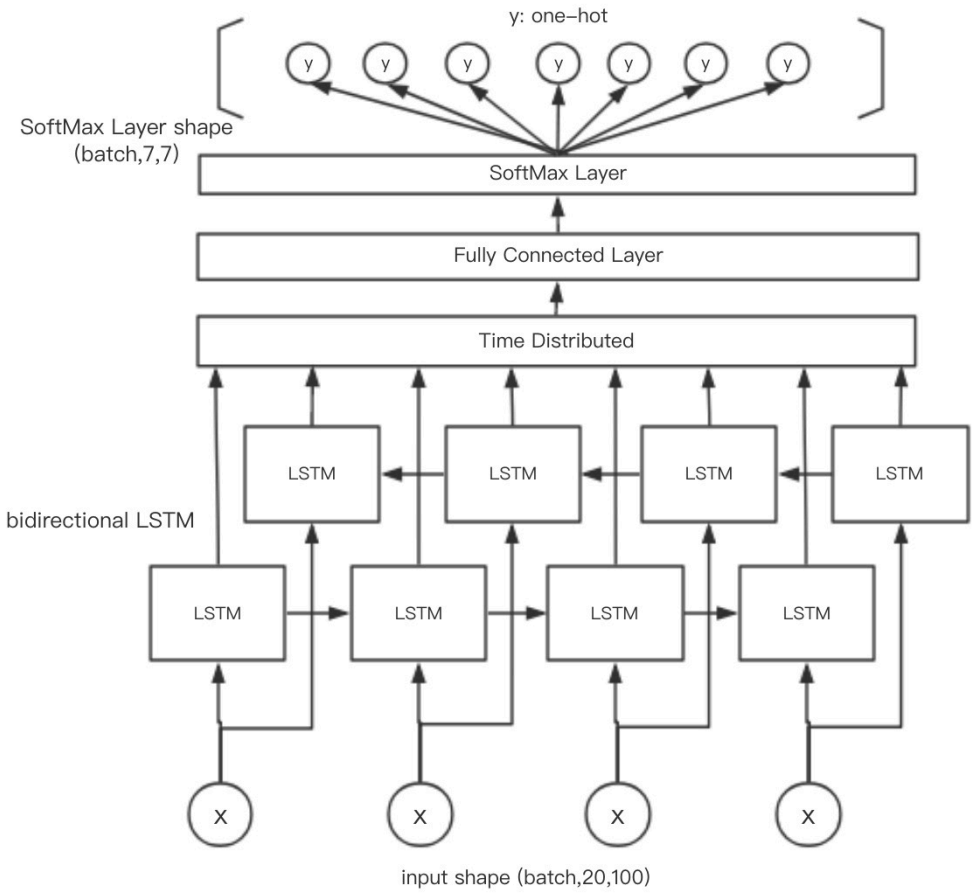

Figure 4: Architecture of LSTM model

From this pipeline, we build up a model to classify a user's request for some specific intentions.

However, there are some drawbacks of our works that can be improved in the future:

- Quality of labeling. As clustering is a unsupervised learning algorithm, it's hard to define a meaning for a cluster. And as the data is not pure at all, some arbitrary sentences is likely to be wrongly labeled. So

it' s crucial to find out a better way for labeling samples. Maybe it' s a feasible way to combine clustering with rule matching.

- Fine-tuning the RNN classification model. As there are lots of hyperparameters in the model, we should pay more effort on the choice of them. Furthermore, the architecture of the network can also be changed for exploration.

- Fine-tuning the RNN classification model. As there are lots of hyperparameters in the model, we should pay more effort on the choice of them. Furthermore, the architecture of the network can also be changed for exploration.

5 Meeting Minutes

5.1 Feb. 7

Participants: Prof. Fangzhen Lin, all registers of the project.
Content:
1. Introduction and intuition of the project
2. Data source introduction and preview
3. Q&A session.

5.2 Feb. 28

Participants: Prof. Fangzhen Lin, Huang Yilun, Lu Guannan, Mi Lan
Content:

1. Project details discussions, including each steps of user intention classification, model candidates, automatic tagging using clustering

2. Feedback from Prof.Lin: take rules into consideration, learn from IBM knowledge grapf building process, ensemble learning.

5.3 Apr. 9

Participants: Prof. Fangzhen Lin, Xiao I Robot, Lu Guannan, Mi Lan
Content:

1. Middle report of our project.

2. Prof.Lin and Xiao I Robot raised question about data cleaning and automatic tagging part of our project.

3. Xiao I Robot introduced their methods to solve this problem in current stage

5.4 May. 11

Participants: Prof. Fangzhen Lin, Huang Yilun, Lu Guannan, Mi Lan

Content:

1. Report the progress of the project. It mainly includes four parts: data preprocessing, word/sentence embedding, clustering, classification. We show our results of part 1-3. Prof. Lin carry out some ideas and questions of our solution.

2. Discuss about how to enhance the performance of clustering (labeling).

3. Prof. Lin emphasize the classification is the main part of the project. We should pay more attention on it.

4. Discuss some details and requirements of the final report.