

About

This assignment is a part of evaluation process for candidates who applied for a role at our organization.

Data

You are being provided with the required information in seven datasets.

Background

Consider a user talking to an advanced conversational system. The system has many complex algorithms incorporated in it. Whenever user asks the system to perform a certain task, the system takes in the text input, classifies it to a particular intent (understanding which task the user wants it to perform, example: Finding a chinese restaurant, booking a flight, opening account balance etc.), finds slots from the query, take appropriate actions and give response accordingly.

Problem

You need to implement from scratch in Python/ PySpark/ TensorFlow.

You have to create an intent classifier model from the data provided. This classifier should correctly classifies intent of user. This model need not take previous utterances into consideration assuming the data to be context free. Also create a slot filling model from the data provided that fills the slots for the test data.

Steps to follow:

1. Prepare a proper training data from the raw data provided. Seven data files have been provided:

- (1) intents.csv : Containing name of 26 intents.
- (2) Vocab.csv : Vocabulary containing all the words present in the data.
- (3) Train.csv : Containing 4978 queries from users such that each number in a query corresponds to the index of the word in vocabs.csv (starts from 0 index in data)
- (4) train_intent.csv : Containing 4978 intents corresponding to the 4978 queries in train.csv such that each number represents the index of the intent in intents.csv (starts from 0 index in data)
- (5) test.csv : Containing 893 queries from users such that each number in a query corresponds to the index of the word in vocabs.csv
- (6) slots.csv : contains 129 slots for the data.
- (7) train_slots.csv : Contains slots for each word of query such that each number represents the slot index in slots.csv (starts from 0 index in data)

Prepare a training data having two columns 'query' and 'intent' having the original text form instead of the numbers. Similarly prepare data for slot filling.

2. Apply proper preprocessing techniques to clean both the data and convert it to a suitable representation (example word2vec, tfidf, n-grams etc.) for the model to accept.
3. Apply feature engineering to include more important features necessary.
4. Create a RNN model (encoder-decoder) that classifies the intent. Similarly create a RNN seq2seq model that will sequence label the slots.

OR

Create a single encoder-decoder model that will classify intents as well as label slots.

5. Answer these questions (a to d) in a separate document:
 - a. Which representation method did you use and on what basis did you decide to select that method?
 - b. What all methods did you try to increase the accuracy for intents?
 - c. Suppose the data required the model to take context into consideration.
How can we incorporate that into our current model?

(Explanation of context: Take this example,

user: Book me a flight.

System: What is your destination?

User: Bangalore

System: What is the date and time?

User: Tomorrow 6 pm.

System: Booking a flight to bangalore for tomorrow 6 pm.

In this conversation, when user first says he wants to book a flight, the system classifies the intent from your classifier model as book_flight. Then, when the user says bangalore, your classifier model fails here since it cannot classify "banglore" into "book_flight" intent because it is just a city name. The classifier needs to know the context in which the user is saying 'bangalore'. In other words, the classifier should 'remember' the conversation that is happening with the user since start.

- d. What approach can you think to solve this?)

Submission

Develop the code as optimized as possible including comments at each stage.
Also print the resulting output at each stage above.

Evaluation

You shall be scored on the following:

1. Approach
2. Understanding
3. Implementation

Confidential