**Part 1a: text classification**

Attached Files:

* [[File](https://uu.blackboard.com/bbcswebdav/pid-4605938-dt-content-rid-63502065_2/xid-63502065_2) dialog\_acts.dat](https://uu.blackboard.com/bbcswebdav/pid-4605938-dt-content-rid-63502065_2/xid-63502065_2) (689.584 KB)

**Dataset description**

The data used in the project consists of the dialogs from the second Dialog State Tracking Challenge (DSTC 2, see <https://www.microsoft.com/en-us/research/event/dialog-state-tracking-challenge/>). The data consist of 3235 dialogs in the restaurant domain. Each dialog represents an interaction between a user and a system. The user is given a task to obtain a recommendation for a restaurant according to a number of preferences. The dialog itself consists of an exchange of utterances between the system and the user in order to communicate the preferences and the recommendation, as well as general communicative statements. Note that the dialogs are collected using automatic speech recognition, which influences the quality of the transcriptions as well as the level of formality of the utterances. An example dialog is as follows:  
  
Task 03888: You want to find an expensive restaurant and it should be in the south part of town. Don't go for the first venue the system offers you, ask if there is anything else. Make sure you get the address, phone number, and postcode.  
  
system: Hello , welcome to the Cambridge restaurant system? You can ask for restaurants by area , price range or food type . How may I help you?  
user:   I'm looking for an expensive restaurant in the south part of town  
system: What kind of food would you like?  
user:   I don't care  
system: frankie and bennys is a nice place in the south of town and the prices are expensive  
user:   is there anything else  
system: the good luck chinese food takeaway is a nice place in the south of town and the prices are expensive  
user:   can I have the phone number  
system: The phone number of the good luck chinese food takeaway is 01223 244149 .  
user:   address  
system: Sure , the good luck chinese food takeaway is on 82 Cherry Hinton Road Cherry Hinton  
user:   post code  
system: The post code of the good luck chinese food takeaway is C.B 1, 7 A.G  
user:   thank you good bye

In the first part of the project you will implement a machine learning classifier for classifying **dialog acts** in dialog data. Dialog acts play an important role in conversations: utterances can be considered actions, for example greeting, asking, confirming, and so on. There are 15 dialog acts in the dataset, listed in the following table:

|  |  |  |
| --- | --- | --- |
| **dialog act** | **description** | **example sentence** |
| ack | acknowledgment | *okay um* |
| affirm | positive confirmation | *yes right* |
| bye | greeting at the end of the dialog | *see you good bye* |
| confirm | check if given information confirms to query | *is it in the center of town* |
| deny | reject system suggestion | *i dont want vietnamese food* |
| hello | greeting at the start of the dialog | *hi i want a restaurant* |
| inform | state a preference or other information | *im looking for a restaurant that serves seafood* |
| negate | negation | *no in any area* |
| null | noise or utterance without content | *cough* |
| repeat | ask for repetition | *can you repeat that* |
| reqalts | request alternative suggestions | *how about korean food* |
| reqmore | request more suggestions | *more* |
| request | ask for information | *what is the post code* |
| restart | attempt to restart the dialog | *okay start over* |
| thankyou | express thanks | *thank you good bye* |

The dataset is available in the format  dialog\_act [space] utterance\_content as attachment to this assignment description. Split the full dataset in a training part of 85% and a test part of 15%. Note that this dataset contains a simplification compared to the original data. In case an utterance was labeled with two different dialog acts, only the first dialog act is used as a label. When performing error analysis (see below) this is a possible aspect to take into account. Convert the data to lower case for training and testing, this will increase the accuracy of the classifier. Note that this implies that you also need to convert all user input to lower case, both when manually entering test cases in this part of the assignment and, importantly, during user input for the dialog system in Part 1b.  
  
**Baseline systems**  
Implement two baselines:

1. A baseline system that, regardless of the content of the utterance, always assigns the majority class of in the data. In the current dataset this is the inform label (almost 40% of all utterances).
2. A baseline rule-based system based on keyword matching. An example rule could be: anytime an utterance contains ‘goodbye’, it would be classified with the dialog act bye. This baseline can be made iteratively, create an initial version, test the performance, and then add or remove keywords for specific classes to improve the results. A reasonable performance for this baseline is at least 0.80.

In both cases, think about the data you’re working with to develop your systems (i.e. think about making sure you’re not (accidentally) ‘training’ on your test data). Your code should offer a prompt to enter a new utterance and classify this utterance, and repeat the prompt until the user exits.  
  
**Machine learning**  
  
Train a minimum of two different machine learning classifiers on the dialog act data. Possible classifiers include Decision Trees, Logistic Regression, or a Feed Forward neural network.  
  
You can use the keras or scikit-learn library for Python to implement the machine learning models. Take a look at online documentation (e.g.,  <https://keras.io/getting-started/sequential-model-guide/> and <https://scikit-learn.org/stable/user_guide.html>). Use a bag of words representation as input for your classifier. Depending on the classifier that you use and the setup of your machine learning pipeline you may need to keep an integer (for example 0) for out-of-vocabulary words, i.e., when a test sentence is entered that contains a word which was not in the training data, and therefore the word is not in the mapping, assign the special integer. After training, testing, and reporting performance, the program should offer a prompt to enter a new sentence and classify this sentence, and repeat the prompt until the user exits.

You may notice that many utterances in the dialogs are not unique, i.e., the exact same sentence is spoken by users in different dialogs. This influences machine learning, because even with a train-test split the same sentence may appear in both the train and test set. For this assignment, create a second dataset with duplicates removed (i.e., if a sentence is said more than once, only keep one occurrence), and create a second train-test split after removing the duplicates. Build and evaluate two different variants of each model, one with the original data and split, one with the deduplicated data and split. Discuss the differences in performance between each pair of variant models in the report.  
  
**Evaluation**  
Carry out an evaluation of the systems, do at least the following:

* Quantitative evaluation: Evaluate your system based on one or more evaluation metrics. Choose and motivate which metrics you use.
* Error analysis: Are there specific dialog acts that are more difficult to classify? Are there particular utterances that are hard to classify (for all systems)? And why?
* Difficult cases: Come up with two types of ‘difficult instances’, for example utterances that are not fluent (e.g. due to speech recognition issues) or the presence of negation (I don’t want an expensive restaurant). For each case, create test instances and evaluate how your systems perform on these cases.
* System comparison: How do the systems compare against the baselines, and against each other? What is the influence of deduplication? Which one would you choose for your dialog system?

**Deliverables**

* Python code that implements a majority class baseline and a keyword matching baseline
* Python code that implements two or more machine learning classifiers, each in two variants with and without data deduplication