MAIR Part 1: System Implementation

EFRAIM DAHL, Studentnumber
JUUL PETIT, 6446981
MATTHEW SCHEERES, 4545966
CHENGHONG YANG, 2274035
RUBEN VAN DER ZEIJDEN, studentnumber

1 MODEL EVALUATION

This section contains analyses and comparisons on the used Machine Learning models. We implemented the following systems; Decision tree model, K-nearest neighbor model and Ridge regression model. We will evaluate the systems based on a quantitative analysis and an error analysis, we will discuss difficult cases and we will compare the systems to each other as well as to the baseline models.

1.1 Quantitative Analysis

Quantitative evaluation: Evaluate your system based on one or more evaluation metrics. Choose and motivate which metrics you use.

To perform the quantitative analysis, the following metrics were chosen: precision, recall, f1 score and accuracy. The *weighted?* means of precision, recall, F1-score and accuracy from the 15 labels are presented in this report. The formulas of the means of these metrics are defined below:

$$Precision = \frac{1}{15}\sum_i \frac{TP_i}{TP_i + FP_i},$$

$$Recall = \frac{1}{15}\sum_i \frac{TP_i}{TP_i + FN_i},$$

$$F1\ score = \frac{1}{15}\sum_i \frac{2TP_i}{2TP_i + FP_i + FN_i},$$

$$Accuracy = \frac{1}{15}\sum_i \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i}.$$

In the above formulas, TP, FP, TN and FN respectively represent true positives, false positives, true negatives and false negatives. $i = 1, 2, ..., 15$ represent the ith label of our system.

The models were evaluated both on the complete training data set and the deduplicated training data set. The experimental results of the quantitative evaluation on the complete data set are listed in Table 1.

Table 1. Results of quantitative evaluation on the complete dataset for the baseline models, decision tree model, K-nearest neighbors model and ridge regression model. *weighted average?!?*

Authors' addresses: Efraim Dahl, Studentnumber; Juul Petit, 6446981; Matthew Scheeres, 4545966; ChengHong Yang, 2274035; Ruben van der Zeijden, studentnumber.

Metric	Majority class baseline	Keywords baseline	Decision Tree	KNN	Ridge
Precision	cell5	cell6	0.87	0.87	0.81
Recall	cell8	cell9	0.88	0.85	0.87
F1 score	cell8	cell9	0.88	0.86	0.83
Accuracy	cell8	cell9	*0.88 ?*	*0.85 ?*	*0.87 ?*

We can see that the Decision Tree model scores highest on all four metrics. *I don't know if we have to mention this here or discuss it in the section 'System comparison'.*
Table 2 shows the results of the quantitative evaluation on the deduplicated data set.

Table 2. Results of quantitative evaluation on the deduplicated dataset for the baseline models, decision tree model, K-nearest neighbors model and ridge regression model. *weighted average?!?*

Metric	Majority class baseline	Keywords baseline	Decision Tree	KNN	Ridge
Precision	cell5	cell6	0.75	0.71	0.73
Recall	cell8	cell9	0.76	0.73	0.79
F1 score	cell8	cell9	0.75	0.71	0.73
Accuracy	cell8	cell9	*0.76 ?*	*0.73 ?*	*0.79 ?*

We can see that the models perform better on the complete dataset than on the deduplicated one. *I don't know if we have to mention this here or discuss it in the section 'System comparison'.*

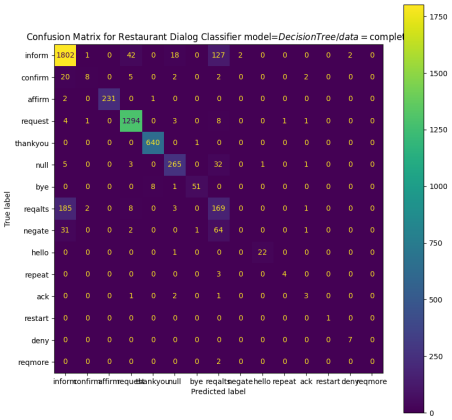
1.2 Error Analysis

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?

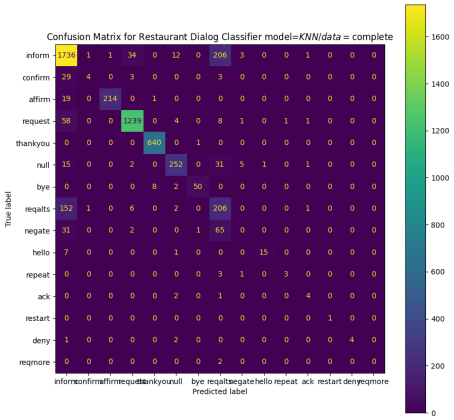
For the error analysis we will discuss two dialog acts that are more difficult to classify and why. Additionally, we will review whether there are particular utterances that are hard to classify for all models and why.

The classification report clearly shows that the results for the 'request' and 'repeat' dialog acts on all metrics is almost always below 0.50. Thus, these dialog acts are more difficult to classify than other dialog acts that often score 0.70 or higher on all metrics. The reason for this difficulty could be the following. Utterances such as 'can you repeat the postal code please?' or 'could you repeat the phone number?' should be classified as request, but instead are classified as repeat. This results in wrong classifications, leading to a low result on the precision metric for the repeat label and a low result on the recall metric for the request label. *Maybe we should delete this part since this is not derivable from the confusion matrices.*

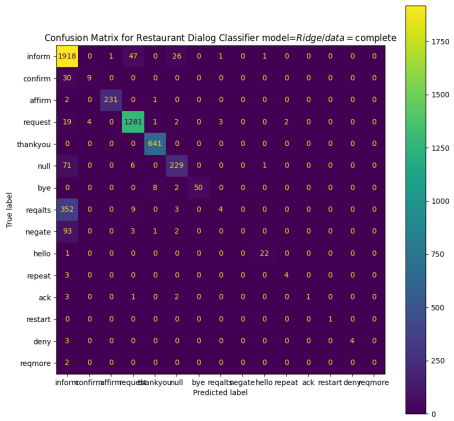
In addition, a lot of utterances that should be labeled as 'reqalts' are labeled as 'inform'. Figure 1, 2 and 3 show the Confusion Matrices on the complete dataset for the Decision tree model, the K-nearest neighbors model and the Ridge regression model, respectively. All figures show a high number of 'inform' as predicted label while the utterance should have been labeled as dialog act 'reqalts'. This error occurs, among others, at the following utterances: 'no id rather find a moderately priced restaurant' or 'no is there anything else'. Both are labeled with dialog act 'inform' while they should actually be labeled with dialog act 'reqalts'. The error also occurs the other way around for the Decision tree



(a) Fig. 1 Confusion Matrix for Decision Tree model on complete dataset



(b) Fig. 2 Confusion Matrix for K-nearest neighbors model on complete dataset



(c) Fig. 3 Confusion Matrix for Ridge regression model on complete dataset

model and the K-nearest neighbors model. We can see in figure 1 and 2 that there are a lot of utterances labeled with 'reqalts' while they should be labeled with dialog act 'inform'.

In de confusion matrices above, we can also see that utterances are almost never classified as 'negate'. Often, dialog acts that should be labeled 'negate', are labeled as 'inform'. Utterances as 'no i want spanish food' or 'no moderately priced' are labeled as dialog act 'inform' while they should be labeled 'negate'. Thus, this dialog act is difficult to classify.

Furthermore, there are some labels that have few samples in the training data. The dialog acts 'repeat', 'ack', 'restart', 'deny' and 'reqmore' for example, do not have more than seven samples in the training data. This automatically complicates classifying these dialog acts correctly.

1.3 Difficult Cases

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.

Concerning the difficult cases, five sentences were defined:

(1) 'no i dont want vietnamese food i want italian food'
Predicted label: deny/inform

(2) 'luck the missing sock'
Predicted label: inform

(3) 'thank you and bye'
Predicted label: thankyou

(4) 'street'
Predicted label: inform

(5) 'greec'
Predicted label: inform

We will evaluate the difficult cases that are listed above one by one.

First, there is the case where there is a negation followed by an informative utterance. Some of our models classify this utterance as the dialog act 'deny', since it would make more sense to classify it as 'inform'. The keyword baseline model classifies this utterance as 'deny' on both the complete dataset as the deduplicated dataset. The decision tree model and the K-nearest neighbors model classifies this assertion as 'deny' only on the deduplicated set. The other models classify it 'correctly' as dialog act 'inform'.

Second, our systems need to handle speech recognition issues such as the assertion: 'luck the missing sock'. Obviously, the keyword baseline model classified this utterance as 'null' since it did not recognice any keywords. Remarkable is that the decision tree model trained on the deduplicated dataset also returns 'null' as label and de decision tree model trained on the complete dataset returns 'reqalts'. All other models return the dialog act 'inform' on this assertion.

Third, there is the case in which a sentence could be labeled as two different dialog acts. For example, the utterance 'thank you and bye', coudl be labeled as both 'thankyou' and 'bye'. All our models, except of the majority class baseline which labels it as 'inform', classified this assertion with 'thankyou'. You may ask yourself why 'thankyou' is given priority as a label to 'bye'.

Fourth, some assertions are simply words instead of full sentences. When the model is asked to classify the utterance 'street', most of the models return the label 'inform', while it is quite obvious for a human being that the user is making a request. This is probably the case because the word 'street' does not once occur in the training dataset. If we, for example, feed the model the utterance 'address', it does classify it correctly as 'request'. Still, a perfect model should recognize this assertion and classify it as 'request'.

Last, the test data could consist spelling mistakes. There are some spelling mistakes contained in the training dataset, such as 'venesian' instead of 'venetian' and 'tha' instead of 'thai'. The model should be able to recognize the mistake and label the assertion correctly. The utterance 'greec' was labeled by our model as follows: 'null' by the keywords baseline model trained on both datasets, 'reqalts' by the decision tree model trained on the complete dataset and 'inform' by all the other models. Thus, our model is recognizing this spelling mistake pretty well.

1.4 System Comparison

To view the results of different systems, run the main.py file of the accompanied code, and view the confusion matrices (saved as .png files), classification reports and results of the difficult instances (saved as .txt files) inside the results directory.

When comparing the models' confusion matrices visually, it is rather clear that the chosen models perform much better than the baselines. After deduplication this is still the case, but it is immediately visible that the amount of samples in each category reduces drastically, with the total amount of samples dropping from 20400 and 4534 to 5101 and 1589 in the stratified training and test sets, respectively.

When comparing accuracy scores between the models, it can be seen that the decision tree classifier outperforms the other models on both the regular and deduplicated test sets, with KNN coming in second for both datasets. Logically, it would make sense to use the decision tree classifier for further development of the dialog system. Consequently, it might also be interesting to explore similar methods, like a bagging or random forest classifier.

1.5 Questions

1. There is a vague difference between Error Analysis and Difficult Cases. What exactly should we discuss in the two sections?
2. Section one, Quantitative Evalutation, just gives information about metrics and the results. We do not take up any evaluation of the results in this section as we do this in section four, System Comparison. Is this the right way to do it?
3. For the difficult cases, did we correctly understand what was meant from this section?
4. Should we add the results of the sklearn classification reports in the report?

REFERENCES