Project 1b: Text Classification

Emiel Scheffer, Filippo Maria Libardi, Igor Kuczuk Modenezi, Jacqueline Isabel Wagner

September 25, 2019

1 Quantitative Evaluation

Evaluation on test set containing 223 samples:

	baseline classifiers		
	keyword classifier	random classifier	machine learning classifier
accuracy	80.2%	22.8%	98.5
top 3 categories	ack (100%) deny (100%) negate (100%) restart (100%) thank you (100%) bye(100%)	inform (39.2%) request (24.9%) thankyou (16.3%)	restart (100%) thankyou (99.8%) request (99.5%)
worst 3 categories	negate (39.3%) confirm (69.2%) affirm (73.7%)	hello (0%) ack (0%) repeat (0%) reqmore (0%) restart (0%)	reqmore (0%) ack (50%) hello (72%)
average precision	72.1%	7.7%	92% - 99%
average recall	82.3%	6.4%	84% - 99%

While it is undoubtedly interesting to know how many cases have been classified correctly overall, employing diverse metrics sheds light on the systems true performance. We decided to calculate the average precision, as well as the average recall, to answer some very relevant questions:

1. What proportion of assigned categories was actually correct? (precision)
2. What proportion of ground truth categories was identified correctly? (recall)

We obtained answers for these questions by (1) calculating the precision and recall for each category and (2) averaging over all possible categories.

In addition, we chose to highlight the three best performing categories, along with the top three underperforming categories. We obtained the three best categories by analyzing the recall for each category. Using this information we were able to quickly identify commonly incorrectly classified sentence-patterns to improve our systems.

In the above table average precision and recall of the ML classifier states respectively the micro average and macro average. While macro average computes the metric independently for each single class and then evaluates the average, the micro average will add together each class contribution and then take the average.

In our case is preferable to value more the micro average as the occurrence of each class is not evenly distributed among the test set (e.g. there are 2048 instances of "inform" and 0 of "reqmore").

Additionally the evaluation of the ML classifier has been a little more challenging. This is due to the numerous different possible ways of classifying text. The cautious evaluation has also been based on the following parameters:

1. Classifier prediction time
2. Classifier fitting time
3. Logarithmic loss
4. Accuracy
5. Labels pre-processing

Surely point 1 and 2 are hardware-dependent, but nevertheless they make a good performance metric. Comparison in the following table is between a Random Forest Classifier (the one we ended up using) with 500 trees/estimators, a Feed Forward Neural Network Classifier and a Logistic Regression Classifier.

Classifier	RFC	MLPC	LRC
Prediction Time	191.8	419.5	98.6
Fitting Time	4.9	6.7	4.5
Accuracy	97.9	99.7	93.4
Log loss	0.48	55.4	1.1

Table 1: comparison of different classifiers

As shown in the above table the difference between the classifiers is not very large. The real reason why we decided to opt for the RFC is its decent prediction time, good accuracy and discrete logarithmic loss.

The latter has played a fundamental role in deciding which classifier to use. Log loss heavily penalises classifiers that are confidently incorrect about a classification. Linear regression classifier had a very good time performance but scored a very bad log loss, reason why it has been discarded.

During the various approaches we also tried different encoding and preprocessing of our labels. We firstly excluded every stop word in the data set, but this eventually led to a far worst performance (accuracy around 90) and when the classifier was asked to classify a word in run-time, it would most probably classify it as "inform" (even mi-spelled words).

After removing the stop-words exclusion we decided to stem all the labels (reducing inflected words to their word stem), this did not significantly improve any of the metrics, but it did return a smaller data set.

2 Error Analysis

Following the quantitative analysis, we take a more in-depth look at the mistakes made by each of the three classifiers. However, since the random classifier randomly assigns categories, there are no sentences which are especially difficult to detect. Therefore we refrain from listing any examples for this specific classifier and instead focus on the keyword classifier in subsection 2.1 and the machine learning classifier in subsection 2.2.

2.1 Keyword Classifier

During our evaluation under-performance was mostly attributed to three different categories: ambiguous categories, overlapping keywords and extremely short messages.

For some instances the ground truth categorization, while correct, could easily be extended to include additional categories. For clarity, two examples are listed in Table 2.

sentence	'what about international food'	'okay thank you good bye'
ground truth	reqalts	bye, thankyou
classification result	reqalts, inform, request	ack, bye, thankyou

Table 2: difficulties related to ambiguous categories

While 'okay thank you good bye' was correctly classified as corresponding to the categories 'thankyou' and 'bye', it additionally retrieves the category 'ack'. Considering that the sentence contains the phrase 'okay', this categorization seems plausible.

Furthermore, using certain keywords was necessary to correctly classify large amounts of data belonging to a given category. However, in some instances, these keywords are used in a different context. Hence, resulting in a wrong classification when employing a keyword classifier. An example of this is given in Table 3.

Lastly, to prevent overfitting, certain very specific keywords could not be assigned to any category. Thus, extremely short messages only containing highly specific keywords are not classified correctly. Underlining examples are given in Table 4.

sentence	’no i said Irish’
ground truth	negate
classification result	inform, negate

Table 3: difficulties related to overlapping keywords

sentence	’any’	’phone’	’center’
ground truth	inform	request	inform
classification result	null	null	null

Table 4: difficulties stemming from extremely short messages

2.2 Machine Learning Classifier

The Machine Learning classifier has different error types than the keyword classifier. A relevant problem with the classification of utterances by our model is that it often doesn’t recognise words belonging to numerous different categories. Therefore if a word is present in our dataset but is classified differently almost every time, then the model will not know what to do with it and probably assign the whole sentence to null. For instance a sentence composed by three words from an inform utterance and three words from a request one, will most likely result in null. An example follows:

Table 8.

sentence	’phone number’	’italian place’	’i want the number of the italian place’
ground truth	request	inform	request
classification result	request	request	null

Table 5: difficulties related to ambiguous categories

One more significant issue the system encounters is that every word that is not in our data set at all will be recognised as null. This is not a deficiency by itself but it becomes one as the model predicts the whole sentence to be null. This is very significant especially because some very important words are missing in the data set. For instance a sentence like "can you tell me more" will be identified as null because tell is not in the knowledge base of the model. While "please say more" is correctly identified as reqmore.

Table 6.

sentence	’whats the price range again’	’again please’	’start again’	”
ground truth	request	repeat	restart	
classification result	request	request	null	

Table 6: problem related to multi-classified words

3 Difficult Cases

The cases that are usually more difficult for the system to detect are utterances that never appear in the training data, such as random noises detected by the microphone or the user saying something that shouldn’t be interpreted by the system (such as a side conversation). This can be seen in the example below:

	random classifier	keyword classifier	machine learning classifier
"afk"	inform	null	null
"noise"	request	null	affirm

Table 7: unusual utterances and their results

The random classifier presented the worst result, as expected, giving a different classification every time, having predominantly classified as "inform" and "request". What should be noted, however, is the performance of the machine learning classifier. Although "noise" appears in the training data, it mistakenly classify as "affirm".

Another difficult case is when the utterance has multiple possible classifiers, such as including information when requesting an address.Some examples can be seen below:

	random classifier	keyword classifier	machine learning classifier
"what is chinese restaurant address"	inform	inform	request
"how about mexican telephone start over"	inform	inform	null

Table 8: utterances with multiple categories and their results

With this difficult cases, the machine learning classifier fared better than the others, successfully recognizing the first one as a request. It should also be noted that the "null" classification is logical, once it could be classified in four different categories ("reqalts", "inform", "request" and "restart").

What can be concluded through the analysis of this cases is that the machine learning classifier can handle both sentences and words with high accuracy (although it may present one mistake). The keyword classifier can handle separate words and categories well, but has a lower accuracy when presented with sentences with multiple possible classifiers. Finally, the random classifier had the worst result, having a different result for the same input. This was expected, once the system is random and the classifiers that had the most occurrences where also the ones that were more frequent in the results.

4 System Comparison