UNIVERSITY OF GOTHENBURG

Lab IV: NLU for dialogue systems

Student name: *Dominik Künkele*

Course: *Dialogue Systems (LT2216)*
Due date: *March 4th, 2022*

Results

Task 1: Evaluate NLU exploratively. Figure 1a shows the result for giving *Rasa* a message, it has been trained on. As can be seen, the evaluated intent is correct. The confidence is under 50%, but lies arround 18% over the confidence for the next likely intent. For the untrained message, *Rasa* still identifies the correct intent (Figure 1b), but the confidence is much lower and closer to another intent. The margin is now only arround 5%.


```
Next message:
clean the room
{
  "text": "clean the room",
  "intent": {
    "name": "vacuum",
    "confidence": 0.47829106172685293
  },
  "entities": [],
  "intent_ranking": [
    {
      "name": "vacuum",
      "confidence": 0.47829106172685293
    },
    {
      "name": "turn_on_light",
      "confidence": 0.19977465697814703
    },
    {
      "name": "turn_off_light",
      "confidence": 0.16412802649895453
    },
    {
      "name": "move_to_trash",
      "confidence": 0.10795068861475329
    },
    {
      "name": "give",
      "confidence": 0.04985556618129193
    }
  ]
}
```


(a) trained utterance


```
tidy up
{
  "text": "tidy up",
  "intent": {
    "name": "vacuum",
    "confidence": 0.2820786228445965
  },
  "entities": [],
  "intent_ranking": [
    {
      "name": "vacuum",
      "confidence": 0.2820786228445965
    },
    {
      "name": "move_to_trash",
      "confidence": 0.23179380294033725
    },
    {
      "name": "turn_off_light",
      "confidence": 0.21123318846835176
    },
    {
      "name": "turn_on_light",
      "confidence": 0.16369256056608475
    },
    {
      "name": "give",
      "confidence": 0.1112018251806298
    }
  ]
}
```


(b) untrained utterance

Figure 1: utterances for specified intents

Looking at Figures 2a and 2b, it can be seen that *Rasa* has problems, when fed with a message, which doesn't fit a specified intent. It just associates the utterance with one of the specified intents, even though it doesn't fit at all. This can happen also with a quiet high confidence as in Figure 2a.

```
call my friend
{
  "text": "call my friend",
  "intent": {
    "name": "give",
    "confidence": 0.54550954053458
  },
  "entities": [],
  "intent_ranking": [
    {
      "name": "give",
      "confidence": 0.54550954053458
    },
    {
      "name": "move_to_trash",
      "confidence": 0.1338784465345795
    },
    {
      "name": "turn_off_light",
      "confidence": 0.12975370340991346
    },
    {
      "name": "turn_on_light",
      "confidence": 0.09572336879530852
    },
    {
      "name": "vacuum",
      "confidence": 0.0951349407256186
    }
  ]
}
```


(a) unspecified intent 1


```
book the hotel
{
  "text": "book the hotel",
  "intent": {
    "name": "move_to_trash",
    "confidence": 0.33371844245830945
  },
  "entities": [],
  "intent_ranking": [
    {
      "name": "move_to_trash",
      "confidence": 0.33371844245830945
    },
    {
      "name": "turn_on_light",
      "confidence": 0.22061821321908262
    },
    {
      "name": "turn_off_light",
      "confidence": 0.18855813815397005
    },
    {
      "name": "vacuum",
      "confidence": 0.1704547487084861
    },
    {
      "name": "give",
      "confidence": 0.08665045746015206
    }
  ]
}
```


(b) unspecified intent 2

Figure 2: utterances for unspecified intents

Task 2: Add an intent. For formatting reasons, I ommited the existing intents:

```
nlu:
...

- intent: cook
  examples: |
        - make food
        - cook the meal
        - prepare something to eat
        - grill the steak
        - warm up the leftovers
```

Listing 1: additional intent 'cook'

Task 3: Evaluate NLU systematically.

without cross-validation. Precision: 1.0, Recall: 1.0

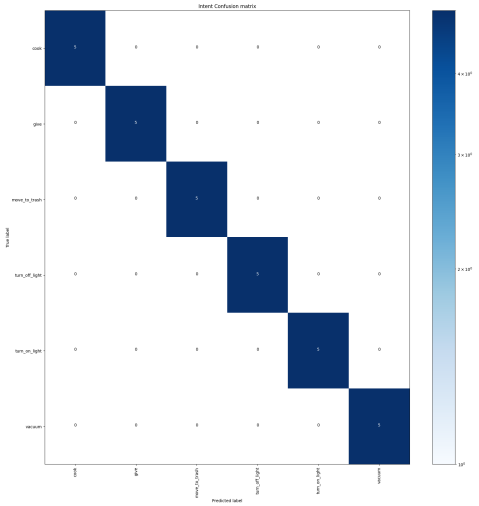


Figure 3: confusion matrix without cross validation

In the confusion matrix without the cross validation can be seen that all test utterances were classified correctly. Since the test data is the same as the training data, this is not very surprising and meaningful. The test does not show, how the system can generalize the training and use it on unseen utterances. Therefore, there should always be used a different set of test data to evaluate the systems performance.

with cross-validation. Precision: 0.668, Recall: 0.633

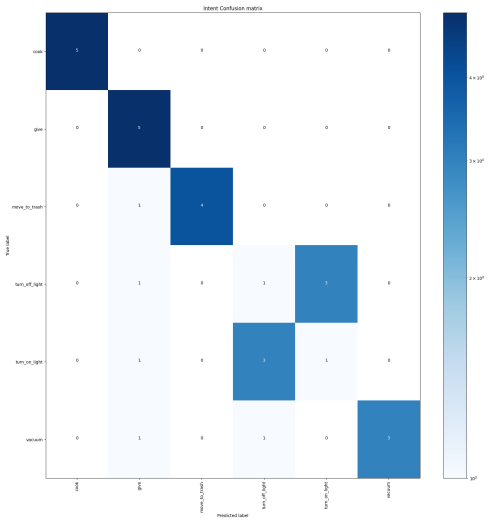
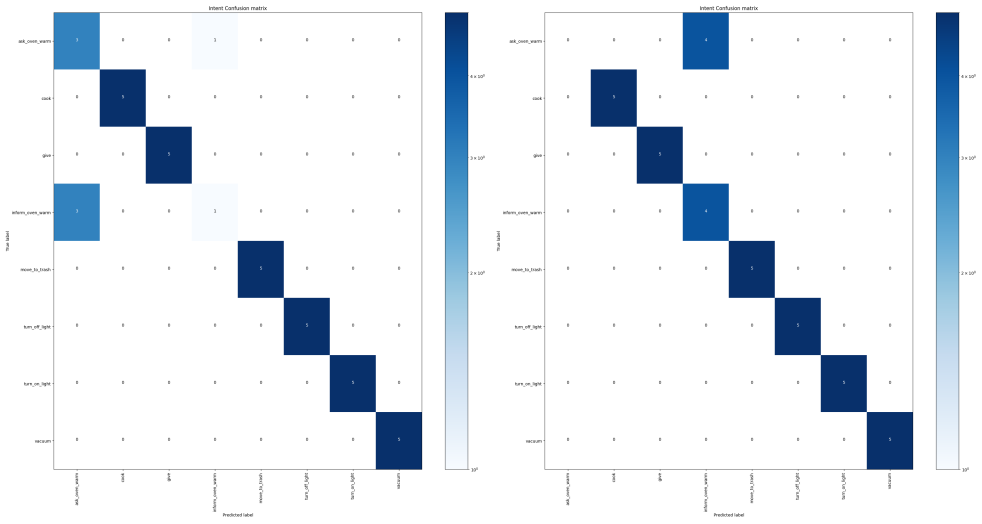


Figure 4: confusion matrix with cross validation

With cross validation, the system performs worse on the first look. But now, the system is tested on unknown utterances. This has a much higher significance than using training data as test data, since it also shows the generalization. A bigger test/training data set would help to increase the performance.

Task 4: Overcome a limitation. The system confuses the *ask* and *inform* intents for the stove. When using *mean pooling*, almost all stove related utterances are assigned to the *ask* intent. Changing to *max* pooling does the opposite and assigns all utterances to the *inform* intent.

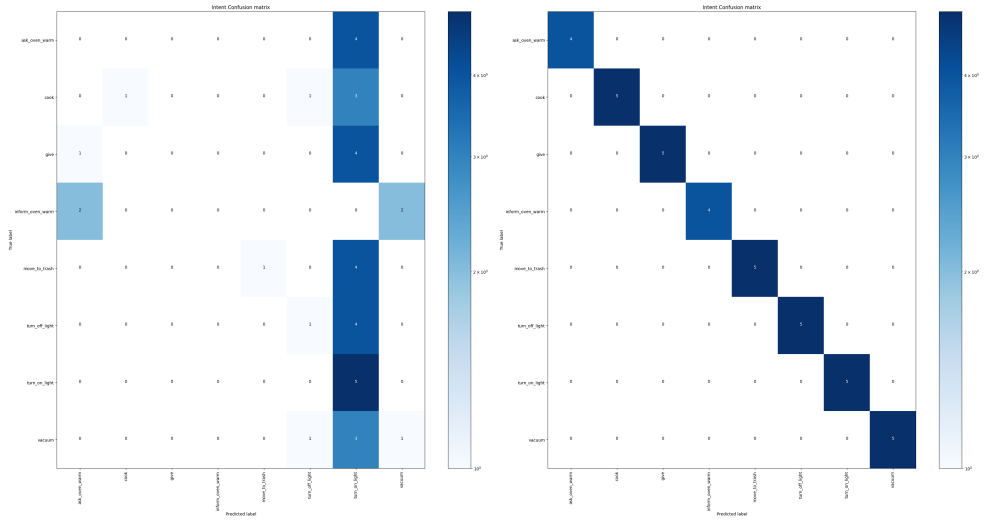


(a) mean pooling (b) max pooling

Figure 5: confusion matrix with stove intents

The problem hear is probably that the example utterances for both stove related intents are using the same words, only have a different order. The order does not play a role in neither the *max pooling* nor the *mean pooling*. As a consequence, the example sentences appear the same for both intents and are then assigned to only one intent.

Using the *DIET classifier* with 10 epochs leads to many classification errors (Figure 6a). These appear across all intents. The only intents that get classified quiet well is *turn_on_light* in this run. (Rerunning the training changes the result completely). Increasing the number of epochs could decrease the confusions a lot. For me, it took arround 56 epochs, too loose all confusions (also here did rerunning the training change the required number of epochs).



(a) 10 epochs (b) 56 epochs

Figure 6: confusion matrices with DIETClassifier

One central part of the pipeline is probably the *LanguageModelFeaturizer*. This is converting the utterance into a vector using the pre-trained model *BERT*. *BERT* is taking the context of each word into account, to represent it in a vector. The order of the words in the utterance are therefore also encoded. The following *CountVectorsFeaturizer* and *DIETclassifier* can then classify the utterances based on the "context-sensitive" vectors produced by *BERT*.

Task 5: Pinpoint confidence threshold. For the selection of a threshold value, multiple considerations should be taken into account. One important factor is the importance of the precision. Does a wrong classification have big consequences? This would apply for example in the medical area, where each classification error should be avoided. Setting a high threshold would reduce the overall clarification errors (and implicitly increase the precision) by asking for approval. In our case with a assisting robot, the consequences may not be as crtitical and the user experience has a higher value (less approval questions). Therefore, the threshold value can be lower.

Furthermore, we can take the performance of the system into consideration. For this, I tested the system with cross-validation (see Figure 7). The macro average precision was 0.639 and the macro average recall was 0.633. Looking at the confusion matrix reveals that espacially the intent *turn_off_light* is confused. (A reason for this may alos be the very small data set). With such a performance, the threshold should also be lower.

In a next step, I experimented in the shell with different utterances to see, which confidence scores are usually reached. Even though, this wes not really "scientific", it helped to get a feeling for the system.

Given these considerations and experimenting with different utterances in the shell, I would

propose a threshold value of 0.35.

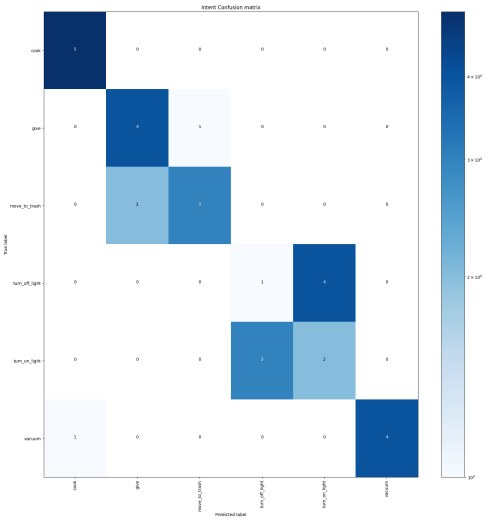


Figure 7: confusion matrix with DIET classifier and cross validation

further considerations. An option, to tackle different precisions for each intent may also be to use different threshold values for each intent. Namely here, a low higher threshold for the intent *turn_off_light* but a low threshold for the intents *cook* or *vacuum*.

Furthermore, one could not only look at the confidence of the highest value, but also at the difference to second highest confidence. This may be hint if the system is undecided between multiple options.