# Summary

# ROC Results



All the ROC.v2\_05 results are zero.

## Explanation:

The ROC metrics are aim to evaluate the first correct hypothesis scores.

The basic idea is to evaluate the first correct hypothesis under different thresholds.

For example, for a hypothesis with confidence score , under a threshold , we define TA (true-accepts), FA(false-accepts), FR(false-rejects) and TR(true-reject) as follows:

* TA = 1 if s >= t and H = True (if the hypothesis is correct and the score is not less than the threshold )
* FA = 1 if s >=t and H = False
* TR = 1 if s <t and H = False
* FR = 1 if s <t and H = True

Therefore, under different thresholds, they can have different number of TA, FA, TR, FR (just like confusion matrix).

And, the ROC metrics are defined:

N is the total number and t is the threshold. #TA(t) means the number of TA under the threshold.

As similar,

However, depends on the accuracy. The maximal is the accuracy when t=0. To counter this problem, another metric is defined:

Now, the maximal is 1 when t = 0, which is no longer depended on accuracy.

Come to my model, since it always has only predict with score 1.0, then changing the threshold doesn’t change the metrics.

At the same, time, is always bigger than 0.05, that’s why is always 0.

## Tried to fix it but it doesn’t work

Instead of always predicting with score 1.0, I give a score with ratio. Here are two ways I tried:

1. Give the score with the SLU confidence score
2. Give the score with the average of SLU confidence score and the 3way prediction score

However, neither of them works.

An example of number of TA, FA, TR, FR is shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| TA | FA | TR | FR |
| 9898 | 85 | 8 | 94 |

So, can never less than 0.05 under this method.

## Another thing to try:

Now, the prediction score is always 1 because of the voting method. For svm, the prediction score is 0.667, 0.333, 0, which doesn’t differentiate the prediction and the rest very much.

I can combine ASR score, SLU score, Prediction score with a linear combination and try to search the best weights when training.

# 2-way model

## No annotation for correct SLU

The issue to apply 3wayModel for this problem is that there is no annotation to mark which SLU is correct. They only annotated the slots that they are interested and only the accumulated results. (By “accumulated” I mean, the current slots depend on pervious turns. For example, if they use said nothing, the slots don’t change.)

There are three slots that annotated in this challenge:

1. “goals” (is the “joint” of all the four slots.) (what specific the user wants: area, food, name, pricerange).For example, "pricerange"= "moderate"
2. “requested” (what the user want the system to response). For example, "requested"=”phone, address”
3. “method” (In what way the user asks the system). For example, “method”= "byconstraints"

## Only certain Dialog Acts matter

They only annotated the interested slots and ignored all other stuff. At the same time, only a certain types of SLU affect the interested slots. For example,

* “goals” depends on only on “inform” act in the user turn.
* “requested” depends only on “request” in the user and “inform” in the system
* “method” depends only on “bye”, “reqalts”, “inform” in the user and “inform” in the system

Thus, theoretically, only the correctness of SLU that contains these dialog acts can be inferred from the annotation. On the other hand, only this SLU matters.

## 3 hypotheses to infer correct SLU

Although it is difficult to infer which SLU is correct, we can still work on it. Here are three hypotheses to infer which SLU is correct

* H1: assume that the SLU is correct if and only if all the SLU hyps (slot,value) appear in the correct answer
  + assume NONE is -1
  + assume slu that doesn't matter is -1
* H2: assume that the SLU is correct if and only if it matches the differences between the pervious correct answer and the current one
* H3: assume that the SLU is correct if and only if it will turn into the correct answer based on the previous one and the new SLU

An example:

Turn 1:

Computer: “yippee noodle bar is a great restaurant serving asian oriental food and it is in the moderate price range”

User: “is there anything else”

*Annotation: request = []*

*Top SLU: []*

Turn 2:

Computer: “Can I help you with anything else?”

User: “could i have the address and phone number”

*Annotation: request = phone;addr*

*Top SLU: request(slot=addr)*

Consider the correctness of SLU in the second turn. Under H1, it is correct. Because, “*slot=addr*” appears in the annotation; Under H2, it is wrong, because, the difference between Turn 2 and Turn 1 is “*request = phone;addr*” but the SLU only matches *addr*. Under H3, it is wrong, too. Because, given *request(slot=addr),* the result should be “request=addr” but it doesn’t match the annotation.

## Results under H1

## 2-way model

Only two labels are considered: {0, -1}

If the top SLU is correct, it is 0; else, it is -1

The result of this model is shown below:

* Model: SVM
* Trained on “train”
* Feature:
  + Res\_act (only the dialog acts)
  + Actngram (dialog acts + unigram)



### Combined Model:

Use the prediction rank to pick up SLU. If it is ‘0’, pick up the first one; if it is ‘-1’, ignore the SLU.

Models:

* baseline: given by the organizer (consider all SLU and pick up one with highest confidence score)
* baseline\_focus: given by the organizer
* HWUbaseline: the adaptive version for DSTC2 from DSTC1 from one of the previous group
* baselineTop1: always take the Top SLU
* 2waymodel\_act: (2way model with act features)
* 2waymodel\_actngram (2way model with act + unigram features)

Conclusion:

* Only Accuracy of “request” and “method” beat the other baseline.
* 2way model helps by comparing 2waymodel and the baselineTop1 (consider only the top SLU)

Highlighted one are the best among all the methods.



# Topline:

It is bad the 2way model doesn’t beat the HWUbaseline on the joint goals. I’d like to see way. Therefore, I compute the topline by considering different number of SLU.

# How details affect performance?

# TODO: