# Summary

## New Announcement for DSTC2

### *About ROC.v2*

“We are going to keep the roc.v2\_ca05 as a featured metric when reporting results. It seems the most fair not to change the featured metrics during the challenge. However we will be sure to explain in publication the fact that these numbers are only comparable across systems of similar accuracies, and that **the accuracy and l2 metrics are therefore more useful in the big picture**.”

### *About submission*

1. The test set for DSTC2 will be released unlabelled next Monday, the 20th of January as of 0:00h GMT on that day.
2. It is due by 0:00h GMT on Monday the 27th January
3. Each team of participants may submit up to **5 trackers** for evaluation.

## Are the goals conditional dependent (Wiz-of-oz)?

In the previous models, all the goals are assumed to be independent. Knowing the food type doesn’t help to determine the price range of the food.

However, it might not be the case. The goals might depend with each other.

To test the idea, for each goal, I add the gold-standard labels for the rest goals as features. For example, when prediction pricerange, I add the feature like “food=chinese”, “area=north”, “name=yes”.

### *Result:*

This model is called “2waymodel\_goals\_wizoz”. It is wiz-of-oz because we cannot know the gold-standard for other labels. It is used to test whether they are dependent with each other.

|  |  |  |  |
| --- | --- | --- | --- |
| method | test | Joint\_Goals\_accuracy | Joint\_Goals\_l2 |
| 2waymodel\_enrich\_more\_goals | dstc2\_train | 0.7113547 | 0.5772907 |
| 2waymodel\_enrich\_more\_goals | dstc2\_dev | 0.561001 | 0.8779979 |
|  |  |  |  |
| 2waymodel\_goals\_wizoz | dstc2\_train | 0.7167909 | 0.5664182 |
| 2waymodel\_goals\_wizoz | dstc2\_dev | 0.5669969 | 0.8660063 |

The results show that they do have some relationship with each other, however, the improvement is not that much (0.711->0.717).

## Is it Dialog Manager (DM) or Speech Recognizer (SR) depended?

The dialog manager will affect the turn transitions and the speech recognizer will after the performance of ASR. Thus, to test whether it helps when we know the DM and SR when prediction.

### *Statistic info about DM\*SR*

The 3 dialog managers (DM) are:

1. an MDP for tracking the dialog state, and a hand-crafted policy
2. a POMDP dialog state tracking, and a hand-crafted policy [1]
3. a POMDP a policy learnt using reinforcement learning

The 2 speech recognizers (SR) are:

1. GMM-HMM model with artificially degraded acoustic models
2. full GMM-HMM model optimized for the domain

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # of turns | |  | # of calls | |  | average call length | |
| dstc2\_train | DM0 | DM1 |  | DM0 | DM1 |  | DM0 | DM1 |
| SR0 | 2075 | 3591 |  | 302 | 435 |  | 6.87 | 8.26 |
| SR1 | 2865 | 3146 |  | 443 | 432 |  | 6.47 | 7.28 |
|  |  |  |  |  |  |  |  |  |
| dstc2\_dev | DM0 | DM1 |  | DM0 | DM1 |  | DM0 | DM1 |
| SR0 | 593 | 1220 |  | 81 | 140 |  | 7.32 | 8.71 |
| SR1 | 938 | 1183 |  | 141 | 144 |  | 6.65 | 8.22 |

### *Observations:*

* Different SR and DM do have an influence on the length of a dialog.
* The call length (# of turns in a call) is bigger when using SR1 than one with SR0. It is probably because that SR1 is better than SR0
* A Dialog with DM1 is longer than DM0. After taking a look at the data. I found, for POMDP, there are more repeating during the dialog. It is probably due to the fact that POMDP considers the probability. If the confidence score is lower than a threshold, it will ask the question again.

### *Results:*

To test the idea, I tried three combinations:

* Add Dialog Manager as a feature
* Add Speech Recognizer as a feature
* Add both as features

|  |  |  |  |
| --- | --- | --- | --- |
| method | test | Joint\_Goals\_accuracy | Joint\_Goals\_l2 |
| 2waymodel\_enrich\_more\_goals | dstc2\_train | 0.711 | 0.577 |
| 2waymodel\_enrich\_more\_goals | dstc2\_dev | 0.561 | 0.878 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_sr | dstc2\_train | 0.712 | 0.576 |
| 2waymodel\_goals\_enrich\_sr | dstc2\_dev | 0.561 | 0.877 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_dm | dstc2\_train | 0.713 | 0.573 |
| 2waymodel\_goals\_enrich\_dm | dstc2\_dev | 0.567 | 0.865 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_sr\_dm | dstc2\_train | 0.712 | 0.577 |
| 2waymodel\_goals\_enrich\_sr\_dm | dstc2\_dev | 0.562 | 0.877 |

### Observations:

* SR doesn’t change the performance
* DM has a bigger influence with the performance than SR. Unfortunately, we cannot use the DM as a feature because the test set has a different DM.
* However, neither of them makes a big difference.

## Sequence Labeling Model for Goal classifiers

In the previous models, I also didn’t consider the history of the dialog. A natural way for history modeling is to consider the problem as a sequence labeling task. In this model, a whole dialog is a data sample point, which in previous models, each turn is a single sample point.

Model: CRF

Features:

* Unigram (that appears at least twice; # of unigram is 381)
* All the Dialog acts (51)
* The previous label (1)

### *Results:*

|  |  |  |  |
| --- | --- | --- | --- |
| method | test | Joint\_Goals\_accuracy | Joint\_Goals\_l2 |
| 2waymodel\_enrich\_more\_goals | dstc2\_train | 0.711 | 0.577 |
| 2waymodel\_enrich\_more\_goals | dstc2\_dev | 0.561 | 0.878 |
|  |  |  |  |
| 2waymodel\_goals\_crf | dstc2\_train | 0.718 | 0.564 |
| 2waymodel\_goals\_crf | dstc2\_dev | 0.560 | 0.881 |

### Observations:

It doesn’t improve the performance.

## More Features

* Add bigrams
* Log ASR score (The intuition is that the given ASR scores are negative. It turns out it is the log version of the probability. Therefore, using log asr score might be better. For example, if the score is too slow, the classifier is not reliable.)

### *Results:*

Add the log ASR score improves a little bit.

However, adding bigrams decrease the performance.

|  |  |  |  |
| --- | --- | --- | --- |
| method | test | Joint\_Goals\_accuracy | Joint\_Goals\_l2 |
| 2waymodel\_enrich\_more\_goals | dstc2\_train | 0.711 | 0.577 |
| 2waymodel\_enrich\_more\_goals | dstc2\_dev | 0.561 | 0.878 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_logasr | dstc2\_train | 0.715 | 0.570 |
| 2waymodel\_goals\_enrich\_logasr | dstc2\_dev | 0.564 | 0.872 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_more\_bigram\_logasr | dstc2\_train | 0.661 | 0.679 |
| 2waymodel\_goals\_enrich\_more\_bigram\_logasr | dstc2\_dev | 0.484 | 1.032 |

## Error Analysis

It is disappointed for the results. So I decide to do a small-scale error analysis.

I picked 271 errors and organized the errors in two ways.

The first one is by type of methods, whether is due to ASR or the classification. Here is the error numbers.

* The most errors are “Pre” (it means it is wrong because it was wrong in a previous turn)
* The second most is “ASR”. If the ASR type means that the correct goal can never be found in all the ASRs, which causes a lot of errors.
* The classifiers are not bad. Most of the time, it is wrong because the ASR is wrong.



The second one is by the goals. The most type of error is “Food”. “Food type” usually appear in the beginning of a dialog. Thus, if it is wrong, it will make all the following turns wrong.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Food | Price | Area | Name | Others |
| type of goals | 170 | 52 | 42 | 0 | 7 |

## Leveraging ASR: combing N-Best

From the error analysis, I found the error is caused by ASR mostly. Therefore, I want to work on this direction in advance. Here are the methods I tried.

* Combine 10 ASRs with 1 sentence and Just use it as features
* Majority Voting: Classify each ASR and vote them

For the classifier, I also include the Speech recognizer ID as a feature. But I excluded the SLU as features because the SLU rank doesn’t match the ASR rank.

* Weighted Voting: each ASR has a confidence score, which can be used as a weight

### *Result:*

* 2waymodel\_goals\_enrich\_asrs (Use one sentence including all ASR, and binary indicator)

For each word, a binary feature is used to indicate whether it appears or not

* 2waymodel\_goals\_enrich\_asrs\_counts (Same with above, but use the number of word instead of a binary feature)
* nbestmodel\_goals\_nbest\_top1 (Training on all the N-Best, but only use the top 1 to predict)
* nbestmodel\_goals\_nbest\_majority (Majority Voting)
* nbestmodel\_goals\_nbest\_weighted (Weighted Voting)
* nbestmodel\_goals\_nbest\_voting\_top3 (Majority Voting with just the top 3)
* nbestmodel\_goals\_nbest\_voting\_top5 (Majority Voting with just the top 5)
* nbestmodel\_goals\_nbest\_voting\_top7 (Majority Voting with just the top 7)

|  |  |  |  |
| --- | --- | --- | --- |
| method | test | Joint\_Goals\_accuracy | Joint\_Goals\_l2 |
| 2waymodel\_enrich\_more\_goals | dstc2\_train | 0.711 | 0.577 |
| 2waymodel\_enrich\_more\_goals | dstc2\_dev | 0.561 | 0.878 |
|  |  |  |  |
| HWUbaseline | dstc2\_train | 0.732 | 0.451 |
| HWUbaseline | dstc2\_dev | 0.623 | 0.601 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_asrs | dstc2\_train | 0.737 | 0.525 |
| 2waymodel\_goals\_enrich\_asrs | dstc2\_dev | 0.544 | 0.911 |
|  |  |  |  |
| 2waymodel\_goals\_enrich\_asrs\_counts | dstc2\_train | 0.718 | 0.564 |
| 2waymodel\_goals\_enrich\_asrs\_counts | dstc2\_dev | 0.574 | 0.851 |
|  |  |  |  |
| nbestmodel\_goals\_nbest\_top1 | dstc2\_train | 0.705 | 0.590 |
| nbestmodel\_goals\_nbest\_top1 | dstc2\_dev | 0.572 | 0.857 |
|  |  |  |  |
| nbestmodel\_goals\_nbest\_majority | dstc2\_train | 0.736 | 0.527 |
| nbestmodel\_goals\_nbest\_majority | dstc2\_dev | 0.583 | 0.835 |
|  |  |  |  |
| nbestmodel\_goals\_nbest\_weighted | dstc2\_train | 0.704 | 0.593 |
| nbestmodel\_goals\_nbest\_weighted | dstc2\_dev | 0.572 | 0.855 |
|  |  |  |  |
| nbestmodel\_goals\_nbest\_voting\_top3 | dstc2\_train | 0.743 | 0.514 |
| nbestmodel\_goals\_nbest\_voting\_top3 | dstc2\_dev | 0.582 | 0.835 |
|  |  |  |  |
| nbestmodel\_goals\_nbest\_voting\_top5 | dstc2\_train | 0.741 | 0.518 |
| nbestmodel\_goals\_nbest\_voting\_top5 | dstc2\_dev | 0.590 | 0.820 |
|  |  |  |  |
| nbestmodel\_goals\_nbest\_voting\_top7 | dstc2\_train | 0.739 | 0.522 |
| nbestmodel\_goals\_nbest\_voting\_top7 | dstc2\_dev | 0.582 | 0.836 |

### *Observations:*

* Considering N-Best ASR does help
* “nbestmodel\_goals\_nbest\_voting\_top5” is the best model under this framework
* Weighted Voting is not better
* Now, it beats the HWUbaseline on the accuracy in the training but not in dev.

# TODO:

## Combine HWUBaseline and N-Best Model: Leveraging both SLU and ASR

The HWUBaseline considers only the SLU and the N-Best considers only the ASR. Therefore, combining them might get a better performance.

## Different Classifiers for different dialog acts

Inspired by the Short Answer Assessment Task, using different classifiers for different questions might help. Now, the number of questions asked by the system is limited. Thus, it is possible to build one classifier for one question.

## Speech Recognition Error Correction

Just add some hand-craft rules to handle high-frequent ASR errors.

# Reference

[1] Steve Young, Milica Gasic, Blaise Thomson, and Jason Williams, [POMDP-based Statistical Spoken Dialogue Systems: a Review](http://research.microsoft.com/apps/pubs/default.aspx?id=185321), in *Proceedings of the IEEE*, vol. PP, no. 99, pp. 1-20, Proceedings of the IEEE, 2013