# Summary

# DSTC1: Linear Combination of Scores

## Result:

# New Challenge: DSTC2

# Challenge Topline

* Topline1: If the correct one appears in one of the ***SLU***, add it to the output; if it doesn’t appear, ignore it.
* Topline2: If the correct one appears in ***ASR***, it is correct
* Topline3: If the correct one appears in ***Transcription***, it is correct

For the “topline2” and “topline3”, only the results with all candidates are reported because there is only one transcription and the ASR is not sorted as the NLU. In other words, the top NLU might not come from the top ASR.

## Results

## Observation

### “joint goals”

* For the Topline3 > Topline2 > Topline1 > 2wayTopline>HWUBaseline>Baseline
* Only considering the top1 is not enough. It is because the HWUbaseline is very close to topline1 with top1 and it will be really hard to beat it.
* It is possible to improve the performance by considering N-Best. For topline1, the performance for top10 is 84.9% compared to 74.6% for top1. It can get to 80.7% even by considering just top2.
* ASR error is a big issue. The result for “topline3” is much better than topline2. The best one is 95.8% based on the transcription. [It is not 100% because of some of slot-value cannot be found in the transcription, such as “pricerange = moderate”, “area = dontcare”]

### “method” & “request”

The “method” and “request” are inferred by the SLU, so that their topline2 and topline3 results are the same as topline1. [ASR and Transcription don’t help. For example, for “request=phone”, “phone” might not appear in the transcription.]

However, we can treat them as an independent problem by building classifiers for them. For “method”, it is just a 5-way classification problem [The only issue is that the current result depends on the previous one. For example, if there is no method mentioned at the current turn, the method is assumed as same as the old one]; for “request”, only 8 things are “requestable”, so we can build a classifier for each one by asking “Is address requested?”, “Is phone requested?”, etc.

# H2

# H3

# Method Classification

# Request Classification

TODO

# Considering top1 + top2

TODO