# HWU Baseline Belief Tracker for DSTC 2 & 3

Zhuoran Wang
MACS, Heriot-Watt University, Edinburgh
zhuoran.wang@hw.ac.uk

October 16, 2013

## 1 Description

The HWU belief tracker `baseline_HWU.py` is an extended version to the work presented in [2]. The fundamental idea is to estimate the belief on a user goal hypothesis by computing the overall probability of it being stated during multiple dialogue turns, and can be summarised as follows.

For a user goal hypothesis $h$ (consist of a slot-value pair), if an observation that agrees with $h$ is observed in the current turn with probability (confidence score) $p$, then the belief on $h$, $b(h)$, is updated as:

$$b(h) \leftarrow 1 - (1 - b(h))(1 - p) \tag{1}$$

otherwise, if an observation conflicting with $h$ is observed in the current turn with probability (confidence score) $\tilde{p}$, then we updated the belief $b(h)$ as:

$$b(h) \leftarrow b(h)(1 - \tilde{p}) \tag{2}$$

To adapt this method to the DSTC 2 and DSTC 3 tasks, one of the key issues is to handle user goal change. The strategy here is that when an observation presents a goal hypothesis (i.e. a slot-value pair), we assume that it will have a conflicting contribution to the other (previously observed) goal hypotheses for the same slot. In addition, we apply two different strategies to compute the beliefs on 'method' and 'requested slots'. For 'requested slots', we still use Eq. (1) for belief update, which estimates the probability of a slot being ever requested during a dialogue. But to compute beliefs for 'method',

a same procedure as the 'focus' baseline is employed, except that we always use the residual probability (1 minus the overall probability for the rest four explicit method labels) for the 'none' method. Our intuition here is that in comparison with Eq. (1), the belief update approach in the 'focus' baseline tends to make the previous belief on a hypothesis to contribute less to its current belief. Since 'method' is a 'turn-wise' issue, an event observed in a turn should have less long-term influence on future turns. Hence, the 'focus' baseline approach would be more suitable for 'method' tracking.

Another major modification in this version is that there is a rule for noise adjustment. This rule is to handle the cases that for some dialogues, most of the top SLU hypotheses are correct, however the confidence scores in every turn are very low. We adopt a simple intuitive assumption that in each dialogue, there exist a noise factor that proportionally reduces the confidence scores for non-NULL SLU hypotheses. Without the noise factor, the marginal confidence scores for all non-NULL SLU hypotheses in most turns should be close to 1. Therefore, at each turn of a dialogue, we fit its marginal confidence score on non-NULL SLU hypotheses together with all those marginal confidence scores for its previous turns (within the same dialogue) to a beta distribution, of which the mean is taken as the noise factor and is used for re-scaling the confidence scores of those non-NULL SLU hypotheses in that turn. To do the beta fitting, a cheap but less accurate approximation is used to compute the maximum likelihood estimation (MLE) of the two beta distribution parameters [1], because an accurate solution here will be too slow for turn-based belief update.

In addition, there was a relatively artificial rule proposed in [2], which assumes that there exists 'implicit affirm' to 'impl-conf' system actions. But such a rule is removed from this new version of HWU baseline, as we found that it is slightly harmful to the results this time (causing about 0.0005 accuracy reduction).

# 2 Running the script

The HWU baseline tracker requires the `dataset_walker` class, so should be placed in the same directory as the other Python scripts. To run it:

```
python baseline_HWU.py --dataset dstc2_dev --dataroot data
--ontology scripts/config/ontology_dstc2.json --trackfile
```

`baseline-hwu.json`

There is also an 'original' version of the HWU baseline tracker that is implemented exactly as described in [2] (without goal change or noise adjustment, but with the 'implicit affirm' rule). By default, this version is switched OFF, but one can use the following command to run it:

```
python baseline_HWU.py --dataset dstc2_dev --dataroot data
--ontology scripts/config/ontology_dstc2.json --trackfile
baseline-hwu.json --original true
```

# 3 Performance

The performance of the HWU baseline trackers on `dstc2_dev` evaluated using the featured metrics in comparison with the top-hypothesis and the 'focus' baselines is sketched in Table 1.

| | | Joint Goals | Requested | Method |
|---|---|---|---|---|
| HWU | Accuracy | **0.6230448** | 0.9026325 | **0.8607016** |
| | L2 | **0.6013564** | 0.1642966 | **0.2171417** |
| | ROC.v2_ca05 | 0.0000000 | 0.0000000 | 0.0000000 |
| Baseline (focus) | Accuracy | 0.6120959 | 0.8936170 | 0.8312150 |
| | L2 | 0.6318690 | 0.1743412 | 0.2652702 |
| | ROC.v2_ca05 | 0.0000000 | 0.0000000 | 0.3293578 |
| HWU (original) | Accuracy | 0.5643900 | 0.9026325 | 0.8556177 |
| | L2 | 0.6445889 | 0.1630160 | 0.2177850 |
| | ROC.v2_ca05 | **0.1782910** | 0.0000000 | **0.4869281** |
| Baseline (top-hypothesis) | Accuracy | 0.5010428 | **0.9029931** | 0.7989324 |
| | L2 | 0.9608538 | **0.1546287** | 0.3621605 |
| | ROC.v2_ca05 | 0.0000000 | 0.0000000 | 0.0000000 |

Table 1: Performance of the baseline trackers with respect to featured metrics on `dstc2_dev`.

# References

[1] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions, Volume 2*. Wiley, 2nd edition, 1995.

[2] Z. Wang and O. Lemon. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the 14th annual SIGdial Meeting on Discourse and Dialogue*, pages 423–432, Metz, France, 2013.