We first evaluate our model on MultiWOZ 2.0 dataset as shown in Table 1. We compare with five published baselines. TRADE (Wu et al., 2019) is the current published state-of-the-art model. It utilizes an encoder-decoder architecture that takes dialogue contexts as source sentences, and takes state annotations as target sentences. SUMBT (Lee et al., 2019) finetunes a pre-trained BERT model (Devlin et al., 2019) to learn slot and utterance representations. Neural Reading (Gao et al., 2019) learns a question embedding for each slot, and predicts the span of each slot value. GCE (Nouri & Hosseini-Asl, 2018) is a model improved over GLAD (Zhong et al., 2018) by using a slot-conditioned global module. Details about baselines are in Section 6.

| | Joint | Slot |
|---|---|---|
| GLAD | 35.57 | 95.44 |
| GCE | 36.27 | 98.42 |
| Neural Reading | 42.12 | - |
| SUMBT | 46.65 | 96.44 |
| TRADE | 48.62 | 96.92 |
| DSTQA w/span | **51.36** | 97.22 |
| -graph | 50.89 | 97.17 |
| -gating | 50.38 | 97.14 |
| -bi att +avg | 49.74 | 97.11 |
| -bi att | 49.51 | 97.07 |
| -ELMo +GloVe | 49.52 | 96.96 |
| DSTQA w/o span | **51.44** | 97.24 |
| -ELMo +GloVe | 50.81 | 97.19 |

Table 1: Results on MultiWOZ 2.0 dataset.

For our model, we report results under two settings. In the DSTQA w/span setting, we do span prediction for the five time related slots as mentioned in Section 5.1. This is the most realistic setting as enumerating all possible time values is not practical in a production environment. In the DSTQA w/o span setting, we do value prediction for all slots, including the five time related slots. To do this, we collect all time values appeared in the training data to create a value list for time related slots as is done in baseline models. It works in these two datasets because there are only 173 time values in the training data, and only 14 out-of-vocabulary time values in the test data. Note that in all our baselines, values appeared in the training data are either added to the vocabulary or added to the domain ontology, so DSTQA w/o span is still a fair comparison with the baseline methods. Our model outperforms all models. DSTQA w/span has a 5.64% relative improvement and a 2.74% absolute improvement over TRADE. We also show the performance on each single domain in Appendix A.3. DSTQA w/o span has a 5.80% relative improvement and a 2.82% absolute improvement over TRADE. We can see that DSTQA w/o span performs better than DSTQA w/span, this is mainly because we introduce noises when constructing the span labels, meanwhile, span prediction cannot take the benefit of the bidirectional attention mechanism. However, DSTQA w/o span cannot handle out-of-vocabulary values, but can generalize to new values only by expanding the value sets, moreover, the performance of DSTQA w/o span may decrease when the size of value sets increases. Table 2 shows the results on MultiWOZ 2.1 dataset. Compared with TRADE, DSTQA w/span has a 8.93% relative improvement and a 4.07% absolute improvement. DSTQA w/o span has a 12.21% relative improvement and a 5.57% absolute improvement. More baselines can be found at the leaderboard.[4] Our model outperforms all models on the leaderboard at the time of submission of this paper.

| | Joint | Slot |
|---|---|---|
| TRADE | 45.60 | - |
| DSTQA w/span | **49.67** | 97.10 |
| -graph | 49.48 | 97.05 |
| -ELMo +GloVe | 48.15 | 96.98 |
| DSTQA w/o span | **51.17** | 97.21 |
| -ELMo +GloVe | 50.03 | 97.12 |

Table 2: Results on MultiWOZ 2.1 dataset.

**Ablation Study**: Table 1 also shows the results of ablation study of DSTQA w/span on MultiWOZ 2.0 dataset. The first experiment completely removes the graph component, and the joint accuracy drops 0.47%. The second experiment keeps the graph component but removes the gating mechanism, which is equivalent to setting $\gamma$ in Equation (2) to 0.5, and the joint accuracy drops 0.98%, demonstrating that the gating mechanism is important when injecting graph embeddings and simply adding the graph embeddings to context embeddings can negatively impact the performance. In the third experiment, we replace $B_i^{QD}$ with the mean of query word embeddings and replace $B_j^{CD}$ with the mean of context word embeddings. This is equivalent to setting the bi-directional attention scores uniformly. The joint accuracy significantly drops 1.62%. The fourth experiment completely removes the bi-directional attention layer, and the joint accuracy drops 1.85%. Both experiments show that bidirectional attention layer has a notably positive impact on model performance. The fifth experiment substitute ELMo embeddings with GloVe embeddings to demonstrate the benefit of using contextual word embeddings. We plan to try other state-of-the-art contextual word embeddings such as BERT (Devlin et al., 2019) in the future. We further show the model performance on different context lengths in Appendix A.4.

[4] http://dialogue.mi.eng.cam.ac.uk/index.php/corpus/