Task Oriented Dialogue State Tracking with GPT-2

MLMI 8 - Neural Machine Translation and Dialogue Systems

April 23, 2022

Candidate no.: J902C

Word count: 2320[1]

UNIVERSITY OF CAMBRIDGE
Department of Engineering

[1]Excluding appendix, tables, footnotes, images, code and titles

Trumpington Street
Cambridge CB2 1PZ
Tlf.: +44 01223 332600
http://www.eng.cam.ac.uk/

UNIVERSITY OF CAMBRIDGE

Table of contents

1. Overview

In this practical Natural Language Understanding (NLU) and Dialogue State Tracking (DST) are investigated using GPT-2 transformer models [1].

Natural Language Understanding is the automatic annotation of dialogue acts within individual utterances or dialogue turns. NLU is performed and evaluated at the turn level within each dialogue. On the other hand, Dialogue State Tracking consists of determining at each turn of a dialogue the full representation of what the user wants at that point in the dialogue.

Transformers are usually language models trained on a huge text collection and they are described by a large amount of parameters. Training a transformer model is unfeasible for the computational power available, but a pre-trained model (GPT-2 model from HuggingFace [1]) can be downloaded and fine-tuned with relatively small amount of in-domain data. One of the keys to successfully develop this project is to determine how the fine-tuning task is formulated and executed.

The Multi-Domain Wizard-of-Oz (MultiWOZ) data sets are going to be used to report performance of in Natural Language Understanding and Dialogue State Tracking after fine-tuning several GPT-2 models on the in-domain data. MultiWOZ is a collection of human-human written conversations over multiple topics and it is widely used in the literature for reporting progress in DST. There exist different versions, because the dataset has been cleaned several times. Here, in this practical, the results will be reported just in the MultiWOZ 2.1 version [4].

	Train	Dev	Test
Turns	54971	7374	7368
Dialogues	7888	1000	999

Table 1.1: MultiWOZ 2.1 training, development, and test set sizes in this practical.

Exercises 1 and 2 (sections 2 and 3 respectively) will focus on fine-tuning, decoding and scoring for NLU. Exercise 3 (section 4) will consist on DST reporting the joint accuracy of the investigated models and, finally, in exercise 4 (section 5) we will build our own GPT-2 DST system.

2. Exercise GPT2DST.1

The very first step and the first exercise is to fine-tune a pre-trained GPT-2 model for NLU. The GPT-2 model is pre-trained in vast datasets, but the fine-tuning is done just on the MultiWOZ 2.1 dataset. We are provided with the dataset in the expected format to fine-tune the GPT-2 model.

A script named `train.nlu.mwoz21.slurm.wilkes2` is provided to fine-tune a HuggingFace GPT-2 model. The script can be set to train from models provided at batch 60000 or from scratch, i.e., a flat-start fine tuning from GPT-2 as distributed by HuggingFace [1]. In our case, we decide to execute the first option and keep fine-tuning the GPT-2 model based on the fine-tuned model until batch 60000 (continued training). To do so, we just have to submit a SLURM job to the server using the following command:

```
    sbatch train.nlu.mwoz21.slurm.wilkes2
```

After approximately 2 hours the script is done and results are shown in table 2.1. Losses (negative log-likelihood) are reported in both training and developing sets. Results until batch 60000 are collected from the provided log at
`$BDIR/checkpoints/nlu_flat_start/logs/train-gpt2-dst.log`
The rest of the results report development set loss every 20,000 batches, up to 200,000 batches, and training set loss every epoch.

Batch	Train loss	Dev loss
20000		0.4037
40000		0.2639
54971	0.4144	0.2180
60000		0.2093
80000		0.2323
100000		0.2031
114971	0.2325	0.2112
120000		0.2254
140000		0.2088
160000		0.2285
169942	0.1916	0.2105
180000		0.2454
200000		0.2866

Table 2.1: Replication of MultiWOZ 2.1 NLU executing fine-tuning as continued training. Development set loss is reported every 20000 turns and training loss is reported every epoch (54,971 turns).

Comparing table 2.1 with the one provided in the practical statement, we can analyze results after batch 60000 and see that they are pretty similar. Comparison can be done just up to batch 120000 because the provided table is incomplete, but analyzing batch 80K and 100K we can see that dev. losses are in the same page. In general, the model tend to improve its performance and it is *expected* that continued training performs better in test set after 200K batches than just fine-tuning the model up to 60K batches.

3. Exercise GPT2DST.2

After fine-tuning HuggingFace GPT-2 models on MultiWOZ 2.1 NLU training sets, a turn level belief state annotation can be generated for each dialogue turn in the input file by decoding.

It is provided another SLURM script `decode.nlu.mwoz21.slurm.wilkes2` to run the decoder over the data. The script has to be modified to specify the fine-tuned GPT-2 model path, including the line:

```
 CHECKPOINT=/rds/user/as3159/hpc-work/checkpoints/nlu_continued_training
/model.80000/
```

In the previos example, the variable `CHECKPOINT` stores the path where a GPT-2 model has been fine-tuned with 80000 batches. The script can be submitted to the HPC servers by executing the command:

```
 sbatch decode.nlu.mwoz21.slurm.wilkes2
```

The output of this script includes the dialogue turn itself, copied directly from the input field `dst_input`, and the predicted linearized belief state follows the input, appearing between the `<BOS>` and `<EOS>` tokens.

After decoding, the provided python script `multiwoz_dst_score.py` can be executed to score the NLU decoder output against the reference turn level belief states. Table 3.1 shows the DST Average Joint Accuracy after fine-tuning a GPT-2 model by continued training. The results are reported following the practical statement, focusing from batch 60000 to 200000 every 20000 batches.

Batch	Dev	Test
5000	68.66	66.30
52000	70.56	67.44
54000	70.90	69.46
60000	69.24	67.13
80000	71.94	70.14
100000	70.22	68.59
120000	71.74	69.84
140000	72.20	70.28

160000	73.27	71.20
180000	73.07	71.85
200000	74.48	72.58

Table 3.1: NLU Joint Accuracy for MultiWOZ 2.1 found via decoding with GPT2NLU using continued training models

In general, DST Average Joint Accuracy in the development set tends to increase. Similarly, it also gradually increases on the test set, what means the GPT-2 is improving while being fine-tuned up to 200000 batches, reaching a top DST Average Joint Accuracy of 72.58%.

4. Exercise GPT2DST.3

The investigation of turn level Natural Language Understanding is extended to Dialogue State Tracking. As the base case, DST Joint Accuracy of the turn level annotations against the dialogue level reference can be measured, but it is expected that this accuracy will be considerably low because of the fact that the predicted belief state is based only on the current turn. This can be improved by simply concatenating turn level NLU belief state predictions, implementing a Cheap and Cheerful (*something that does not cost a lot but is attractive and pleasant*) Dialogue State Tracker.

The provided python script `cc_dst.py` concatenates the turn level belief states to generate dialogue level belief states. It is expected that the overall performance is limited, since it depends on the independence of the generated turn level annotations.

Once the dialogue level annotations have been generated, they can be scored, i.e., the DST Joint Accuracy can be measured. The following command would score the fine-tuned GPT2NLU CC-DST model by continued training (200000 batches) contrasting its annotations with the reference:

```
 python $BDIR/multiwoz_dst_score.py --field dst_belief_state  \
 --dst_ref $BDIR/data_preparation/data/multiwoz21/refs/test/test_v2.1.
json  \
 --h hyps/test/cc_dst/nlu_continued_training/model.200000/belief_states.
json
```

Using this command on the development and test reference, the DST Joint Accuracy can be obtained for different GPT-2 models that have been fine-tuned for different number of batches. The results are shown in table 4.1.

Batch	Dev	Test
5000	34.26	31.46
52000	39.03	34.22
54000	41.36	37.76
60000	35.72	31.79
80000	41.69	38.57
100000	37.75	34.73
120000	40.29	37.19
140000	41.13	36.92
160000	42.26	39.45
180000	44.78	41.26
200000	46.46	42.55

Table 4.1: MultiWOZ 2.1 DST Joint Accuracy: GPT2NLU (continued training) and Cheap and Cheerful DST

In general, DST Joint Accuracy tend to increase on the development set, reaching a top accuracy of 46.46%. Focusing on the test set, the behaviour is similar. The DST Joint Accuracy usually improves if the GPT-2 model is fine-tuned for more batches. The best result on the test set, 42.55% DST Joint Accuracy, is obtained after fine-tuning for 200000 batches.

5. Exercise GPT2DST.4

5.1. Building our own GPT-2 DST system

So far we have investigated the models described in the statement, but now we are interested in designing, building and evaluating our own GPT-2 Dialogue State Tracker on the MultiWOZ 2.1 dataset.

The proposed model in this section merges NLU and DST into a single system. To do so, a modified version of the given dataset (turn level MultiWOZ 2.1) has to be built. From the turn level data provided a dialogue level dataset is crafted using a similar script to `cc_dst.py` (used in exercise 3, section 4). This script was a simple implementation of hypothesis concatenation, but in this exercise this script is extended to concatenate training, development and testing data.

The snippet 5.1 shows a fragment of the new dialogue level training dataset, where a turn contains information about the current turn itself and the previous part of the dialogue between the user and the system. This format is logically also used for development and testing data.

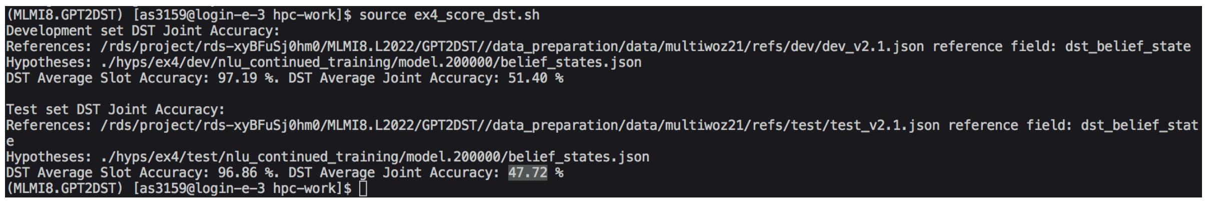
```
    [
        {
            "example_id": "MUL0003-0",
            "dst_input": "<USR> i am looking for a place to stay . it needs
                to be a guesthouse and include free wifi . "
        },
        {
        "example_id": "MUL0003-1",
        "dst_input": "<USR> i am looking for a place to stay . it needs
            to be a guesthouse and include free wifi .  <SYS> there are
            23 hotel -s that meet your needs . would you like to narrow
            your search by area and and or price range ? <USR> i would
            like for it to be cheap and include free parking . "
      },
      {
        "example_id": "MUL0003-2",
        "dst_input": "<USR> i am looking for a place to stay . it needs
            to be a guesthouse and include free wifi .  <SYS> there are
            23 hotel -s that meet your needs . would you like to narrow
            your search by area and and or price range ? <USR> i would
            like for it to be cheap and include free parking .  <SYS>
            there are 9 guesthouse hotel -s in various area -s . what
            part of town are you hoping for ? <USR> nothing in particular
            . i just need it booked for 6 people for a total of 4 nights
            starting from sunday . i would also like the reference number
            , please . "
      },
    ]
```

Listing 5.1: Example of the new dialogue level data

The system is trained using the new dataset and executing a modified version of the `train.nlu.mwoz21.slurm.wilkes2` script, setting the variables `TRAIN_DATA` and `DEV_DATA` to the new training set and development set respectively. In this case, the model is fine-tuned from scratch (from GPT-2 as distributed by HuggingFace [1]) up to 200000 batches. After training, decoding can be run using a modified version of the script `decode.nlu.mwoz21.slurm.wilkes2`, where the trained model and the directory for the decoded dialogues are specified. Finally, scoring (on the test set) can be done by executing:

```
    echo "Development set DST Joint Accuracy:"
    python $BDIR/multiwoz_dst_score.py --field dst_belief_state \
    --dst_ref $BDIR/data_preparation/data/multiwoz21/refs/dev/dev_v2.1.json
    \
    --h ./hyps/ex4/dev/nlu_continued_training/model.200000/belief_states.
   json
    echo "Test set DST Joint Accuracy:"
    python $BDIR/multiwoz_dst_score.py --field dst_belief_state \
    --dst_ref $BDIR/data_preparation/data/multiwoz21/refs/test/test_v2.1.
   json \
    --h ./hyps/ex4/test/nlu_continued_training/model.200000/belief_states.
   json
```

The results are shown in figure 5.1 and summarized in table 4.



```
(MLMI8.GPT2DST) [as3159@login-e-3 hpc-work]$ source ex4_score_dst.sh
Development set DST Joint Accuracy:
References: /rds/project/rds-xyBFuSj0hm0/MLMI8.L2022/GPT2DST//data_preparation/data/multiwoz21/refs/dev/dev_v2.1.json reference field: dst_belief_state
Hypotheses: ./hyps/ex4/dev/nlu_continued_training/model.200000/belief_states.json
DST Average Slot Accuracy: 97.19 %. DST Average Joint Accuracy: 51.40 %

Test set DST Joint Accuracy:
References: /rds/project/rds-xyBFuSj0hm0/MLMI8.L2022/GPT2DST//data_preparation/data/multiwoz21/refs/test/test_v2.1.json reference field: dst_belief_stat
e
Hypotheses: ./hyps/ex4/test/nlu_continued_training/model.200000/belief_states.json
DST Average Slot Accuracy: 96.86 %. DST Average Joint Accuracy: 47.72 %
(MLMI8.GPT2DST) [as3159@login-e-3 hpc-work]$ 
```

Figure 5.1: Terminal output after scoring our own GPT-2 DST model, showing DST Average Joint Accuracy

Model	**Batch**	**Dev**	**Test**
CC-DST	200000	46.46	42.55
NLU+DST	200000	51.40	47.72

Table 5.1: Comparison of Cheap and Cheerful DST system (detailed in section 4) and the new proposed model that merges NLU and DST into a dialogue level system.

It is easy to check that the proposed system improves the results of the previous exercise (section 4), achieving a 47.72% DST Average Joint Accuracy on the test set. This was expected since in exercise 3 dialogue level belief states were just generated by concatenating the predictions, whether in this exercise the DST procedure was refined by building new dialogue level dataset and training, decoding and scoring using these enhanced data.

5.2. Comparison to UBAR

UBAR is a end-to-end system with GPT-2 ([3]) that models task-oriented dialogues on a dialog session level. UBAR is trained and evaluated fine-tuning a GPT-2 model in a more realistic setting than our proposed model in this section 5, since its dialog context has access to user utterances and all content it generated such as belief states, system acts, and system responses of every dialog turn. In contrast, the proposed NLU+DST system in this practical only uses dialogue level belief states for training and testing.

UBAR makes the process of inference easier for the current turn by conditioning on the previous belief states and system acts in the dialogue. Figure 5.2 shows an example of two-turn interaction and what information is UBAR processing.

The creators of UBAR also conducted experiments on the MultiWOZ 2.1 dataset, achieving state-of-the-art performances accordingly to [3]. This is perfect to compare the capability of UBAR with our system of section 5.1. The performances are compared in table 5.2.

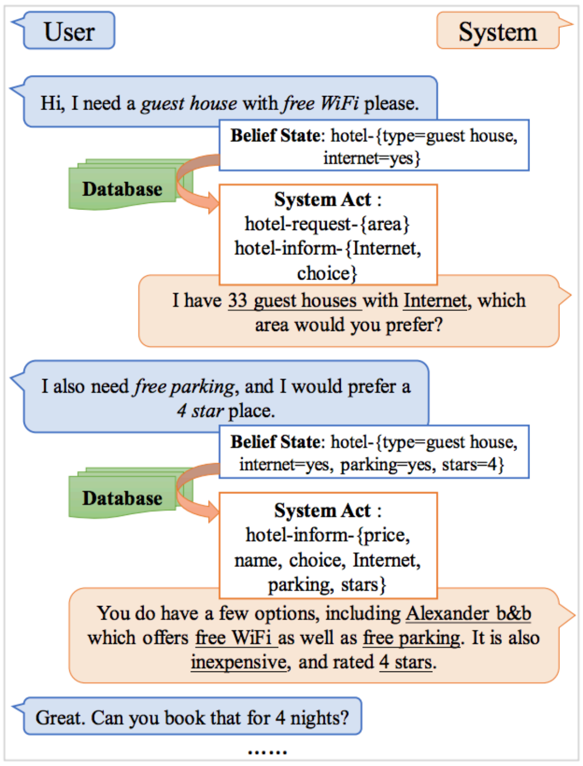

User
System
Hi, I need a *guest house* with *free WiFi* please.
Belief State: hotel-{type=guest house, internet=yes}
Database
System Act : hotel-request-{area} hotel-inform-{Internet, choice}
I have 33 guest houses with Internet, which area would you prefer?
I also need *free parking*, and I would prefer a *4 star* place.
Belief State: hotel-{type=guest house, internet=yes, parking=yes, stars=4}
Database
System Act : hotel-inform-{price, name, choice, Internet, parking, stars}
You do have a few options, including Alexander b&b which offers free WiFi as well as free parking. It is also inexpensive, and rated 4 stars.
Great. Can you book that for 4 nights?
......

Figure 5.2: An example of the first two-turn interactions between a user and a TOD system. UBAR uses all this information for training and inference, whether the proposed system in 5.1 only uses dialogue level belief states. This image is sourced from [3].

Model	**Dev**	**Test**
NLU+DST	51.40	47.72
UBAR	-	56.20

Table 5.2: Comparison of UBAR and the new proposed model that merges NLU and DST into a dialogue level system. Units are measured in DST Average Joint Accuracy.

As it can be seen, UBAR outperforms our system, reaching a 56.20% DST Average Joint Accuracy on the test set, compared to the 47.72% of the prosed system in section 5.1. This was expected since UBAR uses much more information than the trained model in this project.

5.3. Model Card of our own GPT-2 DST system

The following model card describes our own GPT-2 DST system described in section 5.1. There are several formats for a model card, but here it is used the one proposed in [2].

Model Details:

- Built by 2022 MPhil. MLMI Cambridge student, using scripts elaborated by professor Bill Byrne to fine-tune and evaluate GPT-2 models.
- Dialogue State Tracking system with GPT-2. HuggingFace [1] GPT-2 transformer models are employed.
- Pre-trained in a huge general dataset and fine-tuned in this practical for task-oriented dialogue state tracking.
- LICENSE: MIT license.

Intended Use:

- Automatic annotation of dialogue acts within dialogue turns.
- Determining at each turn of a dialogue the full representation of what the user wants at that points in the dialogue.
- Not intended to be deployed to a final system, since it has several limitations.

Metrics:

- Performance reported using DST Average Joint Accuracy of the dialogue level annotations.

Training Data:

- Dialogue level belief states of several human-system interactions.
- MultiWOZ 2.1 training set [4].

Evaluation Data:

- Dialogue level belief states of several human-system interactions.
- MultiWOZ 2.1 test set [4].

Ethical Considerations:

- Dialogue annotations, which span 7 distinct task-oriented domains, are collected and publicly published by Amazon and Google researchers [4] to evaluate different DST systems.

Caveats and Recommendations:

- Does not use other sources of information, such as system acts or system responses.
- To improve the performance is worth checking UBAR system [3].

- Synthetic data has a limited range of domains. It is worth evaluating its performance in real-word situations.

Quantitative Analyses:

- Achieved 51.40% DST Average Joint Accuracy on the development set and 47.72% DST Average Joint Accuracy on the test set.
- The system out-performed previous investigated models, such as a CC-DST system based on concatenating turn level annotations to generate dialogue level predictions.
- There is room for improvements (e.g. UBAR system [3]).

6. Summary and Conclusion

In this practical Natural Language Understand (NLU) and Dialogue State Tracking (DST) were investigated using provided GPT-2 transformer models by HuggingFace [1]. These transformer models were fine-tuned on the MultiWOZ 2.1 dataset [4] in order to evaluate performance in DST.

In the first 2 exercises we fine-tuned, decoded and scored GPT-2 systems for NLU. The results were quite similar to the ones provided in the practical statement as an example. The little variations in number could be due to randomness in initialization or optimization. In addition, exercise 3 consisted on DST, concatenating turn level belief states of the predictions to generate dialogue level belief states. The overall performance was limited since it depends on the independence of the generated turn level annotations.

The system investigated in exercise 3 got a 42.55% DST Average Joint Accuracy on the test set, and in exercise 4 we tried to improve this result by further investigating on DST. A new dataset was crafted in order to fine-tune a GPT-2 system using train, development and test data using dialogue level annotations. This approach achieved a final DST Joint Accuracy of 47.72%, outperforming the base result of exercise 3.

Future work could consist on improving the current top performing model including more information to use for training and inference or taking inspiration from other DST models, such as UBAR [3].

To sum up, in this practical we have tried to automatically annotate dialogue acts within dialogue utterances and also how to determine at each turn of a dialogue the full representation of what the user wants at that exact moment. The achieved results were not outstanding, but in the same page as other state-of-the-art models results.

Bibliography

[1] Thomas Wolf et. al. (2019). ’HuggingFace’s Transformers: State-of-the-art Natural Language Processing’. *Computing Research Repository (CoRR)*. http://arxiv.org/abs/1910.03771

[2] Margaret Mitchell et. al. (2019). ’Model Cards for Model Reporting’. *Association for Computing Machinery (ACM)*. https://arxiv.org/abs/1810.03993

[3] Yunyi Yang et. al. (2021). ’UBAR: Towards Fully End-to-End Task-Oriented Dialog Systems with GPT-2’. *Computing Research Repository (CoRR)*. https://arxiv.org/abs/2012.03539

[4] Mihail Eric et. al. (2019). ’MultiWOZ 2.1: Multi-Domain Dialogue State Corrections and State Tracking Baselines’. *Computing Research Repository (CoRR)*. https://arxiv.org/abs/1907.01669