Supplementary Material For Submission "Validating Predictive Models Of Evaluative Language For Controllable Data2Text Generation"

1 Empirical study

The empirical study is implemented as a django web application (Python > 3.8 and django 4.1.7) and can be deployed with nginx and gunicorn with little effort. Premise is a small publicly visible server and the rights to set up the app under a specific url for reproduction of the experiment in a real production environment. The app is lightweight and only needs a python interpreter and a django installation. For testing and reviewing, the app can be run locally. All necessary files are contained in the Github repository and this supplementary material folder in **/empirical_study**.

1.1 Usage

1. open terminal / cmd.exe and navigate to the directory /studyEvalAdv, which contains "manage.py"
2. execute "python3 manage.py makemigrations"
3. execute "python3 mangage.py migrate"
4. execute "python3 mangage.py runserver"
5. open browser and visit url "http://127.0.0.1:8000/evalAdv/"
6. enter dummy prolific id of format `^[a-f\d]{24}$` for testing

The first two commands may not be necessary, since the folder already contains pre-initialized databases, which were active in our production environment. For simple usage, please use one of the databases in **/databases**, copy to **/studyEvalAdv** and rename to "db.sqlite3".

1.2 Advanced usage

You can modify items etc. by manipulating the csv files "items.csv", "items_ablation.csv" and "distractors.csv". In order to transfer changes to the database, you should follow the steps in Figure (1).

2 Data analysis

All experimental data, which is persistently stored in databases, is available online in the Github repository and in this supplementary materials folder in **/Data_analysis_and_regression**. All source code for data analysis and for creating the graphs and tables depicted in the paper are available online in the same Github repository as well as in this folder.

Please use source code file **main_study_data_evaluation.py** in combination with the database file **studyEvalAdv_GER_50.sqlite3** for reproducing the analysis of the main study. For the ablation study, please use **ablation_data_evaluation.py** and the database file **studyEvalAdv_GER_ablation.sqlite3**.

The items for both studies are contained in csv files.

3 ADAC technical database

The ADAC technical database is proprietary data, which is why we are not allowed to distribute it. Please contact us as authors for more information on how to obtain a license from ADAC. We added a subfolder to the Github repository and to this supplementary materials folder which contains graphs and distribution plots for visualisation of the technical data in so far as this is relevant for the content of the paper, but no raw data from the database is provided.

4 Regression models

The regression models described in this paper are a linear regression model, a polynomial regression model and deep neural network for regression. Linear and polynomial models are implemented using scikitlearn, the DNN is implemented in Keras / Tensorflow. The DNN is implemented as a Keras sequential model consisting of a normalisation layer and dense layers, the network is trained using Adam optimizer, a learning rate of

```
# enter django shell
$ python3 manage.py shell
# execute the following commands in the shell
shell$ > from evalAdv.models import ExpectationItem, DistractorItem
shell$ > from imports import load_csv_items
shell$ > a = load_csv_items("<item_file_name>", ExpectationItem)
shell$ > for x in a:
shell$ >       x.save()
shell$ > b = load_csv_items("<distractors_file>", DistractorItem)
shell$ > for y in b:
shell$ >       y.save()
shell$ > quit()
```

Figure 1: Code Listing for advanced modification of read-only database tables of items and distractors

0.001 and mean absolute error as loss function. Source code for all regression models is available in the Github repository and here in the subfolder **/Data_analysis_and_regression**.

The models were trained on a local Unix workstation computer with an I9 10 Core CPU, 48 GB RAM, a Titan RTX 24 GB VRAM GPU and an Nvidia CUDA installation. Training of the models took an insignificant time of only a few minutes, slightly varying with the number of predictors chosen (1-15). The models are small and generally, they may also be run on a laptop without dedicated GPU and CUDA.

These models are trained on a subset of the ADAC database containing technical data of numerous cars. As already stated above, the database is proprietary and we are not allowed to distribute it. If you like to gain access to the data and train the models, please contact us authors for further information on how to obtain the license from ADAC. If you have a corresponding license, we authors can supply you with the data in the necessary input format.