Setting up the CCOBRA Framework

Nicolas Riesterer, Lukas Elflein, Paulina Friemann

October 30th, 2018

Cognitive Computation Lab,
Department of Computer Science,
University of Freiburg
Philosophy of CCOBRA

1. Modeling the experiment instead of abstract processed data
2. No constraints with respect to the underlying formalisms
3. Strong focus on predictive accuracy/precision
1. Framework presents task
2. Model generates prediction
3. Response is compared with ground truth
4. Model adapts to the human
5. Framework presents next task
Today’s Topics

1. Installing CCOBRA
2. Implementing Models for CCOBRA
3. Evaluating Models in CCOBRA
4. Submitting Models for the Seminar
Installing CCOBRA

Installing from the PyPI index:

$> pip install ccobra
$> python -c "import ccobra; dir(ccobra)"
[‘CCobraData’, ‘CCobraModel’, ..., ’model’, ’syllogistic’]

Attention

CCOBRA is only tested on Python 3 and depends on pandas and numpy.

Since the CCOBRA project is young, changes can and will happen. We will keep you posted.
Python class interface consisting of four methods:

- **start_participant:**
  Initializes internal structure for a new participant

- **pre_train:**
  Trains the model on training data prior to the prediction phase

- **predict:**
  Generates a single prediction for a given task

- **adapt:**
  Adapts the model based on a previous task-response combination
Random Model

- Subclassing the CCOBRA model interface
- Predicting a randomly chosen response
Evaluating the model

Download CCOBRA’s Github-Repository:
https://github.com/CognitiveComputationLab/ccobra

Use the contained CCOBRA benchmarking tool:

$> cd /path/to/ccobra-repository/ccobra-bench
$> python runner.py benchmarks/full.json
$> python runner.py benchmarks/full.json -m /path/to/model.py
Adaptive Random Model

- Remembers response frequencies
- Samples predictions based on individual response frequencies
- Relies on task encodings
Pre-Trained Adaptive Random Model

- Is initialized based on given training data (of other individuals)
- Relies on adaption as training
- Definition of a custom benchmark
Online Evaluation

https://orca.informatik.uni-freiburg.de/orca_sylwebsite/orca/

Syllogistic Reasoning Models

Please upload your syllogistic model below. The uploaded zip archive is required to contain a file called "model.py" in which a model class definition (subclassing cobra.syllogistic.SModel) is located. Please pay attention that this file is in the top-level directory of your archive and not part of a subfolder.

Choose File  no file selected

Upload

CCobra (Cognitive Computation for Behavioral Reasoning Analysis)
Daniel Brand <daniel.brand@cognition.uni-freiburg.de>
Nicolas Riesterer <riestern@cs.uni-freiburg.de>
CCOBRA Evaluation Demonstration

Accuracy

Computes the accuracy of the models, i.e., the percentage of correct predictions.

- MPSModel
- UniformModel
- Matching
- PHM
- Atmosphere
- NMT
- NVCModel
- PSYCOP
- Conversation
- VerbalModels

https://orca.informatik.uni-freiburg.de/orca_sylwebsite/orca/
Model Submission

- The online evaluation website requires zip archives containing the model definition in a file called `model.py`.
- Submissions (midterm, final) are required to be in exactly this archive format.
- Use the website to validate your submissions!
Next Steps

1. Decide on a general topic and on whether you want to work together with a teammate (send us a mail within the next week)
2. Try to implement rapid prototyping strategies and evaluate as often as possible
3. Contact us if you encounter any problems or questions (in particular with respect to CCOBRA)
4. Your first model will be due December 2nd, 23:59