# Chat-bot Prototype for Truck data

## Building blocks:

* brobot.py: The script containing the chatbot.
* network.py: The script containing the neural network for intent classification.
* dataloader.py: The dataloader, which feeds the data into the neural network.
* conversations: folder which contains the user-bot interactions.
* model: folder containing the saved model for the neural network.
* data: folder containing the input data, and the pickled pre-processed/embedded data.

## Data pre-processing

The data is read from a json file, which contains intents (5 classes with 20 query examples each). Since there was no data available, the data was generated manually based on common sense.

After loading, the each query is tokenized, lower-cased and lemmatized. The stop-words are kept in the queries, because removing these hurts the performance.

After feature extraction, each query is embedded by feeding these into a word2vec model (word2vec-google-news-300), and then padded to a length of at least 20 (if there are query examples that are larger than this, then to the length of the largest query). At the same time, the intent-labels for each query are mapped to a number value.

## Model

The mode used in the chatbot is a Neural Network created using the pyTorch framework. The data is divided into a training and validation sets using the sklearn package.

The network consists of:

* 1D Conv layer with a kernel\_size of 2, and a filter size of 64.
* ReLu activation function
* Fully connected layer with 64 \* 19 in\_features and 256 out\_features.
* ReLu activation function
* Fully connected layer with 256 in\_features and 128 out\_features.
* ReLu activation function
* Fully connected layer with 128 in\_features and 5 out\_features.
* Softmax activation function

Due to the manually generated data, the low amounts of data available, and the randomness of the test-train split, the results of the training has a high variance, however most of the time the accuracy is between 66%-77%.

## User-Bot interaction

The purpose of the chatbot is to provide information of trucks based on the input license plate number. For this, the following logic is coded into thet the bot:

* The bot asks how it can help at the beginning.
* It can recognize when someone tries to say hi to it, and responds in kind.
* It can also identify if a general information request is given to it, and produces a response which describes that it needs the license plate number to provide information about a truck.
* If information specific truck related information requests are identified, the response given indicated that this cannot be provided without the license plate number.
* The input plate number is checked against the available trucks.
  + If found, the response clarifies the available data about the trucks.
  + If not found, this is communicated to the user.
* After the plate number is given, the following intents are answered:
  + Location of the truck
  + Cargo of the truck
  + Fleet number and other specs of the truck.
* If other requests are given, these are classified into the general request category, or taken as a greeting.

Each conversation is recorded into the conversations folder into separate txt files.

## Known issues:

* Since the data was artificially generated, the model could not be optimized, and in turn the performance of the neural net is not ideal.
* The available resources (cpu, gpu, memory), prevents the training of deep neural networks.
* The bot lacks the ability to recognize if the user’s intention does not belong to any category. In other words, whatever input is given to the bot, it is always classified into 1 of 5 categories.
* The word2vec model is 1.5 GB, and needs to load each time the bot is executed. This hurts the performance of the bot during loading.