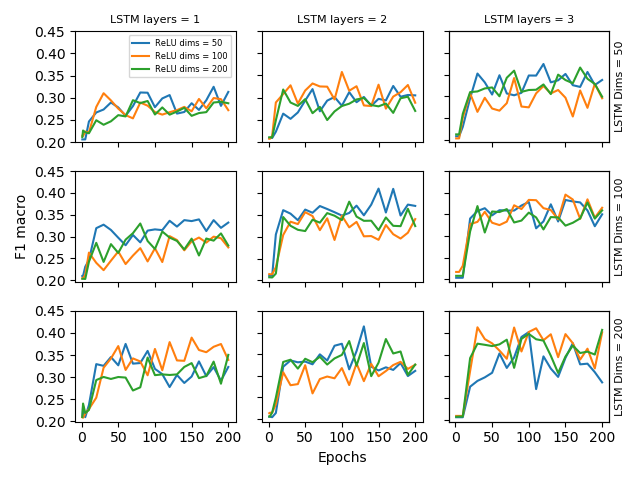
**Current state of the live veracity detection research project, 31.01.20**This is an over-view of the current state of the research project, and not a technical description of the code for the project. See the project README for a more extensive description of project classes and command-line interfaces.

**Stance detection**Stance detection is performed using an LSTM model – no alternative approaches have been tested. The model applies a number of linear layers with ReLU activation and drop-out. Grid-based hyperparameter optimization has been performed over the number of LSTM layers and dimensions, and the number of ReLU layers and dimensions. Below is a visualization of these tests, using a single ReLU layer, as this was found to perform the best.

The model applies the Adam optimizer and negative log loss.

The model works with the DAST dataset and the data generated for this research project through tweet scraping.

*To-do*Test multi-lingual approaches  
Test alternative model types to LSTM  
Test alternative optimizers, loss functions and dropout levels  
Test performance when combining DAST and twitter data  
Test use of alternative features – currently only word embeddings are used  
Misclassification analysis

Veracity determination  
Is performed using stance of conversation trees as features. Only an HMM model has been used, no alternatives have yet been tested. Performance have been tested for the following variations:

* Removal of tweets with “commenting” SDQC label
* Use of timestamps as additional features to SDQC label
* Majority voter approaches
* Casting unverified rumors as “true” or “false”
* Different test datasets:
  + Purely DAST data
  + Purely Pheme data
  + Combination of DAST and Pheme data
* Different train datasets:
  + Purely DAST data
  + Purely Pheme data
  + Combination of DAST and Pheme data
* Different HMM approaches: Gaussian or Multinomial

All tests were performed with five-fold cross validation.



*To-do*Test alternative model types  
Misclassification analysis  
Find and test HMM model type which makes better use of the timestamp features

**Tweet scraping and dataset construction**A tweet scraper has been constructed which can perform scrapes based on a user-defined query, or simply finding popular tweets in Danish from the Danish twitter-sphere. The scraper will navigate to the root node of a given tweet conversation tree before scraping, and scrape the full tree.

Construction of an automation script has begun, found in automation.sh, which will scrape tweets based on a number of hashtags. These are to be labeled for use in both stance detection and veracity prediction. Work on annotation tools has begun, but is far from finished – code from an approach using the web application doccano can be found in the /labeling\_doccano/ folder, and the start of a tkinter GUI can be found in annotation\_gui.py.

*To-do*Finish automation script for tweet scraping  
Finish creating a labeling tool  
Label additional data for stance detection and veracity prediction

**Live veracity detection**A script, live\_veracity.py, has been created, which allows the scraping of the most popular tweet conversation threads from Denmark, predicting stances in comment branches, predicting veracity based on these branch stances, and finally writing potential rumours to a file.

*To-do*  
Make automation for the script, so it can run once or twice a week  
Make a condition within live\_veracity.py that a large majority of conversation branches should agree that a given source tweet is false, before it is saved as a potential rumour. Cut-off suggested to be set at 90% by Leon.   
Combine the automation script for this task with the automation script for general tweet scraping and dataset creation, so that data scraped and automatically labeled using the live\_veracity.py script is also tagged to be subsequently reviewed and manually labeled.