This is the source code for the thesis project: **Evaluation of Crowdsourcing for Automatic Generalised Language Error Detection**

**Project abstract**

The project is aimed to create an English language error detection crowdsourcing experiment on the adopted crowdsourcing platform Amazon Mechanical Turk. The quality and representation of the annotations gathered are then evaluated using a variety of existing state-of-the-art techniques as well as new and innovative methods.

[**Prerequisites**](https://blog.monkeylearn.com/creating-sentiment-analysis-model-with-scrapy-and-monkeylearn/)

Python 3.5

Libraries:

numpy

pandas

tensorflow

matplotlib

itertools

collections

[**Usage**](https://blog.monkeylearn.com/aspect-analysis-from-reviews-using-machine-learning/)

The code in this directory was used for the analysis in the thesis: **Evaluation of Crowdsourcing for Automatic Generalised Language Error Detection**. To use the methods – mainly in agreement.py it is recommended to make a python environment with Virtualenv and run it in an IDE such as Pycharm.

**Code organization**

Most of functions and modules are implemented in agreement\_methods.py file. Besides, there are a series of Python scripts and Jupyter notebooks that implement some necessary scripts. Other modules and dependencies are as follows:

**Directories:**

agreement\_methods - analysis of the agreement methods

batch\_input - input of the AMT experiment sentence batch

batch\_results - results of the AMT experiment sentence batch

error\_counts - counts per error classes

classifier – directory of the MLP classifier, main file mlp\_main\_classifier

experiment\_output\_m2 - output of the experiment in m2 format

experiment\_output\_tsv - LSTM experiment output data

fce\_data\_sets - the fce CLC data set both in LSTM and m2 formats

location\_analysis - location analysis

lstm\_output - output from the LSTM

old\_version - old metric methods

plots - plots

small\_experiment - test data experiment

train\_data\_chunks - chunked data for classifier

weighting\_votes\_results - results from the weighting votes method

env - virtual environment

**Files:**

agreement.py - the main code file for the analysis

convert\_m2.py - helper function

create\_csv.py - helper file creator functions

fce\_api.py - the API for fce data sets

settings.py - configuration properties - file locations