CA – Assignment 2: Argument Mining

- Group: `FakeNews`
- Group members:
  - Adnan Manzoor
  - Sajjad Pervaiz
  - Kevin Taylor
  - Christoph Schäfer

Structure

```
.
└── argument-mining-assignment
    ├── Documentation.pdf
    ├── README
    ├── requirements.txt
    └── code
        ├── conf_bias_evaluation.py
        ├── model.py
    ├── data
        ├── essay_corpus.json
        ├── predictions.json
        ├── sample_prediction.json
        ├── train-test-split.csv
```

Scripts

- `essay_corpus.json`: Data corpus created in Data Acquisition assignment.
- `model.py`: The ML model that we use for generating predictions.
- `conf_bias_evaluation.py`: Script to evaluate the `F1` score of the `ML` model.

How to run the scripts

- On a venv install the requirements specified in `requirements.txt`
- Make sure you have the same directory structure as above otherwise adjust the paths in the scripts accordingly.
- Run `model.py` to generate the predictions in `data/` directory with name `predictions.json`
- Run `conf_bias_evaluation` script with the filepath to the `predictions.json` as predictions

Model Explanation

Inspired by `Stab16` we choose a support vector machine (SVM) as a learner. Therefore we used a linear SGDClassifier (`sklearn.linear_model.SKDClassifier`) with two different features (Adversative Transitions and Unigrams) to classify the confirmation bias. These features are described in more detail below. With this

approach of `SVM uni+adv` we achieve a `F1-Score: 0.6875`, evaluated by the `conf_bias_evaluation.py` file.

Feature Selection

In the approach of `Stab16`, the SVM works best in combination with the features `Unigrams (uni)`, `Adversative transitions (adv)`, and `Production rules (pr)`, whereby the `adv` features seem to yield the best results of all these features. At first, we just worked with the `uni` feature. Therefore we used `TfidfVectorizer` (sklearn.feature_extraction.text.TfidfVectorizer) to create Unigrams and achieved a F1-score of already 0.645. Afterward, we tried to increase this score with `adv`. Therefore, like in the approach of `Stab16`, we added 20 different features: We also used 47 adversative transitional phrases that are grouped in the following categories: concession (18), conflict (12), dismissal (9), emphasis (5) and replacement (3). For each of these categories, we added features for the upper and the lower case as well as for their presence in the surrounding paragraph (introduction+conclusion or in the body). But the results were even worse than just the approach with only `uni`. These results from the opposite labels in `Stab16`, so if we have a `confirmation_bias=true` in the paper, our data has a `confirmation_bias=false`. So we did a deeper analysis of the different adversative transition categories to just use the ones that appear more often in essays with `confirmation_bias=true`. As a result, we detected that the concession and conflict phrases are the best indicator for the `confirmation_bias` in our data. Consequently, we just use the phrases of these categories, such that we came up with 12 different phrases.