Computational Argumentation

Data Acquisition Project Documentation

Team: powerpuffboys

Project Overview:

The main objective of this text mining project is to identifying argumentative discourse structures in persuasive essays.

Project Structure:

A Data Acquisition Project is a folder on file system with a project file and a well-defined structure. The most basic layout looks like this:

```
data-acquisition-assignment
code/
    data/
    src/
material/
```

The code folder contains all the sources that are built into this project. This contains data/ and src/ folder. All the data-file that are required for this project and final output of this project <data.json> is stored in the data folder. In src/ folder contains the files <ArgumentAnnotatedEssay.py> and Jupyter notebook file <ArgumentAnnotatedEssayStat> for statistics of this project.

Work-Flow:

1. The workflow implements main operations of this project: create unified one single json file. Execution of this task involved used of data split file. For this purposed we have using <flag> as an indicator. If the flag is set to 1, it will used only the data that have tag 'TRAIN' will be considered. If the flag is set to 2, only the data that have tag 'TEST' are considered and finally if the flag is set to 3 all the data that have tag 'TRAIN' and 'TEST' will be considered.
2. The other implementation of project statistics is done by the Jupyter notebook file. All the statics like:
   - Number of essays, paragraphs, sentences, and tokens,
   - Number of major claims, claims, premises,
   - Number of essays with and without confirmation bias,
   - Number of sufficient and insufficient paragraphs (arguments),
   - Average number of tokens in major claims, claims, and premises,
   - The 10 most specific words in major claims, claims, and premises.

   are shown in Jupyter notebook file <ArgumentAnnotatedEssayStat>.

3. For most specific words: Most specific words are words that appear frequently in the corresponding argument unit type but not in others. That is most specific word for claims are those that appear more often in the claims but not in other units in the text (including non-argumentative units). To compute the most specific words of each of the arguments units we used following method:
   - We fetch all tokens for MajorClaims, claims and premises in individual lists.
   - Then for each category list, we remove all the tokens that occur in either of the other categories.
   - Now, Find the 10 most common words from each category updated list.

- And we have shown it on a Jupyter notebook file.

Implementing:

1. Under the code/src folder consists a Python file <ArgumentAnnotatedEssay.py>. To reproduce the unified data file, we need to run this python file and it will create a unified json file <data.json> in data folder.
2. The statistics of this project are shown in Jupyter notebook file under code/src folder named <ArgumentAnnotatedEssayStat>.