Feedback Prize - Evaluating Student Writing

Team Name: AI 4 life.

Team members:

Hassan Ali Hassan Ibrahim

Mohamed Haroon Saleh

Abdel-Rahman Ibrahim Saad.

problem statement

Writing is a critical skill for success. However, less than a third of high school seniors are proficient writers, according to the National Assessment of Educational Progress. Unfortunately, low-income, Black, and Hispanic students fare even worse, with less than 15 percent demonstrating writing proficiency. One way to help students improve their writing is via automated feedback tools, which evaluate student writing and provide personalized feedback.

Evaluating Student Writing is a competition sponsored by the Bill and Melinda Gates Foundation where NLP researchers get a chance to build a model to classify student writing into its key argumentative components[1]. The goal is to identify writing structures such as thesis statements and support for claims in essays[2].

In this problem, we will identify elements in student writing. More specifically, we will automatically segment texts and classify argumentative and rhetorical elements in essays written by 6th-12th grade students.

Data set decribtion

The dataset contains argumentative essays written by U.S students in grades 6-12. The essays were annotated by expert raters for elements commonly found in argumentative writing.

Task is to predict the human annotations. need to segment each essay into discrete rhetorical and argumentative elements (i.e., discourse elements) and then classify each element as one of the following:

Lead - an introduction that begins with a statistic, a quotation, a description, or some other device to grab the reader’s attention and point toward the thesis

Position - an opinion or conclusion on the main question

Claim - a claim that supports the position

Counterclaim - a claim that refutes another claim or gives an opposing reason to the position

Rebuttal - a claim that refutes a counterclaim

Evidence - ideas or examples that support claims, counterclaims, or rebuttals.

Concluding Statement - a concluding statement that restates the claims

The training set will consist of individual essays in a folder of .txt files, as well as a .csv file containing the annotated version of these essays. It is important to note that some parts of the essays will be unannotated (i.e., they do not fit into one of the classifications above).

Files

train.zip - folder of individual .txt files, with each file containing the full text of an essay response in the training set

train.csv - a .csv file containing the annotated version of all essays in the training set

id - ID code for essay response

discourse_id - ID code for discourse element

discourse_start - character position where discourse element begins in the essay response

discourse_end - character position where discourse element ends in the essay response

discourse_text - text of discourse element

discourse_type - classification of discourse element

discourse_type_num - enumerated class label of discourse element

predictionstring - the word indices of the training sample, as required for predictions

test.zip - folder of individual .txt files, with each file containing the full text of an essay response in the test set

sample_submission.csv - file in the required format for making predictions - note that if you are making multiple predictions for a document, submit multiple rows

Train samples

15645 train sample

purposed architecture

Transformer models.

Evaluation metric

Submissions are evaluated on the overlap between ground truth and predicted word indices.

For each sample, all ground truths and predictions for a given class are compared.

If the overlap between the ground truth and prediction is >= 0.5, and the overlap between the prediction and the ground truth >= 0.5, the prediction is a match and considered a true positive. If multiple matches exist, the match with the highest pair of overlaps is taken.

Any unmatched ground truths are false negatives and any unmatched predictions are false positives.

Example:

Ground Truth

```
id,class,predictionstring
1,Claim,1 2 3 4 5
1,Claim,6 7 8
1,Claim,21 22 23 24 25
```

Prediction

```
id,class,predictionstring
1,Claim,1 2
1,Claim,6 7 8
```

The first prediction would not have >= 0.5 overlap with either ground truth and would be a false positive. The second prediction would overlap perfectly with the second ground truth and be a true positive. The third ground truth would be unmatched, and would be a false negative.

The final score is arrived at by calculating TP/FP/FN for each class, then taking the macro F1 score across all classes.

The word indices are calculated by using Python's .split() function and taking the indices in the resulting list. The two overlaps are calculated by taking the set() of each list of indices in a ground truth / prediction pair and calculating the intersection between the two sets divided by the length of each set.