Executive Summary: Essay annotation with deep learning.

Team: Mariner, Team members:Irati Hurtado, Konstantinos Karatapanis, Sammy Sbiti

Introduction

This project originates from kaggle ("Evaluating Student Writing") [https://www.kaggle.com/competitions/feedback-prize-2021/overview]. We are given a dataset that contains argumentative essays written by U.S students in grades 6-12. These essays were annotated by expert raters for elements commonly found in argumentative writing. The goal of the project is to predict human annotations. It was a very challenging problem that required many techniques, namely nlp techniques to preprocess the data, an appropriate word embedding, and a suitable neural network upon it.

Model description

Each student essay was sliced, by expert graders, into annotated segments, where, namely, the different annotations where, lead, position, evidence, clam, concluding statement, counterclaim and rebuttal of the argument. The task was to perform supervised learning so that we could accurately predict such annotations.

Preprocessing

For the preprocessing part we had to tokezine all the essays. This process was streamlined, after having chosen a trasnformer, by using the corresponding keras tokenizers. Each token received a number from 0 to 7 based on the type of discourse element it belonged to:

- 0 = Lead
- 1 = Position
- 2 = Evidence
- 3 = Claim
- 4 = Concluding statement
- 5 = Counterclaim
- 6 = Rebuttal
- 7 = None of the above

In this stage we also tried to correct the grammar of the essays, however there were many challenges and not enough time to implement them appropriately. Firstly, since the essays were written by young students there were many evolving elements of the language that are not easily accounted for in pre existing databases. Many of these elements originate from the emerging social media nomenclature, as well as the inherent ever evolving nature of language among young people. A candidate spelling corrector we implemented (which can be found in our repository) uses from the nltk module the Jaccardi_distance on ngrams of lenght two for a candidate word (a word not native in the chosen dictionary which in our case was brown corpus), combined with edit_distance in cases where the former couldn't achieve adequate accuracy. These proved most accurate for spelling mistakes of letter rearrangement or missing/adding letters.

Modeling & Learning

The basic form of our model takes as input the list of tokens in a given document, and outputs the class probability vectors for each token. We feed our token to the pre-trained Longformer model from the transformers module as a first layer in a neural network. The Longformer is a pretrained word embedding based off the well known transformer BERT, however its attention mechanism scales linearly with document length as opposed to BERT which scales quadratically, making Longformer useful for analyzing large documents. For context, to obtain the word embedding, via longfomer, using a 7 core 8th gen cpu took 3 hours.

We then feed the word embeddings to a dense layer followed by a classification layer using softmax to output class probabilities for each token. We then use these class probabilities along with certain prior assumptions about the distribution of discourse elements throughout the essays to predict classes at the sentence level.

For future imporovements we wanted our neural network to tune the distribution assumptions of the discourse elements to predict the classes at the sentence level. However, materializing this idea concretely proved to be techniquely very challenging, so we had to hone our intution between the dynamics of the different discourse types, whithin the context and structure of an argumentative essay, to modify the token distributions appropriately.

Furthermore as mentioned the Longformer model came pre-trained and we used it as a word embedding. However, for this to be most successful, the database used for training the Longformer would potentially have to be appropriately selected for this idiosyncratic type of text that is written from very young students.

Performance & App usability

We use the metric provided by Kaggle to evaluate performance. The metric is binary, i.e. it takes the value 1 if the to corresponding annotations are overlapping by more than 50$ of their true lenght. Using that metric we calculate the F1 score for each class and then take the average across all classes. And then it is evaluated according to F1 score which is the harmonic mean of precision and recall.

Running our predictions against our test set, which comprisesd of 1000 essays gaves us the following statistics. Namely the F1-scores for Claim is .084, for Concluding Statement is .32, for Counterclaim is 0.0, for Evidence is .56, for Lead is .42, for Position is .21, for Rebuttal is 0.0, for an overall .23 average across classes.

This problem proved very difficult to achieve high accuracy, as it was not enough to figure the meaning of the text but also how a human grader decides that it is most ideally represented.

We further developed a web app on streamlit that allows users to input their essays and outputs an annotated essay. The annotation were visually represented by highlighting the different annotations with different colors. We hope this can be helpful to students who want quick feedback on the structure of their argumentative writing. We imagine that it could be used as a part of a larger program for providing feedback on students' writing, as well as potentially used for extracting ideas and facts in written text.

Thank you!!

We wish to thank the Erdos Institute, Matt and along with everyone everyone else for the ample support and an excellent learning adventure!