Evaluating Student Writing

Deep Learning - CS 7150
Spring 2022
Course Project - Group 9

Northeastern
Khoury College of
Computer Sciences

Presented By:

- Apoorva Surendra Malemath
- Sravya Burugu

Outline

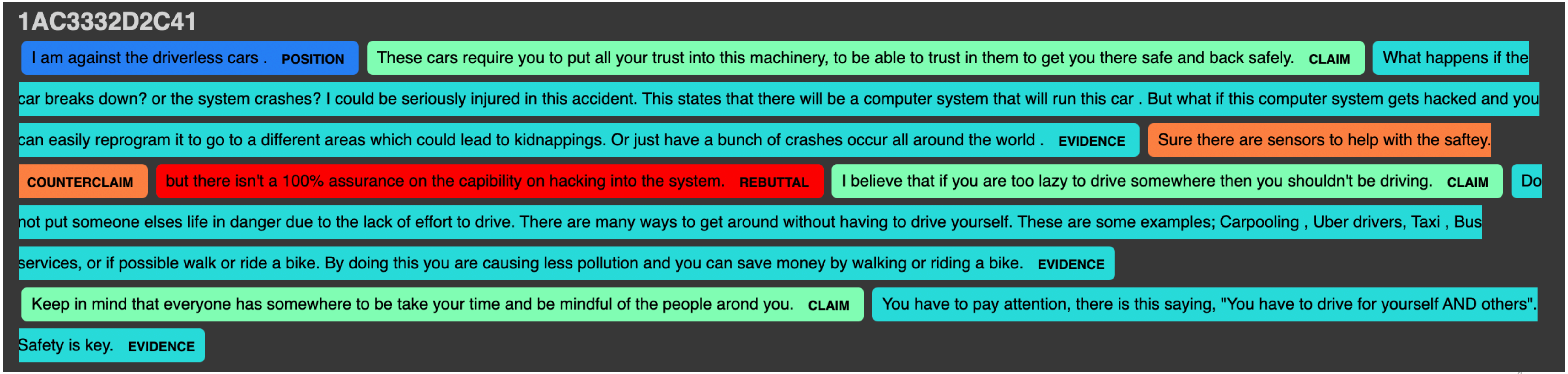
- Introduction
- Understanding the Data
- Exploratory Data Analysis
- Methodology
    - ✓ Longformer
    - ✓ BigBird
- Results
- Conclusion and Future Scope

Introduction

- Named Entity Recognition
- We aim to identify elements in student writing i.e. we segment text and classify argumentative and rhetorical elements.
- We analyze argumentative writing elements from students grade 6-12.
- The scope of the project is to make it easier for students to receive feedback on their writing and increase opportunities to improve writing outcomes.
- The project will allow any educational organization to better help young writers develop.
- The problem statement is hosted by Georgia State University on Kaggle.

Understanding the Data

- The dataset contains argumentative essays written by U.S students in grades 6-12.
- The essays are annotated by expert raters for elements commonly found in argumentative writing.
- Each text segment can be classified in either of the 7 classes i.e.
  1. Lead
  2. Position
  3. Claim
  4. Counterclaim
  5. Rebuttal
  6. Evidence
  7. Concluding


1AC3332D2C41

I am against the driverless cars . POSITION

These cars require you to put all your trust into this machinery, to be able to trust in them to get you there safe and back safely. CLAIM

What happens if the car breaks down? or the system crashes? I could be seriously injured in this accident. This states that there will be a computer system that will run this car . But what if this computer system gets hacked and you can easily reprogram it to go to a different areas which could lead to kidnappings. Or just have a bunch of crashes occur all around the world . EVIDENCE

Sure there are sensors to help with the saftey. COUNTERCLAIM

but there isn't a 100% assurance on the capibility on hacking into the system. REBUTTAL

I believe that if you are too lazy to drive somewhere then you shouldn't be driving. CLAIM

Do not put someone elses life in danger due to the lack of effort to drive. There are many ways to get around without having to drive yourself. These are some examples; Carpooling , Uber drivers, Taxi , Bus services, or if possible walk or ride a bike. By doing this you are causing less pollution and you can save money by walking or riding a bike. EVIDENCE

Keep in mind that everyone has somewhere to be take your time and be mindful of the people arond you. CLAIM

You have to pay attention, there is this saying, "You have to drive for yourself AND others". Safety is key. EVIDENCE

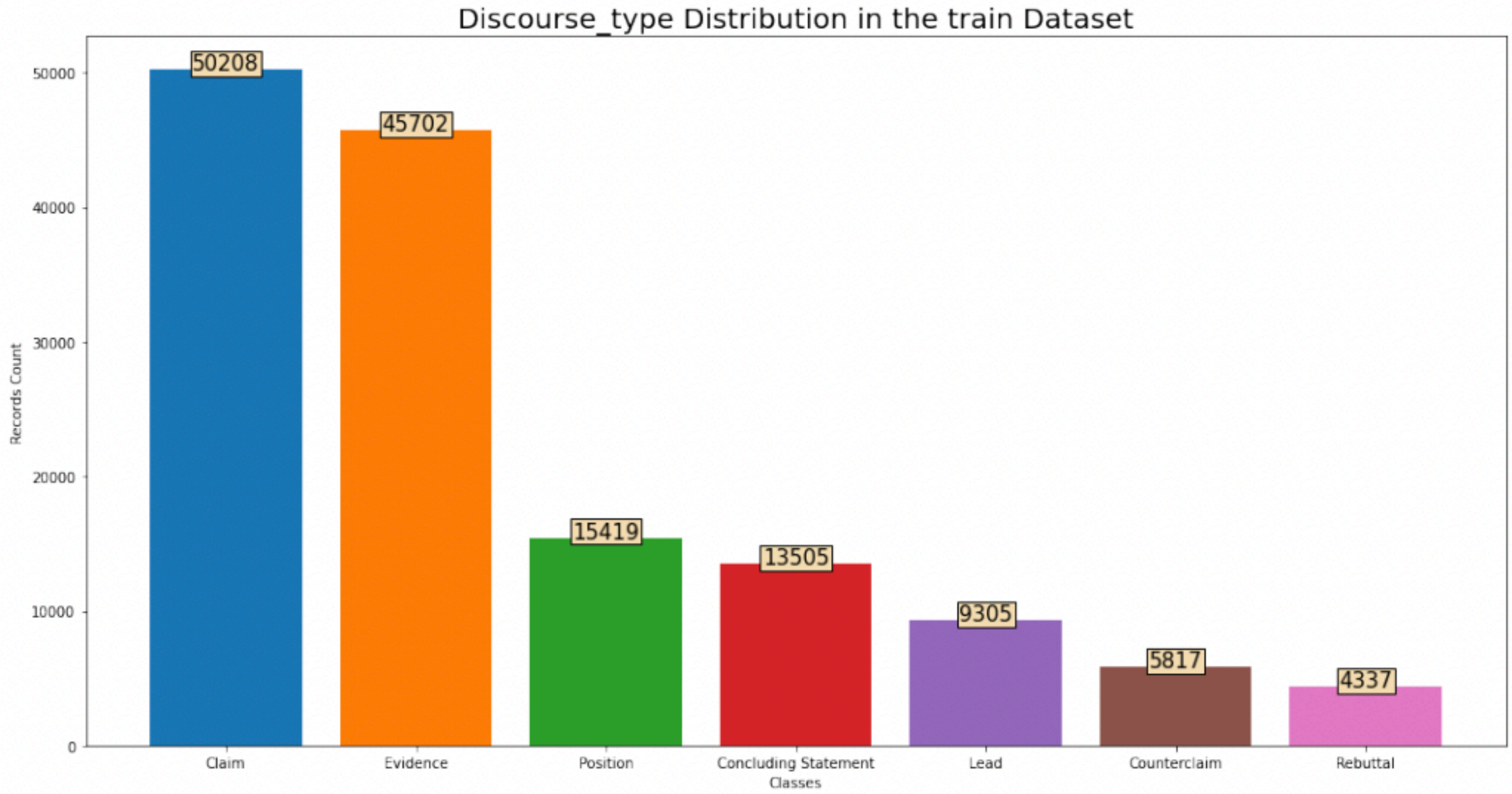
Understanding the Data

- The dataset consists of 15594 files i.e. each file has the contents of one essay.
- Supporting information i.e. the annotation for all the essays in the training set comprises of the following attributes.

id	ID code for essay response
discourseId	ID code for discourse element
discourseStart	character position where discourse element begins in the essay response
discourseEnd	character position where discourse element ends in the essay response
discourseText	text of discourse element
discourseType	classification of discourse element
discourseTypeNum	enumerated class label of discourse element
predictionString	the word indices of the training sample, as required for predictions

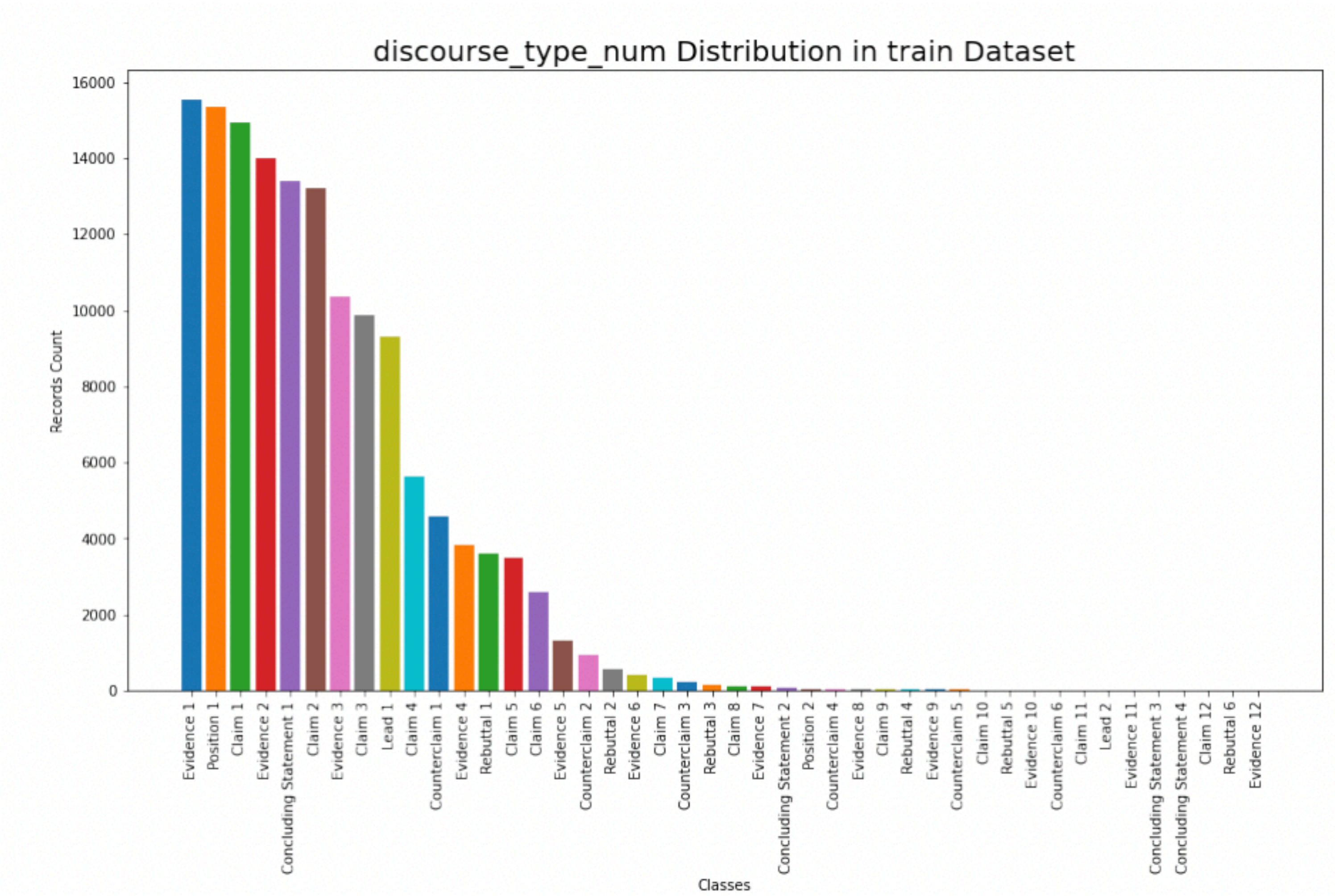
Exploratory Data Analysis

Distribution of the disclosureType



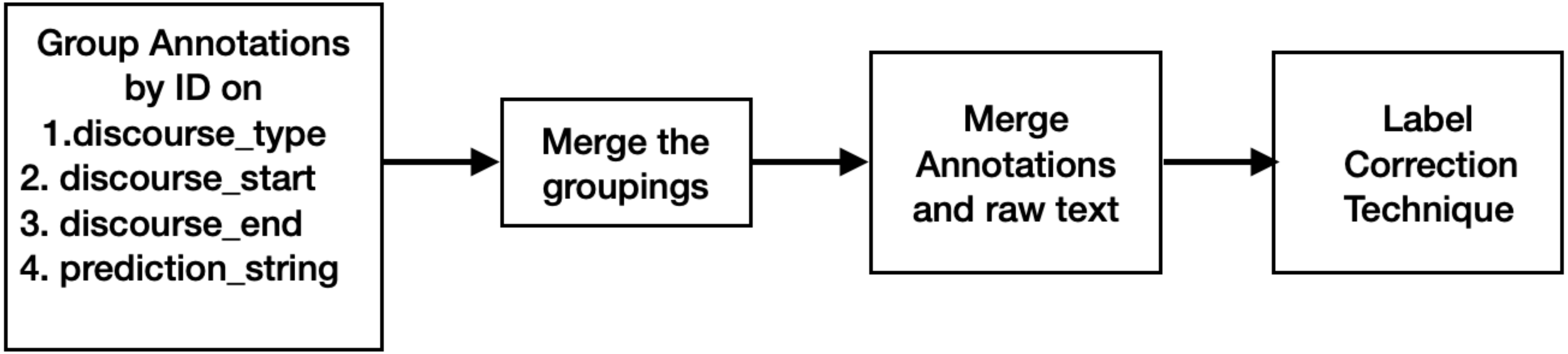
Exploratory Data Analysis

Distribution of the disclosureTypeNum



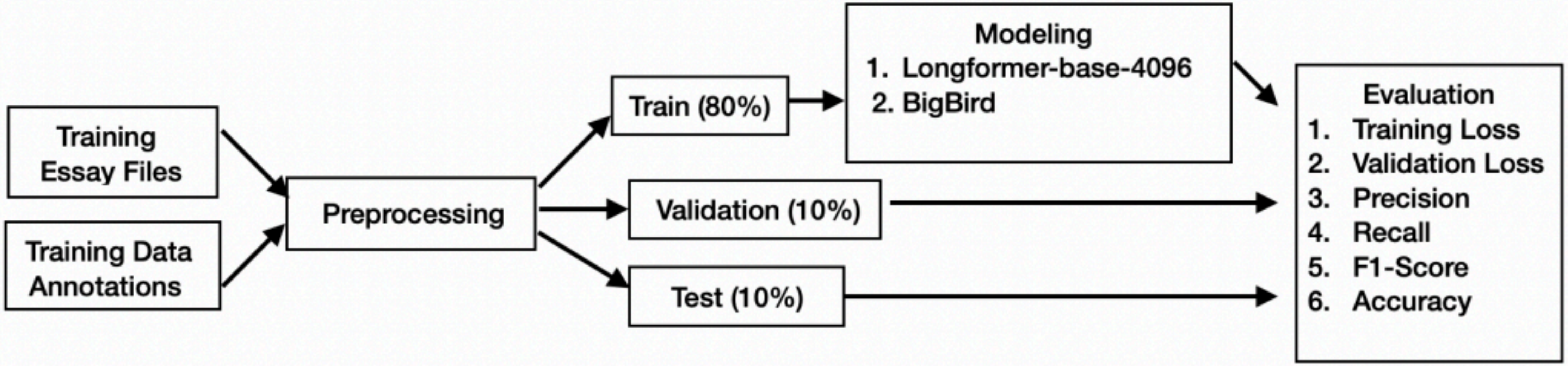
Methodology

Pre-Processing



Methodology

Modeling and Evaluation Pipleline



Methodology

Train Test Split: We have a total of 15594 essays, and we use 80% i.e. 12475 essays of the data as our train data, 10% as the validation and 10% i.e. 3119 essays as the test data.

Post pre-processing:

- Tokenize the data
- Apply the data collection technique from the transformer using DataCollatorForTokenClassification.
- Data collectors form a batch by using a list of dataset elements as input.
- DataCollatorForTokenClassification dynamically pads the inputs and labels.
- We then apply the AutoModelForTokenClassification pre-trained model.

LVX VERITAS VIRTVS

Methodology

Longformer:

- BERT-like model started from the RoBERTa and pretrained for long documents.
- It supports sequences of length up to 4096.
- Longformer uses a combination of a sliding window (local) attention and global attention.
- It is a transformer-based architecture that reformulates the self-attention computation to reduce the model complexity.

- **Longformer Self Attention** : It applies self attention in both local and global context. And it takes individual values for query, key, and value for both local and global attention.

- **Longformer Tokenizer** : It is derived from RoBERTa tokenizer and uses byte-level Byte-Pair encoding.

- **Longformer Model for token classification**: It is a linear layer on top of of the hidden-states output, which is used for Named-Entity-Recognition task. It uses word embeddings, position embeddings and token type embeddings.

LVX VERITAS VIRTVS

Methodology

BigBird:

- BigBird computes attention along the diagonal, sides and a few random places of the matrix.
- BigBird runs on sparse attention mechanism enables it to process sequences of length up to 8x more than what was possible with BERT.
- Attention mechanism is applied token by token, unlike BERT where the attention mechanism is applied to the entire input just once.
- BigBird Model makes use of word embeddings, position embeddings and token type embeddings.

LVX
VERITAS
VIRTVS

Methodology

- Dropping outliers
- Tested with different tokenizers - BERT, RoBERTa, RoBERTa-Large
- The below mentioned hyper parameters are used for both the models.

Maximum Length	1024
Stride	128
Minimum number of Tokens	6
Batch Size	4
Learning Rate	5e-5
Weight Decay	0.01
Gradient Accumulation Steps	8
Warm Up Ratio	0.1
Number of Epochs	5

LVX VERITAS VIRTVS

Results

Evaluation Metrics for Longformer

Epoch	Training Loss	Validation Loss	Precision	Recall	F1-Score	Accuracy
0	0.9516	0.6140	0.1719	0.3052	0.2199	0.8040
1	0.5780	0.5584	0.1889	0.3324	0.2409	0.8144
2	0.4916	0.5522	0.2276	0.3580	0.2783	0.8193
3	0.4248	0.5814	0.2292	0.3766	0.2850	0.8118
4	0.3707	0.5962	0.2297	0.3736	0.2844	0.8132

F1-Scores for each class on Test Data

Claim	0.5571
Concluding Statement	0.8041
Counterclaim	0.4946
Evidence	0.6915
Lead	0.7817
Position	0.6558
Rebuttal	0.3990

LVX VERITAS VIRTVS

Results

Evaluation Metrics for BigBird

Epoch	Training Loss	Validation Loss	Precision	Recall	F1-Score	Accuracy
0	1.0197	0.6547	0.2275	0.3626	0.2796	0.7904
1	0.5944	0.5830	0.2968	0.4009	0.3411	0.8126
2	0.4969	0.5840	0.2778	0.4238	0.3356	0.8110
3	0.4164	0.5955	0.2831	0.4349	0.3430	0.8113
4	0.3574	0.6215	0.2827	0.4383	0.3437	0.8078

F1-Scores for each class on Test Data

Claim	0.5276
Concluding Statement	0.7075
Counterclaim	0.4695
Evidence	0.6525
Lead	0.7941
Position	0.6394
Rebuttal	0.3701

LVX VERITAS VIRTVS

Conclusion and Future Works

- **Conclusion**: We observed that RoBerta tokenizer performed the best. Looking at the evaluation metrics we observe that both LongFormer and BigBird have similar performance on the data.

- **Future Works**:
  - UI for better user experience.
  - Improve the metrics by using combination of LongFormer and BigBird.
  - Build a ML pipeline using MLOps.

Thank you

Open to Questions