Documentation:

We start our program by first gathering id, preceding post, and labels of each thread for the training, validation, and test datasets. Then we have removed punctuation, spaces, and URLs from the text.

We have done feature engineering firstly on individual dialogues to find semantic similarity between each comment of a thread and lastly on the combined the vectors all together.

For individual checking, for each dialogue, we obtain a feature vector. Here, we have used stemming, the length of an argument, the number of insults, part of speech (POS), and sentimental analysis, and merged it into one vector. We considered this vector as a primary vector which would be used later to model the dialogical nature of the conversation.

Stemming: Because stemming is the most significant procedure in NLP, it has been used here. It assists us in retrieving and extracting crucial information from a large text. It will aid in the recognition and retrieval of various types of words that may have been overlooked previously.

Length of an argument: We read about Godwin's Law, which is responsible for the length of the argument used in this case for feature engineering. “It is an Internet adage asserting that as an online discussion grows longer (regardless of topic or scope), the probability of a comparison to World War II gets more likely” [1].

Number of insults: Common insults are the next feature used here. We checked for four straightforward insults (ass, idiot, fuck and shit) in the text which are plainly detectable. The list of insults in the list of foul words can be modified at any time. Ad hominem attacks are more likely when there are a greater number of insults.

Parts of Speech Tags: We have done POS and counted the number of verbs, nouns, pronouns and modal verbs. It aids in the derivation of hidden linguistic qualities. These POS can identify parts where ad hominem can be used. Like pronouns like ‘you or your’ is used while employing ad hominem technique. Similarly, verbs, nouns and modal verbs can be used in identifying ad hominem as proposed by Stab and Gurevych (2014b). For Eg: “Could you be dumber?” has a modal verb and a pronoun.

Sentiment Analysis: Following that, sentimental analysis was employed. It assists in determining the author's mood. Polarity and subjectivity are two characteristics of sentimental analysis. Subjectivity provides information about personal emotions, judgments, opinions etc. Polarity tells us whether a statement is negative or positive. If the author is utilizing ad hominem, it helps us comprehend by looking at the author's feelings.

Our approach to model the dialogical nature of the conversation consists of combining these feature vectors (for individual dialogues) to generate one secondary feature vector for each thread in the given dataset. In this vector we calculated the cosine similarity, average of polarity, if the polarity is increasing in the next comment, average number of insults, if the number of insults is increasing in the next comment.

Cosine Similarity: We calculated the cosine similarity between the feature vectors of each dialogue in the post. It helps us in determining how similar or distinct two dialogues are. Based on the features (mainly polarity, subjectivity, insults, number of POS tags) we calculated for individual dialogues, we can estimate if two dialogues have similar or contrasting point of views. It was our approach to estimate “semantic similarity” between two dialogues as mentioned in Habernal et al(2018) and Stab and Gurevych (2014b).

Polarity: We have employed polarity averages. The average aids in determining the mid-value, or the overall tendency of the conversation. It means that if the trend +0.5, it indicates a positive trend, and if it -0.5, it indicates a negative trend. We have also looked at the increase and decrease in polarity.

Insults: We have counted the average number of insults in each thread. As the dialogues which leads to ad hominem have high chances of having insults, we have employed this feature. We have also checked the increase and decrease in number of insults between the two individual dialogues.

Finally, we have combined the secondary vectors (for each thread) with Bag of Words (BOW).

BOW is used here as it aids in finding frequency of words. The words which help out in finding ad hominem.

After doing feature engineering, we have passed the feature vector to the SVC classifier.

Generating Output File:

After classification is completed, we output the evaluation result on the console.
Then we take the predicted values and map it to the text id and write them in the desired format to ***<dataset>_result.json*** file.

PS: The execution time on our local machines was 10-12 minutes. If the test set is larger than the validation set it may require more time to execute. Please be patient.

Steps:

To run classifier:

```
python arg_assessment.py -x <TRAINING SET> -y <VALIDATION SET> -z <TEST SET>
```

Output generated by the classifier in for validation data: ***val_result.json***
Output generated by the classifier in for test data: ***test_result.json***

References:

[1] https://en.wikipedia.org/wiki/Godwin's_law#External_links

[2] Stab, C., & Gurevych, I. (2014). Identifying Argumentative Discourse Structures in Persuasive Essays. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. doi:10.3115/v1/d14-1006

[3] Habernal et al. 2018: “Before Name-calling: Dynamics and Triggers of Ad Hominem Fallacies in Web Argumentation”