Sl.No	Matriculation Nr.	Name	Email address
1	6868664	Akshit Bhatia	abhatia@mail.uni-paderborn.de
2	6896703	Harshith Srinivas	harshith@campus.uni-paderborn.de
3	6852329	Suraj Manjunatha	surajm@mail.uni-paderborn.de
4	6866035	Vinaykumar Budanurmath	budanurm@mail.uni-paderborn.de

Table of contents:

1. Goal:

Given an essay recognize whether the confirmation bias is true or false for the essay.

2. Running instructions:

Environment:

Dependency	Version
Python - 3.7.3	3.7.3
TextBlob	0.15.3
Tensorflow	2.2.0
Tensorflow-estimator	2.2.0
Tensorflow-hub	0.8.0
tensorboard-plugin-wit	1.6.0
Tensorboard	2.2.2

When ".zip" file is extracted then two folders are seen.

- Code folder: This folder contains the python notebook file which has the code to execute. This file requires as input, the files "essay_corpus.json" and "train-test-split.csv" for its execution. These input files are placed in the data folder.

- data folder: Below table shows the contents of this folder:

Sl no	File name	Description
1	essay_corpus.json	Corpus
2	train-test-split.csv	Train test split
3	raw_polarity_test.txt raw_polarity_train.txt test_essay.json train_essay.json predictions.json	All these are generated files, which are generated when the python code is executed. The file predictions.json contains the predictions which should be used as input for the evaluation.

3. Features explained:

Definition of Confirmation Bias:

When an essay *has opposing arguments,* then confirmation bias flag would be "True".

When an essay *does not have opposing arguments,* then confirmation bias flag would be "False".

After reading through the essays we came up with the following plan:

We planned to recognize the confirmation bias by answering the following questions:

1. Whether Claims are opposing the Major Claims?
2. What is the stance of each claim with respect to the major claim?
3. Whether Claims within themselves oppose each other? Whether there is an opposing claim in the essay?
4. Which claims speak about an opposing topic?
5. What is the sentiment of the premises, towards the claims or the major claim?
6. How to associate a premise with a specific claim?
7. How to associate the claims with the sentiment of the premises?
8. How to weigh sentiment of the opposing claims?
9. Whether the paragraphs contain the adversity words such as "On the other hand", "however", "but", "not only" etc., ?

The table below lists all the features. Each of the features answer a combination of the questions listed above.

sl. No	Feature name	Brief Description of the feature	Questions answered from above list
1	majAndclaimSimilarity	This value indicates the semantic similarity between major claim and all claims.	Answers Question 1 and partly Question 2
2	minClaimSimilarity	This value indicates the most "semantically dissimilar" claim from all the other claims. This can be found by calculating the	Answers Question 3 and partly Question 4

		minimum similarity within all the claims	
3	PremiseSent_Polarity	This value indicates the overall sentiment of all the premises of an essay	Answers Question 5
4	weightedSimilarity	This is an experimental value. Our goal was to use a mathematical operation to relate minimum claim similarity and premise sentiment.	Partly Answers Question 6,7,8
5	Presence or absence of adversity words such as: “on_the_other_hand” “however” “but” “nor” “not_only” “moreover” “nevertheless” “though” “yet” “either” “therefore” “consequently” “admittedly” “argue_that”	This value is a Boolean value which indicates the presence and absence of the adversity words	Answers Question 9

3.1 Feature 1: majAndclaimSimilarity

This feature attempts to answer the following questions:

1. Whether Claims are opposing the Major Claims?
2. What is the stance of each claim with respect to the major claim?

To answer these questions, it is necessary to understand the semantic relationship between the sentences to recognize the intention of the author. So, we decided to find the semantic similarity between the “major claim” and “all the claims”.

Given an essay this similarity value, ideally would be a higher score, if the claims and major claim have the following characteristics:

- The claims and the major claims discuss about the same topics
- The claims do not have very varying set of words vs the words used in the major claim.
- Etc.,

To calculate the semantic similarity, we have used “Universal sentence encoder”. Unlike embedding models like word2vec, Glove, etc. which provide a relationship score for a bag of words, the “Universal Sentence Encoder” from TensorFlow Hub gives the semantic similarity scores between the sentences. Irrespective of the length of the sentences, the “embed()” function of the “Universal Sentence Encoder”, would output a vector in 512 dimensions.

```
embeddings = embed( [ majorClaimText, claimText ] )

vA = embeddings[0]

vB = embeddings[1]
```

The above statements would give embeddings having two vectors – one for each item in the input list. Once we obtain the vectors, we calculate the similarity between these vectors.

majAndclaimSimilarity = 1 - spatial.distance.cosine(A, B)

With the above calculation, we understand the semantic similarity between "major claim" and "claim".

It is noted that, with this attempt we are not able to precisely indicate the stance of a claim towards the major claim, but the value calculated with this step, when combined with the features 2, feature 3 and feature 4 explained below, helps the classifier to subtly point towards the stance of the claims towards the major claim.

3.2 Feature 2: Minimum claim similarity

This feature attempts to answer the following questions:

3. Whether Claims within themselves oppose each other? Whether there is an opposing claim in the essay?
4. Which claims speak about an opposing topic?

To answer these questions, it is necessary to understand the semantic relationship among all the claims. So, we decided to find the semantic similarity between "all the claims".

To find the most "semantically dissimilar" claim from all the other claims, we must calculate semantic similarity score within all the combinations of the claims and finally chose the minimum value.

For example,

If there are 3 claims – c1, c2 and c3, then we should know similarity of (c1, c2), (c1, c3), (c2, c3).

Once we find this, we must get the minimum value among the calculated values.

To achieve this, we decided to calculate the similarity correlation matrix for the claims of each of the essay and get the minimum value of the matrix.

To calculate the semantic similarity, we have used "Universal sentence encoder" as used for the feature number 1 mentioned above.

As mentioned above, the function "embedding=embed(list_of_sentences)" would take as an input the list of sentences for preparing the embeddings.

In this case we input all the claims of an essay to the embed function. If the length of the list input for the "embed()" function is 3, then this function returns three vectors. Using these three vectors, we can use the below numpy function - "np.inner(embedding, embedding)", to calculate the correlation matrix. The output of this function would be a 3*3 matrix of the similarity scores. We can then extract the minimum value of this matrix using numpy's "np.min()" function.

Please see below an example output:

```
Essay id:  365 has 2 claims. So, the below matrix is a 2x2 matrix

[[0.99999976,0.0344076]
 [0.0344076,1.0000001]]
```

```
Essay id:  134 has 3 claims. So, the below matrix is a 3x3 matrix.

[[1.0000001, 0.11030761, 0.40365237]
 [0.11030761, 1.0000007, 0.25565603]
 [0.40365237, 0.25565603, 1.0000004]]
```

Given an essay this similarity value, ideally would be a low score, if the claims the following characteristics:

- A claim, with respect to another claim, has a very varying set of words vs the words used other claims.
- Etc.,

However, it is noted that, with this attempt we are not able to precisely indicate whether any of the claim is opposing major claim, but the value calculated with this step, when combined with the feature 1 (explained above), feature 3 and feature 4 (explained below), helps the classifier to classify an essay into one of the two categories of "conf_bias" true or false.

3.3 Feature 3: Sentiment polarity of the premise

Essays are monological arguments. The premises in an essay will either support or attack the claims of the essay.

We assume that, the act of '*support*' or '*attack*' is always tightly coupled with sentiment of the author/speaker. So, whenever a person speaks about an aspect, then the person expresses '*support*' with positive sentiment and on the other hand, expresses '*attack*', with negative sentiment. We based our solution on this assumption.

Since we knew that, the premises in the essay corpus were used as reasons to support or attack the author's claim, we planned to quantify the sentiment of each premise.

For example:

An essay has the following components: Introduction, major claim, claim, premises and conclusion.

We focus on the premise component of each of the essays, to quantify the overall sentiment of the author towards the claim and major claim.

Consider an essay has 5 premises:

Premise 1: support statement (a positive sentiment) – sentiment value – s1

Premise 2: Attack statement (a negative sentiment) – sentiment value – s2

Premise 3: support statement (a positive sentiment) – sentiment value – s3

Premise 4: support statement (a positive sentiment) – sentiment value – s4

Premise 5: support statement (a positive sentiment) – sentiment value – s5

With this we calculate the overall impact of the premises by adding the quantified sentiment value of each of the premise.

Therefore, overall sentiment of the essay is: s1 + s2 + s3 + s4 + s5

In our code we have used the TextBlob library of Python to get the sentiment value.

Concrete example:

Let us consider two statements with opposite sentiment polarity.

positiveSentence = "excellent social skills means best customer satisfaction."

negativeSentence = "worst social skills means very bad customer satisfaction."

When we feed these two sentences to the TextBlob library we get the output as shown in the below image:

```
1
2  from textblob import TextBlob
3  positiveSentence = "excellent social skills means best customer satisfaction."
4
5  blob = TextBlob(positiveSentence)
6
7  for sentence in blob.sentences:
8      print("Positive statment sentiment polarity:",sentence.sentiment[0])
9
10 negativeSentence = "worst social skills means very bad customer satisfaction."
11
12 blob = TextBlob(negativeSentence)
13
14 for sentence in blob.sentences:
15     print("Negative statment sentiment polarity:",sentence.sentiment[0])
16
17
```

```
Positive statment sentiment polarity: 0.6777777777777777
Negative statment sentiment polarity: -0.6255555555555555
```

Concrete examples from the essay corpus:

Premises of Essay 365 – confirmation bias: False	Sentiment Value
Buses and trains have been phased in the streets for quite a long time and have proven themselves the best candidates among all options	0.475
a bus can carry no less than 30 people, while a train can carry as much as nine or even ten times of that number	0.09444444444444444
they will lessen the crowded image in most cities nowadays	0.5
Not to mention the fact that highway train is operated underground, which leaves the street above with buses and non-fuel-based vehicles	0.0
huge centers of attention distributed equally in a wide region means that residential areas will come along thus exert less pressure on the public facilities	0.03333333333333335

if some of the enormous shopping malls or universities were relocated in the commuter belt, the city street would be les burstling and overly-teemed as it is now	0.0

Premises of Essay 134 – confirmation bias: True	Sentiment Value
It is of importance that workers should be well educated and trained to perform duties of their job	0.0
people who work in professional fields such as medicals are required to have a university degree and years of experience	0.05
Social skills seem to be less important as they cannot contribute to the success of a surgery	0.14166666666666666
organization have been transformed to customers oriented	0.0
Excellent social skills can help employees maintain strong relationship with their clients which thus lead to a better customer satisfaction	0.4916666666666667
people nowadays are required to work as a team in their jobs frequently	0.1
Interpersonal skill plays an essential role in helping colleagues to cooperate with each other, especially when conflicts arise	-0.041666666666666664 (NEGATIVE value)
Better social skills allow them to resolve disagreements in a much more harmony way	0.3444444444444444

Premises of Essay 388 – confirmation bias: True	Sentiment Value
fewer guns available mean less crime	-0.02638888888888888
this is not as simple as it sounds	0.0
most gun violence is committed with guns obtained illegally	0.0
People who are intent to commit crime and control cities through gang violence and other means will find a way to continue to use guns, regardless of gun control	-0.125
the root of violence is criminal mind not the weapon itself	-0.4
there will be law non-abiding people anywhere in the world, either polices or common citizens	-0.3

With the above example, one may conclude that if the essay topic is in itself negative (by using words such as crime, guns,), then the overall sentiment will be negative. But the below explanation will clarify this doubt.

Let us consider the first premise of Essay 388 above.

"fewer guns available mean less crime". This statement has the negative score as indicated above.

But now if the sentence is changed to

"fewer guns available mean ***no*** *crime"* then the value is positive: 0.04375000000000001.

Therefore, with these findings we conclude that the way the words are framed decides the sentiment and the overall polarity of the premises for 'support' or 'attack' action. It is also observed that, the

quantified values though subtle sometimes, but when combined with the other features helps the classifier to categorize the essay into its appropriate category.

3.4 Feature 4: weightedSimilarity

As explained in the above features, the features are not very precise in their goals. So, we wanted to come with a value which can increase the precision of the above-mentioned features. Our goal was to use a mathematical operation to relate most opposing argument with and sentiment of the essay.

With a hit and trial method we were able to come up with the below formula which can increase the overall accuracy of the classifier.

weightedSimilarity = (minClaimSimilarity) / (100*overallpolarity)

Other math operations which we had tried were addition and multiplication. Since these were decimal values division made more sense.

Our assumption with this calculation is that given an essay, more sentiment of an essay is utilized to emphasize the opposing claim.

3.5 Feature 5: Presence of adversity indicator words

After doing some statistics of the “adversity” indicator words in the essay corpus, we have observed the statistics as shown in the below image (Figure 1). We used few of these words to indicate the presence of the word as another feature.

```
but : 402
however : 235
nor : 67
not only : 96
moreover : 85
on the other hand : 57
disagree : 35
nevertheless : 35
though : 140
either : 15
yet : 12
do not : 105
not : 916
therefore : 159
consequently : 31
admittedly : 26
argue that : 44
```

Figure 1. Statistics of adverse or positive words

4. Classifer:

We used SVM classifier for our task. Please see below part of the screen shot of the feature table that is fed to the classifier.

X_Train input for the SVM

```
In [8]: 1 X_train.head()
```

Out[8]:

	majAndclaimSimilarity	minClaimSimilarity	PremiseSent_Polarity	weightedSimilarity	on_the_other_hand	however	but	nor
0	0.337829	0.034408	0.275694	0.001248	False	False	True	True
1	0.258606	0.110308	0.181019	0.006094	True	False	True	False
2	0.564156	0.056060	0.094975	0.005903	False	True	False	False
3	0.565267	0.021634	0.120370	0.001797	False	False	True	False
4	0.640624	0.245322	-0.011905	-0.206070	False	True	True	False

5. Evaluation:

We fed the feature table to SVM classifier. Based on the above features we are able to achieve the below score.

	precision	recall	f1-score	support
FALSE	0.84	0.82	0.83	51
TRUE	0.70	0.72	0.71	29
accuracy			0.79	80
macro avg	0.77	0.77	0.77	80
weighted avg	0.79	0.79	0.79	80

The python code generates a file called “predictions.json”. This file has “encoding=utf-8”. This file is used as input to run the evaluation python file. When we ran the evaluation python file with this file as input, we have observed a score of 0.712

Please note: When running the evaluation python file we faced the problem of charmap encoding. So we changed the line number 30 of the evaluation python file to:

with open(ESSAYS_PATH, "r",**encoding='utf-8'**) as f:

6. Conclusion:

This approach tries to quantify the sentiment, semantic similarity and calculate the minimum semantic similarity among the claims. Since all these features are not very precise, this approach finally relates some of these with a mathematical operation.

We realize that this approach is not very precise in the main tasks such as – calculating “stance” of a claim and pointing the opposing argument, but when the features are combined, we get very good results.