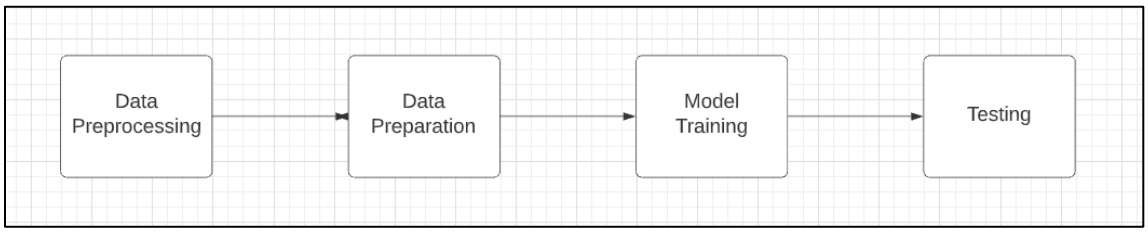
1. Baseline Architecture

We have built two architectures for the two subtasks. For both the tasks we designed two different data preparation strategy. After preprocessing of the data, it is passed through the data preparation stage where we get the data in a form which can be further used to input into ML models to train the model.



As part of **data preprocessing** following techniques are used:

1. Remove any punctuations or limited set of special characters like, or. or # etc.
2. Convert the words to lowercase
3. Remove digits and words which contains digits and remove Stop words

Task-1 – Span Identification Task:

Approaching the Span Identification task as tagging the tokens at sentence level using two tags Propaganda(P) and Non-Propaganda (NP). For the baseline model, the propagandistic span which is present across two different sentence is taken as two different spans with proper tagging.

Data Preparation

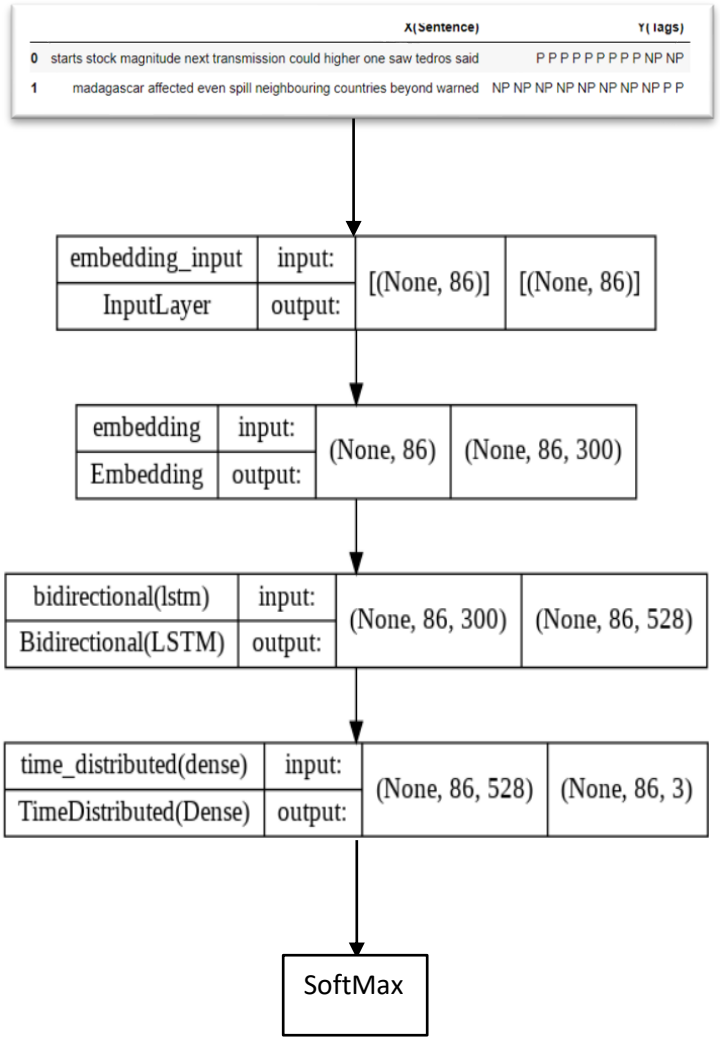
The tagged dataset at sentence level is created as part of the data preparation stage using following steps:

- Divide a news article into sentence(X) and tags(Y)
- Each Data point will have multiple words in the input sequence
- Each word will have its corresponding tag in the output sequence

	X(Sentence)	Y(Tags)
0	starts stock magnitude next transmission could higher one saw tedros said	P P P P P P P P P NP NP
1	madagascar affected even spill neighbouring countries beyond warned	NP NP NP NP NP NP NP NP P P

Model Architecture

The prepared dataset is then used to train Bi-directional LSTM model.



Steps:

1. Encoded X and Y
2. Padded X and Y using the max sequence length. The padded zeros are added at the end.
3. Created embedding matrix to initialize embedding layer using pretrained model "**GoogleNews-vectors-negative300**".
4. **Model Architecture**
   a. First layer is input layer
   b. Second layer is embedding layer which converts each word into 300 dim vectors
   c. Embedded vectors are passed to Bidirectional Layer with 264 units
   d. Time distributed layer is used to get output from each units
   e. Output will be three probabilities for NP, P, Padded Words
5. **Model Configuration**
   Optimizer – **Adam**(lr=0.01)
   Loss – Categorical cross entropy
   Metric – F1 score

Post-Processing:

We developed an algorithm to create spans from the predicted output from the model. At this stage we are creating spans by labeling all tokens between the first sentence-wise 'P' label and the last sentence-wise 'P' label as propaganda.

Task-2 – Technique Classification Task:

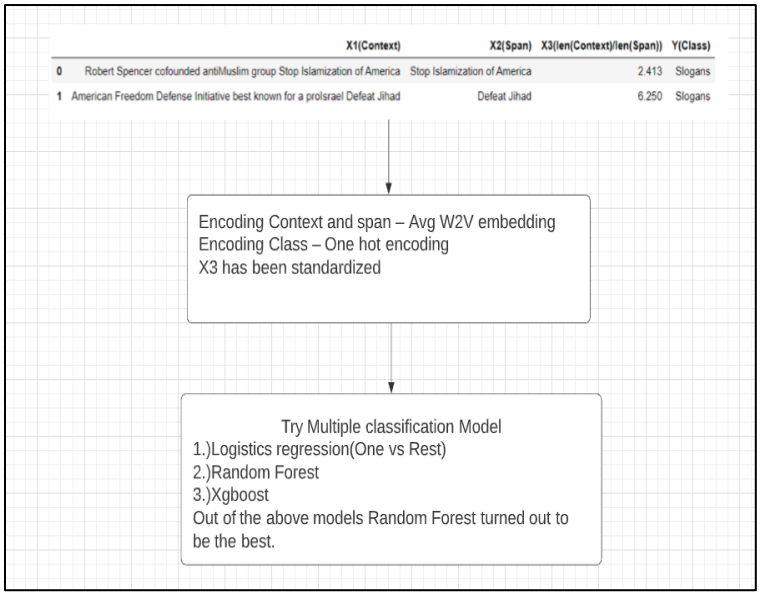
We approached this task as multi-class classification task with 14-classes. As features we used context, the span present in the context and ratio of length of context and length of span.

Data Preparation

- Divide a news article into context(X1), Span(X2), Len(context)/Len(span)(X3) and Class(Y)
- For the baseline model we considered the sentence in which the span is present as context. This to avoid influence of lot of contexts to the span where the propaganda is present.

	X1(Context)	X2(Span)	X3(len(Context)/len(Span))	Y(Class)
0	But Tedros voiced alarm that "plague in Madagascar behaved in a very, very different way this year."	a very, very different	4.25	Repetition
1	Pamela Geller and Robert Spencer co-founded anti-Muslim group Stop Islamization of America.	Stop Islamization of America	3.00	Slogans

Model Architecture



Steps:

1. The prepared dataset has two text features which is converted to vectors using Average word2vec method.
2. "**GoogleNews-vectors-negative300**" is used to convert word into 300 dim vectors, which is then averaged to get vector for the sentence.
3. The numerical feature "Ratio" is standardized
4. Amongst the multiple model we tried Random Forest gave the best performance.
5. GridSearch Cross validation is performed for hyperparameter tuning to get the best value of depth of the tree and number of estimators for the random forest model.

2. Results

To measure the performance of our model on dev-articles (development dataset) we used **task-TC_scorer.py** and **task-SI_scorer.py** which are provided along with the dataset**.** Below screenshots are from the output of the scorer:

2.1. Span Identification Task –

```
2022-03-27 10:28:34,456 - INFO - Checking user submitted file
2022-03-27 10:28:34,836 - INFO - Scoring the submission with precision and recall method
2022-03-27 10:28:35,044 - INFO - Precision=198.213604/1064=0.186291     Recall=329.823395/940=0.350876
2022-03-27 10:28:35,044 - INFO - F1=0.243370
```

Table2.1

Task	Precision	Recall	F1-Score
Span Identification	0.1862	0.3508	0.2433

2.2. <u>Technique Classification Task</u> –

```
2022-03-27 10:33:31,313 - INFO - Checking format: User Predictions -- Gold Annotations
2022-03-27 10:33:31,313 - INFO - OK: submission file format appears to be correct
2022-03-27 10:33:31,355 - INFO - Scoring submission
F1=0.409219
Precision=0.409219
Recall=0.409219
F1_Appeal_to_Authority=0.08219178082191782
F1_Appeal_to_fear-prejudice=0.14814814814814814
F1_Bandwagon,Reductio_ad_hitlerum=0.12500000000000003
F1_Black-and-White_Fallacy=0.1142857142857143
F1_Causal_Oversimplification=0.058823529411764705
F1_Doubt=0.2482758620689655
F1_Exaggeration,Minimisation=0.23529411764705882
F1_Flag-Waving=0.5350318471337581
F1_Loaded_Language=0.6442432082794307
F1_Name_Calling,Labeling=0.46786632390745503
F1_Repetition=0.04878048780487805
F1_Slogans=0.045454545454545456
F1_Thought-terminating_Cliches=0.1111111111111113
F1_Whataboutism,Straw_Men,Red_Herring=0.05714285714285715
```

F1-score = 0.409219

Precision = 0.409219

Recall = 0.409219

Class	F1-score	Class	F1-Score
F1_Appeal_to_Authority	0.0821	**F1_Flag-Waving**	0.5350
F1_Appeal_to_fear-prejudice	0.1481	**F1_Loaded_Language**	0.6442
F1_Bandwagon,Reductio_ad_hitlerum	0.125	**F1_Name_Calling,Labeling**	0.4678
F1_Black-and-White_Fallacy	0.1142	**F1_Repetition**	0.0487
F1_Causal_Oversimplification	0.0588	**F1_Slogans**	0.0454
F1_Doubt	0.2482	**F1_Thought-terminating_Cliches**	0.1111
F1_Exaggeration,Minimisation	0.2352	**F1_Whataboutism,Straw_Men,Red_Herring**	0.0571

<u>Table2.2</u>