Hate, did you fake it? Deep learning model with multitasking for the detection of fake content in Code-Mixed hate speeches.

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
Dharwad
ज्ञानेन विकासः

CSE ,2018-2022

MAJOR PROJECT

Group Members:

V Harsha Vardhan -18bcs118

Rama Dundi Saketh -18bcs076

Pokala Dattatreya -18bcs067

Sampath Sakala -18bcs087

Under Guidance of:

Dr. Sunil Saumya
Asst. Professor
IIIT-Dharwad.

Problem Statement

Problem Statement:

- Detection of fake,hate news that are widely spreading these days by people using social media platforms and other networks to share their views on many topics. While sharing their views,they includes fake,hate news to gather more attention towards it which effects many individuals or organizations.
- Because of huge data manual verification is difficult.So,we can use AI and DL models to overcome this difficulty.

Context & Motivation

Hate Speech

- Initially hate speech is not defined in any law in india.
- But later law commission of India on 23 March,2017 in its 267th report stated that "Hate speech is any word written or spoken,signs,visible representations within the hearing or sight of the person with the intention to cause fear or alarm,or incitement to violence".
- Hate speech can be derived as any sort of communication which derogates any person or a community based on nationality, race, caste, gender, ethnicity etc...
  - *Tell those idiots to go hell!*
  - *That women should mind her own business.*

Fake News

- **Fake news is the news which is not based on truths and facts.**
- **News plays an important role in our day to day life and that is why fake news can create a significant problem to our society.**
- **Ex:**
  - ***Virat is dropped from T20 world cup.***
  - ***10 rupee coin is no more a legal tender.***

- **The laws that exist or not powerful enough to stop hate content spreading through web and of course it is tough to identify & catch people behind it.**

- **There we had Automatic hate speech detection techniques to find such speeches easily which are in single language.**

- **But in a diverse country like India having around more than 24 recognised languages the hate content would be in code-mixed form which is little challenging.**

- **Hindi being most speaking language in india usually most of hate speeches are in Hindi-English (Hinglish) combination.**

Motivation

- By using modern technologies like AI and ML it is possible to detect and control the hate and fake news increasing drastically.
- Although there are some implementation of AI and ML models where they are specific to either hate speech or fake news or specific to single language.So,this type of implementations consume more resources and time.
- It is benefit to have a single model which can do multi-task for prediction.

Dataset

Dataset

- Datasets were not found for code-mixed hate speeches labelled for fake,hate, sentiment classes.
- Then started finding datasets for each class of data required individually.
- And finally found datasets for all three required classes in various platforms.

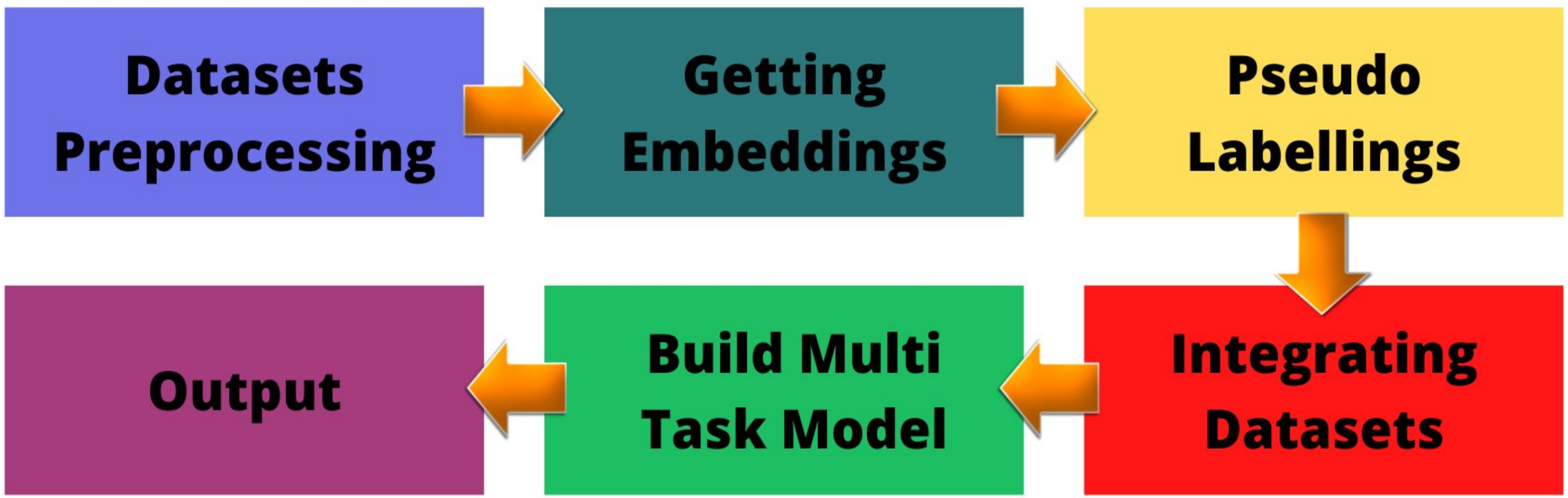
Dataset

Dataset	size(# of samples)	#of classes	modality	file format	Source
Fake news	2012	2	Hindi text	csv	Google scholar
Hate news	4580	2	Hinglish text	csv	Git-hub
Sentiment	3877	3	Hinglish text	txt	Git-hub

Description of Datasets

Proposed Methodology

Proposed Methodology



Data Pre-Processing

NOT FAKE	FAKE
HATE	NOT HATE

Dataset Transliteration:

- **Code-mixed data which we found for fake data is in Hindi text.So In order to make all the documents being processed to be in same language(english text) it was transliterated using Indic NLP.**

Dataset Transliteration:

- **Code-mixed data which we found for fake data is in Hindi text.So In order to make all the documents being processed to be in same language(english text) it was transliterated using Indic NLP.**

Dataset Transformation:

- **Code-mixed data which we found with sentiment labels is in txt file format.It was transformed into csv format.Thus,dataset will be structured and accessible for further process.**

Datasets Balancing:

- All the datasets found for fake,hate are imbalanced but sentiment we found balanced.
- Generally for imbalanced datasets models get biased towards majority class and minimizes the accuracy.
- So to maximize accuracy we used SMOTE oversampling technique to balance the datasets.

DATASET	BEFORE	AFTER
FAKE	(0,761) (1,1250)	(0,1250) (1,1250)
HATE	(0,2918) (1,1661)	(0,2918) (1,2918)

- **Thus,we had balanced datasets for fake ,hate and sentiment after balancing fake and hate datasets.**

Fakedata	label
bhaarata ke videsha ma.mtraalaya paakistaa raajaduuta bula paa..	0
bihaara taarakishora prasaada biijepa vidhaayaka dala chu ke ...	1
...	...

Hatedata	label
Knowing ki Vikas kitna samjhata hai Priyanka aur Itch Guard Luv..	0
I am Muhajir .. Aur mere lye sab se Pehly Pakistan he .. agr	1
...	...

Stmdata	label
baa.mglaadesha kii shaanadaara vaapasii, bhaarata ko 314 rana ..	0
saba ra.mDii naacha dekhane me vyasta jaise hii koii #shaa...	0
...	1

Embeddings and Pseudo labeling.

Embeddings:

- Embeddings for all three datasets have been generated using mBert and XLM-R based sentence transformers.
- Sentence transformers maps sentences to 768 dimensional vector space.
- Thus we get shape of embeddings like (no:of sent,768) and embeddings of type numpy.

```
In [18]: embeddings_f_xlm

Out[18]: array([[ 0.49724126,  0.62162393,  0.50529695, ..., -0.07838535,
          -0.18122661, -0.06042475],
         [ 0.04995521,  0.33718002, -0.3626911 , ..., -0.06306948,
          -0.21589772,  0.06186014],
         [ 0.31021848,  0.31006628,  0.6642327 , ..., -0.05180714,
          -0.3390372 ,  0.06049879],
         ...,
         [ 0.5168515 ,  0.17653647,  0.09194178, ..., -0.24003209,
          -0.46960288, -0.07383722],
         [-0.10863463,  0.25409207,  0.5928139 , ..., -0.2886395 ,
          -0.2744195 ,  0.17723785],
         [ 0.3963737 ,  0.5829972 ,  0.50839925, ...,  0.04402175,
          -0.21898633, -0.0798969 ]], dtype=float32)

In [19]: type(embeddings_f_xlm)

Out[19]: numpy.ndarray

In [20]: embeddings_f_xlm.shape

Out[20]: (2010, 768)
```

Embeddings:

- Then averaged the embeddings generated both models for each class of datasets.
- Then reduced dimensions of embeddings passing it through dense layers to

(no:of sent,200)

```
ye.shape
```

```
(2010, 200)
```

```
ye.head()
```

	0	1	2	3	4	5	6	7	8	9	...	
0	0.000000	0.018750	0.0	0.0	0.118426	0.0	0.084805	0.070893	0.334041	0.000000	...	0.290
1	0.006322	0.018838	0.0	0.0	0.035249	0.0	0.043300	0.037982	0.319422	0.017381	...	0.337
2	0.000000	0.026553	0.0	0.0	0.096972	0.0	0.116328	0.000000	0.327529	0.000000	...	0.328
3	0.000000	0.042453	0.0	0.0	0.065719	0.0	0.070164	0.004974	0.299562	0.000000	...	0.400
4	0.000000	0.000000	0.0	0.0	0.000000	0.0	0.101028	0.045421	0.352284	0.000000	...	0.342

5 rows × 200 columns

```
type(ye)
```

```
pandas.core.frame.DataFrame
```

Pseudo Labeling:

- Pseudo labeling is important because we don't have a dataset having three labels(fake,hate,sentiment).
- So,we take one dataset and generate other two labels which are not present in the original dataset.Thus,we will get a dataset which contains all three labels,by using this dataset we can train the multi-task model.
- For each class of dataset pseudo labels have been generated for the other two classes
- Ex:

  To generate fake labels for hate and sentiment class datasets we using LSTM model by passing embedding of fake dataset generated before.

Pseudo Labeling:

- For each class of dataset pseudo labels have been generated for the other two classes
- Ex:

  To generate fake labels for hate and sentiment class datasets we using LSTM model by passing embedding of fake dataset generated before.

Build Model → Train Model with embeddings generated → Test embeddings With 10% test data → Predict fake labels for hate and sntm data

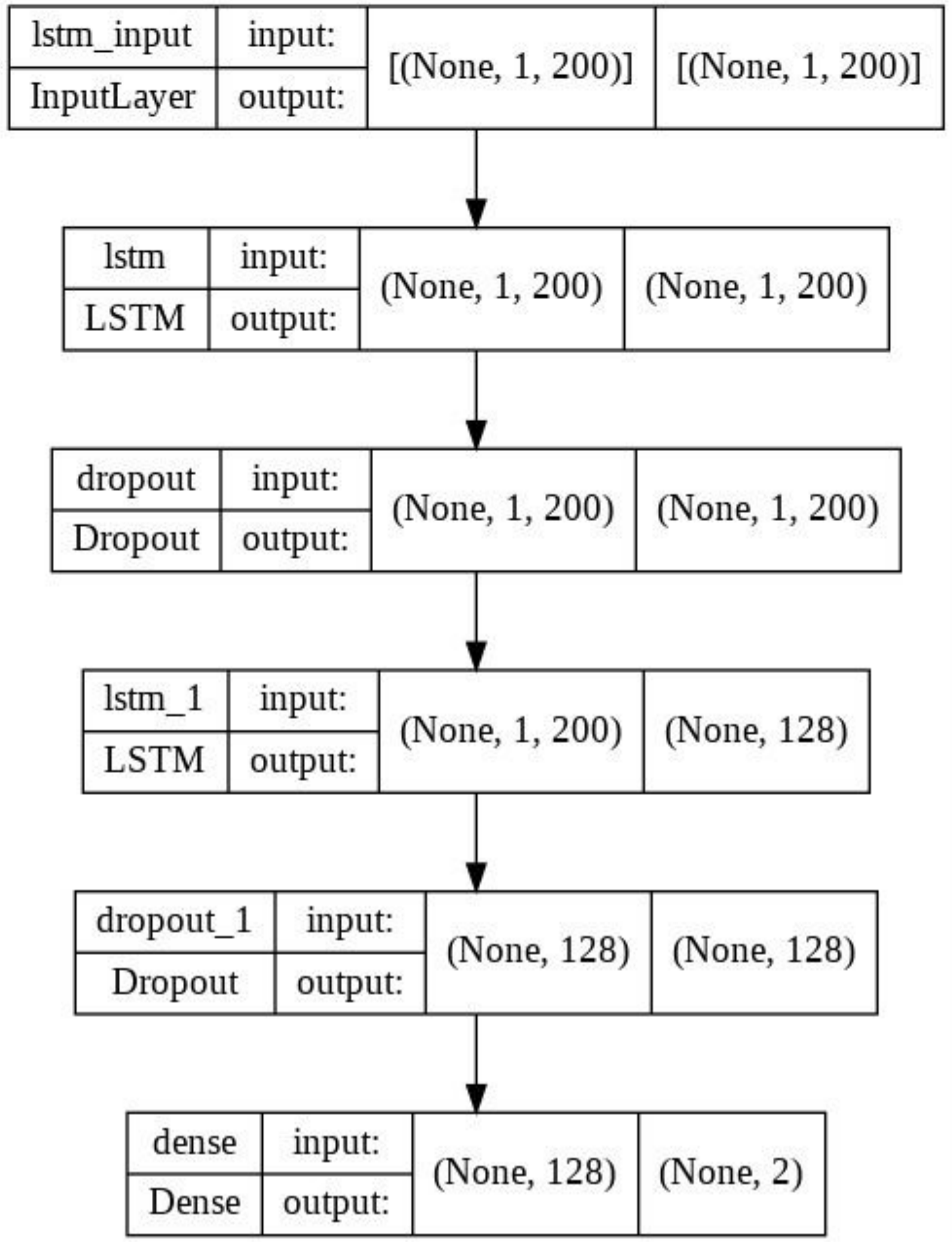
- And the same process repeated for hate labels with sentiment,fake datasets and sentiment labels with hate,fake datasets.

LSTM:

- Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.
- Steps:
- Clean the data
- Get Embeddings

  We split the data

  Train:90% , Test:10%
- Train the model(Epochs:40,Softmax,Embeddings)
- Test the model
- Predict the outputs for hate and sentiment data sets.



Outputs:

- **Predicted Probabilities of sentences belonging to 0 or 1:**

```
[[3.74282463e-05 9.99962568e-01]
 [1.86290722e-02 9.81370986e-01]
 [1.87031850e-01 8.12968194e-01]
 [7.85509467e-01 2.14490578e-01]
 [8.80725026e-01 1.19274952e-01]
 [3.70560676e-01 6.29439294e-01]
 [3.99855395e-07 9.99999642e-01]
 [4.61484888e-04 9.99538541e-01]
```

- **Using argmax function to get the argument having maximum probability :**
- **Ex: argmax[3.7428e-05,9.99e-01] = 1**

```
array([1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0,
       0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0,
       1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0,
       1, 0, 0], dtype=int64)
```

Pseudo Labeling:

LSTM MODEL	EPOCHS	MODEL ACCURACY	MODEL LOSS	VALIDATION ACCURACY	VALIDATION LOSS
HATE	40	0.6969	0.5741	0.6444	0.6392
FAKE	40	0.8272	0.3718	0.8011	0.4181
SENTIMENT	40	0.7059	0.5543	0.6361	0.6505

Fakedata	label
bhaarata ke videsha ma.mtraalaya paakistaa raajaduuta bula paa..	0
bihaara taarakishora prasaada biijepa vidhaayaka dala chu ke ...	1
...	...

Hatedata	label
Knowing ki Vikas kitna samjhata hai Priyanka aur Itch Guard Luv..	0
I am Muhajir .. Aur mere lye sab se Pehly Pakistan he .. agr	1
...	...

Stmdata	label
baa.mglaadesha kii shaanadaara vaapasii, bhaarata ko 314 rana ..	0
saba ra.mDii naacha dekhane me vyasta jaise hii koii #shaa...	0
...	1

Fakedata	label
bhaarata ke videsha ma.mtraalaya paakistaa raajaduuta bula paa..	0
bihaara taarakishora prasaada biijepa vidhaayaka dala chu ke ...	1
...	...

Hatedata	label
Knowing ki Vikas kitna samjhata hai Priyanka aur Itch Guard Luv..	0
I am Muhajir .. Aur mere lye sab se Pehly Pakistan he .. agr	1
...	...

Stmdata	label
baa.mglaadesha kii shaanadaara vaapasii, bhaarata ko 314 rana ..	0
saba ra.mDii naacha dekhane me vyasta jaise hii koii #shaa...	0
...	1

Fakedata	label_f	label_h	label_s
bhaarata ke videsha ma.mtraalaya paakistaa raaja....	0	0	0
bihaara taarakishora prasaada biijepa vidhaayaka ..	1	0	1
...	...	...	...

Hatedata	label_h	label_f	label_s
Knowing ki Vikas kitna samjhata hai Priyanka aur Itch..	0	1	0
I am Muhajir .. Aur mere lye sab se Pehly Pakistan	1	0	0
...	...	...	...

Stmdata	label_s	label_f	label_h
Saba ra.mDii naacha dekhane me vyasta jaise hii koi..	0	1	0
baa.mglaadesha kii shaanadaara vaapasii, bhaarata ..	0	0	1
...	1	...	...

Analysing Pseudo labels:Fake

Sentence	Label_F	Label_H (G,M)	Label_S (G,M)
bhaarata ke videsha mamtraalaya paakistaa raajaduuta bula paakistaa bala dvaara nirdosha naagarika jaanabuujha nisha bana kada shabda nimda	0	(0,1)	(0,1)
bihaara taarakishora prasaada biijepa vidhaayaka dala chu ke charcha sushiila moda jagaha vo niitiisha kumaara ke upakaptaana raha	0	(0,0)	(1,0)
uttara pradesha ke sambhala jaila kisaana shaamta bhamga aashamka ke kaarana laakha rupaya ke bnda bhara sambamdha notisa jaara kiya gaya	0	(1,1)	(1,1)
baabara mugala saltanata sthaapa bulamda ke dina usaka paasa duniya chautha adhika daulata saltanata kshetraphala aphagaaanistaana sameta lagabhaga puura upamahaadviipa phaila	0	(0,0)	(0,0)
mugala saltanata ke samsthaapaka baabara jaha vije ke ruupa dekha varnita kiya duusara ora unha bada kalaakaara lekhaka maa	0	(1,0)	(0,0)

Analysing Pseudo labels:Hate

Sentence	Label_H	Label_F (G,M)	Label_S (G,M)
muslamno ko mecca aur madina k nam pe cash karwata raha	0	(1,1)	(0,0)
bad politics ek hindu bhai dusre hindu bhai ka khoon pene k liye taiyar hogaya he	0	(1,1)	(0,0)
vivad hone pr freedom of speech khatre mein aa jata hai	0	(0,0)	(0,0)
admi acha kaam kare to b bhakto k chati pe saanp lotne lagte k ye acha kyun karra bhakto ko padwaoo to ye hate mongring lies chor dege	1	(0,0)	(1,0)
punjab me terrorism ki jimmebar to bjp ki dost log he jhooth ki bhi ek had hoti he	0	(1,1)	(1,0)

Analysing Pseudo labels:Sentiment

Sentence	Label_S	Label_F (G,M)	Label_H (G,M)
isna muslim ko chalang kiya ha iska movie koi mat dekhna agr tum sab muslim hoga to nhi dekho ge	0	(0,0)	(1,1)
bakwas actor ki bakwaas film	0	(1,1)	(1,1)
jo sachcha musalman hoga vo iski movie nahi dekhega bhai kyu k isne muslamano ko chalange kiya hai so mai to nahi dekhunga jiski jo marzi bhai	0	(0,0)	(1,0)
eid mubarak bhaijan	0	(0,0)	(0,0)
salman khan bolliwood ki shan banchuka back to back blockbuster superhit movi name dabbang redy bodyguard ek tha tiger dabbang jai ho kick bajarangi bahizan superhit jayagi fix	0	(1,1)	(0,0)

- **Then finally combined all the 3 pseudo labels generated datasets into single final dataset which having all three lables for each sentence.**

Data	label_f	label_h	label_s
bhaarata ke videsha ma.mtraalaya paakistaa raaja....	0	0	0
bihaara taarakishora prasaada biijepa vidhaayaka ..	1	0	1
Knowing ki Vikas kitna samjhata hai Priyanka aur Itch..	0	1	0
I am Muhajir .. Aur mere lye sab se Pehly Pakistan	1	0	0
Saba ra.mDii naacha dekhane me vyasta jaise hii koi..	0	1	0
baa.mglaadesha kii shaanadaara vaapasii, bhaarata ..	0	0	1
...	1	...	...

Building Multi-Task Learning model.

Models Used:

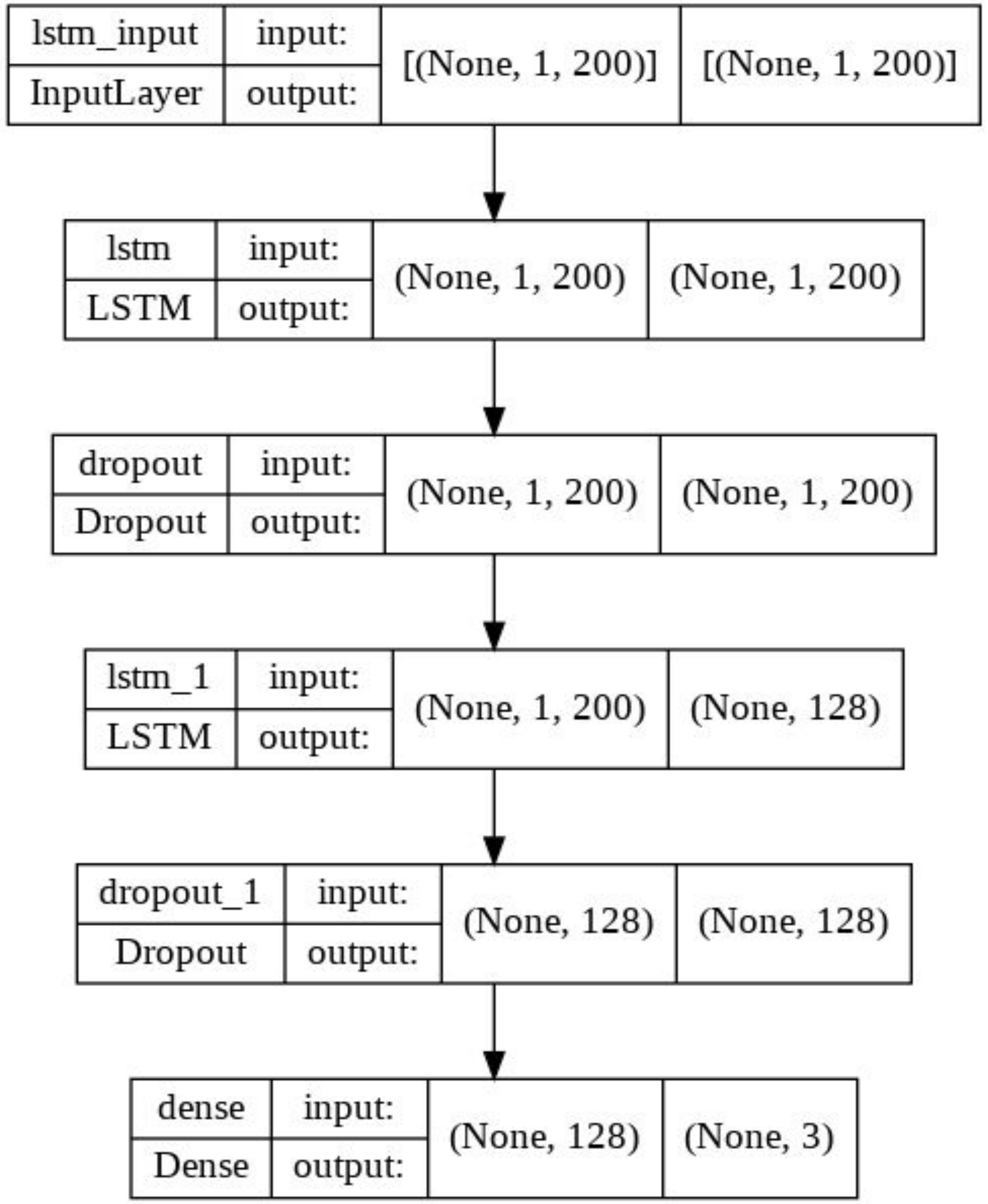
- LSTM
- Bi-LSTM
- Bert-Base
- Bert-Large
- Multilingual-Bert
- Roberta Base
- Roberta Large

LSTM Model:

- Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.
- Steps:
- Clean the data
- Get Embeddings

  We split the data

  Train:85% , Test:15%
- Train the model(Epochs:30,Softmax)
- Test the model



Output:

- **Predicted Probabilities of sentence belonging to one of the 8 possibilities as shown below :(0,0,0)(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1) :**

```
array([[0.5795595 , 0.5975832 , 0.16293538],
       [0.57913035, 0.01303385, 0.5236353 ],
       [0.90666157, 0.01318333, 0.4030353 ],
       ...,
       [0.8257415 , 0.25289512, 0.3799547 ],
       [0.92484635, 0.09241248, 0.05563275],
       [0.8121611 , 0.08322582, 0.29576597]], dtype=float32)
```

- **As we should do multilabel classification So, at each index:**
  if probability<0.5 then labeled as: 0
  else labeled as : 1
  Ex: [0.761,0.476,0.03792] = [1,0,0]

```
[[0 1 0]
 [1 0 0]
 [1 0 0]
 ...
 [1 0 0]
 [1 0 0]
 [1 0 0]]
```

```
[[1 1 0]
 [1 0 1]
 [1 0 0]
 ...
 [1 0 0]
 [1 0 0]
 [1 0 0]]
```

Results:

Lstm with SoftMax:

- Model Accuracy:0.7310
- Model Loss:0.4999
- Validation Accuracy:0.7225
- Validation Loss:0.4900

Lstm with Sigmoid:

- Model Accuracy:0.7302
- Model Loss:0.4994
- Validation Accuracy:0.7371
- Validation Loss:0.4907

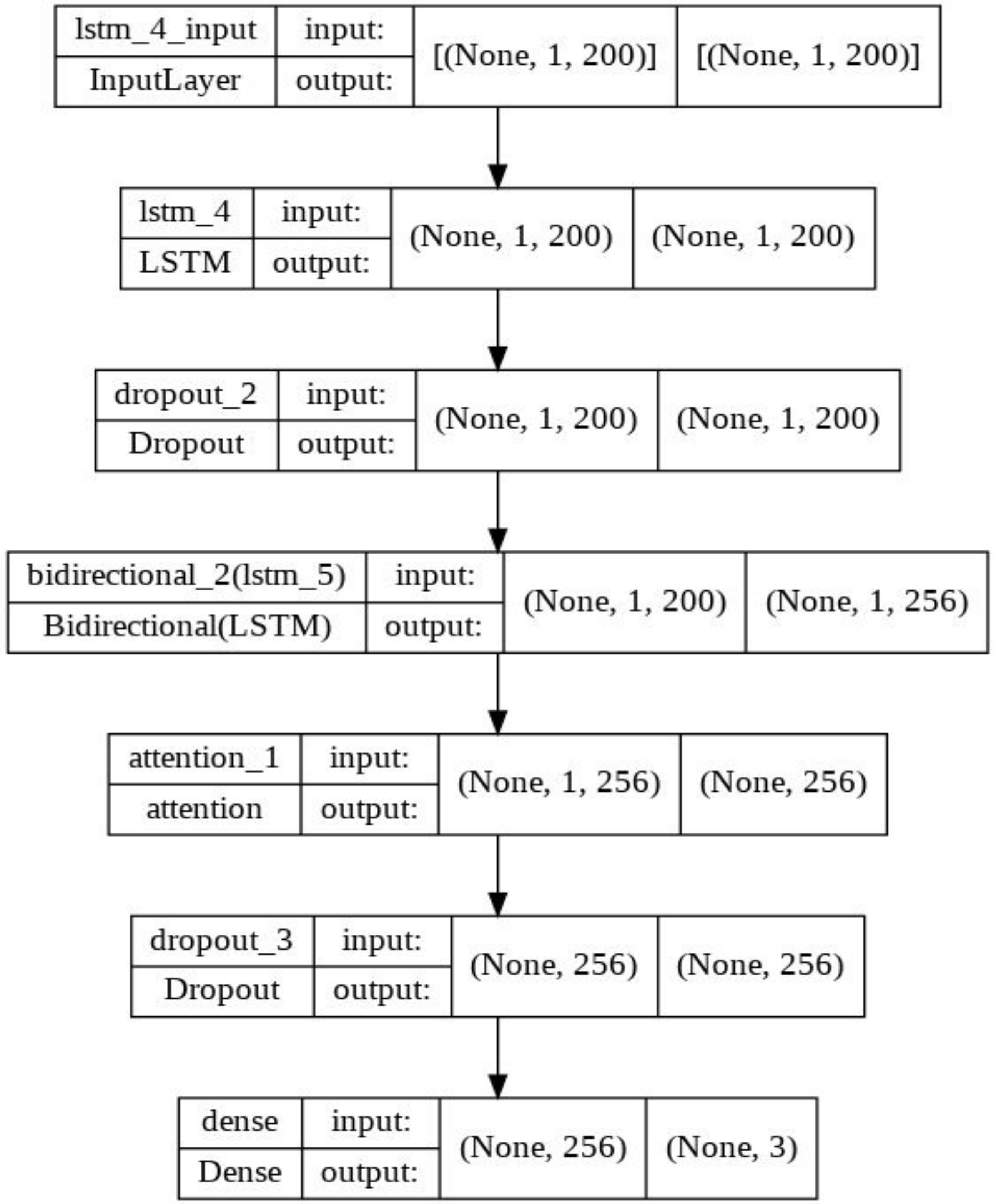
Bi-LSTM with Attention:

- BiLSTM is simply bidirectional LSTM, which means the signal propagates backward as well as forward in time.
- **Steps:**
- **Clean the data**
- **Get Embeddings**

We split the data

Train:90% , Test:10%

- **Train the model(Epochs:22,Softmax)**
- **Test the model**



Output:

- **Predicted Probabilities of sentence belonging to one of the 8 possibilities as shown below :(0,0,0)(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1) :**

```
array([[0.7614588 , 0.47644165, 0.03792679],
       [0.25400865, 0.00119931, 0.82639146],
       [0.855026  , 0.010306  , 0.27480638],
       ...,
       [0.65121114, 0.41550076, 0.23309052],
       [0.93155885, 0.09468693, 0.07481083],
       [0.8357703 , 0.07398084, 0.14388382]], dtype=float32)
```

- **As we should do multilabel classification So, at each index:**
  if probability<0.5 then labeled as: 0
  else labeled as : 1
  Ex: [0.761,0.476,0.03792] = [1,0,0]

```
[[1 0 0]
 [0 0 1]
 [1 0 0]
 ...
 [1 0 0]
 [1 0 0]
 [1 0 0]]
```

```
Predicted Labels
[[1 0 0]
 [1 0 0]
 [1 0 1]
 ...
 [1 0 0]
 [1 0 0]
 [0 0 0]]
```

Results:

Bi-Lstm Attention with SoftMax:

- Model Accuracy:0.7286
- Model Loss:0.4941
- Validation Accuracy:0.7146
- Validation Loss:0.4880

Bi-Lstm Attention with Sigmoid:

- Model Accuracy:0.7295
- Model Loss:0.4949
- Validation Accuracy:0.7331
- Validation Loss:0.4877

Bert

- BERT, which stands for Bidirectional Encoder Representations from Transformers

Roberta

- RoBERTa stands for Robustly Optimized BERT Pre-training Approach.

Implementation of Bert and Transformer Models

- Simple Transformer models are built with a particular Natural Language Processing (NLP) task in mind.
- To create a task-specific Simple Transformers model, you will typically specify a `model_type` and a `model_name`

Sl.No	Model Name	Layers	Hidden Layers	Attention Heads
1	**Bert-Base**	12	768	12
2	**Bert-Large**	24	1024	16
3	**Multilingual Bert**	12	768	12
4	**Roberta-Base**	12	768	12
5	**Roberta-Large**	12	1024	12

Comparing Parameters

Steps:

- Divide the train data and test data

  We split the data:

  Train:60%

  Evaluation:20%

  Test:20%

- Load a pre trained Models

  Bert,Roberta

- Train the model

  Epochs:5,Max_SequenceLength:200,Learning_Rate:3e-5

Steps:

- **Evaluate the model(Below are the parameters we get after evaluating the model):**

```
[ ] result3

{'LRAP': 0.9299586119070342, 'eval_loss': 0.49817745000806474}

model_outputs3

array([[0.96142578, 0.68115234, 0.02365112],
       [0.0413208 , 0.01834106, 0.79345703],
       [0.99267578, 0.04467773, 0.13378906],
       ...,
       [0.28222656, 0.00475693, 0.61474609],
       [0.99121094, 0.02522278, 0.17712402],
       [0.96875   , 0.01173401, 0.15771484]])

[ ] wrong_predictions3

[{'guid': 3, 'text_a': 'aisa kuch nahi i hate it', 'text_b': None, 'label': (0, 1, 0)},
 {'guid': 4, 'text_a': 'swachchh bharat aviyaan ke liye danyabaad', 'text_b': None, 'label': (1, 0, 0)},
 {'guid': 5, 'text_a': 'advance eid mubarak salman bhai', 'text_b': None, 'label': (0, 0, 1)},
 {'guid': 8, 'text_a': 'bhaii ab avi jao', 'text_b': None, 'label': (0, 0, 1)},
 {'guid': 9, 'text_a': 'aj ai nijum raate mone pore she fale asha din gulo kotha mone pore she sobuj a path mathe tmr hat dore hate chola din golo', 'text_b': None, 'label': (1, 0,
 {'guid': 12, 'text_a': 'salman khan dest actar bollywood', 'text_b': None, 'label': (0, 0, 0)},
 {'guid': 13, 'text_a': 'jumme ka din hai  chatting ki baat hai allah bachaye mujhe  fake pages se', 'text_b': None, 'label': (1, 0, 1)},
```

- **Test the model**
- **Predict the Output**

Output:

- **Test Model**

  Get the predictions using to_predict

  Predicted Probabilities of sentence belonging to one of the 8 possibilities as shown below :(0,0,0)(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1) :

  pred:[[0.152,0.578,0.68]........[0.95,0.32,0.56]

- **As we should do multilabel classification So, at each index:**

  if probability<0.5 then labeled as: 0

  else labeled as : 1

  Ex: [0.761,0.476,0.03792] = [1,0,0]

  o/p: [[0,1,0]....[1,0,0]]

```
Result
Predicted Values
[[0.061309814453125, 0.0120086669921875, 0.179443359375]
Predicted Labels
[[0 0 0]
 [1 0 0]
 [0 0 0]
 ...
 [1 0 1]
 [0 0 1]
 [1 0 0]]
```

Comparing Results.

NOT FAKE	FAKE
HATE	NOT HATE

Comparing LSTM models:

MODEL	EPOCHS	MODEL ACCURACY	MODEL LOSS	VALIDATION ACCURACY	VALIDATION LOSS
LSTM with Soft Max	30	0.7310	0.4999	0.7225	0.4900
LSTM with Sigmoid	30	0.7302	0.4994	0.7371	0.4907
Bi-LSTM with Attention using Soft Max	22	0.7286	0.4941	0.7146	0.4880
Bi-LSTM with Attention using Sigmoid	22	0.7295	0.4949	0.7331	0.4877

Comparing models:

MODEL	EPOCHS	ACCURACY SCORE	MICRO AVERAGE F1 SCORE	MACRO AVERAGE F1 SCORE
LSTM using Softmax	30	0.3649	0.60	0.45
LSTM using Sigmoid	30	0.4070	0.65	0.51
Bi-LSTM with Attention using Softmax	22	0.3605	0.60	0.49
Bi-LSTM with Attention using Sigmoid	22	0.3847	0.62	0.48
Bert-Base	5	0.4728	0.70	0.63
Bert-Large	5	0.4788	0.70	0.64
Bert-Base-Multilingual	5	0.4985	0.72	0.66
Roberta-Base	5	0.4692	0.71	0.64
Roberta-Large	1	0.4713	0.66	0.59

Conclusion & Future Scope

Conclusion & Future Scope

- **Identifying fake and hate news generating in social media platforms can help to reduce its impact over that thing,area,person or group of people targeted by such news.**

Conclusion & Future Scope

- Identifying fake and hate news generating in social media platforms can help to reduce its impact over that thing,area,person or group of people targeted by such news.
- And it can further developed along with image processing to read the thumbnails of videos,social media posts to avoid misleading public towards fake news.
- We can further develop by getting more and better data by extracting data using web-scraping from social-media APKs.And also improved with better computational resources

References

References

- htpps://www.analyticsvidhya.com/blog/2020/01/3-important-nlp-libraries-indian-languages-python/
- https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1
- https://simpletransformers.ai/docs/multi-modal-classification-model/

THANK YOU!


HATE CRIME
RUMOR
FAKE NEWS
STOP THE HATE
NEWS
BREAKING NEWS
FAKE