UE17CS490B – Capstone Project Phase – 2

SEMESTER - VIII

END SEMESTER ASSESSMENT

Project Title :Rumour Detection and Veracity Verification
Project ID :PW21BJD01
Project Guide :Bhaskarjyoti Das
Project Team :PES1201700214 Sukanya Harshvardhan
PES1201700294 Sirisha Lanka
PES1201701261 Prajna Girish

Outline

- Abstract
- Team Roles and Responsibilities.
- Summary of Requirements and Design
- Summary of Methodology / Approach
- Design Description
- Modules and Implementation Details
- Project Demonstration and Walkthrough
- Results and Discussion
- Lessons Learnt
- Conclusion and Future Work
- References

Abstract

- Rumour detection and veracity verification in Twitter using socio-linguistic data and social graphs

- **Identified gap** :
  Existing research identifies rumours based on either content based feature or features of the underlying social graph, not a combination of both

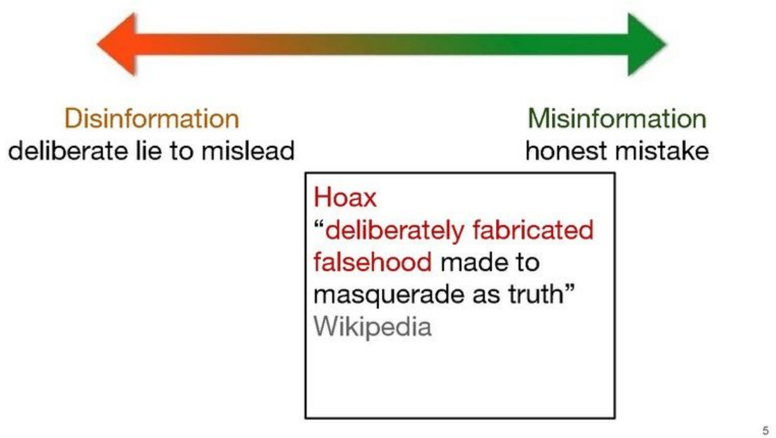
- **In scope** :
  Building a joint model utilizing both features of a rumour graph as well as content features from Tweets

- **Out of scope** :
  We only deal with rumours, and not fake news
  Our model only considers with domain specific tweets and may not produce accurate results if tried with any random tweet

Rumour vs Fake News

Rumour:

Any unverified piece of information that spreads from person to person



Rumour vs Fake News

Fake news is any piece of information that is a 100% untrue. It has a target network where it propagates with a purpose.

Rumours have an element of ambiguity in their veracity. They can be dropped randomly onto any network. The main focus would fall on the way it spreads rather than who spreads it.

LITERATURE SURVEY SUMMARY

TYPE	KEY TAKEAWAYS
RUMOUR DETECTION USING TEXT	• To capture contextual variation of tweets over time • Consider the textual content of the tweet, its timestamp, as well as the sequential conversation structure leading up to the target tweet
RUMOUR DETECTION USING GRAPH	• Homophily used as a feature for detecting rumours; a user is more likely to post rumours if he/she follows other rumour mongers • Focus on the strongest dependencies among immediate neighbours • Uses POS Tagging and classifies words as anxiety or rumour based on tags
STANCE CLASSIFICATION	• Combining structural and conversation based features gives good accuracies • Hierarchical framework to tackle rumor stance classification and veracity prediction jointly, • Encodes conversation structures for learning stance features
VERACITY VERIFICATION	• Stance of the tweet helps us determine its veracity • Leveraging the relationship between the tasks from the rumour classification pipeline in a joint multitask learning setup

Team Roles and Responsibilities

SL.NO	NAME	ROLE
1.	SUKANYA HARSHVARDHAN (PES1201700214)	• Stance classification using Bi-LSTM • Veracity verification using Bi-LSTM • Demonstration code for stance and veracity using bi-lstm module • Final Integrated Demonstration code
2.	SIRISHA LANKA (PES1201700294)	• Rumour Detection Using Random Forest Classifier • Stance classification using Torchmoji module • Demonstration code for graph module • Demonstration code for stance using emoji detection module • Weekly reports
3.	PRAJNA GIRISH (PES1201701261)	• BERT model for text based rumour detection • Demonstration code for BERT module • Project Report • IEEE Draft • LLD and Implementation Document

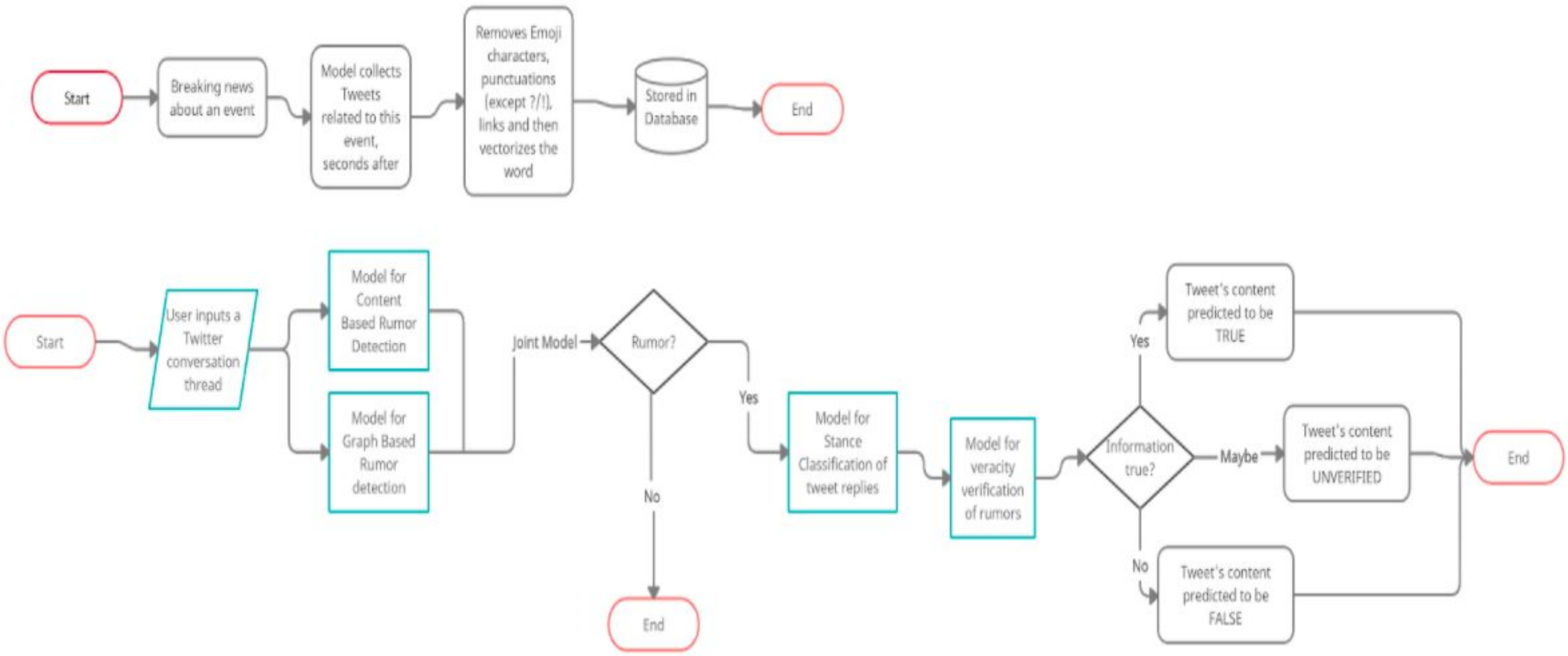
Summary of Requirements and Design

- **Dataset**

  PHEME:This dataset contains a collection of Twitter rumours and non-rumours posted during breaking news. It contains rumours related to 9 events and each of the rumours is annotated with its veracity value, either True, False or Unverified.

- **OS Requirements**
  - Windows 10 OS with the intel core i5 8th Gen processor
- **Functional Requirement**
  - The function of the model built is to detect a rumourous tweet and predict its veracity.
  - Inputs: Tweet id, Tweet propagation structure, .npy files of the conversation thread
  - Output: Detecting rumour or non rumour and veracity predicted as true, unverified, false
- **Legal Requirements**
  - All data data used in this application must be in in compliance with the Twitter Developer Agreement and Policy.

Summary of Methodology / Approach



Pre-Processing

- Remove all hashtags, user mentions, links, punctuations (except for '?' and '!'),non-alphabetic characters
- Convert to lower case, tokenize the tweet and converting to vectors
- Padding vectors to make them equal length
- Convert labels to one hot vectors
- .npy files are created using conversation thread
- Feature dictionary consists of 5 features:
    - User verification
    - Number of followers
    - Number of following
    - Number of retweets
    - Favourite count
- Conversation tree built by creating feature dictionary for each branch of the conversation using structure.json available in PHEME dataset
- Feature vector for graph model created using the sklearn.feature_extraction.DictVectorizer
- Propagation context learnt using a Decision Tree Classifier

Design Description

- Rumour Detection - Text Based : *BERT Classifier*
  - Accuracy Obtained: 71%
  - Input: Content of source tweet along with the replies
  - Output: Rumour/Non-Rumour
- Rumour Detection - Graph Based : *Random Forest Classifier*
  - Accuracy Obtained: 73%
  - Input: Propagation structure
  - Output:Rumour/Non-Rumour
- Stance Classification : *BiLSTM Model + Torchmoji*
  - Accuracy Obtained: 78%
  - Input: .npy files of conversation thread and content of the replies
  - Output: support/deny/query/comment
- Rumour Veracity Verification : *BiLSTM Model*
  - Accuracy Obtained: 92%
  - Input: stance of the users in .npy file
  - Output: True/Unverified/False

Modules and Implementation Details

- **os module** (*inbuilt*): provides a portable way of using system functionality. It is mostly used to parse and process the dataset.
- **json module** (*ver 2.0.9*): used to process data in json format.
- **scikit-learn module** (*ver 0.23.2*): machine learning library that supports supervised and unsupervised learning.
- **nltk** (*ver 3.5*): platform to build python programs to work with human language data.
- **transformers** (*ver 3.5.1*): library with NLP-oriented architectures of pre-trained models
- **pytorch** (*ver 1.7.0*): open-source library that is based on the Torch machine learning library.
- **numpy** (*ver 1.19.0*): module that will assist in working with large numerical data.
- **re** (*ver 2.2.1*): helps evaluate regular expressions in textual data.
- **word2vec** module to capture the content of a tweet
- **torchmoji** to capture the emotions associated with a particular tweet

Modules and Implementation Details

o

Input: $conversationthread$
Output: $Non-Rumour/TrueRumour/FalseRumour/UnverifiedRumour$

[1] $n \leftarrow$ no of replies
$graphstruct \leftarrow$graph struct
$setofreplies \leftarrow$[$r_1, r_2, ..., r_n$]
full text $\leftarrow$ source text
$i \leftarrow 1$ to N

    $replytext \leftarrow$set of replies$_i$

    $fulltext = fulltext.concatenate$(reply text)

    $i \leftarrow i + 1$

$detectiontext = BERTModel(fulltext)$
$detectiongraph = RFClassifier(graphstruct)$
$finaldetection = JointModel(detectiontext, detectiongraph)$

IF $finaldetecion == NonRumour$ **then**
EXIT

$stanceemoji = []$
i $\leftarrow 1$ to N

    $stance = Torchmoji$(set of replies$_i$)

    $stanceemoji$ =stance emoji.append($stance$)

stance bilstm= []
$i \leftarrow 1$ to N

    $stance = Bilstm$(set of replies$_i$)

    $stancebilstm$ =stance bilstm.append($stance$)

$setofstances$ =joint stance($stanceemoji$,stance bilstm)
veracity= $veracitybilstm$(set of stances)

Project Demonstration

Demo

Results and Discussion

- Result obtained after combining both the detection models.
- They were combined using confidence scores.

	precision	recall	f1-score	support
0	0.96	0.93	0.95	1241
1	0.89	0.93	0.91	687
accuracy			0.93	1928
macro avg	0.92	0.93	0.93	1928
weighted avg	0.93	0.93	0.93	1928

- Result obtained for final veracity is

	precision	recall	f1-score	support
0	0.91	0.96	0.94	187
1	0.91	0.87	0.89	82
2	1.00	0.89	0.94	64
accuracy			0.92	333
macro avg	0.94	0.91	0.92	333
weighted avg	0.93	0.92	0.92	333

- The results obtained were as expected. The combination of two models has given us high accuracies.

Schedule

TASK	TIMELINE	GOALS MET?
Literature Survey	11 Jan - 22 Jan	YES
Going through already existing codes/ implementations	22 Jan - 31 Jan	YES
Stance Classification	1 Feb - 8 Feb	YES
Veracity Verification	8 Feb - 15 Feb	YES
Stance Using Torchmoji	15 Feb - 22 Feb	YES
Combined Stance using BiLSTM and Torchmoji	23 Feb - 28 Feb	YES
Graph Based Detection	1 March - 25 March	YES
Combined Detection Model	26 March - 4 Apr	YES
Testing on Twitter API	5 Apr - 20 Apr	YES
Analysis/ Improvements	21 Apr - 1 May	YES

Documentation

SL.NO.	DOCUMENT	STATUS
1.	Project report approved by guide	YES
2.	IEEE format of paper ready for submission	YES
3.	Project video	YES
4.	Github repository link https://github.com/sirishalanka181/Capstone-2021--Rumour-Detection-and-Veracity-Verification	YES
5.	A3 Poster of Project	YES
6.	Artifacts uploaded to repository	YES
7.	ESA Presentation	YES
8.	Team Details Document	YES

Lessons Learnt

- We learnt the use cases of different deep learning models and the situations in which the models are relevant
- Familiarized with different Python modules - such as Torchmoji, nltk and keras.
- We learnt how to use Twitter APIs
- How to capture propagation patterns in a Tweet
- Time distributed LSTMs and their performance on sequential data
- How to combine different models to improve results
- Transformers and Transfer Learning

Issues Faced

- In the rumour detection aspect, we were making decisions based on the source tweet in isolation ie. with no context

  Based on the feedback given by the panel, we decided to incorporate the replies to the source tweet, in addition to the source itself into the decision making process.

- While combining the models, we faced issues on what weightage to assign to them

  We dealt with this issue by looking into the confidence levels that the different models had to offer.

Conclusion and Future work

- Extracted features from the dataset help analyse the ego centric network of an individual in a social network to help detect rumour tweet
- Confidence scores of the linguistic model and graph based model are used to build the joint model to detect rumours on Twitter
- A final analysis of the rumour detection and veracity verification model proves that that this surpasses other existent models.

Future Work

Extending the scope to include other social media networks such as Reddit, Instagram and Facebook

References

- Zubiaga, Arkaitz Kochkina, Elena Jiakata, Maria Procter, Rob Lukasik, Michal Bontcheva, Kalina Cohn, Trevor Augenstein, Isabelle Discourse-Aware Rumour Stance Classification in Social Media Using Sequential Classifiers
- Pamungkas, Endang, Basile, Valerio Patti, Viviana 2019/01/07 Stance Classification for Rumour Analysis in Twitter: Exploiting Affective Information and Conversation Structure
- Wei, Penghui, Xu, Nan Mao, Wenji 2019/09/18 Modeling Conversation Structure and Temporal Dynamics for Jointly Predicting Rumor Stance and Veracity
- Huang, Qi, Zhou, Chuan, Wu, Jia, Mingwen, Wang, Wang, Bin, 2019/07/01 Deep Structure Learning for Rumor Detection on Twitter
- Angelova, Ralitsa, Weikum, Gerhard,Jaervelin, Kalervo, 2006/01/01, SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 485-492 (2006) Graph-based Text Classification:Learn from your neighbour

References

- MA, Jing; GAO, Wei; and WONG, Kam-Fai. Rumor detection on Twitter with tree-structured recursive neural networks. (2018). *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. 1980-1989. Research Collection School Of Computing and Information Systems.

- Nguyen, T. T.. “Graph-based Rumour Detection for Social Media.” (2019).

- Zhou, Kaimin, C. Shu, B. Li and Jey Han Lau. “Early Rumour Detection.” *NAACL-HLT* (2019).

- https://github.com/huggingface/torchMoji

- https://medium.com/huggingface/understanding-emotions-from-keras-to-pytorch-3ccb61d5a983

Thank You