IR Term Project-Rumour Detection

TEAM ID:-18

Harshavardhan Alimi	-	18EC10021
Nalla Siva Krishna Sai	-	18EC10037
Vunnam Yashwanth Babu	-	18EC10069
Ravi Karthik	-	18EC3AI19

Individual Contributions:-

Common task(Task-1):- Done by Harshavardhan Alimi, Ravi Karthik, both have done the task while on the call.

Contribution:-

- Merged the available data on stance labels into 1 file and used it for labeling tweets, as each source tweet is represented in tree-format, we have added stance(key) for each tweet in the dictionary,
    - In some cases, The stance dataset also contains the stance label for source tweets, so we removed those stance labels while generating trees.
    - Initially we have done label for stance as the one-hot encode:= [0, 0, 0, 0,]:- No stance ; [1, 0, 0, 0]:-comment ; [0, 1, 0, 0]:-deny ; [0, 0, 1, 0]:-support ; [0, 0, 0, 1]:-query. Lot of tweets doesn't have stance label, we managed them giving separate encode label([0,0,0,0])
    - Later we had labeled for stance as normal stance label:= 0:- No stance 1:-comment; 2:- deny; 3:-support; 4:- query, Because one-hot encoded values are not helpful in the leaving-out no-stance label(as they were source nodes they shouldn't be considered in loss calculation step of stance classification.) during the calculation of loss function. The tweets which don't have a stance label in the given dataset were labeled as '0'.
- Read the paper and Run the Tree-LSTM code for the tree dictionary to generate a Rumour label for source-tweet.
- The Tree-LSTM will find hidden representations for each node in the tree(i.e. Each tweet)[which is already implemented], they will be passed into a feed-forward dense layer, drop-out layer(to remove the overfitting), layer normalization layer, and then softmax layer to give 4 outputs from there we can get the stance for each node.
- The model will generate 4-outputs for each node, but it will consider those nodes whose stance labels are given(don't consider) for calculation of stance loss.
- But the model had always given a label comment (because of the imbalance stance dataset), To address this issue weight vector is calculated, which will be given as an input for the loss function(to assign weightage for each label, The class labels which are low in the number[in the stance dataset] assigned higher weightage and the class labels which are High in number[in the stance dataset] assigned lower weightage).
- This weight vector can be calculated in two ways:-

    - By considering the entire dataset:- which gives total class labels available in number as [a,b,c,d]
    - Considering only Training dataset:- which gives total class labels available in number as [a,b,c,d]
    - Then the weight vector becomes [k/a , k/b , k/c , k/d]
    - where k = 1 / (1/a + 1/b + 1/c + 1/d).
- Also implemented logging at each iteration and results.

Rumour_RVNN(Task-2) :- Done by Vunnam Yashwanth Babu, Nalla Siva Krishna Sai both have done the task while on the call.

Contribution:-

- Read the paper and executed the code for 'Rumor Detection on Twitter with Tree-structured Recursive Neural Networks' paper on their dataset for a few epochs.
- The data set was very large so had to run only a few epochs
- Understood the columns used in the processed data set used in the paper for Bottom-up and Top-down trees
- Pre-processed the PHEME dataset, created a vocabulary, and assigned indices to all the words used in the tweets.
- Indexed every tweet id separately for each tree and also mentioned the index of the parent of the tweet id, current tweet id, parent number (in case of TD tree) / max degree (in case of BU tree), maximum text length, and index(of the word in the whole vocabulary)-count(of the word in the current tweet) pairs for every word in the tweet.
- Created another text file for the labels for every tweet id present in the dataset. There are 4 labels: non-rumor, false, true, unverified.
- Created test and train split into 5 folds
- Ignored retweets in pre-processing as the paper suggested they have no influence on the result in this algorithm as they do not provide any extra information or evidence content wise.
- As the entire PHEME data set is very large and we do not have enough resources to run it. We ran the model on gurlitt-all-rnr-threads (a dataset in PHEME) and found the results. However, preprocessed data is provided for the entire PHEME dataset separately.
- Provided two separate directories, one for the complete PHEME dataset and the other for the gurlitt-rnr-threads dataset. Both are very similar except for very few minor changes in the extraction script and word-dimension in the model.

Results:-

Common Task:-

1. Stance labels for sub-set in each dataset:- (remaining tweets in the respective dataset don't have stance labels) The value in the bracket shows the number of stance labels mentioned in the Dataset for source tweets in each class.

Dataset	comment	deny	support	query
charliehebdo	775(2)	55(2)	61(0)	174(74)
ebola-essien	21(0)	6(0)	1(0)	4(2)
ferguson	754(5)	86(2)	108(0)	144(43)
germanwings	228(1)	14(1)	43(0)	52(27)
gurlitt	0(0)	0(0)	0(0)	0(0)
ottawashotting	561(0)	76(2)	82(0)	108(61)
price-toronto	62(2)	6(1)	11(0)	12(9)
putinmissing	32(1)	5(1)	5(0)	11(7)
sydneysiege	760	88(1)	110(0)	152(73)

2. Comparison of results :- (with initial rumour detection) In stance classification weights(for loss function) are [0.0436, 0.4141, 0.3305, 0.2118]

	Initial Results(from paper)		weights :- [0.0436, 0.4141, 0.3305, 0.2118]			
	Rumour Detection @ root		Rumour Detection @ root		Stance classification @non-root	
Test-Dataset	accuracy	f1-score	accuracy	f1-score	accuracy	f1-score
charliehebdo	0.7508772	0.5867827	0.7326981	0.6231748	0.4769953	0.3282978
ebola-essien	0.7333333	0.44	0.4285714	0.3000000	0.2812500	0.1500000
ferguson	0.3934589	0.3934762	0.6617063	0.5471394	0.3635531	0.2107704
germanwings	0.5210918	0.4896692	0.5136476	0.5050877	0.5548961	0.3209847
gurlitt	0.3125	0.3055555	0.6000000	0.4886364	-	-
ottawashotting	0.5497076	0.5503488	0.5152225	0.4510443	0.4486094	0.3089019
price-toronto	0.1865716	0.1672078	0.2631579	0.2208274	0.5054945	0.3565760
putinmissing	0.4448529	0.4467213	0.4444444	0.4444444	0.4905660	0.3554894
sydneysiege	0.5994876	0.5975113	0.6085470	0.5676107	0.3576577	0.2820080

3. Comparison of results :- stance classification weights are calculated in code for each stance class vs constant weights [0.0436, 0.4141, 0.3305, 0.2118]

	weights :- [0.0436, 0.4141, 0.3305, 0.2118]				weights calculated based on training set			
	Rumour Detection @ root		Stance classification @non-root		Rumour Detection @ root		Stance classification @non-root	
Test-Dataset	accuracy	f1-score	accuracy	f1-score	accuracy	f1-score	accuracy	f1-score
charliehebdo	0.7326981	0.6231748	0.4769953	0.3282978	0.7617854	0.5833060	0.3126761	0.2706941
ebola-essien	0.4285714	0.3000000	0.2812500	0.1500000	0.4285714	0.3000000	0.2187500	0.1835637
ferguson	0.6617063	0.5471394	0.3635531	0.2107704	0.5307540	0.4953465	0.5000000	0.3012603
germanwings	0.5136476	0.5050877	0.5548961	0.3209847	0.5310174	0.4961600	0.3976261	0.3090617
gurlitt	0.6000000	0.4886364	-	-	0.6666667	0.6411483	-	-
ottawashotting	0.5152225	0.4510443	0.4486094	0.3089019	0.5866511	0.5816903	0.2986699	0.2495856
price-toronto	0.2631579	0.2208274	0.5054945	0.3565760	0.1503759	0.1345235	0.4505495	0.3263400
putinmissing	0.4444444	0.4444444	0.4905660	0.3554894	0.5407407	0.5185228	0.4905660	0.2818262
sydneysiege	0.6085470	0.5676107	0.3576577	0.2820080	0.5974359	0.5550788	0.3540541	0.2862467

Rumour_RVNN (Individual Task):-

1. Results obtained on the Twitter -15 dataset :

BOTTOM-UP TREE			
#epoch	Accuracy	F1-score	RMSE
1	0.38	0.34	0.45
6	0.53	0.51	0.43
11	0.55	0.56	0.41
16	0.59	0.62	0.39
600*	0.70	0.71	-

TOP-DOWN TREE			
#epoch	Accuracy	F1-score	RMSE
1	0.25	0.10	0.55
6	0.24	0.09	0.55
11	0.25	0.10	0.54
16	0.26	0.13	0.54
600*	0.72	0.73	-

* → from the paper
- → not reported

2. Results obtained on the gurlitt-all-rnr-threads (a dataset in PHEME) dataset :

BOTTOM-UP TREE			
#epoch	Accuracy	F1-score (avg)	RMSE (avg)
1	~0	~0	0.33
6	0.49	0.17	0.32
11	0.50	0.17	0.30
21	0.99	0.25	0.29
36	~1	0.25	0.29

Note : Extreme accuracies because of small dataset

TOP-DOWN TREE			
#epoch	Accuracy	F1-score (avg)	RMSE (avg)
1	0.53	0.23	0.37
6	0.89	0.44	0.29
11	0.78	0.38	0.25
21	0.82	0.40	0.22
36	0.79	0.39	0.21