SMAI Project Report

Thapaswini Kancharla
Issac Balaji
Ali Hussaini
Animesh Das

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

1

Problem Statement

Detect Rumor and Stance Jointly by Neural Multi-task Learning

The challenge is aimed at considering the two tasks namely rumour and stance classification dependent and reinforce the two tasks via mutual feedback in a unified architecture to learn a set of bilaterally friendly features to both of the tasks to facilitate their interaction.

Rumor detection aims to determine the veracity of a given claim about some subject matter. Stance detection aims to determine the different attitudes expressed in a text towards a specific target.

2

Datasets

we used two datasets namely pheme dataset and fakenews dataset for rumour detection and stance classification respectively.

Dataset for rumour detection: pheme

pheme dataset is classified into two classes namely rumours and non-rumours.It is highly unbalanced with non-rumours class being the majority class.

1. total no of instances : 97410

2. non-rumours : 67952

3. rumours : 29458

Dataset for stance classification: fakenews

fakenews dataset is classified into four classes namely agree ,diagree ,discuss and unrelated. It is highly unbalanced with unrelated class being the majority class.

1. total no of instances : 49972
2. agree : 3678
3. disagree : 8909
4. unrelated : 36545

3 Motivation

In recent years we have seen a surge in rumours spreading like wild fires. These have not only affected personal lives but also can have serious global consequences. To combat this menace our group decided to use our deep learning knowledge to curb this menace and implement a model that can detect fake news and rumours with a certain level of accuracy. Previous analyses indicate that false rumors tend to provoke tremendous controversies than normal news report in which denying and questioning stances were found playing crucial role in signaling claims as being rumors. It is noticed that several studies on rumor detection have taken into account such kind of stance-bearing signals in their models. Taking into account of this dependency Multi-task learning approach has been considered rather than training models for two different tasks independently.

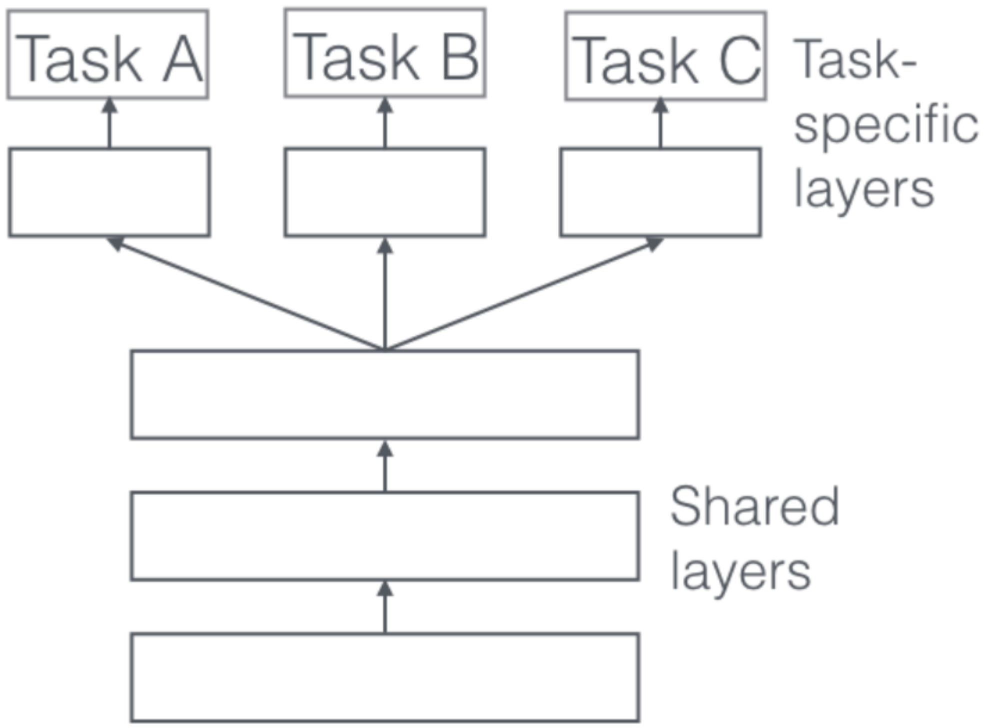
4

Overview

Multi task learning is a approach in which representations are shared between related tasks to enable our model to generalize better on our original task. There are two types of layers in neural multi-task learning. They are

1. Shared layers

2. Task specific layers



5

Approach

5.1 Data preprocessing

1. CAPITALIZATION: Text often has a variety of capitalization reflecting the beginning of sentences, proper nouns emphasis. The most common approach is to reduce everything to lower case for simplicity but it is important to remember that some words, like "US" to "us", can change meanings when reduced to the lower case.

2. STOPWORDS: A majority of the words in a given text are connecting parts of a sentence rather than showing subjects, objects or intent. Word like "the" or "and" cab be removed by comparing text to a list of stopword.

3. EXPANDING CONTRACTIONS: Contractions are the shortened version of words. They exist in either written or spoken forms. The shortened version of existing words are created by removing specific letters and sounds. In case of English contractions they are often created by removing one of the vowels from the word. Examples would be "is not" to "isn't" and "will not" to "won't". By nature contractions do pose a problem for NLP and text analytics because to start with we have special apostrophe character in the word . One way to deal with this problem is to create an dictionary which stores the mapping and can used as reference for cleansing.

4. HASHTAGS AND URLs: Tweets contains lots of hashtags and urls which has some data regarding the topic of discussions. These are not necessary while training the model and hence are removed from the dataset.

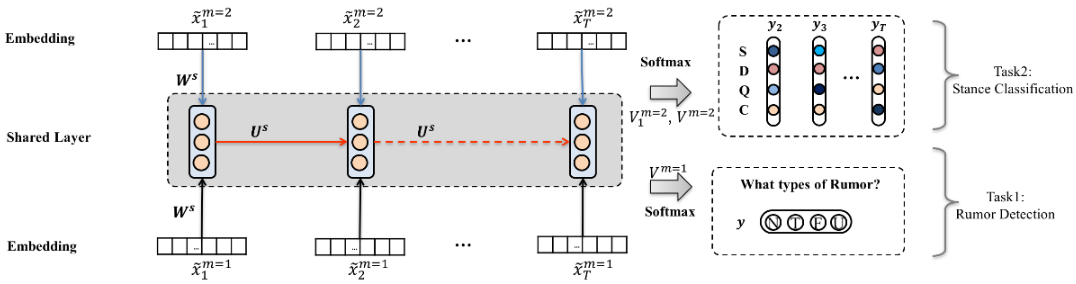
5. SENTENCE EMBEDDINGS: For gettings embeddings of either tweets or each sentence of a paragraph from fakenews dataset. we have use the pretrained GloVe provided by stanford. This GloVe is available at https://nlp.stanford.edu/projects/glove/. These word embeddings are then multipied with the TFIDF values of the words nad averaged to get the sentence embeddings.

Nearly 95% of the words present in both the datasets have their embeddings in the pre-trained GloVe. So we used the embeddings without training to get GloVe correspinding to the particular dataset. Random embeddings are given to the words which dont have embeddings in the pretrained GloVe.

5.2 Architectures

UNIFORM SHARED LAYER ARCHITECTURE:

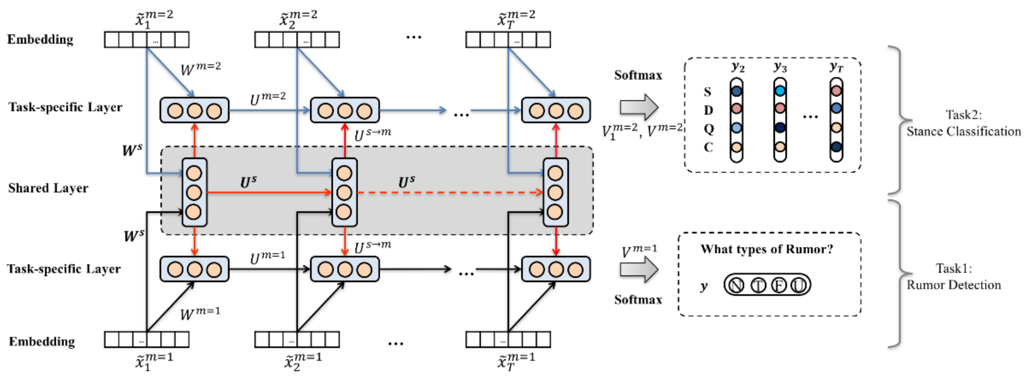
In this model, the different tasks share a same hidden layer and each task has its own task-specific input and embeddings, which is shown as the Uniform Shared-Layer Architecture in Figure.



(a) Uniform Shared-Layer Architecture

ENHANCED SHARED LAYER ARCHITECTURE:

The Enhanced Shared-Layer Architecture adopts two hidden layers for each task: one is used to extract the common pattens via the shared parameters, and the other is used to capture task-specific features via the separate parameter sets. Accordingly, each task is assigned a shared GRU layer and a task-specific GRU layer, which hopefully can be used to capture the shared and local representations for different tasks.



(b) Enhanced Shared-Layer Architecture

5.3 INPUTS

The input to this neural network is a labeled Claim.

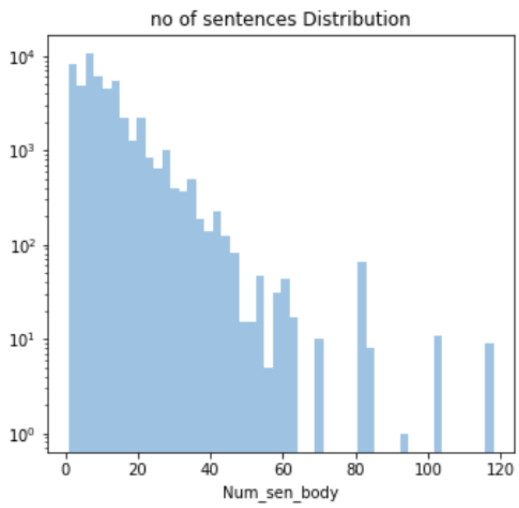
Each claim corresponds to an input sequence of its relevant posts for rumour dataset i.e. one claim consists of a source tweet and it's corresponding reply tweets.

whereas it is a sequence of sentences corresponding to a news article for fakenews dataset i.e. one claim consists of the heading of the article and all the sentences of the body of the article

6 EXPERIMENTS

6.1 No of Timesteps

As a GRU has a fixed a fixed number of timesteps the size of all the claims must be equal.But as different news articles have different number of sentences and a source tweet has different number of reply tweets the claims we obtained are of different sizes.



To make all the claims of equal length we had to either pad small length with zeros or decrease the size of claim by removing some sentences.

we experimented with different timesteps. micro and macro F1-score obtained for different timesteps are:

for number of dense layers = 2

RUMOUR:

- 120 : (0.70,0.68)
- 100 : (0.735,0.669)
- 80 : (0.76,0.72)

FAKENEWS

- 120 : (0.53,0.48)
- 100 : (0.57,0.49)
- 80 : (0.55,0.419)

Based on the above observations we have taken the number of timesteps to be 100.

6.2 NUMBER OF DENSE LAYERS

for timesteps = 100

RUMOUR:

- 2 : (0.735,0.708)
- 3 : (0.72,0.72)

Fakenews:

- 2 : (0.58,0.41)
- 3 : (0.69,0.579)

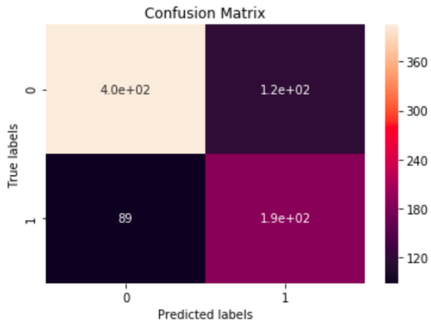
7

RESULTS

7.1 UNIFORM SHARED ARCHITECTURE

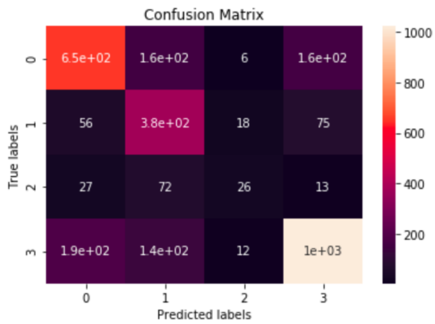
- Rumour :

```
Micro-F1-score 0.74
Macro-F1-score 0.7191447417279351
```



- Fakenews :

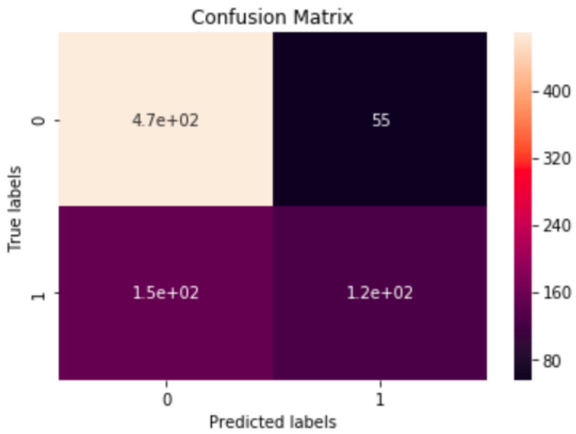
```
Micro-F1-score 0.692
Macro-F1-score 0.5791031006648775
```



7.2 Seperate architecture

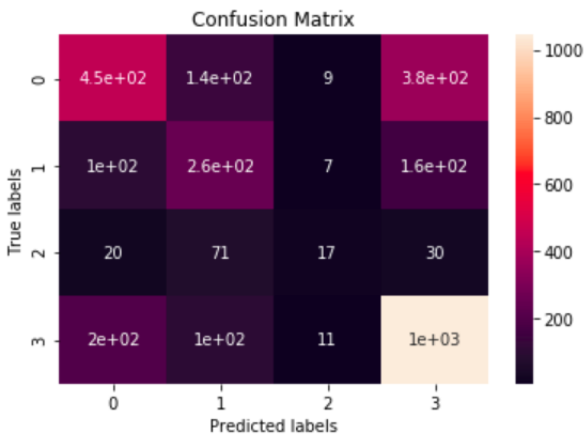
- Rumour :

```
Micro-F1-score 0.7425
Macro-F1-score 0.6840878419825789
```



- Fakenews :

```
Micro-F1-score 0.59
Macro-F1-score 0.4683010896359845
```



7.3 Enhanced Shared architecture

```
Confusion Matrix Test Data - Task 1 :
[[  0   0   0  37]
 [  0   0   0  10]
 [  0   0   0  93]
 [  0   0   0 360]]
Accuracy Score Test Data - Task 1 : 0.72
Report Test Data - Task 1 :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        37
           1       0.00      0.00      0.00        10
           2       0.00      0.00      0.00        93
           3       0.72      1.00      0.84       360

   micro avg       0.72      0.72      0.72       500
   macro avg       0.18      0.25      0.21       500
weighted avg       0.52      0.72      0.60       500
 samples avg       0.72      0.72      0.72       500

Confusion Matrix Test Data - Task 2 :
[[500]]
Accuracy Score Test Data - Task 2 : 1.0
Report Test Data - Task 2 :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       500

   micro avg       1.00      1.00      1.00       500
   macro avg       1.00      1.00      1.00       500
weighted avg       1.00      1.00      1.00       500
```

7.4 Success

Training through Uniform shared layer is working better than training seperately.

7.5 Failure

Training through Enhanced shared layer is not working better than training seperately

8

Analysis of Results

As we could see that with increase in the number of timesteps the performance increased till 100 and then it started to decrease. It might be because of overfitting.

For 2 layer architecture the performance is not better than the 3 layered one may be due to the simplicity of the model.

9

Contributions

- Thapaswini Kancharla : 50%
  (Data pre-processing, Uniform shared architecture,report)

- Issac Balaji : 25%
  (Enhanced shared architecture)

- Ali Hussaini : 20%
  (Enhanced shared architecture, presentation)

- Animesh Das : 5%
  (Project Proposal, presentation)