PROJECT REPORT

PostCom2DR: Utilizing information from post and comments to detect rumors

Submitted to – Sanasam Ranbir Singh

Mentored by – Sujit Kumar

IIT GUWAHATI

07-05-2022

Submitted by - Random vector trees

Atul Bunkar 214101011

Jyotirmoy Malakar 214101024

Rahul Chorotia 214101039

Mohit Sharma 214161005

PHASE 1:

1. INTRODUCTION

(a) Brief introduction and background of research problem.The development of social media has caused a boom in information sharing, but it also provides an ideal platform for publishing and spreading rumors. On social media platforms, there are a lot of comments, which contain the user's most direct views and reactions to the post. They can be utilized as clues to detect rumors. Recently, some methods are proposed to detect rumors through posts and comments which usually focus on the content information. With the rapid development of the Internet, social media has become the main platform for users to obtain information and exchange opinions. Due to a large number of users and easy access to social media, rumors can spread widely and quickly on social media, bringing huge harm to society and causing a great threat to social stability. During the outbreak of COVID-19 in 2020, some people made up a lot of rumors about the spread of the epidemic on the social platform, coupled with the incitement of some comments, causing great social panic. The widespread dissemination of rumors has affected our normal lives in all aspects. Thus, rumor detection on social media is imminent. To debunk rumors and minimize their harmful effects, many efforts have been made. Existing research can be divided into three categories: content-based methods, user-based methods, and propagation-based methods.

1.2 RESEARCH PROBLEM

We propose a novel model, PostCom2DR, for rumor detection. In PostCom2DR, a reply graph between posts and comments is first created. Then a bilevel GCN and self-attention mechanism are built to learn the representation of comments. Secondly, a post-comment co-attention mechanism is introduced to selectively fuse information, and this helps the model focus on more relevant information. Through the above modules, we can get a global representation of posts and comments. In addition, a 1D CNN is built to capture the local topic drift on time series inside the comments. Finally, we concatenate the global representation and local representation for rumor detection. Extensive experiments conducted on Chinese and English datasets show that PostCom2DR significantly outperforms other state-of-the-art models on both rumor detection and early detection.

2. **CHALLENGES IN RESEARCH PROBLEM**

(b) Challenge in research problem: Discuss various challenges in the respective research problem.

The current work of rumor detection is facing the challenge of interpretability, and it is important to provide a direct basis for the detection results. Our model can use the mutual attention mechanism to select more post-related content from comments and vice versa, which is demonstrated in the Case Study. Therefore, the mutual attention mechanism module enables our model to eliminate noise or irrelevant information when aggregating information. However, the model failed to select explanatory comments for the results of rumor detection. We will look into how to add more direct evidence to rumor detection in future work. In addition, there is some other valuable information on social media, such as pictures, videos, and other information, which is undoubtedly valuable for rumor detection tasks. However, we have not used these messages in our current work.

3.**PROPOSED MODEL**

(c) Discuss paper proposed models in detail.

We present experiments to evaluate the effectiveness of the proposed PostCom2DR model. Specifically, we aim to answer the following questions:

RQ1 Can PostCom2DR outperform the state-of-the-art methods on experimental datasets in a rumor classification task?

RQ2 Does each component of PostCom2DR help improve classification performance?

RQ3 Can PostCom2DR show early detection performance?

RQ4 Whether the sequence differences of the three components: reply structure, mutual selection, and topic drift will affect the results?

- We evaluated our method on three datasets, which include two Chinese datasets: Rumdect and Weibo, and one English dataset: GossipCop. Each dataset has a collection of posts from social media.
- To verify our model, we make a detailed comparison of our proposed methods and some of the state-of-the-art baselines on the rumor detection task. The methods can be divided into three categories: machine learning models, non-graph deep learning models, and graph based deep learning models
- The experimental environment is as follows: Intel i7 2.20 GHz processor, 8.0 GB memory, GTX-1050 ti GPUs. And all codes are implemented in Python (3.7.6). We implement all machine learning-based baseline models with scikit-learn libraries (0.22.1). And we implement all deep learning-based baseline models, as well as our model, with PyTorch libraries (1.4.0)

To answer RQ1, we compare our model against several related models shows the performance of the proposed method and all the compared methods. We compare and analyze our overall performance To illustrate the effectiveness of every module of the PostCom2DR, we perform a series of ablation studies (Answer to RQ2) which include four parts: PostCom2DR-G, PostCom2DR-S, PostCom2DR-Co and PostCom2DR-Cn.

To address RQ3 and illustrate the early detection ability of our model, we further report the performance by varying the number of comments per post. We control the scope of detection and only use instances with a small amount of comments to test the model. Specifically, considering that the average number of comments in the Rumdect dataset is 235, we use the top 20, 50, 100, 150, 200 numbers of comments in the test set for rumor early detection.

To answer RQ4 and further analyze the effectiveness of the module, we have experimented with different ways of organizing modules. We have three module combination modes, in which (a) is PostCom2DR.

Mode b: First, we build the reply graph and use Bilevel-GCN and comments self-attention mechanism to get comments representation. Then, CNN is applied to enhance comments representation. Finally, post-comment co-attention is used to get the aggregated representation for classification.

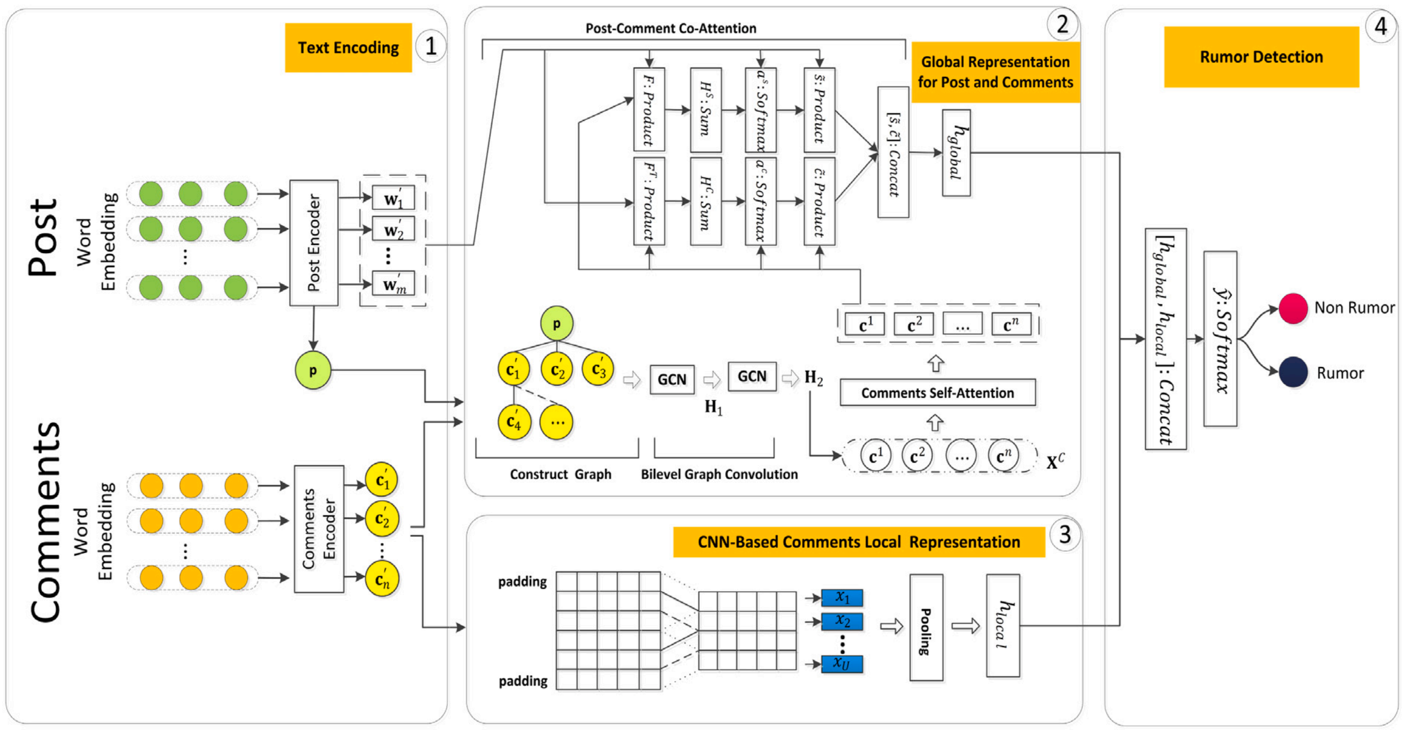
Mode c: First, we capture the comment feature representation $X1$ by Bilevel-GCN and comments self-attention. At the same time, we get the comment representation $X2$ by CNN. Next, we add the corresponding elements of $X1$ and $X2$ to get the final comment representation. Finally, post-comment co-attention is used to consider mutual selection, and the information is aggregated for classification by mutual selection.

Model overview

In this paper, we proposed a novel model, Utilizing Information from Post and Comments to Detect Rumors (PostCom2DR). shows the framework of the proposed model. It consists of four components. (1) Text Encoding, (2) Global Representation for Post and Comments, (3) CNN-Based Comments Local Representation, (4) Rumor detection.

3.1. Text encoding

As rumors are intentionally written to mislead readers, there are differences between rumor and non-rumor in terms of words and language styles, which have the potential to help detect rumors. In order to make full use of the information of each word, post and comments are represented by a word-level encode.



3.2 Global representation for post and comments

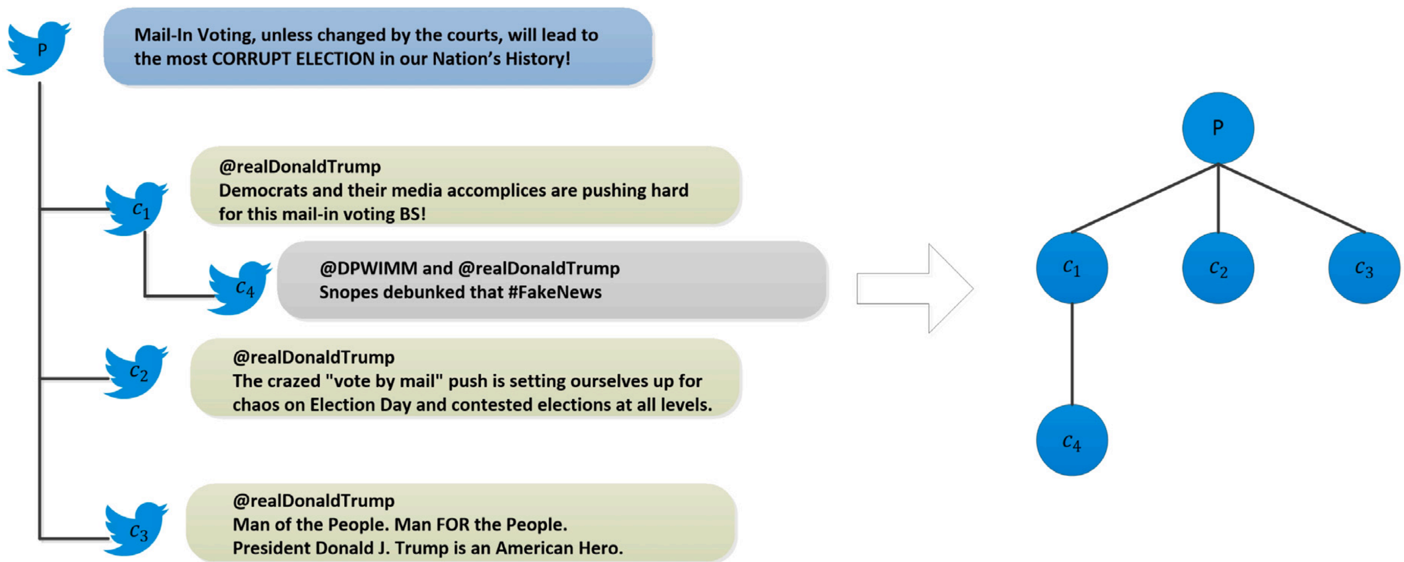
In this section, we set up four steps to get a global representation of post and comments. The first is to construct a reply graph. Then, a bilevel graph convolution network is built in the second part to integrate information through the reply graph and obtain enhanced comments representation. The third part is a comments self-attention mechanism, which is used to consider the influence of any comment. The fourth part is a post-comment co-attention that can learn the mutual selection between post and comments to help grasp the important information in them.

3.2.1 Construct reply graph

3.2.2 Bilevel graph convolutional network

3.2.3 Comments self-attention

3.2.3 Post-comment co-attention



3.3 CNN-based comments local representation

There is the topic drift within the comments. When making comments, users always take into account those topics that have been talked about during this period. Capturing comments in each time period is helpful to get the change of the comment topic and is useful for rumor detection. Here we use a convolutional network (CNN) to enhance the representation of comments, which we call the local representation. For the comment set C. We define $PC = \langle c_1, c_2, \ldots, c_n \rangle$ as the comment sequence arranged in chronological order. We first apply a 1- D convolution on T consecutive comment vectors, i.e., $\langle \mathbf{c}'_t, \ldots, \mathbf{c}'_{t+T-1} \rangle$ with a filter $\mathbf{W}_f \in \mathbb{R}^{T \times l}$, to produce a scalar feature $x_t \in \mathbb{R}$ according to the following formula

$$x_t = \mathrm{ReLU}\left(\mathbf{W}_f \cdot \mathbf{C}_{t:t+T-1} + \mathbf{b}_f\right)$$

where $\mathbf{C}_{t:t+T-1} = [\mathbf{c}'_t, \ldots, \mathbf{c}'_{t+T-1}]^{\mathbf{T}} \in \mathbb{R}^{T \times l}$ is a matrix of vector representations from the tth comment to the $(t + T - 1)$th comment on the comment sequence PC and $\mathbf{b}_f \in \mathbb{R}$ is a bias. Note that the vector representation of c_i is $\mathbf{c}'_i$, as shown in Fig. (Comments Encoder). The symbol ReLU (·) refers to the element-wise rectified linear unit function. We perform the same convolution operation with k filters to produce a multivariate feature vector $\mathbf{x}_t \in \mathbb{R}^{1 \times k}$. At the same time, in order to avoid the loss of information as much as possible, we proceeded with the padding size Pad. In the end, we obtain the final local representation $\mathbf{h}_{local} \in \mathbb{R}^{1 \times k}$ of the comments by the following operations:

$$\mathbf{h}_{local} = \max\left(\mathbf{x}_1, \ldots, \mathbf{x}_{n+2Pad-T+1}\right)$$

PHASE 2:

(a**) Limitations of proposed model in research paper**

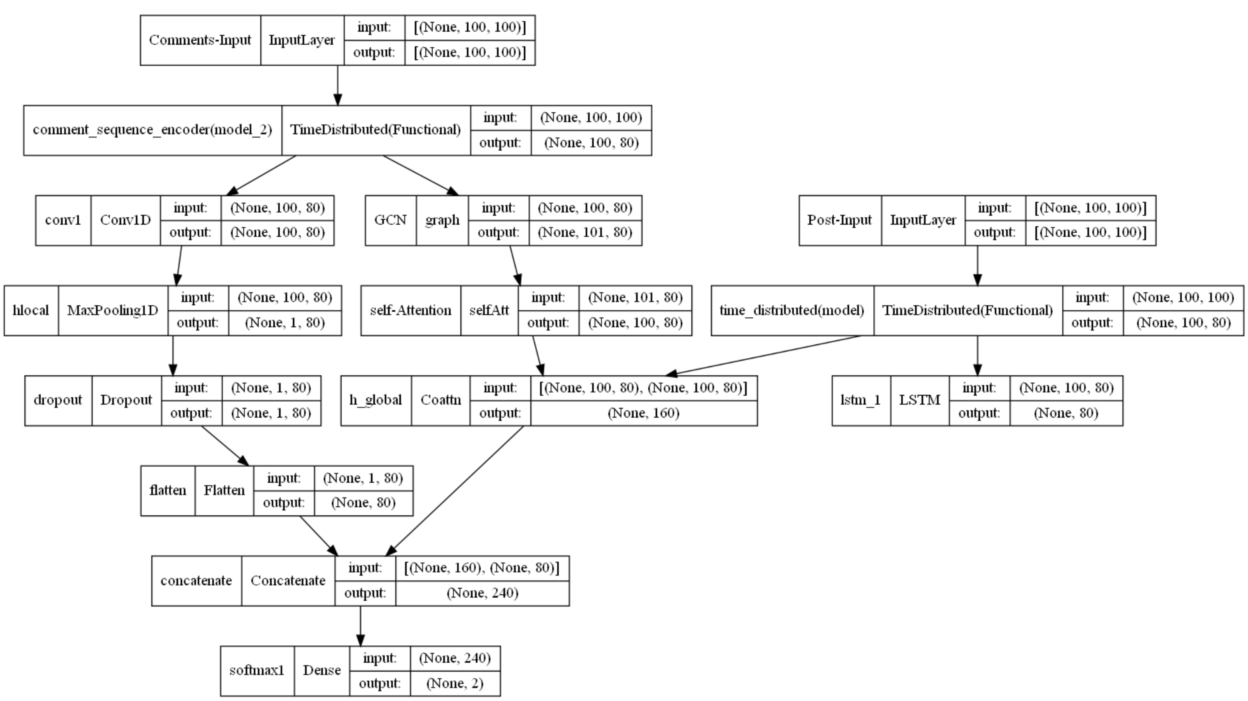
The current work of rumor detection is facing the challenge of interpretability, and it is important to provide a direct basis for the detection results. For this model they can use the mutual attention mechanism to select more post-related content from comments and vice versa. Therefore, the mutual attention mechanism module enables the model to eliminate noise or irrelevant information when aggregating information. However, the model failed to select explanatory comments for the results of rumor detection.

(b) **Objective of phase 2**

Objective is to implement the model given in the paper and improve its accuracy using the dataset GossipCop ,PolitiFact ,Twitter15 and Twitter16

c)**Explanation of intuitions behind objective**

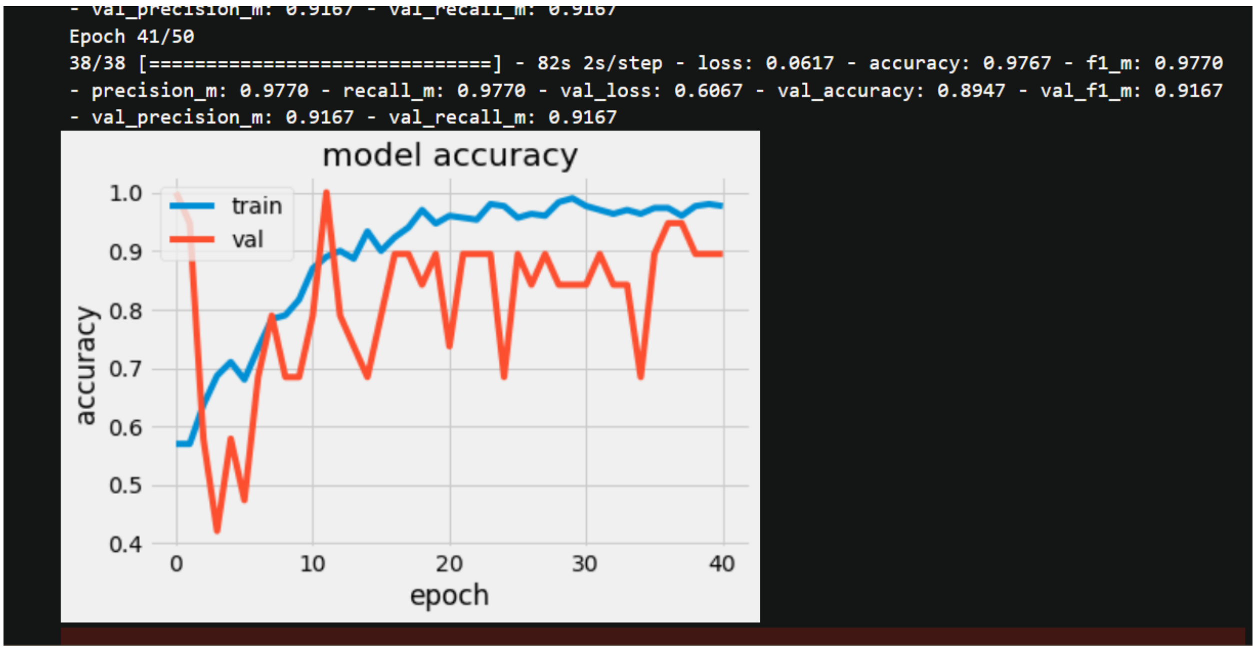
(d)**Supporting experimental setup.**



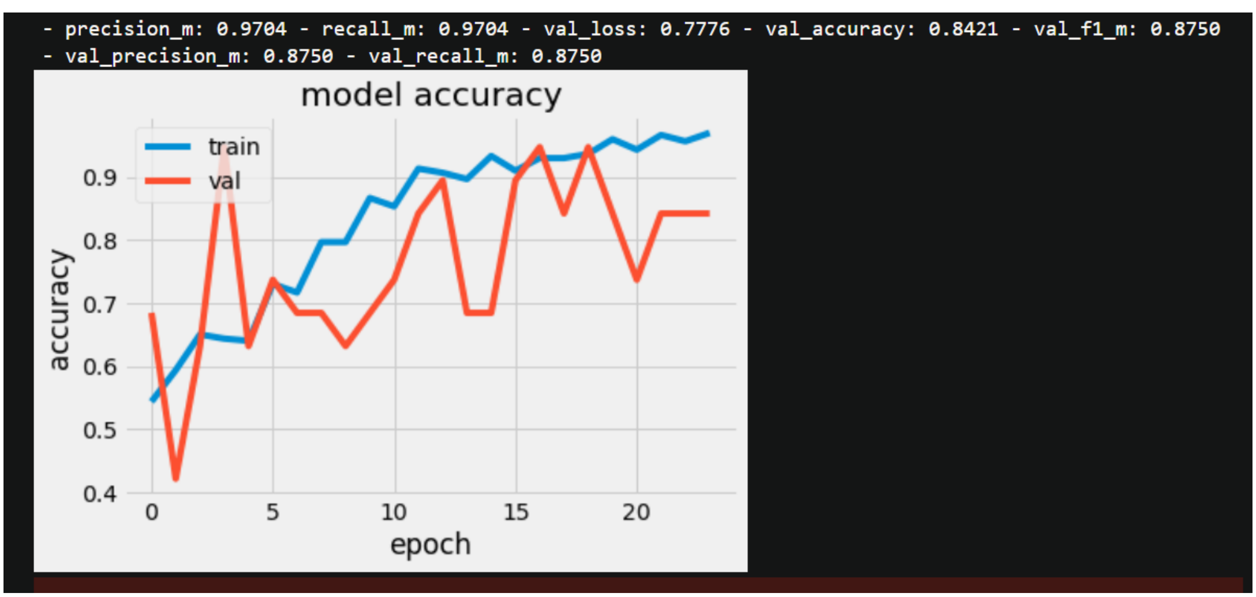
(f)**Results**

Model	Val Accuracy	Val F1 Score
t15	89.4	91.6
t16	84.2	87.5
GossipCop	77.3	77.5
Polit	78.4	79.3

Accuracy/F1 of twitter 15 data



Accuracy of twitter 16 data



Gossipcop after just 5 epochs gives val accuracy 77%

```
8294/8500 [===========================>.] - ETA: 3:53 - loss: 0.5157 - accuracy: 0.7801 - f1_m: 0.7801 - precision_m: 0.7801 - r
8295/8500 [===========================>.] - ETA: 3:51 - loss: 0.5157 - accuracy: 0.7801 - f1_m: 0.7801 - precision_m: 0.7801 - r
8500/8500 [==============================] - 10127s 1s/step - loss: 0.5161 - accuracy: 0.7798 - f1_m: 0.7798 - precision_m: 0.7798 - recall_m: 0.7798 - val_los
s: 0.5131 - val_accuracy: 0.7768 - val_f1_m: 0.7768 - val_precision_m: 0.7768 - val_recall_m: 0.7768
Epoch 3/5
8500/8500 [==============================] - 10114s 1s/step - loss: 0.4878 - accuracy: 0.8031 - f1_m: 0.8031 - precision_m: 0.8031 - recall_m: 0.8031 - val_los
s: 0.5343 - val_accuracy: 0.7774 - val_f1_m: 0.7774 - val_precision_m: 0.7774 - val_recall_m: 0.7774
Epoch 4/5
8500/8500 [==============================] - 10092s 1s/step - loss: 0.4625 - accuracy: 0.8172 - f1_m: 0.8172 - precision_m: 0.8172 - recall_m: 0.8172 - val_los
s: 0.5424 - val_accuracy: 0.7817 - val_f1_m: 0.7817 - val_precision_m: 0.7817 - val_recall_m: 0.7817
Epoch 5/5
8500/8500 [==============================] - 10147s 1s/step - loss: 0.4357 - accuracy: 0.8312 - f1_m: 0.8312 - precision_m: 0.8312 - recall_m: 0.8312 - val_los
s: 0.5751 - val_accuracy: 0.7752 - val_f1_m: 0.7753 - val_precision_m: 0.7753 - val_recall_m: 0.7753
```

(g)**Observation from experimental results.**

We see the model does a wonderful job in accurately predicting the rumor with a val accuracy of 91%

For twitter 15 data. The model is still running for gossipcop and politifact .(For 5 epoch , you can see the above accuracy)