Topics and Tools in Social Media Data Mining

Report on,
Early Fake News Detection

Submitted To: Dr. Sanasam Ranbir Singh

Submitted By:	**Ambuj Guhey**	**204101005**
	Saurabh Jaiswal	**204101052**
	Samiksha Agrawal	**204101069**
	Utkarsh Kahti	**204101058**

INTRODUCTION

Early research on fake news detection mainly focused on the design of effective features from various sources, including textual content, user profiling data, and news diffusion patterns. Linguistic features,such as writing styles and sensational headlines ,lexical and syntactic analysis, have been explored to separate fake news from fake news.

To solve the above problems, many recent studies apply various neural networks like recurrent neural network, convolutional neural network (CNN) , matrix factorization and graph neural network to automatically learn high-level representations for fake news detection.These methods only apply more types of information for fake news detection, but paying little attention to early detection.The main limitation of these methods is that they ignore the importance of publishers' and users' credibility for the early detection of fake news.

1) A design of structure-aware multi-head attention module to learn the structure of the publishing graph and produce the publisher representations for the credibility prediction of publishers.

2)Then application of the structure-aware multi-head attention module to encode the diffusion graph of the news among users and generate user representations for the credibility prediction of users.

3) A convolutional neural network to map the news text from word embedding to semantic space and utilize the fusion attention module to combine the news, publisher, and user representations for early fake news detection.

4) We have added another credibility of Propagation Path, how the news was spread based on the users which reposted the news. Modelled a Multivariate time series where at particular time instance which user has spread the news and it's features are driven by RNN and CNN both and then fully connected layer is used for classification.

Publisher Credibility

The structure-aware multi-head attention module has three input items: the query item, the key item and the value item.

The attention module first takes each node in the query and attends the adjacent relations of the graph structure into the attention module.

$$\mathbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_h = \mathbf{softmax}\left(\frac{\mathbf{Q}\mathbf{W}_h\mathbf{K}^T}{\sqrt{d}} \odot \mathbf{D}^{\mathbf{p}-\frac{1}{2}}\mathbf{A}^{pn}\mathbf{D}^{\mathbf{n}-\frac{1}{2}}\right)\mathbf{V}\,,$$

Now,we expand one head attention to multi-head schema: Q, K, and V are dispensed to h heads.

$$\mathbf{Z_h = Attention(P, N, N)_h\,,\ h \in [1, H]}$$

The output of SMAN is concatenated together through a fully-connected layer to predict publisher's credibility i.e "unreliable", "uncertain", and "reliable".

$$\mathbf{p_i(c|G(V_p, E), P\ ;\ \theta_1) = softmax(\tilde{P}_i W_p + b_p)}$$

User Credibility

The diffusion graph of news G(V u , E) records how news propagated from publishers to other users. The nodes Vu of the graph belongs to the user set and the edges denote the diffusion traces.

Finally, we use these users' representations $\tilde{\mathbf{R}} \in \mathbf{R}^{(|U| \times K \times d)}$ to predict the users' credibility scores, which can be formulated as follows:

$$\mathbf{p_{ij}(c|G(V_u\,, E), U;\ \theta_2) = softmax(\tilde{R}_{ij} W_r + b_r)}$$

where i ∈ [1, . . . , |U |] and j ∈ [1, . . . , K]. W_r ∈ R_d×|c| is a trainable matrix and |c| is the levels of credibility. b_r ∈ R|c| is a bias term.

Fusion Attention Unit

Firstly, we find publisher id p_i from the publishing and diffusion graph by news id m_j. Then, we look up publisher representation from all publisher representations table P˜ by publisher id p_i .

And by the same way, we look up user representations from all user representations table R˜ by publisher id p_i . R˜ $_i$ denotes K different users who had reposted the news m_j. We aggregate the reposted user embeddings $R\tilde{}_i$ ∈ $R^{K \times d}$ by an attention module.

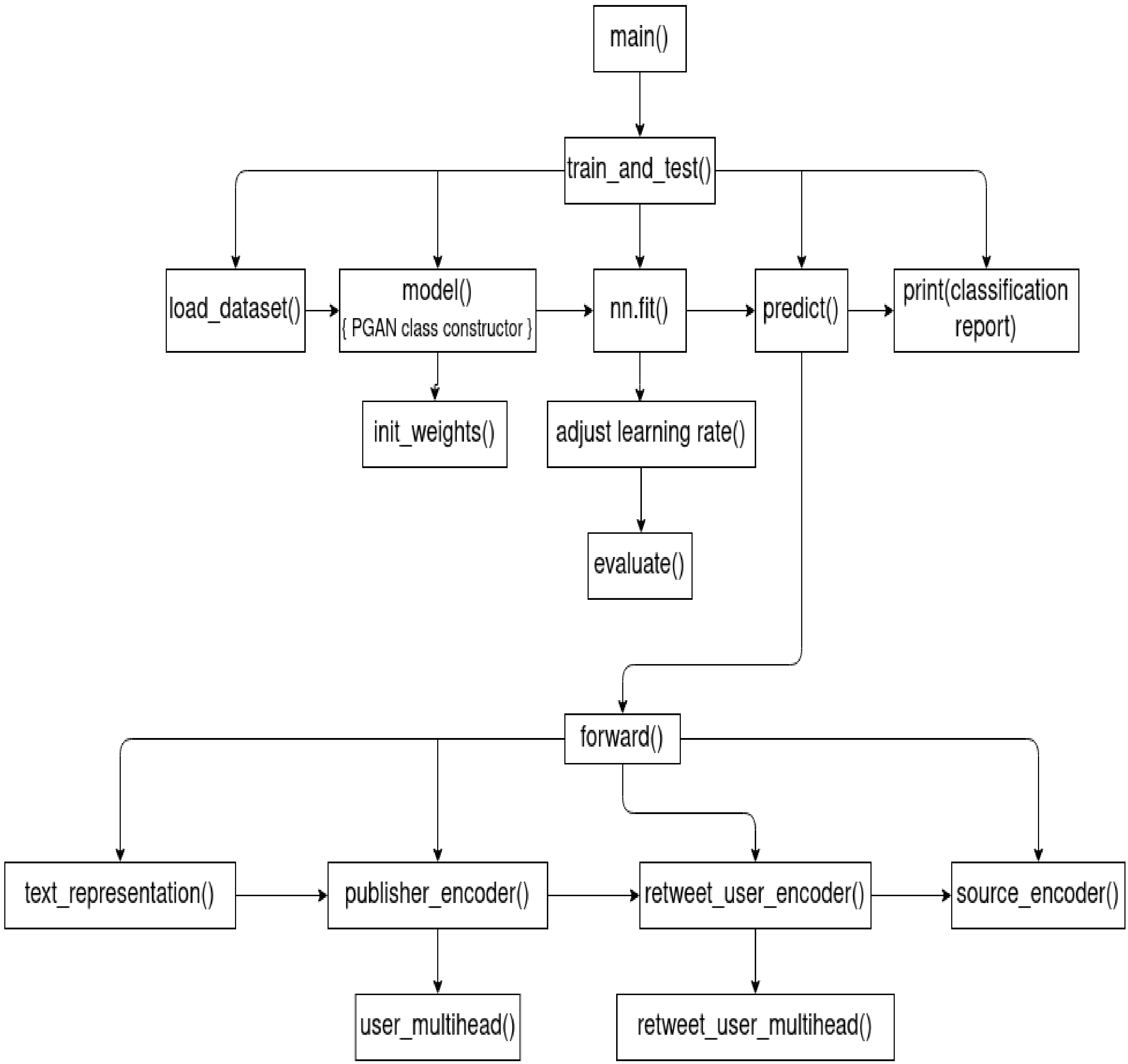
$$\mathbf{R}' = \sum_{k=1}^{K} \alpha_k \tilde{\mathbf{R}}_k \,, \quad \alpha = \mathbf{softmax}(\mathbf{N}_j \tilde{\mathbf{R}}_i^T) \,,$$

Then, we fuse the publisher representation and user combined representation by a heuristic method.

A fully-connected layer is applied to project the final representation into the target space of classes probability.

p(c|m_j ,P, N , U; θ_3) = softmax([m_j ; ˜m_j]W_m + b).

FLOW:


main()
train_and_test()
load_dataset()
model()
{ PGAN class constructor }
nn.fit()
predict()
print(classification report)
init_weights()
adjust learning rate()
evaluate()
forward()
text_representation()
publisher_encoder()
retweet_user_encoder()
source_encoder()
user_multihead()
retweet_user_multihead()

Propagation Path Credibility (Updations)

For early detection of fake news on social media by classifying news propagation paths.We model the propagation path of each news as a multivariate time series, in which each tuple denotes the characteristics of a user who engaged in propagating the news. Then, we build a time series classifier with both recurrent and convolutional networks to predict whether a given news story is fake. Recurrent and convolutional networks can learn global and local variations of user characteristics respectively, which are discriminate clues for fake news detection.

The proposed fake news detection model consists of four major components, i.e., propagation path construction and transformation, RNN-based propagation path representation, CNN-based propagation path representation, and propagation path classification, which are integrated together to detect fake news at the early stage of its propagation.

1. Given a news story propagating on social media, we first construct its propagation path by first identifying the users who engaged in propagating the news. Then, its propagation path is denoted as a variable-length multivariate time series.

2. We utilize a variant of RNN called Gated Recurrent Unit (GRU) to learn a vector representation for each transformed propagation path.

3. We also use convolutional networks (CNN) to learn another vector representation for each transformed propagation path.

4. After propagation paths are obtained through RNNs and CNNs respectively, they are concatenated into a single vector that represents the transformed propagation path.

Classification Report:

Original:

Precision	Recall	F1- Score	Support	
NR	0.912	0.869	0.890	84
FR	0.888	0.940	0.913	84
TR	0.851	0.881	0.865	84
UR	0.950	0.905	0.927	84
Accuracy			0.88	336
Macro avg	0.900	0.899	0.899	336
Weighted avg	0.900	0.899	0.899	336

Updates:

Precision	Recall	F1-score	Support	
NR	0.906	0.917	0.911	84
FR	0.898	0.940	0.919	84
TR	0.887	0.845	0.866	84
UR	0.916	0.905	0.910	84
Accuracy			0.902	336
Macro avg	0.902	0.902	0.901	336
Weighted avg	0.902	0.902	0.901	336

THANK YOU