TICNN for Fake News Detection: An Implementation

1st Naman Goenka 2nd Ayush Singh 3th Himanshu Pandey 4th Harshita Gupta
Department of Computer Science and Information Systems
Birla Institute of Technology and Science, Pilani
Rajasthan, India

***Abstract*—This implementation report has been written in partial fulfilment of the Data Mining Course under Dr. Yashwardhan Sharma, CSIS, BITS Pilani. It starts with problem description of the task, proceeds on to describe about state-of-the-art work in the field, then our implementation of [1] is presented and critically analyzed. Finally, we identify the research gaps, and present detailed description of our novel TI-CNN-TITLE-1000 model concluding with the future scope in the work.**

***Index Terms*—TI-CNN, Fake News, Explicit features, Latent features**

I. Problem Description

Fake News Detection is modelled as a **supervised binary classification problem**. Feature set (X^T_i,X^I_i) of i^{th} datapoint constitutes of text and image features respectively, which further are divided into latent and explicit features, derived by text and image information associated with the article. **Hence, fake news task is to learn a model T:(X^T_i,X^I_i) →y; where y describe predicted binary label of article.**[1]

II. Related State-of-the-art Work in brief

Fake news detection research is supported by various benchmark datasets such as FakeNewsNet, LIAR, BuzzFeedNews, BUZZFACE, CREDBANK, r/Fakeddit, COVID-19 Fake News Dataset, Grover-Mega etc. Different methodologies ranging from supervised to unsupervised learning combined with different type of features such as network related, linguistic, semantic, multimodal are been utilized in this field. Fakedetector [3] uses textual information extracted from politifact.com in a diffusive deep neural network performing better than most of the other models utilizing political textual information only. [4] attempts to use capsule neural network for the task with hypertuned on different n-gram sizes and embeddings. [5] explores a CNN-based semisupervised approach eliminating the need of large supervised datasets.

III. Implementation and its Scope

- Four models from paper were implemented, namely CNN-text-1000, LSTM-400, GRU-400 and TI-CNN-1000[2]. Primitive models using just conv nets on image data or logistic regression on text data exclusively were skipped because they were just used as baselines in [1], and were not deemed fit for the task.
- For CNN-text-1000, only textual explicit and textual implicit features are used; title latent features is not used.
- For LSTM-400 and GRU-400, text branch and content branch was formed with a embedding of 400*100. Finally bidirectional LSTM and GRU layers were added respectively. The two branches were merged using two dense layers followed by dropout in both the models. Finally, a sigmoid activation layer was used for binary label prediction.
- **Significant increase of performance can be observed in TICNN-TITLE-1000 model compared to TI-CNN-1000 model in our implementation as well as from results in [1]**.

[1] 1 - Fake; 0 - Real is the label mapping used in our implementation.
[2] Here the suffix denotes padded length for text inputs used

Fig. 1. Results taken from [1]

Method	Precision	Recall	F1-measure
CNN-text-1000	0.8722	0.9079	0.8897
LSTM-400	0.9146	0.8704	0.892
GRU-400	0.8875	0.8643	0.8758
TI-CNN-1000	0.922	0.9277	0.921

Fig. 2. Our implementation results including the novel TI-CNN-TITLE-1000;all networks using image are trained on dataset containing around 7k points only, rest models are on full dataset

Model	Testing Acc.	F1 Score	Precision	Recall
GRU-400	0.8721	0.8844	0.8783	0.8942
LSTM-400	0.931	0.9414	0.9476	0.9353
CNN-text-1000	0.946	0.9549	0.9589	0.9509
TICNN-1000	0.921	0.922	0.9206	0.9177
TICNN-TITLE-1000	0.9931	0.9946	0.9925	0.9967

IV. Research Gaps Identified

1) In supervised learning large amounts of good quality datasets are required; getting verified real articles is less problematic as they can be easily obtained from well-reputed news houses after minimal human verification. But **procuring articles related to fake or other particular fine grained labels in huge numbers is not an easy task** as illustrated by the collection process of TICNN dataset. 2) While most of available datasets are based on US politics, we need to develop **multilingual datasets and explore methodologies on other aspects of news** such as an socio-cultural emergency, sports,

entertainment, etc. 3) **Level of interaction in networks of different modalities needs to be made more complex** than simple interactions as in case of [1].

V. Our model :- TI-CNN-Title-1000

A. Preprocessing

Since publicly available dataset was not compatible with inputs required by the models mentioned, so decent amount of preprocessing was performed. Key steps in the process are highlighted. 1) Several columns from dataset such as country, language, uuid, id, publication_date, author name etc, are dropped off because either they don't tell any meaningful information about target attribute or are almost unique for an entry. 2) Data points having non empty title and text are only kept, accounting to 20015 data points. Target attribute is encoded via Label Encoder. 3) Keras inbuilt tokenizer is fitted upon **100 dimensional GLOVE embeddings, which is better than word2vec used in [1], however dimensions are kept same**, and text and title are transformed to sequences of indexes of words in GLOVE vocabulary. Glove embedding matrix is constructed and saved for future use. 4) Text and title sequences are post padded with 0 value to a length of 1000 (from [1]) and 93 respectively. 5) Using requests library, a script was written to fetch images urls mentioned in 'main_img_url ' column using custom headers while scraping which was robust against 404 error, connection errors, etc. 6) A blob of size 400 * 400 was used to resize the image with scaling factor 1.2 for preparing image data. 7) A pre-trained model was used to count faces in the extracted image. Using OpenCV [2], the image and its resolution were read. Finally, **7272 out of 20015 data points were extracted**. Rest of the data points were discarded mainly due to broken image links in the original publicly available dataset. **The train-test-validation split of the data - 6:2:2 on a dataset size of 7272 data points.**

B. Model Architecture

1) Text Branch
   - **Textual Explicit Features**- These features are based on the statistics of the text like the length of the news, the number of sentences, question marks, exclamations and capital letters, average number of words in a sentence, exclusive words, negations,First-person pronoun,Second-person pronoun. We represent these statistics as a vector of fixed size of 31. This vector is then passed through a fully connected layer to form explicit features.
   - **Textual Latent Features**- These features are based on the news text itself.Then we use a CNN(convolutional neural network) to extract latent features from the vector representation of news article as mentioned in preprocessing section.
   - **Title Latent Features**- These features are based on the news title. First we convert each word in the text into a 100-dimensional word embedding using GloVe.The title is max-padded to a length of 93 using 0 values. Then we use a CNN(convolutional neural network) to extract latent features from the vector representation of news article.

Fig. 3. TI-CNN-TITLE-1000 MODEL SPECIFICATIONS, for abbreviation refer Table III in [1]

Text Branch			Image Branch	
Textual Explicit	Textual Latent	Title Latent	Image Explicit	Image Latent
Input 31 X 1	Embedding 1000 X 100	Embedding 93 X 100	Input 3 X 1	Input 50 X 50 X3
	Dropout (0.5)	Dropout(0.8)		Conv2D
Dense 128				ReLU
	Conv 1D	Conv 1D	Dense 128	Dropout(0.8)
				MaxPooling 2D
	MaxPooling 1D	Max Pooling 1D		Conv2D
				ReLU
	Flatten	Flatten		Dropout(0.8)
				MaxPooling 2D
BN	Dense 128	Dense 128	BN	Conv2D
				ReLU
	BN	BN		Dropout(0.8)
				MaxPooling 2D
ReLU	ReLU	ReLU	ReLU	Flatten
				Dense 128
	Dropout(0.8)	Dropout(0.8)		BN
				ReLU
Merge			Merge	
Merge				
ReLU				
Dense 128				
BN				
Output				

2) Image Branch
   - **Image Explicit Features**- These features are based on the properties of the image. 3-dimensional vector containing resolution of image(height and width), number of faces in the image. This vector is then passed through a fully connected layer to form explicit features.
   - **Image Latent Features**- These features are based on the image itself. First we convert each image to a fixed size of 50 x 50. The vector of size 50 x 50 is thus passed through a CNN(convolutional neural network) to extract latent features from the vector representation of image.
3) **Experiment**
   - The outputs of textual explicit sub branch,textual latent feature sub branch and title latent feature sub-branch are added to create a single vector of size 128 x 1.
   - The outputs of image explicit and implicit feature sub branches are added.
   - The outputs of textual and image branch are then concatenated. A neural network with sigmoid activation is added which gives the labels of the news. Finally, network is trained using Adam optimizer and binary

VI. Conclusion and Scope of Future Work

Text-Image CNN (TICNN) [1] introduced provides efficient robust technique for multimodal fake news detection with capability of requiring less training time than recurrent network counterparts and being able to concentrate on only important parts of data. **A novel TI-CNN-TITLE-1000 model**

is also presented which helps in better modelling of article's title role as indicative component of an article. This approach was found to perform better than TI-CNN-1000 model from [1] on our preprocessed dataset. Developing defense mechanism against such high fake news dissemination remains a crucial research opportunity for the future. Multilingual research, application of image captioning methods, weight sharing methodology of siamese networks applied to TICNN can be explored in future.

REFERENCES

[1] Yang, Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. "TI-CNN: Convolutional neural networks for fake news detection." arXiv preprint arXiv:1806.00749 (2018).

[2] Bradski, G. (2000). The OpenCV Library. Dr. Dobbx27;s Journal of Software Tools.

[3] Zhang, J., Dong, B., Philip, S. Y. (2020, April). Fakedetector: Effective fake news detection with deep diffusive neural network. In 2020 IEEE 36th International Conference on Data Engineering (ICDE) (pp. 1826-1829). IEEE.

[4] Goldani, M. H., Momtazi, S., Safabakhsh, R. (2021). Detecting fake news with capsule neural networks. Applied Soft Computing, 101, 106991.

[5] Dong, X., Victor, U., Qian, L. (2020). Two-Path Deep Semisupervised Learning for Timely Fake News Detection. IEEE Transactions on Computational Social Systems.