Fake News Detection using Hierarchical Convolutional Attention Network

Karish Grover
Roll No. 2019471
IIIT Delhi
karish19471@iiitd.ac.in

Himanshu Singh
Roll No. 2017291
IIIT Delhi
himanshu17291@iiitd.ac.in

Nirali Arora
Roll No. Phd21002
IIIT Delhi
niralis@iiitd.ac.in

Abstract

In today's digital era, social media platforms' popularity has increased exponentially, leading to an increase in the amount of unreliable information online. Verifying the integrity of a particular piece of news is not easy for any end-user. In this paper, our end goal is to design an efficient model to examine the reliability of a news piece. This paper addresses the problem of Fake News classification, given only the news piece content and the author name. We present, Hierarchical Convolutional-Attention Network (**HCAN**) composed of attention-enhanced word-level and sentence-level encoders and a CNN to capture the sequential correlation. Extensive experiments show that HCAN outperforms the state-of-the-art baselines models on a Kaggle dataset.

1 Introduction

Social media has become a necessitous part of people's lives. Users can interact, express their views, comment on others' views, and access news through various social media platforms. Some people make unjust use of these mediums to spread misinformation.

As fake news pieces are intentionally created to spread inaccurate information and lure people into believing them, they often have opinionated and sensational language styles, which can help verify the integrity of the news piece. In addition, a news document contains linguistic cues at different levels, such as word-level and sentence-level, which provide different degrees of importance for the explainability of why the news is fake. For example, consider an unreliable news piece, *Iranian woman jailed for fictional unpublished story about a woman stoned to death for adultery*. The words *jailed* and *fictional* contributes more in deciding whether the news claim is fake or not, as compared

Figure 1: Motivation behind relative importance of words and sentences in a news piece.

to the other words in the sentence. Further, as shown in Figure 1, the highlighted sentences and colored words are more crucial towards the final prediction as compared to the others parts of the news piece.

Thus, we can significantly improve our model performance by capturing the relative importance of the words and sentences in the news piece towards the final prediction. We consider Fake news detection as a binary classification problem and employ a hierarchical convolutional attention network (**HCAN**) for the issue of fake news classification. A CNN is used to capture the sequential context of the word tokens in the news piece. Then, sentence representations are built by encoding the outputs of the CNN and applying the attention mechanism on them, resulting in a sentence vector. In the second level of the hierarchy, the final news piece representation is built in the same way by attending over the sentence vectors of each sentence of the news piece as input. This way, we capture the semantics and the sequential context of the original news piece.

Another motivation behind employing such an architecture comes from Table 1 which shows that the average length of a news piece in the used dataset is 774. Simple RNN models like LSTM and GRU can lead to information bottlenecks and van-

ishing gradients for such long texts. Thus, splitting the news piece into words and sentences is crucial to get their relative importance towards the final prediction and tackle different parts of an otherwise long text separately. Adding a CNN helps capture the sequential context. There are possible patterns in how fake news is written, and capturing them can boost model performance.

2 Related Work

There are broadly three existing approaches for detecting/curbing fake news, (1) News Content-based, (2) Propagation-based, and (3) Social/User context-based. Typical methods rely on the content of news articles/tweets to extract the textual features, such as TFIDF/n-gram, and apply supervised learning (e.g., random forest, support vector machine, and Naive Bayes) for binary classification (Shu et al., 2017).

State-of-the-art approaches like (Sheng et al., 2021) jointly model the news content to extract user- and fact- preference maps. Some consider the polarization in the social network between trustworthy and non-trustworthy users and observe the patterns in news diffusion (Monti et al., 2019). AENet (Jain et al., 2021) uses attention enabled neural architecture using contextual features related to the user (e.g., location, time of the speech, etc.) and the news piece to detect fake news.

Attention-based mechanisms have proven to be very effective because of their ability to stress upon the most important parts of a text. (Roy et al., 2018) used deep learning ensemble architecture based on CNN and Bi-LSTM to capture hidden features and information in both directions, respectively, and (Long, 2017) proposed a hybrid attention-based LSTM model. In (Popat et al., 2018), the authors concatenate the claim text and content of the article and apply word-level self-attention to distinguish between real and false claims.

3 Methodology

The model architecture is shown in Figure 2. It can be divided into three broad parts: (1) CNN-based Representation, (2) Word-level Encoder, and (3) Sentence-level Encoder.

3.1 CNN-based Representation

Convolutional Neural Networks (CNN) are good at learning the sequential correlation. We use this to our advantage. The input glove embeddings of the tokens are passed through a 1D Convolutional layer. We consider λ consecutive tokens at one time to model their sequential correlation (Lu and Li, 2020), i.e., $\langle \mathbf{x}_t, ..., \mathbf{x}_{t+\lambda-1} \rangle$. Hence the filter is set as $\mathbf{W}_f \in R^{\lambda \times v}$. Then the output representation vector $\mathbf{C} \in R^{d \times (t+\lambda-1)}$ is given by

$$\mathbf{C} = ReLU(\mathbf{W}_f \cdot \mathbf{X}_{t:t+\lambda-1} + b_f) \quad (1)$$

where $\mathbf{W}_f$ is the matrix of learnable parameters, $ReLU$ is the activation function, $\mathbf{X}_{t:t+\lambda-1}$ depicts sub-matrices whose first row's index is from $t = 1$ to $t = n - \lambda + 1$, and b_f is the bias term.

3.2 Word Encoder

We learn the sentence representation via a recurrent neural network (RNN) based word encoder. Though theoretically, RNN is able to capture long-term dependencies, but in practice, it leads to information bottleneck and vanishing gradients and it seems to "forget" the information. To capture long-term dependencies of RNN, we employ a bidirectional GRU. GRUs are known for a more persistent memory.

The CNN-based representation is fed into the word encoder. The word-level representations are obtained by concatenating the forward and backward hidden states of the BiGRU. This helps in capturing the word-level context. Next, by using a one-layer MLP and softmax function, we calculate the normalized importance (attention) weights over the word representations generated by the word encoder. Then, we compute the sentence vector as a weighted sum of the word representations based on the word-level attention weights.

3.3 Sentence Encoder

In a similar way with word encoder, use a bidirectional GRU to encode the sentences to get a sentence-level representation. This helps in capturing the sentence-level context. Sentence Attention. Similar to word-level attention, we use a one-layer MLP and softmax function to get the weights over sentence annotations. Then, we calculate a weighted sum of the sentence annotations based on the sentence level attention weights to get the overall document/news piece vector.

3.4 Final Classification

We finally use the a MLP and softmax function to calculate the probability of all classes.

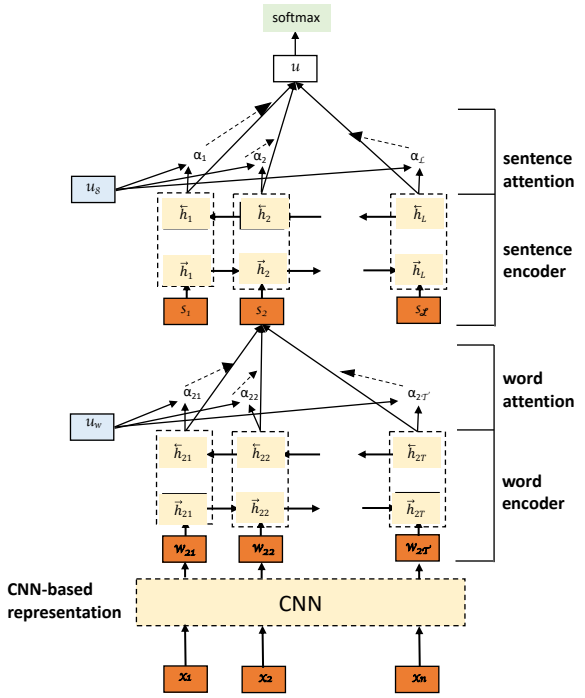

Figure 2: Model Architecture.

4 Experimentation

4.1 Dataset and Preprocessing

We use the Kaggle Fake news detection dataset for our task. The statistics of the dataset are mentioned in Table 1. There are three attributes in the dataset, `author`, `title`, and `text`. We concatenate all the three features in order to make the final predictions. This is because the credibility of an author plays a very crucial role in determining the reliability of a news piece. Further, many times a news title has a particular writing style or phrases, and by detecting such patterns one can be more certain about a news article. Next, we remove the stop words and punctuations to further process our dataset. All the experiments are performed on a 80:20 train-test split.

4.2 Baselines

We evaluate the dataset collected on several baselines, as listed in Table 2. We evaluate the performance of these baselines using F1 score, Recall, and Precision. The implementation of these baselines has been released. We implement three types of baseline models, Simple Linear Classification Models, Deep Neural Network Models, and Pre-trained Language Models.

- **LR** (Logistic Regression), **DT** (Decision Trees), and **RF** (Random Forest). Trained using TFIDF vectors of the input text.

- **CNN**(Convolutional Neural Networks). 1D Convolutional layer with kernel size 3, followed by max pooling and a fully connected Dense layer. All deep neural models have been trained for a maximum input length of 70.

- **RNN**(Recurrent Neural Networks), **LSTM**(Long Short Term Memory cells), **GRU**(Gated Recurrent Networks), **Bi-RNN**(Bi-directional Re-current Neural Networks). Respective RNN followed by dropout and fully connected layers.

- **RCNN**(Recurrent Convolutional Neural Networks). Uses Bidirectional GRU to encode the Glove embeddings of the tokens, 1D Convolutional layer, followed by a max pooling and dropout layer.

- **BERT** (Bidirectional Encoder Representations from Transformers), **RoBERTa** (Robustly Optimized BERT Pretraining Approach). Huggingface implementation of the `bert-base-cased` and `roberta-base` model finetuned using the `AdamW` optimizer, with a learning rate of `4e-5` and batch size of 8 for 3 epochs on NVIDIA Tesla V100 GPU.

Table 1: **Dataset Statistics.**

	Stats
#Reliable news pieces	10387
#Unreliable news pieces	10413
#max. posts by single author	243
max. length of news text	24234
avg. length of news text	774.65
max. length of title	72
avg. length of title	12.43
#NaN values in author	1957
#NaN values in text	39
#NaN values in title	558

4.3 Baselines Ablation Study

We perform a series of experiments to evaluate the baseline models under various conditions. The simple linear classification models are trained on the TFIDF representations of the news text. We perform experiments for word-level, character-level and n-gram level TFIDF vectors. In almost all

Table 2: **Experimentation Results and Baselines**

Model	Prec	Rec	F1
LR *w/* word-tfidf	0.9694	0.9643	0.9669
LR *w/* char-tfidf	0.9613	0.9650	0.9631
LR *w/* ngram-tfidf	0.9718	0.9296	0.9502
RF *w/* word-tfidf	0.9121	0.9675	0.9390
RF *w/* char-tfidf	0.9675	0.9684	0.9680
DT *w/* word-tfidf	0.9517	0.9642	0.9579
DT *w/* char-tfidf	0.9642	0.9777	0.9709
LSTM	0.9322	0.9384	0.9353
GRU	0.9637	0.9202	0.9414
GRU *w/o* Dropout	0.9274	0.9422	0.9348
GRU *w/* trainEmb	0.9522	0.9609	0.9566
BiGRU	0.9326	0.9362	0.9344
BiGRU *w/o* Dropout	0.9656	0.9149	0.9396
BiGRU *w/* trainEmb	0.9646	0.9439	0.9542
CNN	0.9389	0.9221	0.9304
CNN *w/* trainEmb	0.9742	0.9222	0.9475
CNN *w/o* Dropout	0.9680	0.9358	0.9516
RCNN	0.9689	0.9018	0.9341
RCNN *w/* trainEmb	0.9145	0.9642	0.9387
RCNN *w/o* Dropout	0.9274	0.9400	0.9336
RCNN *w/* UniGRU	0.9584	0.9317	0.9449
RCNN *w/* BiLSTM	0.9665	0.8964	0.9301
RoBERTa	0.9781	0.9753	0.9789
BERT	0.9781	0.9772	0.9802
HCAN *w/o* Sent.E.	0.9410	0.9335	0.9373
HCAN *w/o* CNN	0.9877	0.9605	0.9739
HCAN *w/* UniGRU	0.9632	0.9869	0.9749
HCAN (Ours)	**0.9891**	**0.9835**	**0.9863**

cases, the character level TFIDF vectors boost the performance over the other two. Further, in case of deep neural models, we evaluate the effect of dropout layer and the effect of making the embedding layer trainable. It can be seen that making the embedding layer trainable improves accuracy over the untrainable version. The pretrained language models BERT and RoBERTa which were fine tuned on the dataset, perform better than other baselines. This is because of the sophisticated architecture that these models have. Further, all the deep neural models have comparable performance in terms of the F1 score.

5 Results and Analysis

It can be seen from the table that our system HCAN outperforms all the baseline models by a decent margin and gets an F1 score of **0.9856**. We further analyse the effect of the Convolutional layer and the hierarchical sentence encoder on the model performance. It can be seen that on removing the hierarchy i.e. by considering only the word-level encoder i.e. an attention enhanced bi-directional network, the accuracy degrades the most. Further, removing the CNN layer from the model also leads to poor performance. We also analyse the effect of the CNN kernel size and word-encoder max-lengths on the testing and training accuracies. It can be seen from Figure 3, that the testing accuracy first decreases and then increases with the increase in kernel size. Also, from Figure 4 it can be observed that with increase in maximum length of text sequence in encoders, the testing and training accuracies increase.

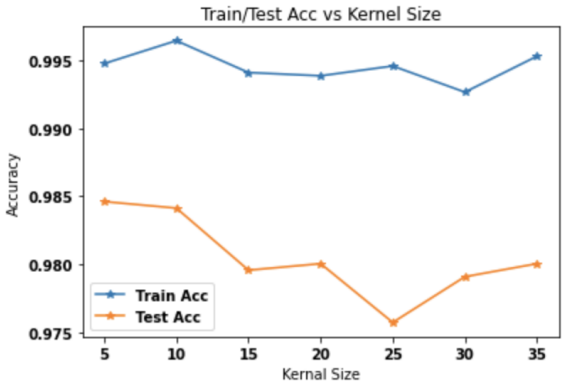

Figure 3: Variation of Accuracy with CNN Kernel Size.

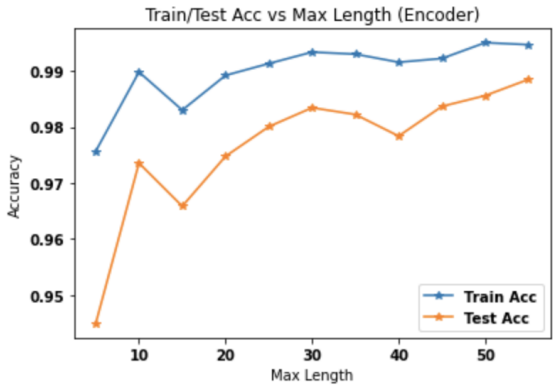

Figure 4: Variation of Accuracy with Encoder Max Length.

The hierarchical structure of our model, enhanced by a CNN, outperforms even the state-of-the-art pretrained language models like BERT and RoBERTa. This can be related back to our starting motivation to model more important parts of a news piece to make the prediction. Our motivations are supported by the experimental results.

References

Vidit Jain, Rohit Kumar Kaliyar, Anurag Goswami, Pratik Narang, and Yashvardhan Sharma. 2021. Aenet: an attention-enabled neural architecture for fake news detection using contextual features. *Neural Computing and Applications*, pages 1–12.

Yunfei Long. 2017. Fake news detection through multi-perspective speaker profiles. Association for Computational Linguistics.

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. *arXiv preprint arXiv:1809.06416*.

Arjun Roy, Kingshuk Basak, Asif Ekbal, and Pushpak Bhattacharyya. 2018. A deep ensemble framework for fake news detection and classification. *arXiv preprint arXiv:1811.04670*.

Qiang Sheng, Xueyao Zhang, Juan Cao, and Lei Zhong. 2021. Integrating pattern-and fact-based fake news detection via model preference learning. *arXiv preprint arXiv:2109.11333*.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.