Rumour Detection on Microblogs Tweets

Submitted by

Fang Zhuohao	G2304641D
Jiang Kuncen	G2304840L
Li Fangyu	G2304806A
Wang Wenxin	G2304475D

1. Introduction

The information explosion in modern society and the popularity of social media have led to the existence and wide spread of fake news. Fake news will not only mislead susceptible people but may also induce adverse social events. Therefore, the detection of fake news is particularly important. Traditional fake news detection methods mainly rely on human judgment, but this method is costly and subjective, and cannot meet the needs of large-scale fake news detection. In recent years, with the development of artificial intelligence technology, automated fake news detection methods have gradually become a research hotspot.

Existing fake news detection methods are mainly divided into social propagation-based methods and news content-based methods. Compared with the method based on social propagation, the method based on news content can predict the authenticity of news only through news content, which is simple and efficient.

In this work, we use three different machine learning models to extract and classify the features of news based on news content-based methods to achieve the identification of fake news. Three models are the BERT text pre-training model from the Hugging Face library, a graphic fusion model based on the pre-training model with a multimodal processor, with the BERT-LSTM combination model. This work uses publicly available datasets Twitter for fake news detection experiments and performance comparison of the models is carried out.

2. Data

2.1 Data Collection

We evaluated our models using the MediaEval Verifying Multimedia Use (MMU) dataset, designed to combat falsified media on social platforms. The dataset contains various types of multimedia content, including images, videos, and audio, sourced from platforms like Twitter, Instagram, and YouTube. Our focus was primarily on the Twitter dataset within MMU.

One key aspect of the MMU dataset is its labels, which cover authenticity, source credibility, potential edits, and copyright status. The core task involves developing algorithms to assess content credibility and authenticity. Evaluation metrics include accuracy, precision, and recall.

The MMU benchmark has sparked interest in multimedia processing and computer vision, benefiting fields like social media, news, and entertainment. This dataset aids in the development of more effective multimedia content verification algorithms.

The dataset comprises a development subset (used for training) and a test subset (used for testing), providing a consistent evaluation environment.

2.2 Data Description

This dataset has two parts: the development set and test set. We use the development as training set and test set as testing set to keep the same data split scheme. The tweets in the Twitter dataset contain text content, attached image/video and additional social context information.

Table 2-1 Data Volume

Method	Twitter
# of fake News	7898
# of real News	6026
# of image	8136

Table 2-2 Dataset Segmentation

Class	Amount
Training Set	3766
Validation Set	620
Testing Set	1045

3. Model

3.1 Pre-trained BERT Model (Base Model – Text only)

Pre-trained models address data scarcity in machine learning. Training models requires ample labelled data, which is expensive and often lacking in domains like healthcare, finance, and law. Pre-trained models learn from unlabelled data, extract general features, and adapt to different tasks, reducing the need for labelled data and offering solutions for data scarcity.

The structure of the BERT pre-training model based on textual information is shown below. Next, this section provides a brief description of the model.

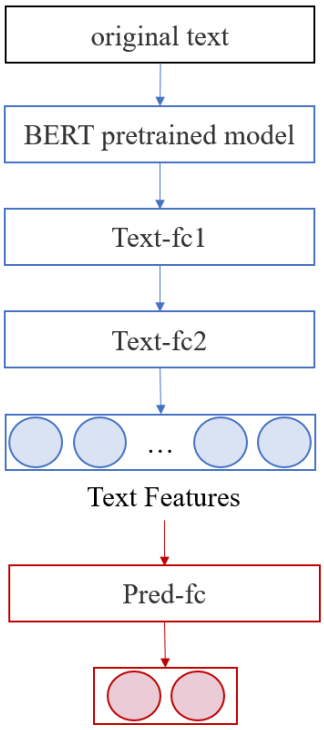

Figure 3-1 Model structure of text-only model

Data Processing

The pre-processing process of the text in the experiment is as follows: read the text into the deactivation word list, use the thesaurus to perform the deactivation operation and text cleaning, filter out some common words with no practical meaning, such as prepositions, conjunctions, pronouns, etc.; the removal of deactivation can reduce the dimensionality of the textual data and the noise, so as to improve the efficiency and accuracy of the text processing.

Textual Feature Extractor (BERT Pre-trained Model)

The pre-processed post text T is defined as $T = \{t_1, t_2, \dots, t_w\}$ (where w is defined as the number of characters in the sentence). T is input to BertTokenizer, ultimately

generates word embeddings $X = \{x_1, x_2, \ldots, x_n\}$ and corresponding masks $M = \{m_1, m_2, \ldots, m_n\}$, where n is the length of the word embedding sequence, and each word vector has a dimension of d. This completes the encoding of the input text.

The word embeddings X and masks M are input to the BERT model to generate 768-dimensional sentence vectors. This model utilizes 12 Transformer encoder layers, with each layer having 12 self-attention heads. Through a multi-head attention mechanism, BERT can simultaneously consider feature representations from different subspaces, better capturing contextual information in the input sequence and obtaining richer representation capabilities.

After multi-head attention, a combination of residual connection and layer normalization generates the Self-Attention layer's output, denoted as $Self - Attention_{\text{out}}$.

$$Self - Attention_{\text{out}} = \text{LayerNorm}(MultiHead(Q, K, V) + X)$$

Next, $Self - Attention_{\text{out}}$ is input to a feedforward neural network, with the feedforward layer involving two linear transformations and an activation function. Typically, the linear transformation dimensions are expanded and then reduced back to the original dimension, increasing the model's representational capacity to capture input features more effectively:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

where x is the input vector $Self - Attention_{\text{out}}$, W_1 and b_1 are the weight matrix and bias vector of the first layer, $\max(0,\ xW_1 + b_1)$ represents the ReLU activation function, and W_2 and b_2 are the weight matrix and bias vector of the second layer.

In the end, the output of the feedforward layer, combined with the Self-Attention layer's output and normalized, is passed to the next Encoder layer to produce the current Encoder layer's final output.

$$Feedforward_{\text{out}} = \text{LayerNorm}\big(FFN(Self - Attention_{\text{out}})\big) + Self - Attention_{\text{out}}$$

The final output of the BERT part is assigned as $R_{T_{Bert}}$.

FC Layer

Through two fully connected layers with sizes 2742 and p, the text feature vector R_T is finally generated. The p-dimensional text feature representation is denoted as $R_T \in R_p$. W_{t1_f} and W_{t2_f} are the weight matrices of the two fully connected layers in the text feature extractor. $\sigma(\cdot)$ represents the ReLU activation function. The operations in the last two layers of the text feature extractor can be expressed as:

$$R_T = \sigma\left(W_{t2_f} \cdot \sigma\left(W_{t1_f} \cdot R_{T_{Bert}}\right)\right)$$

Prediction Layer

Assume the feature of a news text m_i as $R_{T(m_i)}$, which is fed into the fake news detection module for prediction. The output of the model is the probability that a news item is false:

$$P_\theta(m_i) = G_d\left(R_{T(m_i)}, \theta_d\right)$$

where $G_d(., \theta_d)$ is the parameter of the fully connected layer, and θ_d indicates all parameters included. The cross-entropy loss function is adopted to measure the discrepancy between the model's predicted values and the actual values.

$$L_d\left(\theta_f, \theta_d\right) = -E_{(m,y)\sim(M,Y_d)}\left[y \log\left(P_\theta(m)\right) + (1-y)\log\left(1 - P_\theta(m)\right)\right]$$

We minimise the detection loss function $L_d\left(\theta_f, \theta_d\right)$ by seeking the optimal parameters $\widehat{\theta_f}$ and $\widehat{\theta_d}$. The process can be expressed as:

$$\left(\widehat{\theta_f}, \widehat{\theta_d}\right) = \arg\min_{\theta_f, \theta_d} L_d\left(\theta_f, \theta_d\right)$$

3.2 LSTM-BERT Model (Improved Model 1)

In modern deep learning techniques, there's a notable trend: models integrating LSTM with BERT stand out due to their prowess in handling large volumes of text. The key lies in LSTM's ability to manage long temporal sequences and BERT's expertise in grasping the nuances of textual context. This segment provides an in-depth look at the structure and operational details of the pre-trained LSTM-BERT model.

LSTM-BERT architecture also utilizes BERT, a renowned pre-trained model, to derive bidirectional encoder representations from transformers. BERT's main objective is to produce vector representations with fixed sizes for each word, based on its surrounding context. Before the actual processing with BERT, the original text, symbolized as T, undergoes a tokenization process. The outcome of this process is a series of word embeddings, represented as $X = x_1, x_2, \ldots, x_n$, where n represents the number of words in the text. These word embeddings serve as the input to the LSTM-BERT model.

LSTM (Long Short-Term Memory) is an advanced iteration of the Recurrent Neural Network (RNN). This variant is tailored to address issues associated with time series or

sequences. For the LSTM-BERT combination in our study, the LSTM's role is pivotal in recognizing and processing long-term dependencies within the text. After transforming text into word embeddings via BERT, these embeddings are fed into the LSTM. The result is a series of state representations, denoted as h_t, which compile information up to the present moment.

Given the output from BERT as $O = o_1, o_2, \ldots, o_n$, where each o_i is the vector representation of the corresponding word x_i, the hidden state h_t of the LSTM at time step t can be represented as:

$$ht = LSTM(ot, ht - 1)$$

where, h_{t-1} symbolizes the hidden state from the preceding time step. The figure below shows the whole structure of the LSTM-BERT model.

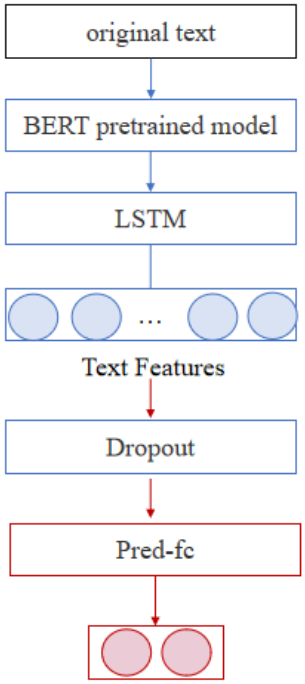

Figure 3-2 Model structure of LSTM-BERT model

Within the LSTM-BERT model, BERT is responsible for generating contextual word vectors, while LSTM captures long-term dependencies within these vectors. First, text T is transformed by BERT into a sequence of word embeddings, O. These vectors are then fed into the LSTM network, resulting in a sequence of hidden states $H = h_1, h_2, \ldots, h_n$. The final output of the model is h_n, which represents the vectorised representation of the whole text.

For classification tasks, h_n undergoes a dropout layer to mitigate overfitting, followed by a fully connected layer (FC) to produce the terminal output. Assuming a binary classification objective, the FC layer would output a dimensionality of 2, indicating the values for the two respective categories.

$$y = FC(dropout(h_n))$$

Within the given formula, 'y' denotes the ultimate result generated by the model, which provides the respective ratings for both classes. Through the integration of BERT's adeptness at understanding context and LSTM's proficiency in handling extended sequences, the LSTM-BERT pre-trained model stands as a robust apparatus for textual categorization endeavours. Such a combination enhances the model's capacity to capture intricate contextual undertones, thereby ensuring heightened accuracy in classification decisions.

3.3 Multi-modal Feature Model (Improved Model 2)

The development of social media promotes the diversity of news content forms, much news contains not only textual information. Multimodal fake news detection methods have gradually emerged in recent years and have shown better performance in detection compared to traditional text detection. Therefore, we attempt to fuse multimodal detection, considering the image information in the dataset, trying to improve the model performance.

The structure of the multi-modal feature model based on textual and visual information is shown below.

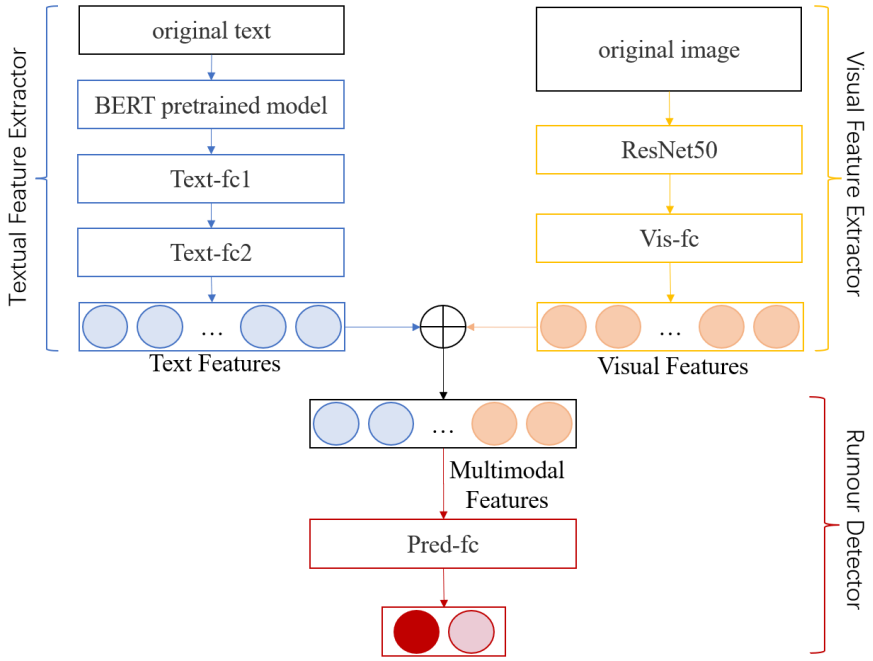

Figure 3-3 Model structure of multi-modal feature model

Data Pre-processing for Image

The pre-processing of a jpg image involves extracting data from the RGB colour channel, resizing the image to 256x256, and subsequently cropping it to 224x224

dimensions at the centre. The image is then converted into a tensor image. Following the tensor conversion, mean and standard deviation adjustments are applied to normalize the tensor image. This normalization ensures consistent pixel distributions across various images, contributing to enhanced model training stability and convergence speed.

Visual Feature Extractor

To efficiently extract visual features, we employ ResNet50, a variant of the deep residual network designed to address gradient-related issues in deep neural network training. ResNet50 excels in tasks such as image classification, object detection, and image segmentation.

The additional image of the post is the input to the visual feature extractor and is denoted as V. The last layer of the ResNet50 network was modified in the experiments to adjust the dimensionality of the final visual feature representation to p. The parameters of the pre-trained ResNet50 neural network were kept unchanged during co-training with the text feature extractor to avoid overfitting.

Denote the p-dimensional visual feature representation as $R_V \in R_p$, the operation of the last layer in the visual feature extractor can be represented as:

$$R_V = \sigma\left(W_{v_f} \cdot R_{V_{ResNet}}\right)$$

where $R_{V_{Resnet}}$ is the visual feature representation obtained from ResNet50 and W_{v_f} is the weight matrix of the fully connected layer in the visual feature extractor.

Feature Fusion

Next, the textual feature representation R_T and the visual feature representation R_V are concatenated to form a multimodal feature representation denoted as $R_F = R_T \oplus R_V \in R^{2p}$, which is the output of the multimodal feature extractor. Denote the multimodal feature extractor as $G_f(M; \theta_f)$, where M is a set of multimodal posts that are fed into the multimodal feature extractor, where θ_f denotes the parameters to be learnt.

4. Result

This task is a binary classification problem, so **Accuracy**, **Precision**, **Recall**, and **F1-measure** are used as the evaluation metrics of the model. The closer the accuracy, precision, and recall scores are to 1, the better the model's classification ability. F1-value combines the precision and recall metrics and is often considered a better measure than accuracy.

Parameter Setting

Table 4-1 Parameters Setting

Parameter	Value
Batch size	20
Learning rate	0.01
Optimizer	Adam
Dropout rate	0.4
Training loop	100

Experiment is based on Google Colaboratory, with Python 3.10.8, PyTorch version 1.13.1 and transformers-4.34.1.

Confusion Matrix

(a) (b) (c)

Figure 4-1 Comparison Confusion Matrix of 3 Proposed Model

Classification Result

Classification report:

	precision	recall	f1-score
0	0.824	0.933	0.875
1	0.667	0.400	0.500
accuracy			0.800
macro avg	0.745	0.667	0.688
weighted avg	0.784	0.800	0.781

Figure 4-2 Classification Report of Text-only Model

Classification Report:

	precision	recall	f1-score
0	0.77	0.92	0.84
1	0.91	0.75	0.82
accuracy			0.83
macro avg	0.84	0.84	0.83
weighted avg	0.84	0.83	0.83

Figure 4-3 Classification Report of LSTM-BERT Model

Classification report:

	precision	recall	f1-score
0	0.863	0.911	0.886
1	0.819	0.734	0.774
accuracy			0.849
macro avg	0.841	0.823	0.830
weighted avg	0.847	0.849	0.847

Figure 4-4 Classification Report of Multi-modal Model

In the task of fake news detection, we compared three different machine learning models. Looking at the overall experimental results, the accuracy of the three models shows an increasing trend. Specifically:

1. The accuracy of the text-only basic model is 0.8. This model primarily relies on BERT's pre-trained architecture for text feature extraction.

2. The BERT-LSTM model achieves an accuracy of 0.83, a significant improvement compared to the text-only model. This indicates that by incorporating LSTM to capture the temporal sequence information in the text, the model's discrimination ability is enhanced.

3. The multimodal BERT model with image features further improves the accuracy, reaching 0.8488. This implies that when we consider both text and image information for news authenticity assessment, the model can produce more accurate detection results.

In summary, the experimental results show that as the model structure is gradually optimized and more sources of information are integrated, we can achieve higher accuracy in fake news detection. This underscores the importance of multimodal information fusion in complex tasks.

- Text-only Basic Model: This model uses BERT's pre-trained architecture for text feature extraction and processes it through fully connected layers. However, relying solely on text features may have limitations in complex fake news detection tasks. Text information may contain ambiguous or easily misinterpreted content, making text-only models prone to misclassification in specific scenarios.

- BERT-LSTM Model: In contrast to the basic model, this model adds an LSTM layer, which means it can extract not only static text features but also capture the

temporal sequence information within the text. The addition of LSTM gives the model a stronger capability to capture the internal logic and contextual relationships within the text, especially for long texts or content with strong logical connections between sentences. This is the main reason for the improved accuracy of this model.

- Multimodal BERT with Image Features: This model, in addition to considering text information, also incorporates image features, allowing the model to comprehensively consider both text and image information when making judgments. Some fake news may appear plausible in terms of text content, but there may be inconsistencies or mismatches between the accompanying images and text. In such cases, a model that includes image features can better detect these discrepancies, thereby enhancing the accuracy of detection. Additionally, using ResNet50 as the image feature extractor effectively extracts meaningful features from images, enhancing the model's discriminative ability.

In conclusion, the experimental results from the three models indicate that with improvements in model structure and the inclusion of more information, the accuracy of fake news detection gradually increases. This validates the significance of multidimensional and multimodal information input in enhancing the model's discrimination ability, particularly in complex tasks like fake news detection.

References

Boididou, C., Andreadou, K., Papadopoulos, S., Dang Nguyen, D. T., Boato, G., Riegler, M., Larson, M., & Kompatsiaris, I. (2015). Verifying multimedia use at mediaeval 2015 in mediaeval benchmarking initiative for multimedia evaluation.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780

Vyas, P., Liu, J., & El-Gayar, O. (2021). Fake news detection on the web: An LSTM-based approach. In Proceedings of the 27th Americas Conference on Information Systems (AMCIS 2021)

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., ... & Gao, J. (2018, July). Eann: Event adversarial neural networks for multi-modal fake news detection. In Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining (pp. 849-857).

Zhu, Y., Sheng, Q., Cao, J., Nan, Q., Shu, K., Wu, M., ... & Zhuang, F. (2022). Memory-guided multi-view multi-domain fake news detection. IEEE Transactions on Knowledge and Data Engineering.