Misinformation Detection in Social Media Texts and LLM Generated Text using Auxiliary Text Supervised Learning

Kongqiang Wang[1,*], Peng Zhang[1,†] and Qingli Tan[2,3,†]

[1]School of Information Science and Engineering, Yunnan University, Kunming 650500, Yunnan, China.
[2]Kunming Academy of Environmental Sciences, Kunming 650032, Yunnan, China.
[3]College of Ecology and Environment, Yunnan University, Kunming 650500, Yunnan, China.

Abstract

Our team wangkongqiang participated in the Prompt RecOvery For MisInformation Detection (PROMID) competition, and the main contributions were in two sub-tasks. They are Subtask 2: Misinformation Detection in LLM generated text. Given a piece of LLM-generated text with misinformation, the objective is to categorize each datapoint into different categories based on the presence of factual incorrectness in the summaries. and Subtask 3: Misinformation Detection in social media texts respectively. The objective of this task is to classify tweets related to the Russo-Ukrainian conflict as either misinformation (positive class) or non-misinformation (negative class). In this competition experiment, our team employed various methods for exploration. This includes the Logistic Regression method of machine learning and the Dense Neural Network and Recurrent Neural Network of deep learning, as well as the method based on the transformer pre-model *models-microsoft-deberta-v3-base*. Through thorough experiments, it has been proved that they have achieved significant accomplishments in these two sub-tasks. In subtask 2, the experiment using the *models-microsoft-deberta-v3-base* model applied to the tamil language achieved the best result with the F1 score of 0.31. The best result F1 score of the experiment using Recurrent Neural Network for the kannada language was 0.34. In subtask 3, Logistic Regression was used to achieve the best result, the precision is 0.81, the recall is 0.83 and the F1 score is 0.82.

Keywords

Text Multi-classification, Binary Classification of Text, Machine Learning, Deep learning, Transformer

1. Introduction

The PROMID [1] shared task aims to explore methods for identifying misinformation in human and LLM generated texs, as well as reconstructing the input prompt that likely led to a given piece of LLM generated misinformative text. There is a lot of motivations in this sharing task. As LLMs become more prevalent, the rate at which misinformation is generated and disseminated is much higher than in the past. It is more crucial than ever to identify ways for combating this spread of misinformation. The PROMID [2] shared task aims at two aspects of combating misinformation. The first is identifying whether a given piece of text has minformation or not. The second is understanding how a specific piece of misinformation was generated using LLMs. If we can trace back or infer the prompt that generated a suspicious text, it could provide insights into the intent, source, or specific instructions used to create misleading content. Both these aspects can aid in developing more robust misinformation detection and mitigation strategies.

It can be seen from the task overview that the organizer mainly offers three subtasks this year for this sharing task:

- `Subtask 1`: Prompt Recovery from LLM generated misinformative text; Given a piece of LLM-generated text with misinformation, participants are tasked to generate a plausible prompt that could have produced the target text.

Forum for Information Retrieval Evaluation, December 17-20, 2025, India
*Corresponding author.
†These authors contributed equally.
wangkongqiang60@gmail.com (K. Wang); zpp1219@gmail.com (P. Zhang); tanqingli@stu.ynu.edu.cn (Q. Tan)
https://github.com/WangKongQiang (K. Wang); https://github.com/zpp1219 (P. Zhang)

- `Subtask 2` : Misinformation Detection in LLM generated text; Given a piece of LLM-generated text with misinformation, the objective is to categorize each datapoint into different categories based on the presence of factual incorrectness in the summaries [3].
- `Subtask 3` : Misinformation Detection in social media texts; The objective of this task is to classify tweets related to the Russo-Ukrainian conflict as either misinformation (positive class) or non-misinformation (negative class). The dataset comprises manually annotated tweets, collected using the Twitter API during the first year of the Russia-Ukraine war. The dataset is highly imbalanced, which is the goal to check how the model's performance works in this setting. Misinfo tweets in multiple languages, and the content can be translated or addressed by LLMs. Misinfo tweets have some extra information (account age, bot account), but that can be extracted for non-misinfo tweets if participants want. The result will be evaluated based on Precision, Recall and weighted-averaged F1.

Participants can choose to participate in one or more subtasks. Here, our group mainly participated in Subtask 2: Misinformation Detection in LLM generated text [4] and Subtask 3: Misinformation Detection in social media texts. With the continuous development and progress of artificial intelligence, we have selected representative works of milestone significance. For examples, Logistic Regression in machine learning, Dense Neural Network and Recurrent Neural Network in deep learning and *models–microsoft-deberta-v3-base* in pre-trained models transformers are taken as the main research objects and have demonstrated excellent performance in these two sub-tasks.

2. Related Work

2.1. Problem Definition and Machine Learning

In recent years, the influence of misinformation/fake news on social media [5] in public opinion and political processes has become increasingly serious. Therefore, automatic detection of misinformation has become an important research direction in natural language processing and computational social sciences. The research [6] not only focuses on the content of a single text itself (such as language style and fact-finding), but also on the assistance of social context (such as dissemination structure and user reputation) and multimodal evidence (such as images) for detection.

Early methods focused on text-based language features (vocabulary, syntax, sentiment, suspicious indicator words) and manual/statistical features, using classifiers such as SVM [7] and LR [8] for discrimination. This type of method is intuitive and has good interpretability, but often has limited ability when dealing with complex semantics, irony and factual inferences.

2.2. Network-based Approaches and Deep Learning

Another important type of work uses how information spreads in social networks to determine its credibility, such as analyzing the shape of the forward/reply tree, user interaction patterns, and time series characteristics. This type of method can capture the dynamics of rumor propagation, but it still poses challenges for cold start (new events) and cross-event generalization.

In recent years, end-to-end methods based on neural networks (CNN, RNN, attention, and pre-trained language models represented by BERT) have significantly improved the effectiveness of text true and false classification. Pre-trained models can capture deeper semantic and contextual information and are often used in combination with meta-information (author, publication time) or structural features to improve performance.

2.3. Multimodal Approaches and Future Work

As social media content often contains images/videos, many studies have proposed multimodal architectures that jointly model text and visual information to enhance robustness. Representative works include EANN (Event Adversarial Network) [9] and MVAE (Multimodal Variational AutoEncoder) [10],

Table 1
Frequency of Special Incorrectness_Type Column in train dataset.

Languages	Incorrectness_Type	Frequency	Comments
Tamil	NaN	3986	This row of data indicates that Incorrect_Summary and Incorrectness_Type are not provided, but only the information of Correct_Summary is provided.
	misrepresentation	294	Given a piece of LLM-generated misrepresentation text with misinformation.
	fabrication	250	Given a piece of LLM-generated fabrication text with misinformation.
	false_attribution	250	Given a piece of LLM-generated false_attribution text with misinformation
	incorrect_quantities	195	Given a piece of LLM-generated incorrect_quantities text with misinformation
	Total	**4975**	**The total number of rows of data.**
Kannada	NaN	3986	This row of data indicates that Incorrect_Summary and Incorrectness_Type are not provided, but only the information of Correct_Summary is provided.
	misrepresentation	294	Given a piece of LLM-generated misrepresentation text with misinformation.
	fabrication	250	Given a piece of LLM-generated fabrication text with misinformation.
	false_attribution	250	Given a piece of LLM-generated false_attribution text with misinformation
	incorrect_quantities	195	Given a piece of LLM-generated incorrect_quantities text with misinformation
	Total	**4975**	**The total number of rows of data.**

Table 2
Frequency of Special IsCorrect Column in test dataset.

Languages	IsCorrect	Frequency	Comments
Tamil	True	140	The relationship between article_ta and summary_ta provided by this row of data is correct.
	False	60	The relationship between article_ta and summary_ta provided by this row of data is incorrect.
	Total	**200**	**The total number of rows of data.**
Kannada	True	140	The relationship between article_kn and summary_kn provided by this row of data is correct.
	False	60	The relationship between article_kn and summary_kn provided by this row of data is incorrect.
	Total	**200**	**The total number of rows of data.**

etc., which improve detection by aligning visual and text representations or learning event-invariant features.

The problems faced by the current evaluation include: inconsistent dataset labeling standards, sensitivity of evaluation metrics to category imbalance, poor generalization of models in new events/domains, and insufficient interpretability and auditability of models. When actually deploying, issues such as latency (online detection), user privacy and ethics also need to be considered.

The important future directions include: cross-language/cross-cultural error message detection, the combination of evidence retrieval & claim verification with generative models, explainable detection mechanisms, multimodal and multi-source fusion strategies, and the improvement of the model's transfer ability in low-resource or new event situations.

3. Exploratory Data Analysis

For Subtask 2: Misinformation Detection in LLM generated text, it contains datasets in four languages, namely telugu, tamil, kannada and malayalam. The two languages we mainly participated in researching were tamil and kannada. Now, statistical analysis is conducted on the datasets of these two languages. Among them, the situation of the Incorrectness_Type Column label in the training dataset is shown in Table 1, and the situation of the IsCorrect Column label in the test dataset is shown in Table 2. They are crucial for data screening before the model training and for the model to evaluate the test data.

For Subtask 3: Misinformation Detection in social media texts, it contains datasets in multiple languages, namely English(en), Spanish(es), German(de), Italian(it), French(fr), Ukrainian(uk), Dutch(nl), Persian(fa), Polish(pl), Russian(ru), Chinese(zh), Japanese(ja) and so on. The objective of this task is to classify tweets related to the Russo-Ukrainian conflict as either misinformation (positive class) or non-misinformation (negative class). The dataset [11] comprises manually annotated tweets, collected [12] using the Twitter API during the first year of the Russia-Ukraine war. This shared task involves two phases: **Development phase** with training and test datasets, **Final phase** for evaluation. All tasks are classification tasks. The dataset is highly imbalanced, which is the goal to check how the model's performance works in this setting. The number of misinfo tag row contents and nonmisinfo tag row contents of the training datasets in the Development phase and Final phase are shown in the table 3 below.

The number of data samples rows in the test datasets of the Development phase and Final phase, see Table 4.

Table 3
The number of misinfo tag lines and nonmisinfo tag lines in the training datasets of the Development phase and Final phase.

Phase	Dataset Folder	Misinfo Label Number	Nonmisinfo Label Number
Development_phase	train_set	364	34174
Final_phase	input_data	364	34174
	reference_data	156	14646

Table 4
The number of undefined tag lines in the test datasets of the Development phase and Final phase.

Phase	Dataset Folder	Undefined Label Number
Development_phase	test_data_without_label	14802
Final_phase	test_final_merge_withoutlabel	2414

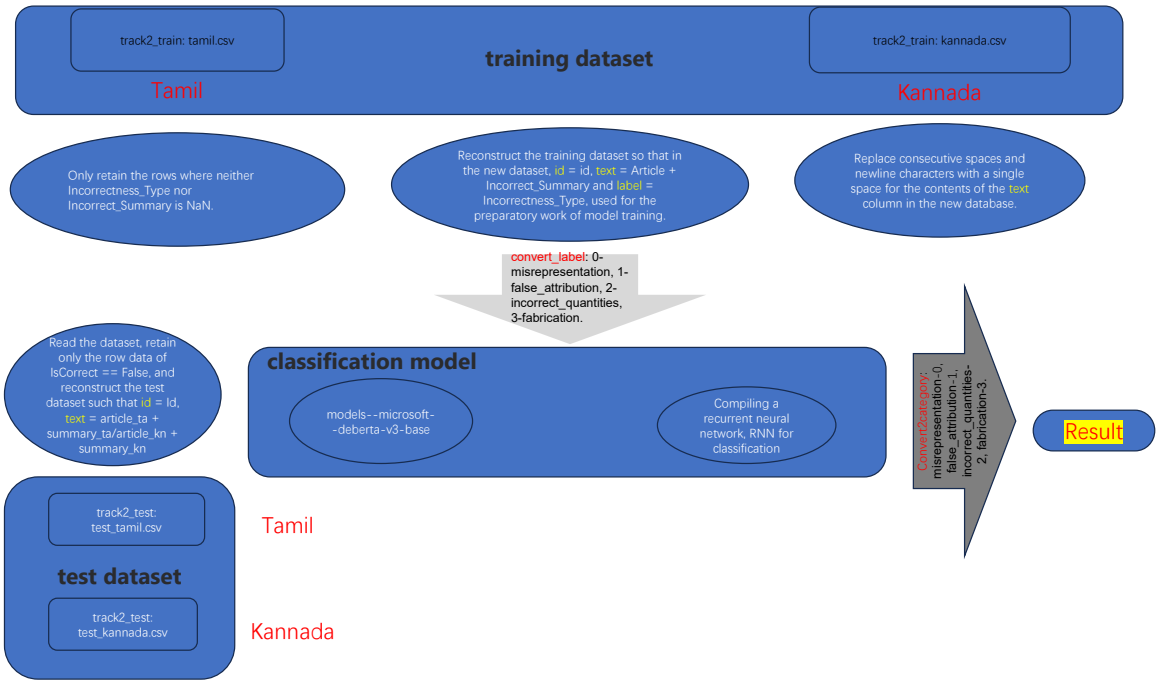

Figure 1: For Subtask 2: Misinformation Detection in LLM generated text, the overall flowchart of the experiment implemented by our group.

4. Methodology

For Subtask 2: Misinformation Detection in LLM generated text and Subtask 3: Misinformation Detection in social media texts. In order to enable the trained model to learn more useful data information, our group recombined the original data set and preprocessed the data respectively. The overall processing flow is shown in Figure 1 and Figure 2. The models respectively adopted the classic Logistic Regression model of machine learning, the classic Dense Neural Network and Recurrent Neural Network of deep learning and DeBERTaV3 : Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing [13]. DeBERTa improves the BERT and RoBERTa models using disentangled attention and enhanced mask decoder. With those two improvements, DeBERTa out perform RoBERTa on a majority of NLU tasks with 80GB training data.

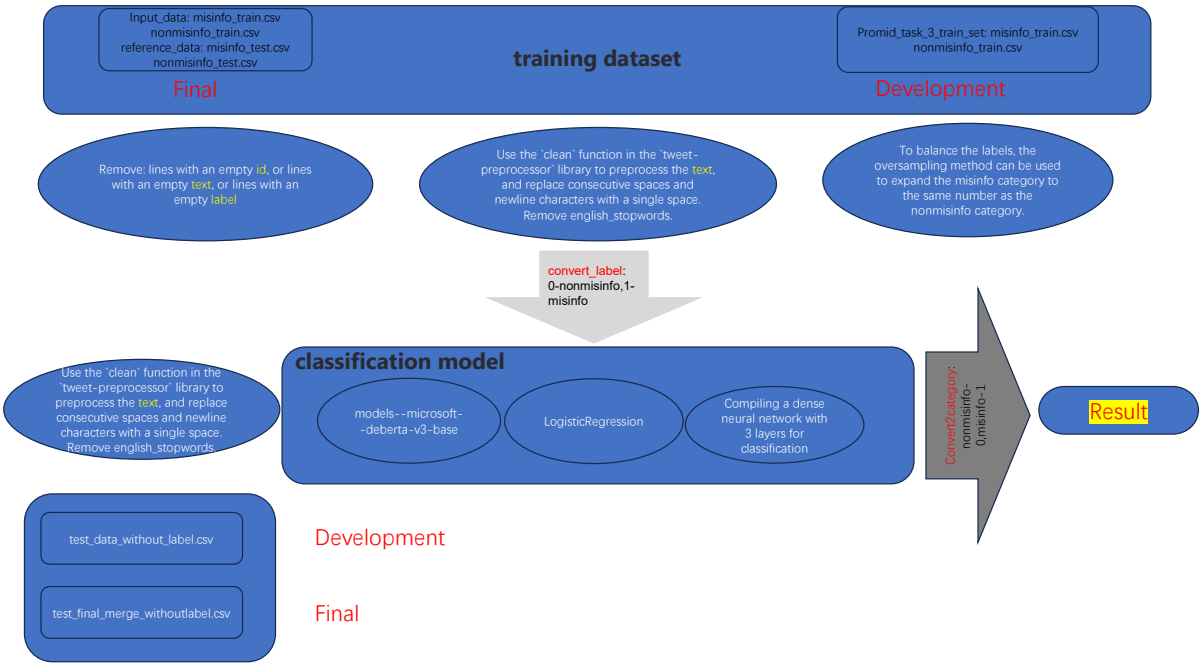

Figure 2: For Subtask 3: Misinformation Detection in Social Media Texts, the overall flowchart of the experiment implemented by our group.

4.1. The Principle of Logistic Regression

Logistic Regression is a type of discriminative model widely used in binary and multi-classification tasks. Its core idea is to linearly combine input features and map the linear results to probability values using the Sigmoid function (or Softmax function), thereby achieving classification decisions. Although the name contains "regression", logistic regression is essentially a linear classification model whose goal is to learn an optimal decision boundary in the feature space.

Conceptual and Format Models. For binary classification tasks, logistic regression assumes that the input feature vector is $x \in \mathbb{R}^d$, It is obtained through linear transformation:

$$z = w^T x + b \tag{1}$$

Among them, w is a weight parameter, b is bias. To map the linear output to classification probabilities, logistic regression uses the Sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

Thus, the probability that the sample belongs to the positive class (labeled as 1) is:

$$P(y = 1 \mid x) = \sigma(w^T x + b) \tag{3}$$

This probability reflects the model's confidence level in the category, and the final classification label can be judged through the threshold (usually 0.5).

Loss Function and Learning Process. Logistic regression learns parameters w with b by maximizing the log-likelihood function of the training data. For a single sample (x,y), Its logarithmic likelihood is:

$$\ell(w, b) = y \log(\sigma(z)) + (1 - y) \log(1 - \sigma(z)) \tag{4}$$

For the convenience of optimization, the opposite number is usually taken as the Loss function, namely the Binary Cross-Entropy loss:

$$\mathcal{L}(w, b) = -\left[y \log(\sigma(z)) + (1 - y) \log(1 - \sigma(z))\right] \tag{5}$$

By performing gradient descent on the loss function (or other optimization algorithms, such as L-BFGS, Newton method, etc.), the model can gradually update the parameters to make the predicted probability as close as possible to the true label.

Regularization. To prevent overfitting caused by overly large parameters, logistic regression often incorporates L1 or L2 regularization terms:

- `L2 regularization (Ridge)` : Encourage smoother weights and reduce model complexity;
- `L1 regularization (Lasso)` : It has the ability of feature selection and can compress some weights to zero.

The addition of regularization can significantly enhance the generalization performance of the model, especially in high-dimensional feature spaces.

Multi-category Extension. Although logistic regression is mainly used for binary classification tasks, it can be extended to multi-classification problems through one-to-many (One-vs-Rest) or Softmax regression (multiple logistic regression). For K Class classification, Softmax regression uses the Softmax function to output the probability distribution of each class:

$$P(y = k \mid x) = \frac{e^{w_k^T x}}{\sum_{j=1}^{K} e^{w_j^T x}} \tag{6}$$

The corresponding loss function is Categorical Cross-Entropy.

Characteristics and Advantages. Logistic regression is widely applied in fields such as text classification, medical prediction, and risk assessment due to its strong interpretability, efficient training, and robust stability. Its decision boundary is linear. Therefore, after the features undergo reasonable engineering processing (such as TF-IDF, embedding vectors), good results are often achieved. The main advantages include:

- The model parameters are interpretable;
- It performs stably on small-scale datasets;
- Fast training speed and low computing cost;
- Overfitting can be effectively prevented through regularization;
- The mathematical form is clear and easy to analyze.

4.2. The Principle of Dense Neural Network

Dense Neural Network (DNN), also known as Fully Connected Neural Network (FCNN), is a type of feedforward neural network composed of multiple layers of linear transformations and nonlinear activation functions. DNN is the most fundamental and classic model structure in deep learning and can be used for various tasks such as classification, regression, and representation learning. The core idea is to gradually learn the complex mapping relationship from input features to output labels through multi-layer abstraction.

Network Structure. A typical DNN consists of the following parts:

- `Input Layer` : Receives input features $x \in \mathbb{R}^d$;
- `Hidden Layers` : Composed of several fully connected layers stacked together, each layer contains several neurons.
- `Output Layer` : Provide the final prediction based on the task type (such as classification or regression).

At each layer, neurons perform linear transformations on the input and then introduce nonlinearity through activation functions, enabling the network to fit complex functions. For the l Layer neurons include:

$$z^{(l)} = W^{(l)} h^{(l-1)} + b^{(l)} \tag{7}$$

$$h^{(l)} = f\left(z^{(l)}\right) \tag{8}$$

Among them, $W^{(l)}$ and $b^{(l)}$ are respectively the weight matrix and the bias vector. $f(\cdot)$ for activation functions, such as ReLU, Sigmoid, Tanh, etc. $h^{(l)}$ for the output representation of this layer.

A multi-layer network can be recursively represented as:

$$h^{(L)} = f^{(L)}\Big(W^{(L)} f^{(L-1)}\Big(\cdots f^{(1)}\Big(W^{(1)}x + b^{(1)}\Big)\Big)\Big) \tag{9}$$

Through multi-layer combination, DNN can gradually extract high-level abstract features from low-level features.

Activation Function. The introduction of the activation function is the key for DNN to represent complex nonlinear mappings. Common activation functions include:

- `ReLU` : ReLU(x) = max(0,x). It has the advantages of stable gradient propagation and simple calculation.
- `Sigmoid` : It is applicable to the binary output layer.
- `Tanh` : It is applicable to some normalized data scenarios.

Nonlinear activation endows the network with the Universal Approximation Theorem, theoretically enabling it to approximate any continuous function.

Forward Propagation. During the forward propagation process, the input features pass through each fully connected layer in sequence to generate the final output. For classification tasks, the last layer typically uses the Softmax function to convert the output into a probability distribution:

$$P(y = k \mid x) = \frac{e^{o_k}}{\sum_j e^{o_j}} \tag{10}$$

Among them, o_k is the output layer for the category k linear response.

Loss Function and Parameter Learning. DNN typically learn network parameters by minimizing the loss function. The commonly used Loss function in classification tasks is Cross-Entropy loss:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k \log(\hat{y}_k) \tag{11}$$

Parameter updates are accomplished through the Backpropagation algorithm. Backpropagation calculates the gradient layer by layer and updates the parameters through the chain rule. The optimization algorithm usually uses: Stochastic Gradient Descent (SGD), Adam, RMSProp, etc. These optimization methods continuously reduce training errors, enabling the model to gradually fit the patterns of the training data.

Regularization and Prevention of Overfitting. To enhance the generalization ability of the model, DNN often incorporate regularization techniques:

- `L2 regularization` : Weight decay.
- `Dropout` : Randomly discard some neurons to reduce co-adaptation.
- `Batch Normalization` : Stable training accelerates convergence.
- `Early Stopping` : Avoid overfitting caused by long-term training.

These techniques can effectively prevent the model from overfitting the training data and improve the generalization performance.

Model Features and Advantages. Dense Neural Network has the following advantages:

- Be capable of learning complex nonlinear mapping relationships;
- It is easy to implement and expand, and serves as the foundation for many deep models;
- It is effective for multiple tasks (classification, regression, text feature learning, etc.);
- After combining regularization and modern optimization algorithms, the training efficiency is high and the effect is stable.

Therefore, DNN is often used as a fundamental module in many deep learning systems and is widely applied in fields such as natural language processing, computer vision, and recommendation systems.

4.3. The DeBERTa Model Principle

DeBERTa [14] (Decoding-enhanced BERT with Disentangled Attention) is an improved Transformer language model proposed by Microsoft in 2021 By introducing the decoupled Attention mechanism (Disentangled Attention) and the Enhanced Mask Decoder structure (Enhanced Mask Decoder), the performance of the language understanding task was significantly improved while maintaining similar model parameter scales. As an enhanced version of BERT and RoBERTa, DeBERTa demonstrates leading performance in multiple NLP benchmark tasks.

Decoupling Attention Mechanism (Disentangled Attention). The traditional BERT model uses the superposition representation of token embedding and position embedding, that is, it processes content and position information in the same way. DeBERTa proposed to separate the content representation of words from the position information representation, thereby enhancing the model's ability to learn language structures. For the i-th token, whose representation is split into:

$$h_i = h_i^{(c)} + h_i^{(p)} \tag{12}$$

Among them, $h_i^{(c)}$ is content embedding, $h_i^{(p)}$ is relative position embedding.

In the self-attention mechanism, DeBERTa divides the attention score into three parts:

$$A_{ij} = Q_i^{(c)} \cdot K_j^{(c)} \ + \ Q_i^{(c)} \cdot K_{ij}^{(p)} \ + \ Q_{ij}^{(p)} \cdot K_j^{(c)} \tag{13}$$

Compared with the simple dot product form of traditional Transformers, decoupled attention can respectively model the dependencies of "content-to-content", "content-to-location", and "location-to-content", making the model more sensitive to language structures.

Advantage: It can better capture sentence structure; Enhance the ability to model long-distance dependencies; Improve the quality of semantic and grammatical learning. This is also the core reason why DeBERTa has significantly improved performance compared to BERT/RoBERTa.

Enhanced Mask Decoder. In the Masked Language Modeling (MLM) task of BERT, the model often fails to make full use of the position encoding information. DeBERTa proposed Enhanced Mask Decoder (EMD), which introduces stronger relative position information when decoding masked tokens, enabling the model to make more accurate judgments based on the relative context structure when predicting mask words. Its improvements include:

- Strengthen the modeling of the Mask position's dependence on surrounding tokens;
- Improve the position modeling ability by using relative position bias;
- Reduce the information loss of tokens after they are masked.

Experiments show that EMD enables DeBERTa to achieve higher training efficiency and prediction accuracy in MLM training.

Removal of Absolute Position Encoding. Unlike the absolute position encoding of BERT, DeBERTa completely eliminates absolute positional embedding and instead uses relative position bias:

$$r_{ij} = \text{relative position embedding}(i - j) \tag{14}$$

This way: It is more in line with the relative position structure of language; It is more suitable for tasks such as sentence rearrangement and fill-in-the-blank; It has better generalization ability for sequence length.

Training and Optimization. DeBERTa adopted RoBERTa's training strategies, such as: Dynamic Masking; Train more data; Larger batch size; Remove the Next Sentence Prediction task. This further enhances the performance of the model.

Summary and Advantages. Compared with BERT [15] and RoBERTa [16], the main advantages of DeBERTa are reflected in:

- `Decouple attention` : Model content and location information separately to capture a stronger language structure.

Table 5
The evaluation results of the test set of Final phase based on the two languages tamil and kannada.

language	model	category	precision	recall	f1-score
Tamil	models-microsoft-deberta-v3-base	fabrication	0.65	0.34	0.45
		false_attribution	0.0	0.0	0.0
		incorrect_quantities	0.22	0.2	0.21
		misrepresentation	0.52	0.64	0.57
		overall	**0.35**	**0.30**	**0.31**
	Recurrent Neural Network	fabrication	0.4	0.19	0.26
		false_attribution	0.18	0.15	0.17
		incorrect_quantities	0.4	0.2	0.27
		misrepresentation	0.38	0.4	0.39
		overall	**0.34**	**0.24**	**0.27**
Kannada	models-microsoft-deberta-v3-base	fabrication	0.0	0.0	0.0
		false_attribution	0.29	0.15	0.2
		incorrect_quantities	0.38	0.6	0.46
		misrepresentation	0.5	0.68	0.58
		overall	**0.29**	**0.36**	**0.31**
	Recurrent Neural Network	fabrication	0.89	0.25	0.39
		false_attribution	0.14	0.08	0.1
		incorrect_quantities	0.3	0.3	0.3
		misrepresentation	0.52	0.64	0.57
		overall	**0.46**	**0.32**	**0.34**

- Enhanced decoder : Improve the prediction quality and training efficiency of mask.
- Relative position encoding : More suitable for sentences of different lengths and structures.
- Strong overall performance : Achieved leading results in benchmark tasks such as GLUE, SQuAD, and SuperGLUE.

Therefore, DeBERTa, as a representative model in language understanding tasks, is widely applied in fields such as classification, text matching, summarization, and information extraction.

5. Result

For Subtask 2: Misinformation Detection in LLM, our group respectively adopted the pre-trained model method of the transformer architecture of models-microsoft-deberta-v3-base and the method of implementing text classification based on the RNN model. Recurrent Neural Network (RNN) is a kind of neural network model specifically for processing sequential data. Unlike traditional neural networks, RNNS have memory capabilities and can capture the temporal dependencies in data. Among them, the precision, recall and f1-score indicators of the two languages tamil and kannada that our group participated in the experiment for the final test results in the two aspects of category and overall are shown in Table 5.

For Subtask 3: Misinformation Detection in Social Media Texts, this subtask of Prompt RecOvery For MisInformation Detection (PROMID) shared task involves two phases: Development phase with training and test datasets, Final phase for evaluation. All tasks are classification tasks.

For **Development_phase**, the test set **test_data_without_label** for verification is provided. This dataset contains a total of 14,802 rows of data content. Our group conducted an initial model development based on this test set, using models-microsoft-deberta-v3-base as the main model and achieved the best results. The evaluation of model performance mainly refers to the weighted avg results of the classification model. The detailed results of each classification model are shown in the table 6 below.

For **Final_phase**, the test set **test_final_merge_withoutlabel** for evaluation is provided. This dataset contains a total of 2,414 rows of data content. Our group conducted a final model development

Table 6
The evaluation results of the test set of Development phase based on weighted average.

model	precision	recall	f1-score
models-microsoft-deberta-v3-base	0.99	0.98	0.99
Logistic Regression	0.99	0.97	0.98
Dense Neural Network	0.99	0.98	0.99

Table 7
The evaluation results of the test set of Final phase based on weighted average.

model	training dataset	precision	recall	f1-score
models-microsoft-deberta-v3-base	Development	0.82	0.84	0.78
	Final	0.84	0.85	0.81
Logistic Regression	Final	0.81	0.83	0.82
Dense Neural Network	Final	0.78	0.83	0.78

based on this test set, using Logistic Regression as the main model and achieved the best results. The evaluation of model performance mainly refers to the `weighted avg` results of the classification model. The detailed results of each classification model are shown in the table 7 below. In this experiment, two training sets were respectively used for the `models-microsoft-deberta-v3-base`. One type is the training dataset (Development) provided by Development_phase. One type is the training dataset (Final) provided by Final_phase. The key difference between these two training datasets is that the test set reference label data of Development_phase is added to the latter dataset. Through experiments, it can be found that these real label data are very useful. He can enhance the learning effect of the model. The other two models, Logistic Regression and Dense Neural Network, were fully trained based on the training dataset (Final) provided by Final_phase.

6. Conclusion

In this paper, several machine learning and deep learning approaches have been used to detect misinformation multiple languages content, and the models have been compared. Several techniques have been employed to increase accuracy. Our proposed `models-microsoft-deberta-v3-base` model achieved good results compared to its simplicity. We believe with proper feature extraction and data augmentation techniques, these will be able to improve our proposed model.

Acknowledgments

We are very grateful to the organizers of the Shared Task on PROMID: Prompt RecOvery For MisInformation Detection and the School of Information of Yunnan University for providing the environment and equipment.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] G. K. Shahi, A. Hegde, S. Satapara, P. Mehta, S. Modha, D. Ganguly, D. Nandini, H. L. Shasirekha, A. K. Jaiswal, G. Pasi, T. Mandl, Overview of the first shared task on prompt recovery for misinformation detection (promid 2025), in: K. Ghosh, T. Mandl, S. Pal, S. Majumdar, A. Chakraborty

(Eds.), Working Notes of FIRE 2025 - Forum for Information Retrieval Evaluation, Varanasi, India. December 17-20, 2025, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[2] A. Hegde, G. K. Shahi, S. Satapara, P. Mehta, S. Modha, D. Ganguly, D. Nandini, H. L. Shasirekha, A. K. Jaiswal, G. Pasi, T. Mandl, Prompt recovery for misinformation detection at fire 2025, in: Proceedings of the 17th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE '25, Association for Computing Machinery, 2025.

[3] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Key takeaways from the second shared task on indian language summarization (ILSUM 2023), in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation (FIRE-WN 2023), Goa, India, December 15-18, 2023, volume 3681 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 724–733. URL: https://ceur-ws.org/Vol-3681/T8-1.pdf.

[4] S. Satapara, P. Mehta, D. Ganguly, S. Modha, Fighting fire with fire: Adversarial prompting to generate a misinformation detection dataset, CoRR abs/2401.04481 (2024). URL: https://doi.org/10.48550/arXiv.2401.04481. doi:10.48550/ARXIV.2401.04481. arXiv:2401.04481.

[5] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (2018) 1146–1151.

[6] X. Zhou, R. Zafarani, A survey of fake news: Fundamental theories, detection methods and opportunities, ACM Computing Surveys (2020).

[7] C. Cortes, V. N. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297. URL: https://api.semanticscholar.org/CorpusID:52874011.

[8] D. R. Cox, The regression analysis of binary sequences, Journal of the royal statistical society series b-methodological 20 (1958) 215–232. URL: https://api.semanticscholar.org/CorpusID:125694386.

[9] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, J. Gao, Eann: Event adversarial neural networks for multi-modal fake news detection, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018). URL: https://api.semanticscholar.org/CorpusID:46990556.

[10] D. Khattar, J. S. Goud, M. Gupta, V. Varma, Mvae: Multimodal variational autoencoder for fake news detection, The World Wide Web Conference (2019). URL: https://api.semanticscholar.org/CorpusID:86785940.

[11] G. K. Shahi, Y. Mejova, Too little, too late: Moderation of misinformation around the russo-ukrainian conflict, Websci '25, 2025. doi:10.1145/3717867.3717876.

[12] G. K. Shahi, T. A. Majchrzak, Amused: An annotation framework of multimodal social media data, in: F. Sanfilippo, O.-C. Granmo, S. Y. Yayilgan, I. S. Bajwa (Eds.), Intelligent Technologies and Applications, Springer International Publishing, Cham, 2022, pp. 287–299.

[13] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. arXiv:2111.09543.

[14] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[16] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training, in: S. Li, M. Sun, Y. Liu, H. Wu, K. Liu, W. Che, S. He, G. Rao (Eds.), Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, Huhhot, China, 2021, pp. 1218–1227. URL: https://aclanthology.org/2021.ccl-1.108/.