# Research on BERT-Based Chinese Offensive Language Detection

**Xubo Zhang, Rouyi Fan, Xiaofeng Li***

*School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China*

**Corresponding Author*

**Abstract: Conventional approaches that rely on dictionaries and rules have been progressively less suitable for real-world use in the task of detecting abusive Chinese language usage. This paper explores the application of the BERT model for Chinese offensive language identification in order to address this problem. The BERT-Base-Chinese pre-trained model is improved by the analysis and processing of a combined collection of offensive Chinese language data. Two independent classification heads are included for subject classification and offensive language identification, respectively, within a parallel multi-task learning architecture that shares low-level feature representations. This method successfully raises the model's overall performance. Strong support and a useful basis for creating safer and more dependable language-generating systems are offered by this research.**

**Keywords: Deep Learning; BERT Model; Data Mining; Offensive Language Detection**

## 1. Introduction

The amount of online debate has increased dramatically due to the quick growth of social media platforms and online communication technologies, which has unavoidably mixed in a sizable amount of aggressive, discriminating, and offensive content. In addition to upsetting the peace in the online community, such unpleasant language can affect individuals psychologically and potentially spark major societal problems. Thus, precisely detecting and removing objectionable content from the internet has emerged as a key area of study in the Natural Content Processing discipline[1].

Traditional machine learning techniques like Support Vector Machines (SVM) and Naive Bayes[2] were the mainstay of early research, which used manually created features like word frequency, syntactic rules, or sentiment lexicons to identify objectionable text. Despite being obvious, these approaches struggle to handle the implicit semantics and unpredictability of online discourse and have poor generalization capacities. As deep learning gained popularity, models built on RNNs, CNNs, and LSTMs became more robust by automatically extracting text features; yet, because of unidirectional information flow, they were still limited in their ability to capture contextual correlations[3]. Text detection tasks have performed much better since the advent of pre-trained language models. Of them, the Transformer-based BERT model has demonstrated good adaptability in multilingual situations. Global dependencies between words in a sentence are well captured by Transformer's self-attention mechanism. It accomplishes collaborative modeling of local semantics and global dependencies when combined with BERT's bidirectional encoding capability. These techniques have greatly increased model generalization and prediction accuracy by using pre-trained knowledge and ongoing structural optimization[4].

However, cultural distinctiveness and a lack of data are the two main issues facing Chinese abusive language detection today. Machine-translated English datasets were frequently used in early studies; however, performance was generally limited due to linguistic convention variations. This void was closed with the publication of the COLD dataset. It provides an excellent annotated resource for model training and addresses delicate subjects like gender and ethnicity using real-world data from Weibo and Zhihu. Still, training on the COLD dataset alone was not enough. As a result, the ToxiCN dataset was combined to create a larger dataset, which enhanced the model's capacity for generalization and allowed it to acquire more precise representations.

Therefore, the COLD and ToxiCN datasets, two recent high-quality datasets on Chinese abusive language, are combined in this research. After that, the merged dataset is examined, purified, and standardized. In the end, 47,490 text samples made up the dataset, which served as a strong basis for training the model. Based on this, this work presents a parallel multi-task learning technique and fine-tunes using the BERT-Base-Chinese pre-trained model. The model's performance and training efficiency are successfully improved by sharing low-level feature representations, which eventually produces acceptable experimental results.

## 2. Basic Information

### 2.1 Architecture of Transformers

In 2017, Vaswani et al.[5] proposed the Transformer architecture. The model's ability to capture long-range relationships was significantly improved by its creative abandonment of conventional recurrent neural network topologies and complete reliance on the attention mechanism for sequence modeling. The feed-forward neural network and the multi-head self-attention mechanism are its two primary sub-modules. Multiple encoder and decoder layers are stacked to accomplish deep semantic modeling. As illustrated in Figure 1, the encoder and decoder are both made up of pointwise fully connected layers and layered self-attention layers.

The model can comprehend contextual semantics from a variety of angles thanks to the multi-head self-attention mechanism, which employs parallelized attention heads to concurrently capture dependencies between various sequence locations across various representation subspaces. Formula (1) depicts the single-head attention mechanism. The multi-head attention mechanism applies a linear transformation after concatenating the output of several heads. This approach efficiently mines semantic associations between words by calculating the attention weights between Query, Key, and Value to accomplish weighted aggregation of information[6].

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

$QK^T$ computes the similarity between the query and the key in Formula (1); $\frac{1}{\sqrt{d_k}}$ avoids too large dot products that might cause the gradient to vanish or explode; softmax normalizes the values into attention weights; and multiplying by V yields a weighted sum to get the output representation.

The feed-forward neural network module, which usually consists of a two-layer fully linked network with an activation function in between, applies a non-linear transformation to the attention output at each point. The Transformer architecture also includes layer normalization and residual connections to further improve the expressive power and training stability of the model. This allows for efficient feature transformation at each layer while maintaining the original data.

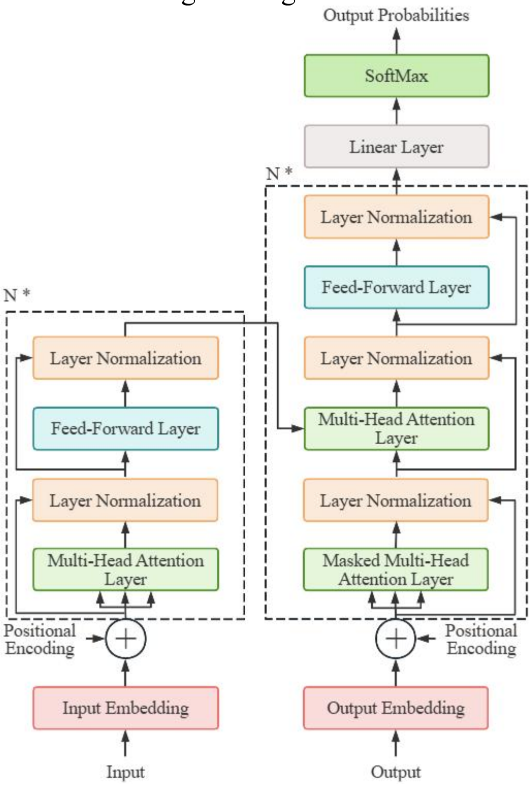

**Figure 1. Transformer Architecture**

### 2.2 Architecture of the BERT Model

In 2018, Google[7] proposed BERT (Bidirectional Encoder Representations from Transformers), a pre-trained language model based on Transformer. In contrast to conventional unidirectional models, its key concept is the use of bidirectional context for text modeling, which enables a better capture of complex semantic linkages inside phrases. Through the complete integration of contextual information, it acquires universal language representations that are subsequently refined on subsequent tasks to enhance performance. The performance of many NLP tasks was improved with the introduction of BERT, particularly in the areas of text classification, named entity identification and machine translation.

Multiple identical Transformer Encoder layers make up BERT's architecture. Token, segment, and position embeddings are three types of embeddings that BERT adds together for input representation. Every word in the text is mapped into a fixed-length vector using token embeddings. BERT breaks words down into smaller subword units using a WordPiece vocabulary. By giving each sentence a distinct embedding value, segment embeddings are utilized to differentiate between two sentences in the input. Position embeddings help the model comprehend the relative order of words in a sentence by providing positioning information for each word through positional encodings. The final input representation for BERT is created by adding these three embeddings, and it is subsequently supplied to the model for additional processing.

A bidirectional encoding technique is used by BERT. By employing Masked Language Modeling during training, it achieves a deep comprehension of contextual information by randomly masking some input tokens and subsequently predicting their original terms, in contrast to conventional left-to-right or right-to-left language models. In order to learn the relationships between sentences, BERT also adds the Next Sentence Prediction (NSP) task at the same time.

The BERT model is optimized for several comparable NLP tasks following pre-training on large datasets. The process of fine-tuning entails supervised training of the previously trained model on annotated data, modifying all parameters, and customizing the output layer for the particular task. The Transformer encoder architecture used for pre-training and fine-tuning is identical, and the fine-tuned model is initialized using the pre-trained parameters, greatly enhancing training effectiveness and model performance[7].

The corpus composition, tokenization technique, and training goals of the BERT-Base-Chinese model are all customized to the peculiarities of the Chinese language. Every Chinese character in BERT-Base-Chinese is handled as a token, while special symbols and words that are not in the lexicon are handled using the WordPiece tokenization process. This strategy improves the model's generalization capabilities for Chinese text by better addressing the problem of Chinese words lacking spaces between words when compared to word-based tokenization techniques.

Overall, BERT is one of the most important models in contemporary mainstream text modeling and offers a strong basis for a variety of NLP tasks because of its extensive contextual modeling capacity and deep semantic representations acquired by pre-training on large-scale corpora. In order to further enhance performance in Chinese offensive language identification, this study develops and applies a parallel multi-task learning technique based on the BERT-Base-Chinese architecture.

## 3. Analysis of Experiments

### 3.1 Information Processing

3.1.1 Understanding and combining data

There are 37,480 Chinese comments in the COLD dataset. These comments cover a variety of issues, mostly race, gender, and area, and each one has a binary designation indicating if it contains offensive content (0 for safe, 1 for offensive). Attacking individuals, attacking groups, counter-bias speech, and other non-attacking content were the four categories into which the test set was further annotated with fine-grained labels in order to further explore the forms of attacks[8]. A group from Dalian University of Technology created the Chinese hazardous language detection dataset ToxiCN, which includes about 12,000 comments from Zhihu and Tieba. It has fine-grained multi-label annotations that include discriminatory subjects, including gender, race, and area, as well as whether it is damaging, language type, and expression form[9].

An initial examination of the information was carried out. Table 1 displays partial data from the COLD dataset.

**Table 1. COLD Partial Data Sample**

| Line | Topic | Label | Text |
|---|---|---|---|
| 3653 | Race | 1 | All Japanese people are bad, to varying degrees |
| 1230 | Gender | 1 | That man is so disgusting |
| 5034 | Region | 0 | A fellow townsman from Ganzhou, Jiangxi passing by |

Table 2 displays partial data from the ToxiCN dataset.

**Table 2. ToxiCN Partial Data Sample**

| Topic | Label | Text |
|---|---|---|
| Race | 1 | Mixed race, blue eyes, mixed with the blood of a white-skinned pig |
| Gender | 1 | Let's take a look at the quality of the grasshopper |

| | | |
|---|---|---|
| Region | 1 | Guangxi has no sense of existence |

The two datasets were combined since they share the same issue categories (race, gender, and area) and offer simple binary objectionable labels. A new dataset called Offensive_Data was created by eliminating other fields from each remark while keeping the text conten, offensiveness label, and related topi. The foundation for further model training was laid via a sequence of processing operations that included data insight, consistent normalization, outlier reduction, and dataset re-splitting.

3.1.2 Processing for normalization

Certain traditional Chinese characters were found in the data. To increase corpus consistency, all traditional characters were changed to simplified Chinese during the preprocessing phase because the two characters have the same semantics and only differ in written form. In a similar vein, as English character case typically has no bearing on textual meaning, all English text was changed to lowercase in order to further minimize the dimensions of incorrect features, which would aid model training[10].

Common Chinese and English characters, digits, and common Chinese punctuation signs were mostly preserved after further preliminary data cleaning. This required removing non-standard characters, combining punctuation marks that appeared frequently, removing superfluous spaces, and standardizing the punctuation format (converting half-width to full-width). Table 3 provides specific examples of processing. The text's uniformity and readability were successfully improved by these actions. The cleaned data was saved into new CSV files for later model training and performance assessment following the aforementioned processes.

**Table 3. Data Normalization Examples**

| Before processing | After processing |
|---|---|
| Emmm, some Asians are good at Chinese | emmm, some Asians are good at Chinese |
| [Don't criticize; let the betrothal gift issue simmer for a while] | Don't criticize; let the betrothal gift issue simmer for a while |
| Speechless...... | Speechless |

3.1.3 Verifying data and eliminating outliers

The results of checks for data outliers are displayed in Table 4.

**Table 4. Data Outliers**

| Line | Type | Value |
|---|---|---|
| 778 | float | NAN |
| 1929 | float | NAN |
| 2054 | float | NAN |
| 11325 | float | NAN |

Four records in the dataset, with the data type being float and the value being NaN, contained null values, according to the check. These four anomalous data points make up a very small percentage of the huge volume of data, and their influence on the distribution of data as a whole and model training is minimal. In order to guarantee the integrity of the model inputs and the seamless operation of further processing stages, a simple method was adopted to remove the data that had null values.

3.1.4 Splitting datasets

Figure 2 displays the results of the analysis of the combined dataset. The label distribution shows that the percentages of Normal and Offensive samples are comparatively equal. Nonetheless, there is some imbalance in the themes' distribution within the topic label. The ratio of Offensive to Normal is comparatively consistent across themes, according to additional examination of the distribution of Offensive vs. Normal within each topic. Furthermore, the majority of text lengths fall between 0 and 150 characters.

Stratified sampling was used for dividing the dataset into training, validation, and test sets in order to improve the model's learning capacity across several categories and the stability of assessment, given that the unbalanced topic distribution may have an impact on model training[11]. To guarantee that the distribution fraction of this combined label in each subset stayed consistent with the original data, the objectionable label and topic category were specifically combined to create a stratification key (e.g., 1_race, 0_gender). The data was initially separated into an 8:2 ratio between the test set and the training+validation set. Next, a 3:1 split between the training and validation sets was made from the training+validation set. In the end, 60% of the data came from the training set, with 20% coming from each of the test and validation sets. By successfully reducing the training bias and evaluation distortion brought on by class imbalance, this stratified splitting technique improved the model's capacity for generalization.

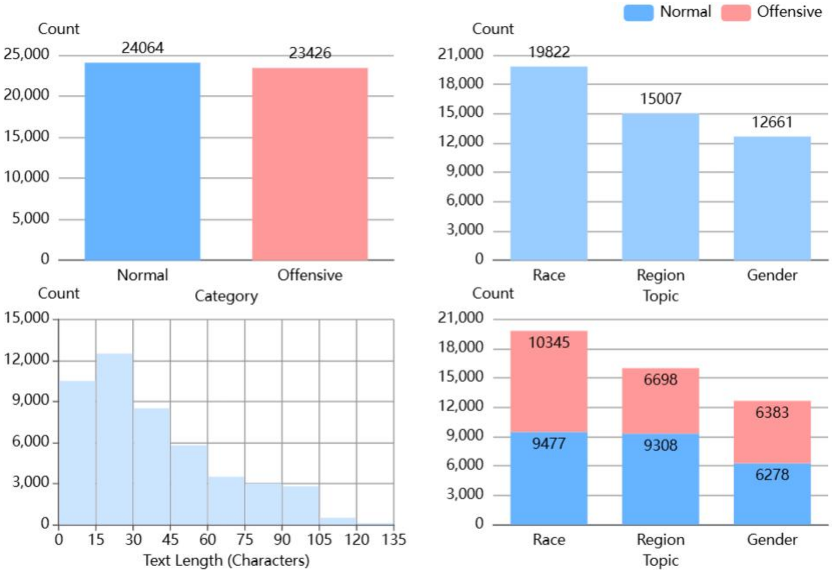

**Figure 2. Analysis of Merged Datasets**

### 3.2 Model Building

In order to extract features from the input Chinese text, the BERT-Base-Chinese model was initially employed as the pre-trained language model during model creation. A concurrent multi-task learning architecture based on the BERT-Base-Chinese model was created because recognizing offensive language in Chinese frequently involves not only judging whether a remark is offensive but also identifying the particular topic involved (e.g., race, area, or gender). The binary classification problem of hostile language detection and the three-class classification goal of topic identification are both concurrently completed by this architecture[12].

The BERT-Base-Chinese model initially encodes the input text in order to extract deep semantic features, as seen in Figure 3. The shared BERT output is then sent into two separate classification heads, the Offensive Language Detection head and the Topic Identification head, after passing through a Dropout layer. During training, these two tasks are completed simultaneously. Dual discrimination of the text and collaborative feature learning are achieved by using a weighted sum of the cross-entropy losses of the two tasks as the loss function[13].

In Formula (2), *loss1* is the cross-entropy loss for offensive detection, *loss2* is the cross-entropy loss for topic classification, and *α* is the weight coefficient for the two loss values. The offensive detection task is given more weight when α is set to 0.7.

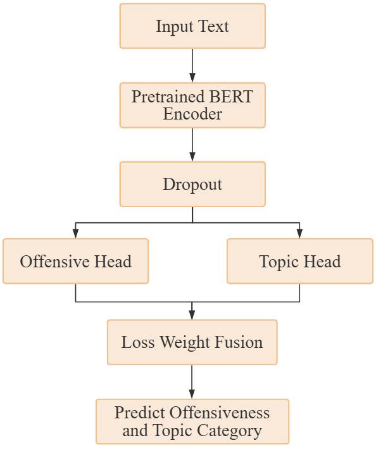

**Figure 3. Architecture Diagram for Parallel Multi-task Learning**

$$loss=\alpha\times loss1+(1-\alpha)\times loss2 \quad (2)$$

Table 5 displays the model's primary parameter setups. In order to improve the speed and efficacy of convergence, the AdamW optimizer was selected to adaptively modify the learning rate during training. To prevent training that is either too quick or too slow, the initial learning rate was adjusted to 2e-5 to guarantee reasonable parameter update steps. To improve generality and avoid overfitting, weight decay was set to 0.01. In order to balance GPU memory utilization and training performance, the batch size was fixed at 16. To avoid gradient explosion and preserve training stability, gradient clipping was used with a maximum norm of 1.0. To prevent overfitting and resource waste, an early stopping approach was used, ending training if the validation accuracy did not increase for three consecutive epochs. To further avoid overfitting and increase model robustness, the dropout rate was set to 0.3, randomly removing some neuron connections[14].

**Table 5. Main Parameter Configuration**

| Parameter | Configuration |
|---|---|
| Optimizer | AdamW |
| Initial Learning Rate | 2e-5 |
| Weight Decay | 0.01 |
| Batch Size | 16 |
| Gradient Clipping | Max Norm:1.0 |
| Early Stopping | Patience: 3 Epochs |
| Dropout | 0.3 |

Table 6 displays the hardware and software environment used to execute the model.

**Table 6. Environment Configuration**

| Name | Configuration |
|---|---|
| Processor | AMD Ryzen 7 5800H with Radeon Grap hics |
| GPU | NVIDIA GeForce RTX 3060 Laptop GPU 6GB |
| Memory | 16GB DDR43200MHz |
| Python | 3.8.5 |
| PyTorch | 2.2.0+cu121 |
| CUDA | 12.4 |
| cuDNN | 8.9.7 |

### 3.3 Model Assessment

A number of frequently used performance indicators in classification tasks, such as accuracy, precision, recall, F1-score, and the confusion matrix, were the main tools used in the model evaluation[15]. As stated in Formula (3), accuracy gauges the model's total capacity for accurate prediction. As stated in Formula (4), precision is the percentage of real positives among all cases that the model predicts as positive. As stated in Formula (5), recall measures the model's capacity to recognize real positive samples. Formula (6) states that the F1-score, which fully reflects the model's performance under class imbalance, is the harmonic mean of precision and recall. With the precise relationships displayed in Table 7, the confusion matrix offers a more user-friendly method of displaying the model's predictions across various categories. This makes it easier to analyze the various kinds of misclassifications and serves as a foundation for further optimization.

**Table 7. Confusion Matrix Table**

| Actual / Predicted | True | False |
|---|---|---|
| Positive | TP | FP |
| Negative | TP | FN |

The following formulas are used in the evaluation:

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

$$precision=\frac{TP}{TP+FP} \tag{4}$$

$$Recall=\frac{TP}{TP+FN} \tag{5}$$

$$F1\ Score=\frac{2TP}{2TP+FP+FN} \tag{6}$$

TP stands for true positives (correctly predicted positives); FN for false negatives (actually positive but predicted negative); FP for false positives (actually negative but predicted positive); and TN for true negatives (correctly predicted negatives) in Formulas (3), (4), (5), and (6).

### 3.4 Analysis of Model Results

The model's performance on each set was acquired following training. Figure 4 displays the model's accuracy on the training set throughout epochs, showing a steady improvement in accuracy.

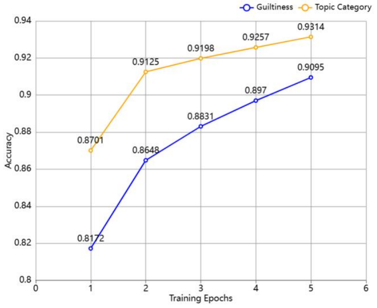

**Figure 4. Accuracy Change of Training Set**

Figure 5 displays the model's accuracy over epochs on the validation set. The findings indicate that the validation accuracy did not increase for three consecutive epochs after the second epoch. Therefore, the model parameters from the second epoch were kept as the best model in accordance with the early stopping strategy, and the test set was used for the final evaluation.

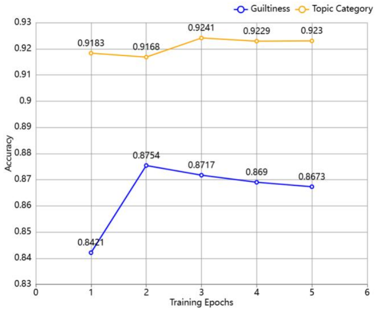

**Figure 5. Accuracy Change of Validation Set**

Tables 8 and 9 display the findings of additional evaluation metrics on the training set.

Tables 10 and 11 display the findings of additional evaluation metrics on the validation set.

Figures 6 and 7 display the confusion matrices for the validation set's second epoch. According to Figure 6, the model is highly effective at identifying the Offensive group; however, the misclassification rate for the Normal category is somewhat higher, suggesting that the algorithm is biased when dealing with borderline instances.

**Table 8. Training Set Offensive Detection Evaluation Results**

| Training Epoch | Precision | Recall | F1-Score |
|---|---|---|---|
| 1 | 81.62% | 81.25% | 81.44% |
| 2 | 85.95% | 86.80% | 86.37% |
| 3 | 87.44% | 89.11% | 88.27% |
| 4 | 88.68% | 90.72% | 89.69% |
| 5 | 89.94% | 91.95% | 90.93% |

**Table 9. Evaluation Results for Training Set Topic Classification**

| Training Epoch | Precision | Recall | F1-Score |
|---|---|---|---|
| 1 | 86.87% | 86.98% | 86.92% |
| 2 | 91.16% | 91.19% | 91.17% |
| 3 | 91.87% | 91.88% | 91.87% |
| 4 | 92.52% | 92.48% | 92.50% |
| 5 | 93.12% | 93.02% | 93.07% |

**Table 10. Validation Set Offensive Detection Evaluation Results**

| Training Epoch | Precision | Recall | F1-Score |
|---|---|---|---|
| 1 | 78.68% | 93.16% | 85.31% |
| 2 | 86.41% | 88.61% | 87.50% |
| 3 | 85.08% | 89.65% | 87.31% |
| 4 | 82.97% | 92.35% | 87.41% |
| 5 | 84.24% | 89.87% | 86.96% |

**Table 11. Evaluation Results for Validation Set Topic Classification**

| Training Epoch | Precision | Recall | F1-Score |
|---|---|---|---|
| 1 | 91.02% | 91.74% | 91.31% |
| 2 | 92.19% | 91.02% | 91.51% |
| 3 | 92.57% | 92.06% | 92.29% |
| 4 | 92.32% | 92.05% | 92.17% |
| 5 | 92.30% | 92.13% | 92.21% |

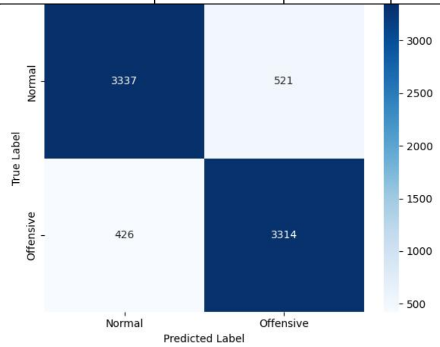

**Figure 6. Offensive Detection Confusion Matrix**

According to Figure 7, the race category exhibits the strongest model recognition impact, indicating that the linguistic elements for this class are more focused and simpler for the model to capture. The gender category, on the other hand, contains more cases that are incorrectly labeled as race. There may be some linguistic overlap between these two subjects. Mainly, gender and race are confused, particularly when gender samples are mistakenly labeled as race. This could be because, in some situations, writings that deal with gender issues also have racial overtones, which causes model misjudgment.

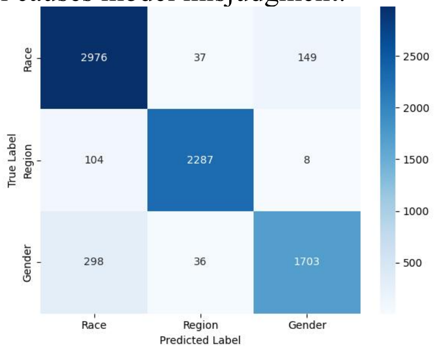

**Figure 7. Topic Classification Confusion Matrix**

The model's performance on the training set improved significantly with increasing epochs, demonstrating high learning potential, according to the different metrics on the training and validation sets. Strong model generalization capabilities without noticeable overfitting were demonstrated by the assessment metrics for both tasks stabilizing on the validation set following the second epoch. In particular, there was a trade-off between memory and precision for the offensive detection task, with a tendency toward stronger recall. Overall, the topic categorization task performed more consistently and precisely. After careful consideration, the 2nd epoch training was chosen as the final model for test set evaluation since it produced a reasonably ideal balanced performance for both tasks. The final model's performance on the test set was as follows: 91.57% accuracy for topic categorization and 87.81% accuracy for offense detection.

## 4. Conclusion

The detection of objectionable language has emerged as a significant area of study in the field of natural language processing due to the growing openness and diversity of the internet.

This study combined two excellent Chinese datasets, COLD and ToxiCN, to create a comprehensive dataset of 47,490 text samples, addressing the problems of data scarcity and linguistic-cultural disparities in Chinese offensive language identification. The data was thoroughly cleaned and normalized, giving the model training process a strong database.

By introducing a parallel multi-task learning framework and expanding on BERT-Base-Chinese, this paper's model design achieved the categorization of the related topic (race, gender, and area) and the identification of offensive content in text. According to experimental data, this model effectively increases detection efficiency and accuracy in a multi-task environment by demonstrating strong robustness and generalization capacity. This confirms that using multitasking techniques in conjunction with pretrained language models for Chinese offensive language identification is both feasible and preferable.

However, Zhihu and Weibo are the key sources of the data, which could cause bias in domain adaptability. Additionally, the model continues to exhibit some poor judgment when it comes to highly implicit or caustic language. In order to further enhance the algorithm's comprehensiveness, future research paths will concentrate on resolving these issues.

## References

[1] Guo Bolu, Xiong Xuhui. A Survey of Offensive Language Detection Methods Based on Deep Learning. Modern Information Technology, 2022, 6(5):5-10.

[2] Su Jinshu, Zhang Bofeng, Xu Xin. Research Progress on Text Classification Technology Based on Machine Learning. Journal of Software, 2006, 17(9):1848-1859.

[3] Wei Xuanling, Sun Xiang. A Review of Research Progress and Development Trends in Natural Language Processing Technology Methods. China-Arab Science and Technology Forum (Chinese and English), 2025, (5):84-88.

[4] He Xuefeng, Zhou Jie, Chen Deguang, et al. A Survey of Deep Learning Models for Natural Language Processing. Computer Applications and Software, 2025, 42(2):1-19+101.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in Neural Information Processing Systems, 2017, 30.

[6] Shi Lei, Wang Yi, Cheng Ying, et al. A Survey of Attention Mechanism in Natural Language Processing. Data Analysis and Knowledge Discovery, 2020, 4(5):1-14.

[7] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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). 2019: 4171-4186.

[8] Deng J, Zhou J, Sun H, et al. COLD: A Benchmark for Chinese Offensive Language Detection. arXiv preprint arXiv:2201.06025, 2022.

[9] Lu J, Xu B, Zhang X, et al. Facilitating Fine-Grained Detection of Chinese Toxic Language: Hierarchical Taxonomy, Resources, and Benchmarks. arXiv preprint arXiv:2305.04446, 2023.

[10]Li Da. Research on Intelligent Detection Method of Offensive Language Based on Fusion of Topic and Semantic Features. Guangxi Minzu University, 2024.

[11]Li Hang. Statistical Learning Methods (2nd Edition). Beijing: Tsinghua University Press, 2019.

[12]Wan Kelan. Research on Detection and Identification of Network Aggressive Speech Based on Multi-Task Learning. Sichuan University, 2021. DOI:10.27342/d.cnki.gscdu.2021.000544.

[13]Ruder S. An Overview of Multi-Task Learning in Deep Neural Networks. arXiv preprint arXiv:1706.05098, 2017.

[14]Qiu Xipeng. Neural Networks and Deep Learning. Beijing: Publishing House of Electronics Industry, 2020.

[15]Zhou Zhihua. Machine Learning. Beijing: Tsinghua University Press, 2016.