The Effectiveness of different model types in Sarcasm Detection

Applied Deep Learning 2023

Seah Ying Xiang T12902136
Dao Lan Ha B09701145
Paul Ong Zesheng T12902130

December 2023

1 Source Code

The implementation of the models, training procedures, and evaluation scripts are available on our GitHub repository. The repository can be accessed at the following URL:

`https://github.com/seahyx/ADL-2023-Project-Sarcasm-Detection`

2 Abstract

The Challenge of Detecting Sarcasm in NLP

Sarcasm detection in natural language processing (NLP) presents significant challenges due to its nuanced and context-dependent nature. Unlike direct sentiment expressions, sarcasm is marked by a complex blend of tone and context, often contrasting literal meaning with intended implication. This complexity becomes particularly pronounced in text-based communication where key non-verbal cues like vocal tone and facial expressions are absent, posing additional challenges for accurate computational detection.

Sarcasm in Online Communications and The Choice of Reddit for Sarcasm Analysis

The prevalence of sarcasm in online social media, where non-verbal cues are lacking, necessitates a nuanced understanding of language and context. Reddit, with its diverse range of communities and unique communication styles, offers an ideal dataset for sarcasm analysis. The platform's threaded conversation structure, linking replies to specific comments, provides essential context, aiding in the identification of sarcasm. This makes Reddit a valuable resource for exploring and understanding the nuances of sarcastic expressions in online discourse.

Objective of Our Study

Our study focuses on evaluating various deep learning models for their efficacy in detecting sarcasm, particularly within Reddit's text data. We aim to leverage the complex dynamics of Reddit's conversational threads to enhance sarcasm detection techniques. By training and evaluating these models on Reddit data, our goal is to contribute to the advancement of NLP systems in interpreting online communication with greater subtlety, mirroring human-like understanding of language nuances.

3 Model & Data

3.1 Model Selection

Model	**BERT**	**RoBERTa**	**XLNet**
Architecture	Transformer-based model with a multi-layer bidirectional Transformer encoder.	Follows BERT's architecture with significant modifications.	Follows BERT's architecture with significant modifications.
Training Strategy	Employs masked language modeling and next sentence prediction, aiding in understanding context from both sides of a token.	Trained on a larger dataset and for more iterations, with altered masking patterns and no next sentence prediction.	Predicts token sequences by considering all permutations of the input, unlike BERT's independent masked token predictions.
Specialization	The base model consists of 12 transformer blocks, 768 hidden layers, and 12 self-attention heads. The uncased version focuses on semantic meaning, ignoring case differences.	Outperforms BERT in various NLP benchmarks, particularly in tasks requiring deep contextual understanding	Excels in capturing long-range dependencies and complex linguistic patterns, beneficial for nuanced tasks like sarcasm detection.
Application	Effective in named entity recognition, sentiment analysis, and question-answering.	Excels in tasks requiring deep contextual understanding.	

Table 1: Comparisons of usage and properties of **BERT**, **RoBERTa** and **XLNet**.

3.2 Data Selection

The dataset utilized for this study is the "daniel2588/sarcasm" dataset from Hugging Face. It comprises 1,010,826 rows, each row representing a comment from Reddit along with various attributes. The key features of this dataset include:

- **Labels:** Binary indicators (0 or 1) signifying the presence or absence of sarcasm in each comment.
- **Comment:** The text of the Reddit comment, which is the primary data used for sarcasm detection.
- **Parent Comment:** The preceding comment in the conversation, providing context for the subsequent reply.
- **Additional Attributes:** Includes metadata such as author, subreddit, score, ups, downs, date, and created_utc.

This dataset was selected for its comprehensive coverage of conversational exchanges on Reddit, offering a rich source of data for understanding and identifying sarcastic remarks.

3.3 Preprocessing

For our study, preprocessing of the dataset was minimal due to the readiness and compatibility of the dataset with our chosen models. The dataset from "daniel2588/sarcasm" on Hugging Face was already structured in a manner conducive to natural language processing, eliminating the need for extensive preprocessing. Moreover, the advanced capabilities of the BERT-Base-Uncased, RoBERTa, and XLNet models, including their sophisticated tokenizers, allowed for efficient processing of the raw text. These models are designed to handle various linguistic nuances inherent in text data, making additional preprocessing steps unnecessary. This approach ensured that the contextual integrity and stylistic elements crucial for sarcasm detection were preserved in their original form.

4 Training

The training process involved six different models: BERT-Base-Uncased, RoBERTa, and XLNet, each trained on two sets of input data - one using only the 'comment' field and the other using a combination of 'parent_comment' and 'comment'. The following subsections detail the hyperparameters and the training procedure used for these models.

4.1 Hyperparameters

The hyperparameters were carefully chosen to optimize model performance while maintaining computational efficiency. Common hyperparameters across all models included:

- **Maximum Sequence Length:** 512 tokens. This limit was set to accommodate the longest sequences in our dataset while avoiding excessive padding.
- **Batch Size:** 8 per device for both training and evaluation. The choice of batch size balanced the need for efficient computation and model stability.
- **Gradient Accumulation Steps:** 8. This setting was used to effectively increase the batch size, aiding in stabilizing the training process.
- **Learning Rate:** 2e-5. This rate was chosen based on standard practices for fine-tuning transformer-based models.
- **Max Steps:** 10,000. This parameter set the limit on the number of training steps to prevent overfitting.
- **Evaluation Metrics:** Accuracy was used as the primary metric for evaluating model performance. Additional metrics such as precision, recall, and F1-score were also considered during the analysis phase.

4.2 Training Procedure

The training was conducted using the script provided, with modifications to accommodate each model and input configuration. Key steps in the training procedure included:

- **Data Shuffling:** The dataset was shuffled with a seed of 12902136 to ensure a random distribution of data across training and validation sets.
- **Input Configuration:** Two configurations were used for each model:
  1. Using only the 'comment' field.
  2. Combining 'parent_comment' and 'comment', separated by a newline character.

- **Model Evaluation:** The models were evaluated during training using a validation split of 20% and a test split of 10%. The best model was selected based on accuracy and further analyzed for performance metrics.

The training output for each model was saved in designated directories (e.g., `./out/roberta-base/` for RoBERTa with 'comment' input), allowing for a detailed comparison of the model performances based on the different input configurations.

5 Evaluation

Note: Models that are trained by combining the 'parent_comment' and the 'comment' fields are denoted with the suffix '_combined', and models that are trained with only the 'comment' field are denoted with the suffix '_notcombined'.

5.1 Training Loss

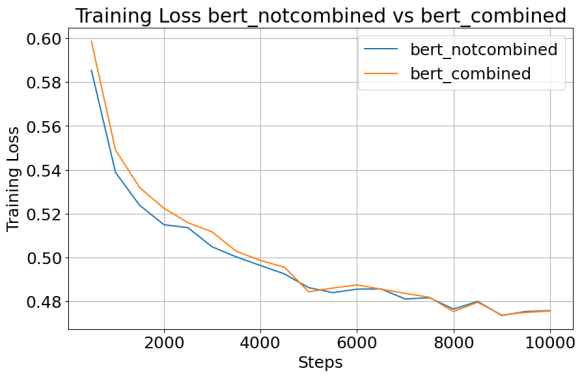

Figure 1: Training loss of **BERT** not-combined vs combined.

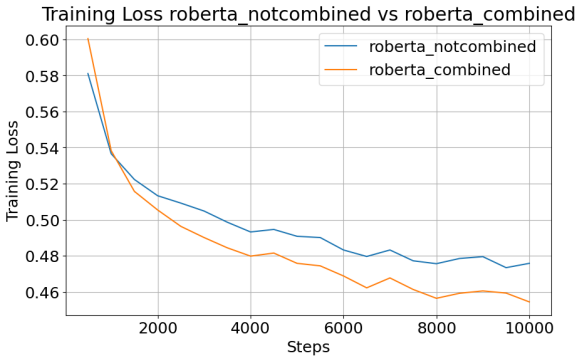

Figure 2: Training loss of **RoBERTa** not-combined vs combined.

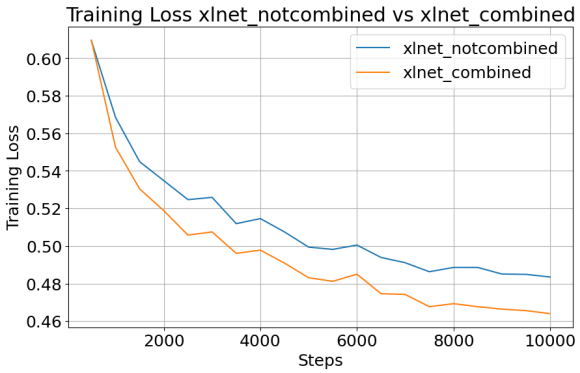

Figure 3: Training loss of **XLNet** not-combined vs combined.

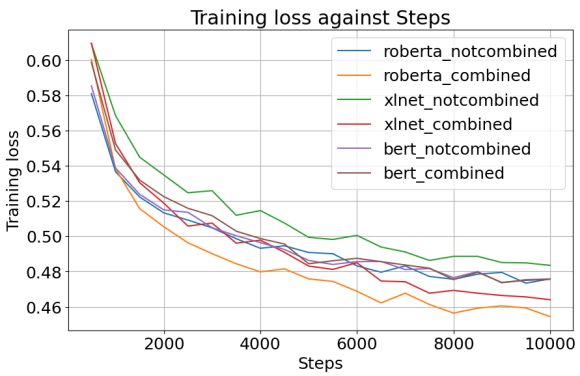

Figure 4: Training loss of all models.

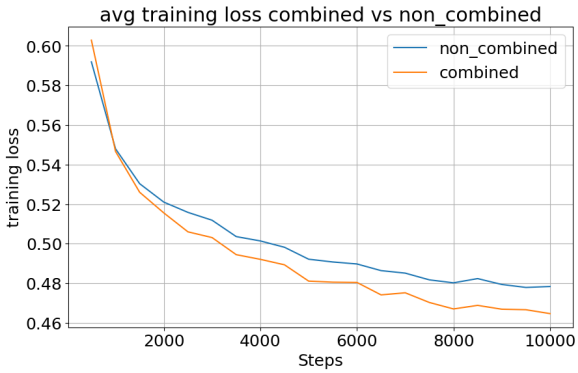

Figure 5: Average training loss of models using not-combined vs combined dataset.

5.2 Evaluation Accuracy

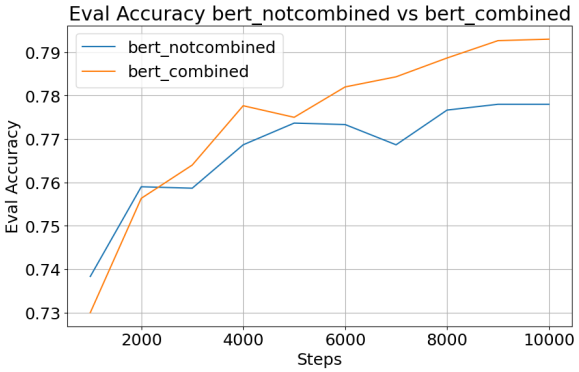

Figure 6: Evaluation accuracy of **BERT** not-combined vs combined.

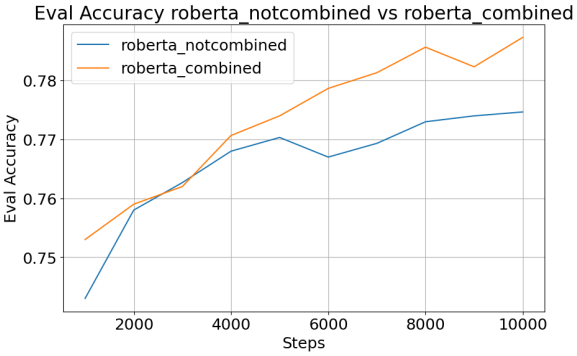

Figure 7: Evaluation accuracy of **RoBERTa** not-combined vs combined.

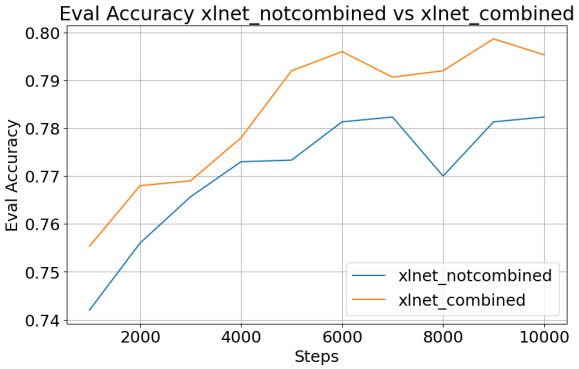

Figure 8: Evaluation accuracy of **XLNet** not-combined vs combined.

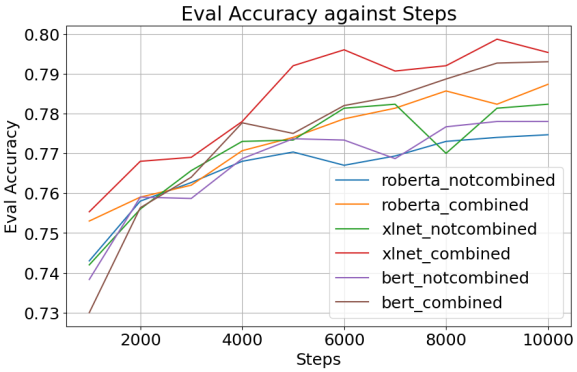

Figure 9: Evaluation accuracy of all models.

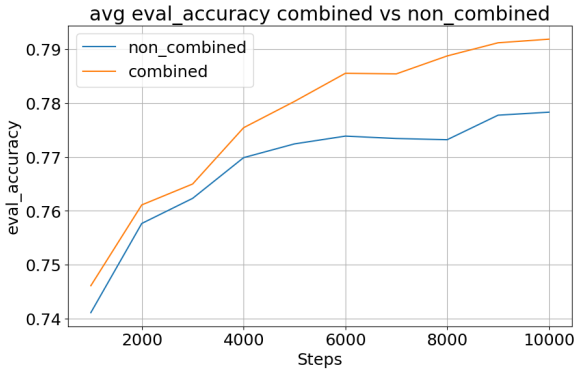

Figure 10: Average evaluation accuracy of models using not-combined vs combined dataset.

Model	**BERT**	**RoBERTa**	**XLNet**
Not Combined	0.778	0.775	0.782
Combined	0.793	0.787	**0.799**

Table 2: Best model evaluation accuracy of each model configuration.

6 Conclusion

It is evident from the training loss graphs that further fine-tuning can improve a model's performance in detecting sarcasm in comments from online forums such as Reddit.

Additionally, the inclusion of comments with the combined datasets display a marked improvement over training with the non-combined datasets. Average training loss showed that models that are trained with additional contextual input (combined) produced lower training losses compared to those without (not-combined). This finding corroborates with the evaluation accuracy results, as models with additional contextual input consistently performed better than models without, as evident in the average evaluation accuracy graph.

Throughout the training process with the three models, it is evident that XLNet emerges as the most well-suited for the task, outperforming BERT-Base-Uncased and RoBERTa. This observation holds true for both combined and non-combined training scenarios, as illustrated by the eval accuracy scores presented in Table 2. XLNet consistently demonstrates superior performance, followed by BERT-Base-Uncased, and RoBERTa lags behind in terms of effectiveness for the given task.