National Yang Ming Chiao Tung University (國立陽明交通大學)

Data Mining HW2

Quentin Ducoulombier (昆丁)
Student Id: 312551811

Data Mining

顏安孜

2024-05-14

1.How do you select features for your model input, and what preprocessing did you perform to review text?

In selecting features for the model input, I focused on the `title` and `text` fields from the dataset. These fields provide substantial context and information about the user's experience and sentiment regarding the product, which are critical for predicting the rating. The `verified_purchase` and `helpful_vote` fields were excluded to simplify the feature set and focus on the textual data, which is the primary source of sentiment and opinion.

Preprocessing Steps

To ensure the textual data is clean, consistent, and compatible with BERT, I performed several preprocessing steps:

1. **Lowercasing**: Convert all text to lowercase to standardize it and reduce variability caused by case differences.
2. **Emoji Transformation**: Converted emojis to their corresponding text representation to capture the sentiment they convey.
3. **HTML Entity Conversion**: Converted HTML entities to normal characters to ensure correct interpretation.
4. **HTML Tag Removal**: Removed HTML tags to eliminate irrelevant content and reduce noise.
5. **Removing Text within Double Square Brackets**: Removed text within double square brackets to discard metadata or extraneous information.
6. **URL Removal**: Removed URLs to reduce noise.
7. **Apostrophe Normalization**: Removed backslashes from apostrophes to ensure correct interpretation during tokenization.
8. **Extra Space Removal**: Removed extra spaces and trimmed leading/trailing spaces for clean text.
9. **Unicode Normalization**: Normalized Unicode characters to ASCII to simplify the text and reduce variability.
10. **Repeated Punctuation Removal**: Reduced repeated punctuation to a single character for consistency.
11. **Standardizing Apostrophes and Quotes**: Replaced different types of apostrophes and quotes with a standard version for consistency.
12. **Handling Partially Decoded HTML Entities**: Removed any remaining partially decoded HTML entities to clean up residual encoding issues.

These preprocessing steps ensure that the textual data is compatible with BERT, which is designed to handle clean and standardized text. By focusing on the `title` and `text` fields and performing thorough preprocessing, I ensured that the input data for the model was clean, standardized, and ready for effective sentiment analysis. This preprocessing helps capture the essential sentiment and context from the reviews, which is crucial for accurately predicting the ratings.

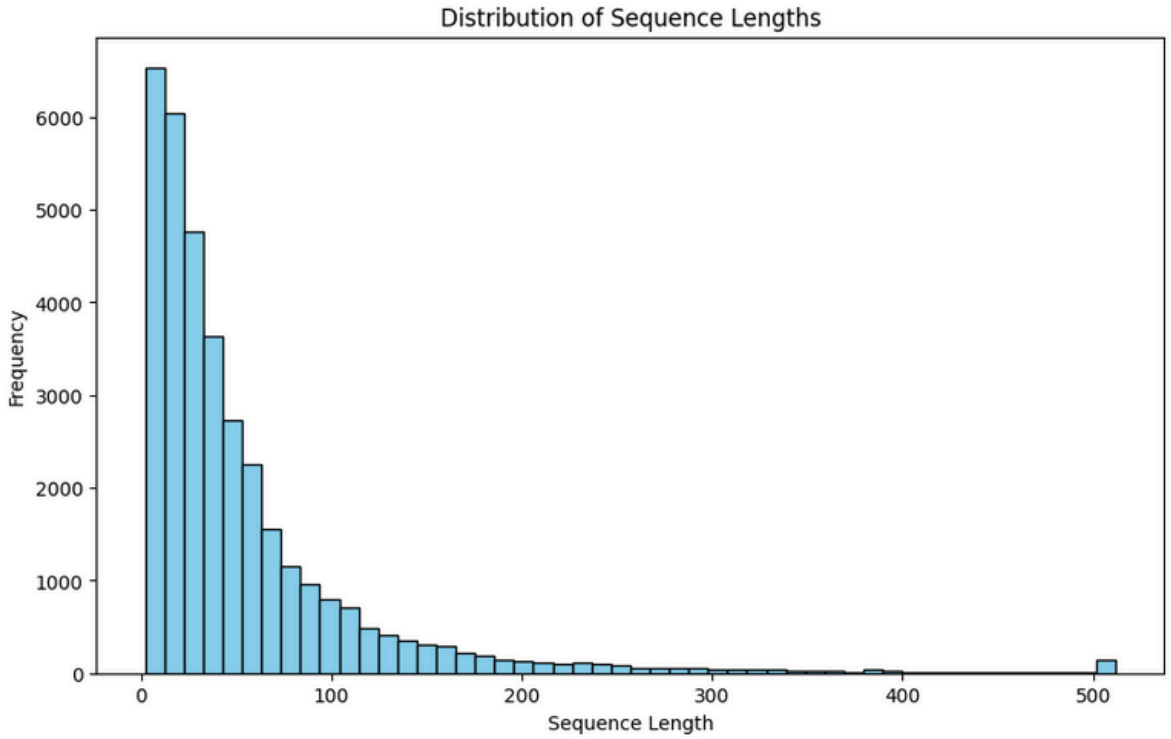
2. Please describe how you tokenize your data, calculate the distribution of tokenized sequence length of the dataset and explain how you determine the padding size

Tokenization Process

To tokenize my data, I used the BERT tokenizer from the Hugging Face `transformers` library. This tokenizer is specifically designed to handle the vocabulary and tokenization rules of the BERT model, ensuring compatibility and optimal performance. The tokenizer converts each text input into a sequence of tokens, which are numerical representations of the words or subwords in the text. It also adds special tokens like `[CLS]` at the beginning and `[SEP]` at the end of each sequence to indicate the start and end of the text, respectively.

Calculating the Distribution of Tokenized Sequence Length

After tokenizing the texts, I calculated the length of each tokenized sequence. This step was crucial for understanding the distribution of sequence lengths within my dataset. By plotting the distribution of these lengths, I could visualize how many tokens each review typically contains.



The histogram of sequence lengths revealed that most sequences were relatively short, with a long tail indicating some very long reviews. Descriptive statistics provided further insights, showing the mean, standard deviation, and various percentiles of sequence lengths. For example, the 95th percentile of the sequence

length was 179 tokens, meaning that 95% of the sequences were 179 tokens or shorter.

Determining the Padding Size

To determine the padding size, I aimed to cover the majority of sequences without unnecessarily increasing the computational load. Based on the distribution analysis, I chose the 95th percentile value, which was 179 tokens, as the `max_len` for padding. This choice ensures that 95% of the sequences fit within this length, minimizing the need for truncation while avoiding excessive padding for the majority of sequences.

```
count    35000.000000
mean        54.765229
std         67.916398
min          2.000000
50%         33.000000
75%         65.000000
80%         77.000000
85%         95.000000
90%        122.000000
95%        179.000000
99%        368.000000
max        512.000000
Name: token_length, dtype: float64
Choose max_len = 179
```

Using a `max_len` of 179, I applied padding to ensure that all sequences had the same length, which is necessary for batch processing in neural networks. Padding adds zeros to the end of sequences that are shorter than `max_len`, ensuring uniform input dimensions for the model.

By tokenizing the texts using the BERT tokenizer, calculating the distribution of tokenized sequence lengths, and choosing a padding size based on the 95th percentile length, I ensured that the input data was well-prepared for the BERT model. This preprocessing step optimized the balance between computational efficiency and the retention of important information from the reviews.

3. Please compare the impact of using different methods to prepare data for different rating categories

To compare the impact of different data preparation methods on predicting rating categories, I experimented with two main approaches:

Method 1: Text and Title Preprocessing (main method)

In this method, I combined the text of the review with its title, performing standard preprocessing steps such as lowercasing, removing HTML tags, handling emojis, and normalizing unicode characters. This ensured the text data was clean and standardized, providing the BERT model with comprehensive context from both the review title and content. This method resulted in a balanced performance across all rating categories, demonstrating the importance of combining the title with the text to capture the full sentiment and context of the review.

Method 2: Incorporating `helpful_vote`

In addition to the text and title preprocessing, I included the `helpful_vote` feature. I normalized this numerical feature and added it to the model. While this additional feature provided extra context about the perceived usefulness of the reviews, it did not significantly improve the model's accuracy or F1-score compared to Method 1. The performance metrics remained relatively similar, indicating that the `helpful_vote` feature might not add substantial predictive value in this context.

Comparison and Conclusion

Through these experiments, I found that focusing on combining the text and title with thorough preprocessing yields significant improvements in model performance. Although incorporating the `helpful_vote` feature adds some context, it does not substantially enhance the predictions. Therefore, the most effective approach for preparing data for different rating categories involves robust text preprocessing and leveraging both the review text and title to ensure the model can accurately interpret and analyze the content.