



Article

# Punctuation Restoration with Transformer Model on Social Media Data

Adebayo Mustapha Bakare <sup>1</sup>, Kalaiarasi Sonai Muthu Anbananthen <sup>1,\*</sup> , Saravanan Muthaiyah <sup>2</sup> ,  
Jayakumar Krishnan <sup>1</sup> and Subarmaniam Kannan <sup>1</sup>

<sup>1</sup> Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia

<sup>2</sup> Faculty of Management, Multimedia University, Cyberjaya 63100, Malaysia

\* Correspondence: kalaiarasi@mmu.edu.my

**Abstract:** Several key challenges are faced during sentiment analysis. One major problem is determining the sentiment of complex sentences, paragraphs, and text documents. A paragraph with multiple parts might have multiple sentiment values. Predicting the overall sentiment value for this paragraph will not produce all the information necessary for businesses and brands. Therefore, a paragraph with multiple sentences should be separated into simple sentences. With a simple sentence, it will be effective to extract all the possible sentiments. Therefore, to split a paragraph, that paragraph must be properly punctuated. Most social media texts are improperly punctuated, so separating the sentences may be challenging. This study proposes a punctuation-restoration algorithm using the transformer model approach. We evaluated different Bidirectional Encoder Representations from Transformers (BERT) models for our transformer encoding, in addition to the neural network used for evaluation. Based on our evaluation, the RobertaLarge with the bidirectional long short-term memory (LSTM) provided the best accuracy of 97% and 90% for restoring the punctuation on Amazon and Telekom data, respectively. Other evaluation criteria like precision, recall, and F1-score are also used.

**Keywords:** punctuation restoration; transformers models; Bidirectional Encoder Representations from Transformers (BERT); long short-term memory (LSTM)



**Citation:** Bakare, A.M.; Anbananthen, K.S.M.; Muthaiyah, S.; Krishnan, J.; Kannan, S. Punctuation Restoration with Transformer Model on Social Media Data. *Appl. Sci.* **2023**, *13*, 1685. <https://doi.org/10.3390/app13031685>

Academic Editor: Javier Hernando

Received: 20 December 2022

Revised: 15 January 2023

Accepted: 18 January 2023

Published: 28 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Several businesses have recently encouraged their customers to provide reviews and feedback on products and brands. Generally, this information is presented in the form of text and sometimes audio data [1,2]. There is plenty of information present in these data that are useful for brands to improve their businesses. The information from these data can be extracted and analyzed using sentiment analysis. Sentiment analysis aims to determine the sentiment of the text concerning the situation, in this case, brands or products. The sentiment analysis outcome is a positive or negative polarity value or, in some cases, neutral [3–5]. Several key challenges are faced during sentiment analysis [6]. One major problem is determining the sentiment of complex sentences, paragraphs, and text documents. A sentence with multiple parts might have multiple sentiment values, which means a combination of positive and negative statements. Predicting the overall sentiment value for this paragraph will not produce all the information necessary for businesses and brands. Therefore, a statement with multiple sentences should be separated into single sentences. The most frequently used method for separating text with multiple sentences is using sentence stoppers, such as periods, exclamation points, and question marks. Due to the nature of data extracted from online sources often containing improper punctuation, separating the sentences may be challenging. As a result, a more sophisticated punctuation-restoration method is required to restore the punctuation before separating paragraphs into simple sentences.

The statement in Table 1 is a properly punctuated sentence. The overall sentiment of this review will be predicted to be positive using the sentiment-analysis technique.

However, if we observe the review closely, we see that there is negative and neutral information in the review. Therefore, this negative and neutral sentiment will be lost, and the company cannot know all the comments or improve the brand based on customer feedback.

**Table 1.** Example sentence before splitting using punctuation and sentiment-analysis information.

Text—Before Splitting	Polarity
I use this every day on my commute. Great battery life. I like the built-in dictionary. It is easy to transfer using pdf format through email or mobile files. But no backlight which makes not readable at night.	Positive

On the other hand, when the text is separated using punctuation (Table 2), the statement above is divided into five sentences. Sentiment analysis will be able to analyze each sentence. The first sentence is considered neutral, the second to fourth are positive, and the last sentence is negative. Hence, splitting paragraphs into simple sentences will give more information. Therefore, the text has to be properly punctuated to split the paragraphs. However, as previously mentioned, most social media text is not properly punctuated, as shown in Table 3.

**Table 2.** Example sentence after splitting using punctuation and sentiment-analysis information.

Text—After Splitting	Polarity
I use this every day on my commute.	Neutral
Great battery life.	Positive
I like the built-in dictionary.	Positive
It is easy to transfer using pdf format through email or mobile files.	Positive
But no backlight which makes not readable at night.	Negative

**Table 3.** Example sentence extracted from social media with no punctuation.

Text	Polarity
The shoe is responsive and that is good but it would be better to have longer shoe laces.	Positive

Table 3 shows an example review extracted from social media about a shoe. The review contains no punctuation, and the sentiment polarity of the whole sentence is produced as a Positive value. However, after reading the text, we see two sentiment polarities in the sentence. Separating the statement into two sentences will give us a positive and a negative statement, as shown in Table 4. Social media texts may contain improper punctuation or no punctuation [7]. As a result, it is required to restore punctuation in the text before condensing the paragraph into simple sentences.

**Table 4.** Possible punctuation placement to produce more sentiment information.

Text	Polarity
The shoe is responsive and that is good.	Positive
But it would be better to have longer shoe laces.	Negative

Punctuation restoration in the text is a challenging natural language processing (NLP) task that enables further text processing, text readability, machine translation effectiveness, etc., requiring the introduction of punctuation marks in the right position into a text [8]. Punctuation marks are used to arrange grammatical structures and explain the meaning of sentences in written language. Punctuation significantly impacts the readability and understandability of text for both human and machine readers. Punctuation restoration has been widely used in automatic-speech-recognition tasks [9]. When speech is transcribed

into text, the information lacks proper punctuation, which makes it challenging to read. Therefore, punctuation must be included to make the text created from the speech more readable. This is one instance where punctuation restoration has been used.

Punctuation has been used to split paragraphs into single sentences with full stops, exclamation marks, and question marks. At the same time, certain language conditions must be comprehended. For example, decimal points between numbers do not represent periods. In [6], the regular-expression (Regex) approach is used in sentence splitting. Tomanek, in 2007, utilized conditional random fields (CRF) in splitting sentences and tokens in a document with biomedical language text [10]. Manning et al., in 2014, developed a natural-language tool kit using annotation, which can be utilized for sentence separation and other natural-language processing tasks [11]. The annotators include conditional-random-field taggers trained on various corpora and two rule-based systems to recognize money and numbers. This library also provides means to add custom annotations to the existing one to help improve the accuracy of the task. Another NLTK library, developed by Loper & Bird [12], is a computational linguistic library designed to compute most natural-language problems, including sentence splitting. All these methods of splitting sentences require a well-punctuated document, but the main problem from social media reviews is that they are not properly punctuated.

In this research, we apply punctuation restoration to the sentiment analysis process to extract all the possible information from the sentences. A properly punctuated statement is necessary to extract more variety of sentiment polarity, so a review statement must be divided into simple sentences. Therefore, this paper focuses on creating a method of restoring punctuation in review statements extracted from social media. We separate the sentences by positioning full stops, exclamation marks, and question marks. This enables us to classify each sentence and provide sentiment value. However, restoring punctuation requires us to consider all the other punctuation, like commas, semicolons, etc., in our model.

This study is organized as follows: Section 2 discusses the related work associated with punctuation restoration. Section 3 is about the methodology employed in developing the punctuation-restoration model. Section 4 focuses on the experimental result, and Section 5 concludes the paper.

## 2. Related Works

Several studies have been conducted on punctuation restoration on transcript text from speech and speech-recognition systems [13–15], but there has been no previous research on automatic punctuation restoration for social media data. Based on the survey by Păis, there are seven approaches to capitalization and punctuation restoration, including the rule-based approach, n-gram-based language models, capitalization as a discrimination decision, hidden-event language models, the boosting approach, conditional-random-fields probabilistic models, and neural-network architectures [16]. The two most frequently used techniques among the capitalization-restoration models are conditional random fields and neural network architectures.

Most early attempts at punctuation restoration relied on lexical, acoustic, prosodic, or a combination of these elements [17,18]. These approaches require several processes, including reader sound interpretation and extensive knowledge of the language to represent the rules. The major drawback is the requirement to represent and capture all the possible variations and exceptions of the language. The models are evaluated based on the F1 score and slot-error rate (SER). Jansche, who used an n-gram model, recorded that increasing the data in the dataset improves the F1 score of the model. Stating that the data size is proportional to the F1 score of the model. Miranda compared two streams of languages, English and Portuguese, in terms of translation checking and comparison of their possible split locations. The automatically transcribed speech is aligned with a manual reference and compares two baseline speech and non-speech components. The model with baseline probability showed an improved 2.4% in SER.

The punctuation-restoration task is considered a sequential problem, and there are statistical models that are known for solving sequential problems, like the hidden Markov model (HMM) and conditional random fields (CRFs). A CRF is considered the most suitable probabilistic method for segmenting and labelling sequential data. Conditional random fields have been frequently employed to predict punctuation position in text documents [8,19]. Lu compared a linear chain CRF (L-CRF) and factorial chain CRF (F-CRF) for both English and Chinese language using the F1 score. The F-CRF outperforms the L-CRF in both language cases, with a 1% increase for Chinese and a 4% increase for English. Marco evaluated his CRF model on different datasets from four sources: *The Wall Street Journal*, English Wikipedia, Competition Organizers, and Reuter RCV1. The results were evaluated based on their F1 scores, with the best-performing dataset being from *The Wall Street Journal*, recording a 62% F1 score. These models are based on the idea that the probability of a punctuation mark at a given position in a sentence can be determined based on the surrounding context. However, these models have limitations in terms of making assumptions of independence between words. For example, in a sentence where the first word determines the punctuation at the end of a sentence, these models do not carry that level of detail attention and dependencies. In addition, these models have difficulty dealing with the out-of-vocabulary word, which are words unknown to the model during training.

Recently, deep-learning models like long short-term memory (LSTM) [20], gated recurrent units [9,21], convolutional neural network (CNN) [22,23], and pre-trained transformer models [13,15] have been utilized for this task. These models have shown state-of-the-art performance and demonstrated the ability to generalize on various forms of data. However, the data used in these models are well-structured and have carefully annotated automatic speech recognition datasets. Other forms of data, like social media, newspapers, etc., have not been explored. In many other natural language processing tasks, the transformer model demonstrated promising results compared to other deep-learning models employing recurrent or convolutional neural networks [24]. Research is ongoing in this area, which has led to the development of transformer-based language models, e.g., BERT [25] and RoBERTa [26], being created. Compared to other NLP tasks, these models have not been thoroughly used to investigate punctuation-restoration problems. As a result, we aim to investigate BERT architectures and fine-tune pre-trained models in combination with the neural-network model, bidirectional LSTM, and gated recurrent unit (GRU). In this paper, we utilize the approach from Tilk and Alumäe (2016) to encode the input text before passing it into the neural network using directional LSTM and [21] transformer models from [13,25].

### 3. Methodology

This section discusses the proposed methodology for the problem we are studying. This includes the dataset to be used, the preprocessing algorithm, and the punctuation-restoration architecture.

#### 3.1. Dataset

The dataset used in this study is for Amazon products, consisting of over 34,000 customer reviews with 22 columns available on Kaggle [27]. The dataset includes basic information about products, ratings, and review text. Only one column out of the 22 columns of the dataset, which is the text-reviews column, is used in this research. The text reviews are extracted and concatenated together to form a document. The overall content of the document is illustrated in Table 5. A total of 60% of the sentences in the document were used to train the algorithm for punctuation restoration and 40% for testing.

**Table 5.** Possible punctuation placement to produce more sentiment information.

Dataset	Amazon Products Reviews	Telekom Malaysia (TM) Reviews
Total Reviews	34,659	500
Total Words	990,326	13,317
COMMA	26,148	251
PERIOD	72,076	730
QUESTION	271	21
OTHERS	891,831	12,315

In addition to this dataset, we used data from Telekom Malaysia (TM) customer-care reviews. The data contains 5000 customer reviews [28]. The performance of the model was also evaluated using these two datasets.

### 3.2. Preprocessing

Both datasets were preprocessed using the method described by [14]. Four labels were used to classify this data: COMMA, PERIOD, QUESTION, and O, which stand for others. The comma “,” is represented by the COMMA class. The punctuation “.” is represented by the PERIOD class. The punctuation “?” is represented by the QUESTION class. The last class, O, denotes a scenario in which there is no punctuation. Other punctuations like exclamation marks and semicolons are classified under PERIOD. Colons and dashes are classified as COMMA.

For example,

*“But no backlight which makes not readable at night.”*

Step 1: The text data was first tokenized, and the punctuation position was detected by getting each token’s last character.

This text was tokenized to form a list of words like

*[‘But’, ‘no’, ‘backlight’, ‘which’, ‘makes’, ‘not’, ‘readable’, ‘at’, ‘night.’].*

Step 2: The punctuation was removed and replaced with a label name for tokens containing punctuation.

*[But O no O backlight O which O makes O not O readable O at O night PERIOD].*

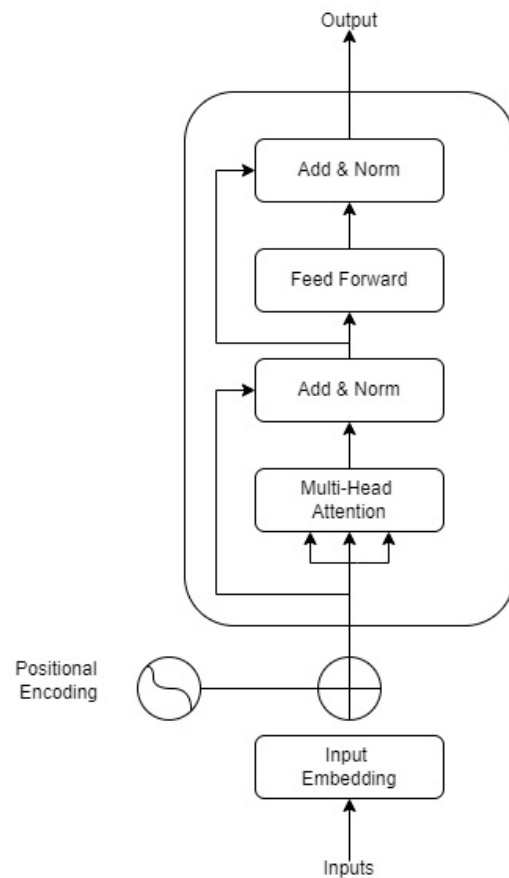
The punctuation was determined by taking each token and taking the last character. If the last character was punctuation, then the punctuation was removed and replaced with its label equivalent, either PERIOD, COMMA, or QUESTION. If no punctuation was found, the label was represented as O. The first eight tokens in this sample have no punctuation, while the last token, “night,” has a period as its last character. The period is removed, and the label PERIOD is attached. The other first eight tokens have the label O attached. This process was repeated for all the text in the dataset, and the results were combined to train the punctuation model.

### 3.3. Transformer Model

Transformer models are generally divided into the encoder and the decoder [29]. The encoder takes the English word simultaneously and generates embeddings for every word. These embeddings are vectors that encapsulate the meaning of these words, and words with similar meanings would have closer numbers in their vector representation. The decoders take the embedding vectors as input and generate corresponding output depending on the task.

This paper concentrates on the transformer encoder used to encode our input text. Figure 1 illustrates a simple encoder as described in [29]. The input embeddings are text converted into vector representation with added positional vectors. The multiheaded attention layer applied self-attention, which associates each word in the input with the other text words. This used a key/query/value concept from information-retrieval systems. The result was added, normalized, and fed into a feed-forward network, which produced encoded information about the input text. The result was added and normalized to form

the output. Stacking multiple simple transformer encoders produced Bidirectional Encoder Representation from Transformers, called BERT. The pre-trained model used is available in Hugging Face's transformer library.



**Figure 1.** Simple Transformer encoder.

### 3.4. Architecture

The proposed system architecture comprised two models, which used a bidirectional LSTM layer and the other a GRU layer. Every other part of the architecture was the same for both models. Figure 2 illustrates a sample training cycle of a text containing fourteen words and the punctuation after each word, if any, present. The transformer model was used to create embeddings for each text, which was represented with  $T_1, T_2, \dots, T_{14}$ . The different transformer models were used in different experiments. The result was fed into the single bidirectional layer of either LSTM or GRU. The result from this layer was passed into a fully connected linear layer,  $y_1, y_2, \dots, y_{14}$ , and the output layer produced one class for each word, which was one of the four classes (PERIOD, COMMA, QUESTION, O).

Step 1: The dataset was passed to the pre-trained transfer model as input and labels. Output was a vector representation of the input text.

Step 2: Vector representation of the text was passed to the neural network layer (LSTM/GRU). The output was fed into a linear, fully connected neural network layer and finally to a SoftMax output layer with four neurons representing each class.

In this paper, we also evaluate different BERT transformer models. The principle of transfer learning has been used in many NLP tasks, as discussed in the literature review. The model is trained for different NLP applications that can be fine-tuned to perform other NLP tasks. We briefly discuss some of the models used in this study.



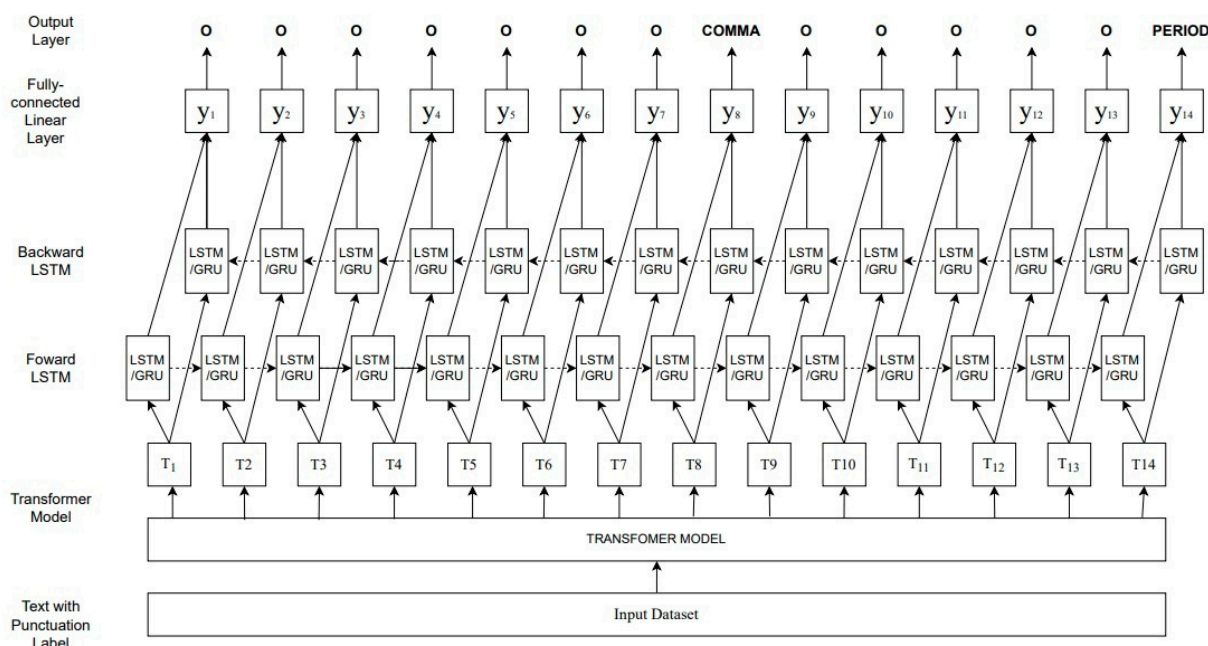


Figure 2. Punctuation-restoration-model architecture.

### 3.4.1. BERT [25]

BERT is a bidirectional transformer for training over plenty of unlabeled data to learn about language representation, which can be fine-tuned for other specific natural-language processing tasks. BERT uses a masked-language model (MLM) and next-structure prediction (NSP) to train the model over many Google data. The data comprises about 3.3 billion words. The BERT model has two versions, which are the base and large models.

### 3.4.2. XLNET [30]

XLNet is an improved version of BERT that introduces permutation language modeling, where all tokens are predicted randomly. This helps improve the model to learn bidirectional relationships and better handle dependencies and relations between words. The XLNet comprises two versions (base and large), just like in the case of BERT.

### 3.4.3. RoBERTa [26]

RoBERTa is also known as the robust optimized BERT approach. As the name implies, this is an improvement in BERT training data, which trains with 1000% more data. A dynamic-masking method is used for the token, just like the masked-language model used in BERT. RoBERTa is a BERT without the next-structure-prediction model (NSP). Like BERT, the RoBERTa comprises two versions (base and large).

### 3.4.4. DistillBERT [31]

The DistillBERT is an approximate version of BERT. The DistillBERT uses a distillation technique to filter training parameters by using only half of the parameters. It is 60% faster than the normal BERT and retains 97% of the language capabilities of the BERT model. The main idea is that the output can be approximated using a smaller network when trained by a large neural network.

### 3.4.5. AIBERT [32]

AIBERT is also known as A Lite BERT. The architecture of AIBERT is similar to BERT, but the input-level embedding and hidden-level embedding are separated, which allows them to have different sizes. The reduction in the hidden-level embedding reduces the training parameter by 80%.

### 3.5. Training

In addition to the pre-trained model, two different layers were used in two different training models. A bidirectional LSTM and a fully connected layer were used in one, and the GRU and a fully connected linear layer were used in the second. These layers were placed on top of the pre-trained network. The dimension of the embedding network was used as the input for the BiLSTM and the GRU layer. The output of the LSTM layer was concatenated at each time step and fed to a fully connected layer that contained the output neuron, which specifies the punctuation to be chosen. This process was replicated for the GRU model as well. During training, the maximum length of the word sequence used was 250, and each sequence had a start token and an end token. We used a batch size of 5, and the sequences were shuffled before each epoch. The learning rate chosen was  $1 \times 10^{-6}$  (0.000001). All the parameters followed the same methods as used in [13], which was proven to have the best performance based on accuracy, with the only changes made in the batch size. Other training parameters were kept at default. The model's performance was measured on accuracy, precision, F1 score, and recall. We tested the model using a review statement from the Amazon reviews dataset. Figure 3b shows the statement without any punctuation. We used the model to restore punctuation to the reviews, as shown in Figure 3b, and compared it with manually assigned punctuations. Based on the comparison, most punctuation was correctly predicted, achieving 95% accuracy.

In addition to the pre-trained model, two different layers were used in two different training models. A bidirectional LSTM and a fully connected layer were used in one, and the GRU and a fully connected linear layer were used in the second. These layers were placed on top of the pre-trained network. The dimension of the embedding network was used as the input for the BiLSTM and the GRU layer. The output of the LSTM layer was concatenated at each time step and fed to a fully connected layer that contained the output neuron, which specifies the punctuation to be chosen. This process was replicated for the GRU model as well. During

(a)

I read a huge number of books in Kindle format. This Kindle Fire is perfect for that. Its just the right balance between screen size and easy carrying. It also is great for Audible users (you will want to install a micro SD card, though). You can also use it to watch Netflix and the like, although I dont. As a tablet, it leaves a lot to be desired. If you are looking for a tablet, give this a pass. It has a lot of system lag, and Wifi reception is terrible, even in areas where other devices show a strong signal. Also, you should know that you cant access any apps on the Google Play store. its locked into Amazon. None of that bothers me, because, like I said, I only use it to read, listen to Kindle books, and this device works perfectly for that. Bottom line, if you just want a Kindle reading platform with some extra functionality, this will work well for you. If youre looking for the full Android tablet experience, look elsewhere.

(b)

**Figure 3.** This figure shows a test-sample of the model against a review paragraph: (a) paragraph without punctuation; (b) restored paragraph punctuation using model.



#### 4. Experimental Result

Table 6 summarizes the experimental results for each neural network and the transfer model used. These findings are based on the XLNET, BERT, AIBERT, RoBERTa, DistillBERT base, and large variants of the models. The GRU+RoBERTaLARGE model achieved the highest training accuracy of 96%.

**Table 6.** Training experiment result (Amazon reviews).

Model	Comma			Period			Question			Others			Training Acc
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
LSTM+BERT <sub>BASE</sub>	0.65	0.14	0.23	0.63	0.53	0.58	0.64	0.31	0.41	0.91	0.99	0.95	0.90
LSTM+BERT <sub>LARGE</sub>	0.47	0.27	0.34	0.75	0.20	0.32	0.55	0.23	0.33	0.91	0.99	0.95	0.89
LSTM+XLNet <sub>BASE</sub>	0.48	0.26	0.34	0.85	0.17	0.28	0.56	0.21	0.31	0.90	0.99	0.94	0.88
LSTM+XLNet <sub>LARGE</sub>	0.46	0.38	0.42	0.86	0.20	0.33	0.54	0.29	0.34	0.92	0.99	0.96	0.89
LSTM+AIBERT <sub>BASE</sub>	0.60	0.15	0.24	0.62	0.53	0.57	0.62	0.31	0.42	0.92	0.99	0.96	0.89
LSTM+RoBERTa <sub>BASE</sub>	0.41	0.28	0.33	0.85	0.06	0.11	0.44	0.17	0.25	0.90	0.99	0.95	0.88
LSTM+RoBERTa <sub>LARGE</sub>	0.77	0.70	0.73	0.83	0.86	0.84	0.80	0.78	0.79	0.98	0.99	0.98	0.95
LSTM+ DistilBERT <sub>BASE</sub>	0.45	0.05	0.09	0.72	0.05	0.09	0.53	0.04	0.08	0.87	0.99	0.92	0.86
GRU+ BERT <sub>BASE</sub>	0.42	0.23	0.30	0.62	0.16	0.25	0.49	0.19	0.27	0.91	0.99	0.95	0.89
GRU+ BERT <sub>LARGE</sub>	0.46	0.17	0.25	0.75	0.19	0.31	0.58	0.18	0.27	0.90	0.99	0.95	0.89
GRU+ XLNet <sub>BASE</sub>	0.45	0.33	0.38	0.85	0.16	0.27	0.52	0.24	0.33	0.91	0.99	0.95	0.89
GRU+ XLNet <sub>LARGE</sub>	0.42	0.41	0.42	0.89	0.06	0.11	0.45	0.24	0.31	0.92	0.99	0.96	0.89
GRU+AIBERT <sub>BASE</sub>	0.62	0.32	0.42	0.71	0.57	0.66	0.67	0.42	0.52	0.93	0.99	0.96	0.91
GRU+AIBERT <sub>LARGE</sub>	0.47	0.23	0.04	0.60	0.31	0.41	0.59	0.15	0.24	0.87	0.99	0.94	0.88
GRU+RoBERTa <sub>BASE</sub>	0.45	0.36	0.40	0.78	0.14	0.23	0.50	0.25	0.33	0.92	0.99	0.95	0.89
GRU+RoBERTa <sub>LARGE</sub>	0.77	0.71	0.74	0.83	0.87	0.85	0.79	0.78	0.79	0.98	0.99	0.98	0.96
GRU+DistilBERT <sub>BASE</sub>	0.41	0.18	0.25	0.66	0.10	0.17	0.47	0.17	0.21	0.90	0.99	0.94	0.87

The training lasted 48 h on a Core i7 CPU 3.60GHz with 16GB RAM for the LSTM+RoBERTaLARGE. Other models trained with lesser time compared to this. The GRU trained faster than the LSTM, but the LSTM showed slightly better accuracy during testing. This is associated with the LSTM model's learning method, allowing it to learn more information than the GRU model. The result was analyzed using the recall, precision, and F1-score metrics.

Tables 7 and 8 show the outcomes of testing the model using the Amazon reviews and TM message-reviews datasets, respectively. Table 7 demonstrates that while the scores from all the models are relatively close, the RoBERTa model stands out among the others. This is similar to the result in Table 8. This result shows that the amount of training data used in the pre-training of the RoBERTa model impacts the model's performance. As a result, the model has a stronger grasp of the English language's context. The only significant downside is the length of time required for model training. The training took around three times as long as the other models in our study. In addition, the nature of the English language plays a vital role in the recall value of the classes. The number of times punctuation appears at the end of a word is extremely low compared to the number of times it does not. Therefore, we can see the recall values of the classes COMMA, PERIOD, and QUESTION are low compared to the OTHERS class in all the models.

Table 9 summarizes the best model based on the results of the Amazon reviews testing. Figure 4 compares the best-proposed model to those described in other studies. For comparison, the same transcribed dataset utilized in other models was employed in testing our model. All other models were trained using annotated transcribed text, whereas we used social media data to train our model. The result indicates that the training data utilized in our model effectively produced the desired outcome during testing and was comparable to that employed by other researchers. Using our model, we achieved much better results in the COMMA class. As a result, our algorithm is significantly better at predicting comma location in a text. When predicting the classes COMMA and QUESTION, LSTM+RoBERTaLARGE has greater precision, recall, and F1 score than the other classes, with values of 0.98, 0.98, and 0.98, respectively. Precision, recall, and F1 score have 0.86, 0.93, and 0.88, respectively, in the QUESTION class. LSTM + RoBERTaLARGE [13] did better to predict the PERIOD class, with precision, recall, and F1 score values of 0.88, 0.92, and 0.91, respectively, compared to LSTM+RoBERTaLARGE, which had precision, recall, and F1 score values of 0.76, 0.74, and 0.75, respectively.

Table 7. Test experimental results (Amazon reviews).

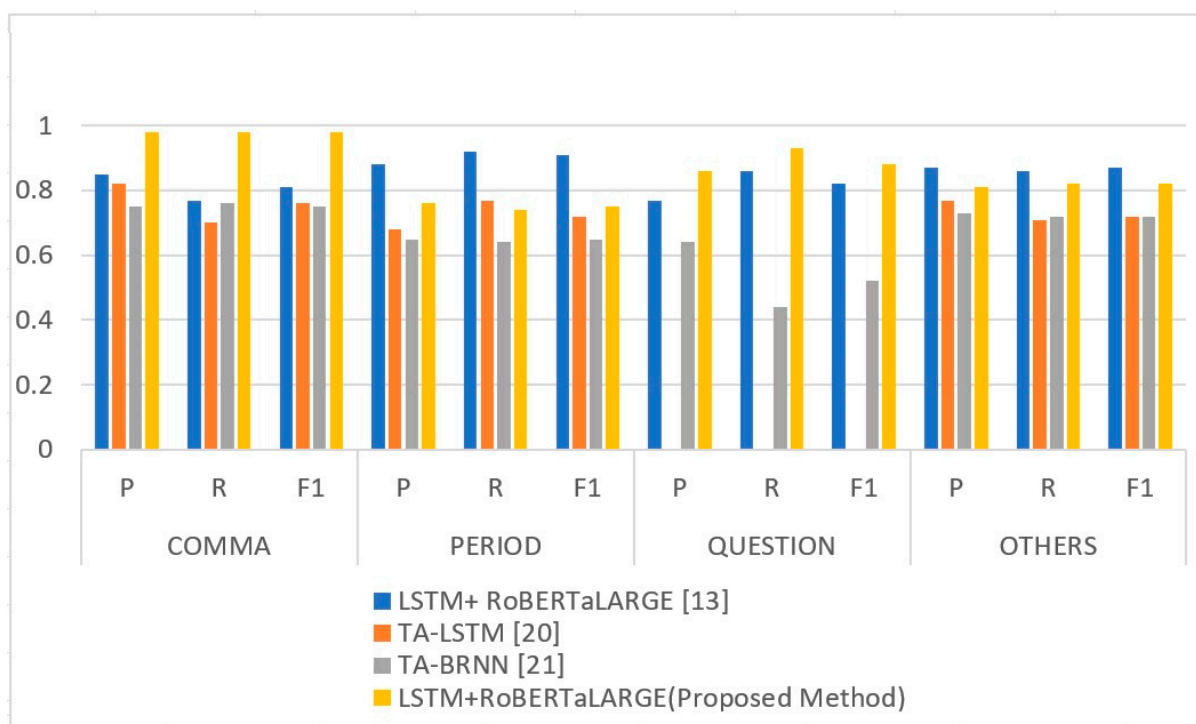
Model	Comma			Period			Question			Others			Testing Acc
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
LSTM+BERT <sub>BASE</sub>	0.63	0.13	0.22	0.61	0.48	0.54	0.61	0.30	0.40	0.92	0.99	0.96	0.90
LSTM+BERT <sub>LARGE</sub>	0.48	0.27	0.35	0.73	0.20	0.32	0.56	0.23	0.33	0.91	0.99	0.95	0.89
LSTM+XLNet <sub>BASE</sub>	0.44	0.21	0.29	0.83	0.17	0.19	0.55	0.20	0.28	0.91	0.99	0.95	0.89
LSTM+XLNet <sub>LARGE</sub>	0.45	0.40	0.41	0.84	0.29	0.32	0.54	0.28	0.37	0.93	0.99	0.96	0.90
LSTM+AlBERT <sub>BASE</sub>	0.60	0.17	0.26	0.60	0.48	0.53	0.60	0.31	0.41	0.92	0.99	0.96	0.90
LSTM+RoBERTa <sub>BASE</sub>	0.41	0.27	0.33	0.84	0.07	0.12	0.45	0.16	0.24	0.91	0.99	0.95	0.89
LSTM+RoBERTa <sub>LARGE</sub>	0.77	0.75	0.76	0.86	0.91	0.88	0.81	0.83	0.82	0.99	0.99	0.99	0.97
LSTM+ DistilBERT <sub>BASE</sub>	0.42	0.04	0.07	0.62	0.07	0.12	0.53	0.05	0.09	0.87	0.99	0.94	0.87
GRU+ BERT <sub>BASE</sub>	0.43	0.24	0.30	0.62	0.16	0.25	0.49	0.19	0.27	0.91	0.99	0.95	0.89
GRU+ BERT <sub>LARGE</sub>	0.46	0.17	0.25	0.75	0.19	0.31	0.58	0.18	0.27	0.90	0.99	0.95	0.90
GRU+ XLNet <sub>BASE</sub>	0.46	0.33	0.39	0.80	0.16	0.26	0.53	0.24	0.33	0.92	0.99	0.96	0.90
GRU+ XLNet <sub>LARGE</sub>	0.42	0.42	0.42	0.92	0.07	0.14	0.45	0.21	0.31	0.93	0.99	0.96	0.90
GRU+AlBERT <sub>BASE</sub>	0.57	0.34	0.42	0.72	0.54	0.62	0.66	0.43	0.51	0.94	0.99	0.97	0.91
GRU+AlBERT <sub>LARGE</sub>	0.48	0.03	0.05	0.59	0.30	0.40	0.58	0.16	0.25	0.89	0.99	0.94	0.89
GRU+RoBERTa <sub>BASE</sub>	0.48	0.36	0.41	0.75	0.16	0.27	0.54	0.25	0.35	0.92	0.99	0.96	0.89
GRU+RoBERTa <sub>LARGE</sub>	0.76	0.76	0.76	0.87	0.91	0.88	0.82	0.84	0.83	0.99	0.99	0.99	0.95
GRU+DistilBERT <sub>BASE</sub>	0.41	0.19	0.25	0.66	0.10	0.17	0.47	0.14	0.21	0.90	0.99	0.95	0.88

Table 8. Test experimental results (TM reviews).

Model	Comma			Period			Question			Others			Testing Acc
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
LSTM+BERT <sub>BASE</sub>	0.36	0.07	0.12	0.59	0.54	0.56	0.57	0.44	0.59	0.95	0.97	0.96	0.92
LSTM+BERT <sub>LARGE</sub>	0.36	0.07	0.12	0.59	0.54	0.56	0.57	0.44	0.59	0.95	0.97	0.96	0.93
LSTM+XLNet <sub>BASE</sub>	0.34	0.15	0.21	0.62	0.63	0.63	0.60	0.53	0.56	0.95	0.97	0.96	0.93
LSTM+XLNet <sub>LARGE</sub>	0.34	0.15	0.21	0.50	0.62	0.63	0.57	0.55	0.56	0.95	0.94	0.96	0.92
LSTM+AlBERT <sub>BASE</sub>	0.04	0.12	0.06	0.45	0.42	0.43	0.29	0.35	0.32	0.95	0.93	0.94	0.88
LSTM+RoBERTa <sub>BASE</sub>	0.21	0.14	0.17	0.65	0.48	0.55	0.57	0.40	0.47	0.95	0.97	0.96	0.93
LSTM+RoBERTa <sub>LARGE</sub>	0.13	0.40	0.20	0.46	0.80	0.59	0.35	0.70	0.46	0.99	0.90	0.94	0.91
LSTM+DistilBERT <sub>BASE</sub>	0.29	0.04	0.07	0.62	0.41	0.49	0.60	0.33	0.43	0.94	0.98	0.96	0.93
GRU+ BERT <sub>BASE</sub>	0.27	0.16	0.20	0.60	0.55	0.58	0.56	0.46	0.51	0.95	0.97	0.96	0.93
GRU+ BERT <sub>LARGE</sub>	0.27	0.12	0.17	0.61	0.57	0.59	0.58	0.48	0.52	0.95	0.97	0.96	0.93
GRU+ XLNet <sub>BASE</sub>	0.39	0.13	0.20	0.63	0.65	0.64	0.62	0.54	0.57	0.96	0.97	0.96	0.93
GRU+ XLNet <sub>LARGE</sub>	0.34	0.15	0.21	0.50	0.62	0.63	0.57	0.55	0.56	0.95	0.94	0.96	0.92
GRU+AlBERT <sub>BASE</sub>	0.07	0.12	0.09	0.58	0.33	0.42	0.37	0.28	0.32	0.94	0.96	0.95	0.90
GRU+AlBERT <sub>LARGE</sub>	0.08	0.02	0.03	0.46	0.11	0.18	0.40	0.09	0.15	0.92	0.99	0.95	0.91
GRU+RoBERTa <sub>BASE</sub>	0.28	0.17	0.21	0.63	0.57	0.60	0.60	0.48	0.52	0.95	0.97	0.96	0.93
GRU+RoBERTa <sub>LARGE</sub>	0.15	0.42	0.20	0.46	0.80	0.59	0.45	0.70	0.56	0.99	0.95	0.95	0.90
GRU+DistilBERT <sub>BASE</sub>	0.27	0.11	0.16	0.61	0.47	0.53	0.57	0.39	0.47	0.95	0.97	0.96	0.93

Table 9. Test evaluations based on models.

Evulation Criteria	Comma	Period	Question	Others
Precision	LSTM+RoBERTa <sub>LARGE</sub>	GRU+RoBERTa <sub>LARGE</sub>	GRU+RoBERTa <sub>LARGE</sub>	LSTM+RoBERTa <sub>LARGE</sub>
Recall	LSTM+RoBERTa <sub>LARGE</sub>	LSTM+RoBERTa <sub>LARGE</sub>	GRU+RoBERTa <sub>LARGE</sub>	LSTM+RoBERTa <sub>LARGE</sub>
F1	LSTM+RoBERTa <sub>LARGE</sub>	LSTM+RoBERTa <sub>LARGE</sub>	GRU+RoBERTa <sub>LARGE</sub>	LSTM+RoBERTa <sub>LARGE</sub>



**Figure 4.** Proposed model comparison with existing models.

## 5. Conclusions

The main aim of this research is to create a punctuation-restoration model using a transformer model. The transformer models evaluated in this study were BERT-based models, which have been used in many NLP problems. To the best of our knowledge, no research has been conducted to restore punctuation on social media. This study explores and evaluates the performance of different BERT transformer models using LSTM and GRU with a linear neural-network layer to restore punctuation of texts. Among all the models in this study, the LSTM+RoBERTaLARGE model produced the highest scores compared to other models for Amazon and TM review datasets. In the future, the dataset imbalance that resulted in high recall values for some classes and lower values for others (comma, period, question) will be investigated. Smoothing methods or any other method that can achieve a better result could be implemented in the future to mitigate this problem. In addition, the punctuation-restoration model will be integrated as one of the pre-processing tasks in sentiment analysis in order to simplify sentences and extract more insight from the text, in addition to other NLP processes, such as automatic speech recognition (ASR), machine translation, and text-to-speech (TTS) systems. They can also be applied to text-editing tools, such as grammar checkers and digital assistants, like chat bots, for clarity and coherence of generated responses. Also, this method can be applied to other languages, for example, the Malay language, which does not have many NLP resources, as most natural-language resources are currently intended for English. However, multilingual transformer models are available, which can be used for other languages, such as Malay. Therefore, the creation of this model for other languages is feasible.

**Author Contributions:** Conceptualization, A.M.B., K.S.M.A. and S.M.; Methodology, A.M.B. and K.S.M.A.; Validation, A.M.B. and S.K.; Formal analysis, A.M.B. and J.K.; Investigation, J.K.; Resources, S.K.; Writing—original draft, A.M.B. and K.S.M.A.; Writing—review & editing, S.M. and S.K.; Visualization, J.K.; Supervision, S.M.; Project administration, K.S.M.A.; Funding acquisition, K.S.M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the Telekom Malaysia Research and Development Grant (TMR&D)(Grant Number: MMUE/210038 RDTC/211021).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** <https://data.world/datafiniti/consumer-reviews-of-amazon-products> (Datafiniti. Data World: accessed on 10 May 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Curran, T.; Treiber, J.; Rosenblatt, M. Building Brands through Social Listening. In *Proceedings of the Northeast Business & Economics Association*; Northeast Business & Economics Association: Worcester, MA, USA, 2018; pp. 74–78.
- Wiener, M.; Hoßbach, N.; Saunders, C. Omnichannel businesses in the publishing and retailing industries: Synergies and tensions between coexisting online and offline business models. *Decis. Support Syst.* **2018**, *109*, 15–26. [[CrossRef](#)]
- Rahat, A.M.; Kahir, A.; Masum, A.K.M. Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset. In *Proceedings of the 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, Moradabad, India, 22–23 November 2019; IEEE: Piscataway, NJ, USA, 2020; pp. 266–270. [[CrossRef](#)]
- Anbananthen, K.S.M.; Krishnan, J.K.; Sayeed, M.S.; Muniapan, P. Comparison of stochastic and rule-based POS tagging on Malay online text. *Am. J. Appl. Sci.* **2017**, *14*, 843–851. [[CrossRef](#)]
- Woldemariam, Y. Sentiment analysis in a cross-media analysis framework; Sentiment analysis in a cross-media analysis framework. In *Proceedings of the 2016 IEEE International Conference on Big Data Analysis (ICBDA)*, Hangzhou, China, 12–14 March 2016. [[CrossRef](#)]
- Dey, L.; Haque, S.K.M. Opinion mining from noisy text data. In *Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data*, Singapore, 24 July 2008; pp. 83–90. [[CrossRef](#)]
- Yuliah, S.; Bandung, P.N.; Purnamasari, Y.; Bandung, P.N.; Yunita, E.; Bandung, P.N. Grammatical Errors in Social Media Caption. *Int. J. Lang. Lit.* **2020**, *8*, 17–20. [[CrossRef](#)]
- Lu, W.; Ng, H.T. Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the EMNLP 2010—Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, USA, 9–11 October 2010; pp. 177–186.
- Kim, S. Deep Recurrent Neural Networks with Layer-wise Multi-head Attention for Punctuation Restoration. In *Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 12–17 May 2019; pp. 7280–7284. [[CrossRef](#)]
- Tomanek, K.; Wermter, J.; Hahn, U. Sentence and Token Splitting Based on Conditional Random Fields. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, Melbourne, Australia, 19–21 September 2007; p. 57. Available online: <https://pdfs.semanticscholar.org/5651/b25a78ac8fd5dd65f9c877c67897f58cf817.pdf> (accessed on 20 June 2022).
- Manning, C.D.; Bauer, J.; Finkel, J.; Bethard, S.J. The Stanford CoreNLP Natural Language Processing Toolkit. *Aclweb.Org.* **2014**, pp. 55–60. Available online: <http://aclanthology.org/P14-5010/> (accessed on 20 June 2022).
- Loper, E.; Bird, S. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, Sydney, Australia, 17–18 July 2006. [[CrossRef](#)]
- Alam, T.; Khan, A.; Alam, F. Punctuation Restoration using Transformer Models for High-and Low-Resource Languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, Online, 19 November 2020; pp. 132–142. [[CrossRef](#)]
- Che, X.; Wang, C.; Yang, H.; Meinel, C. Punctuation prediction for unsegmented transcript based on Word Vector. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Marseille, France, 11–16 May 2016; pp. 654–658.
- Nguyen, B.; Nguyen, V.B.H.; Nguyen, H.; Phuong, P.N.; Nguyen, T.L.; Do, Q.T.; Mai, L.C. Fast and Accurate Capitalization and Punctuation for Automatic Speech Recognition Using Transformer and Chunk Merging. In *Proceedings of the 2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, Cebu, Philippines, 25–27 October 2019. [[CrossRef](#)]
- Päis, V.; Tufiş, D. Capitalization and Punctuation Restoration: A Survey. Available online: [http://mi.eng.cam.ac.uk/research/projects/EARS/ears\\_summary.html](http://mi.eng.cam.ac.uk/research/projects/EARS/ears_summary.html) (accessed on 9 September 2022).
- Jansche, M.; Bacchiani, M. *Restoring Punctuation and Capitalization in Transcribed Speech*; Department of Computer Science, Columbia University: New York, NY, USA, 2009; pp. 4741–4744.
- Miranda, J.; Neto, J.P.; Black, A.W. Improved punctuation recovery through combination of multiple speech streams. In *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, 8–12 December 2013; pp. 132–137. [[CrossRef](#)]
- Lui, M.; Wang, L. Recovering Casing and Punctuation using Conditional Random Fields. In *Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013)*, Brisbane, Australia, 4–6 December 2013; pp. 137–141. Available online: <http://www.aclweb.org/anthology/U13-1020> (accessed on 19 July 2022).

20. Tilk, O.; Alumäe, T. LSTM for punctuation restoration in speech transcripts. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association—INTERSPEECH, Dresden, Germany, 6–10 September 2015; pp. 683–687. [CrossRef]
21. Tilk, O.; Alumäe, T. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In Proceedings of the Annual Conference of the International Speech Communication Association—INTERSPEECH, San Francisco, CA, USA, 8–12 September 2016; pp. 3047–3051. [CrossRef]
22. Tündik, M.A.; Szaszák, G. Joint Word- and Character-level Embedding CNN-RNN Models for Punctuation Restoration. In Proceedings of the 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, Hungary, 22–24 August 2018; IEEE: Piscataway, NJ, USA, 2019; pp. 135–140. [CrossRef]
23. Zelasko, P.; Szymanski, P.; Mizgajski, J.; Szymczak, A.; Carmiel, Y.; Dehak, N. Punctuation prediction model for conversational speech. In Proceedings of the Annual Conference of the International Speech Communication Association—INTERSPEECH, Hyderabad, India, 2–6 September 2018; pp. 2633–2637. [CrossRef]
24. Ramprasath, M.; Dhanasekaran, K.; Karthick, T.; Velumani, R.; Sudhakaran, P. An Extensive Study on Pretrained Models for Natural Language Processing Based on Transformers; An Extensive Study on Pretrained Models for Natural Language Processing Based on Transformers. In Proceedings of the 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 16–18 March 2022. [CrossRef]
25. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
26. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
27. Datafiniti. Consumer Reviews Amazon Product—Dataset by Datafiniti. Data World. April 2019. Available online: <https://data.world/datafiniti/consumer-reviews-of-amazon-productsproducts> (accessed on 10 May 2022).
28. Kalaiarasi, S.M.A.; Surendran, S.; Jaya Kumar, K. The Generation of Malay Lexicon. *Am. J. Appl. Sci.* **2017**, *14*, 503–510.
29. Vaswani, A. Attention Is All You Need. no. Nips. 2017. Available online: <https://arxiv.org/abs/1706.03762> (accessed on 20 June 2021).
30. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive language models beyond a fixed-length context. In Proceedings of the ACL 2019—57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 2978–2988. [CrossRef]
31. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108v4.
32. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.