

Article

DPG-LSTM: An Enhanced LSTM Framework for Sentiment Analysis in Social Media Text Based on Dependency Parsing and GCN

Zeyu Yin ¹ , Jinsong Shao ¹, Muhammad Jawad Hussain ², Yajie Hao ¹, Yu Chen ¹, Xuefeng Zhang ^{1,*} and Li Wang ^{1,2,*}

¹ School of Information Science and Technology, Nantong University, Nantong 226007, China

² Research Center for Intelligence Information Technology, Nantong University, Nantong 226007, China

* Correspondence: xefeng@hotmail.com (X.Z.); wangli@ntu.edu.cn (L.W.)

Abstract: Sentiment analysis based on social media text is found to be essential for multiple applications such as project design, measuring customer satisfaction, and monitoring brand reputation. Deep learning models that automatically learn semantic and syntactic information have recently proved effective in sentiment analysis. Despite earlier studies' good performance, these methods lack syntactic information to guide feature development for contextual semantic linkages in social media text. In this paper, we introduce an enhanced LSTM-based on dependency parsing and a graph convolutional network (DPG-LSTM) for sentiment analysis. Our research aims to investigate the importance of syntactic information in the task of social media emotional processing. To fully utilize the semantic information of social media, we adopt a hybrid attention mechanism that combines dependency parsing to capture semantic contextual information. The hybrid attention mechanism redistributes higher attention scores to words with higher dependencies generated by dependency parsing. To validate the performance of the DPG-LSTM from different perspectives, experiments have been conducted on three tweet sentiment classification datasets, sentiment140, airline reviews, and self-driving car reviews with 1,604,510 tweets. The experimental results show that the proposed DPG-LSTM model outperforms the state-of-the-art model by 2.1% recall scores, 1.4% precision scores, and 1.8% F1 scores on sentiment140.

Keywords: natural language processing; sentiment analysis; social media text; deep learning; attention mechanism



Citation: Yin, Z.; Shao, J.; Hussain, M.J.; Hao, Y.; Chen, Y.; Zhang, X.; Wang, L. DPG-LSTM: An Enhanced LSTM Framework for Sentiment Analysis in Social Media Text Based on Dependency Parsing and GCN. *Appl. Sci.* **2023**, *13*, 354. <https://doi.org/10.3390/app13010354>

Academic Editors: Huan Wang and Wen Zhang

Received: 2 December 2022

Revised: 23 December 2022

Accepted: 23 December 2022

Published: 27 December 2022



Copyright: © 2022 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

With the rapid growth of the information era, digital users now devote an average of 150 min a day to social networks. Social media user numbers have continued to grow by more than 4 billion over recent years [1]. Therefore, relevant businesses utilize social media to stay in touch with customers and market their products [2,3]. Clients also browse other people's reviews to learn about the quality of new services or products that they might be interested in. The massive number of comments generated by social media has caught the interest of corporations, governments, and organizations that are interested in better understanding public opinion on various issues and user behavior for concrete objectives [4–6]. The purpose of sentiment analysis is to extract subjective information from natural language texts to create structured and digital information for further induction and reasoning. Hence, sentiment analysis is critical to social media data analysis.

Deep learning models have particularly achieved remarkable success in various natural language processing (NLP) tasks, including text classification and sentiment analysis [7]. Zhao et al. introduced a separate framework for the domain of sentiment analysis with weighting rules based on rhetorical structure theory [8]. The authors parsed the text into a rhetorical structure tree and, then, used two generic lexicons to calculate the sentiment

scores of the sentences. Finally, they determined the text polarity by summing the scores of the sentences according to the weighting rules. Chen et al. used a convolutional neural network (CNN) to improve the performance of sentiment analysis [9]. They first applied a sequence-based neural network model to obtain word embeddings. The sentences were then classified into three types, based on the number of targets contained in the sentences (non-target, single-target, or multi-target sentences), where each kind of sentence was fed into the model separately.

Social media is mainly in the form of short texts, and the actual sentiment of short texts is difficult to predict. The performance of current methods for sentiment analysis relies heavily on the quality of features extracted from the texts [10]. The features of the original sequence data are extended by merging the syntactic information in the sequence (i.e., syntactic analysis trees, such as dependency trees and component analysis trees). To reduce the distance between words, and words from the perspective of graphically structured data, the syntactic analysis trees can encode complex pairwise relationships between them. Thus, the dependency information of sequences is effectively preserved for learning more informative features [11]. The mainstream sentiment analysis methods use semantic analysis based on sequence structure, which lacks syntactic information in the sequences. In addition, previous studies on social media sentiment analysis used traditional neural networks to extract features. These schemes have algorithmic limitations, in that they cannot extract both semantic and syntactic structural information of a sequence.

To solve the above problems, we proposed an enhanced long short-term memory neural network (LSTM) using dependency parsing (DP) and a graph convolution network (GCN), referred to as dependency parsing graph long short-term memory (DPG-LSTM).

Dependency parsing (DP) refers to a sophisticated NLP task that analyzes the dependency grammar of a sentence, with words and lexemes as input and DTs as output [12]. DP theory assumes that the dominant-subordinate relationship between words is a binary inequality relationship. The modifier is considered dependent when a word affects another word in a phrase; the altered word is called the head, and the grammatical relationship between the two is called the dependency relation.

All word dependencies in a sentence are represented as directed edges in a DT. The DT of the sentence “The Wall Street Journal just published an interesting piece on cryptocurrencies.” is shown in Figure 1. The direction of the arrow points from the dominant word to the subordinate word (the convention when visualizing). Visit the website <https://universaldependencies.org/docsv1> (accessed on 22 December 2022) to understand more about the meaning of dependencies.

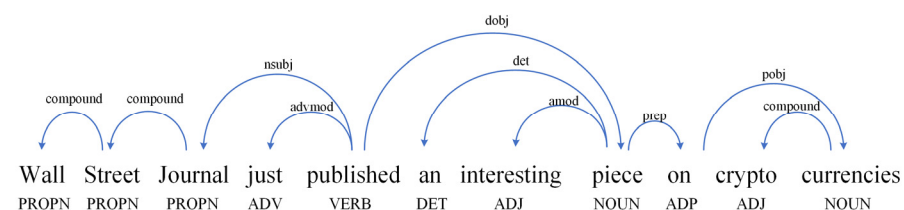


Figure 1. An example of DT.

A DT is a subgraph of a full graph, much as a tree is a particular instance of a graph (a graph composed of all word relations). An algorithm, such as Prim’s algorithm, may be used to determine the maximum spanning tree as a DT if the possibility of each edge in the complete graph belonging to the syntactic tree is scored [13].

Our study considers the syntactic structural information in a text with the help of dependency trees (DTs). To exploit the use of syntactic information in sequences to expand the text features, LSTMs are adopted to capture the semantic information and GCN for the syntactic information [14]. LSTM is the most suitable neural network for capturing word dependencies in text, while a GCN uses parallelism to extract local relevance from the text–space structure. DPG-LSTM gives a fresh insight into the syntactic structure of

sentences from a natural language perspective, allowing more interactions between text elements to be captured than LSTM and GCN alone. Furthermore, because the constructed dependency graph is an arbitrary graph, with numerous relationships that conventional GCN approaches cannot directly exploit, we propose an attention mechanism that combines DTs to extract semantic and syntactic information.

In general, DPG-LSTM can be seen as the intersection between sentiment analysis techniques and social media. The goal of our work is to take advantage of the efforts of DP and DL for analyzing and extracting user sentiment from social media text. Our research can aid in improving the semantic comprehension of social media text. In addition, the attention mechanism we propose can highlight the syntactic information from the contextual information by dependency parsing. The combination of GCN and attention mechanism can further improve classification accuracy. The contributions of this paper are summarized as follows:

- (1) Instead of using traditional semantic analysis methods to classify texts directly, we propose a sentiment analysis method that combines syntactic information.
- (2) We designed an attention mechanism based on dependency parsing, which extracts semantic and syntactic information from texts.
- (3) The best results are obtained on multiple datasets compared to the current advanced models.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the background technical details. Section 4 describes our proposed sentiment analysis method incorporating information about sentence syntactic structure. Section 5 presents datasets and experimental results. Finally, Section 6 concludes the paper and suggests possible future research directions.

2. Research Objective

In this study, our primary objective is to investigate the implications of semantic and syntactic information integration on sentiment analysis in social media. We utilize dependency parsing theory and sentiment analysis techniques to find answers to the following research questions (RQs):

- (1) Considering the sequential structure, how to capture syntactic connections in social media text?
- (2) How can syntactic information be integrated into traditional sentiment analysis techniques, and can syntactic information improve sentiment analysis?
- (3) Compared with the traditional attention mechanism, how to exploit the interplay of semantic and syntactic information to obtain better sentence features?
- (4) Is text length a factor in syntactic information's role in sentiment analysis?

3. Related Work

The majority of existing sentiment analysis methods are based on the semantic information of social media text. Based on the sequential structure of a social media text, these methods train the features of the text and classify it with deep learning. Due to the specificity of graph neural networks in information extraction, GNNs have recently been used for sentiment analysis, as well. Several researchers have attempted to merge GNNs with dependence parsing [11,15,16].

3.1. Sentiment Analysis Methods Based on Semantic Analysis

For sentiment analysis in social media, combining additional content with semantic information has been a hot issue. For example, Li et al. suggested a neural network including conventional economic methodologies to examine public opinion on social media [17]. Their findings suggest that large-scale public opinion on the internet can impact or forecast changes in commodity prices. Sentiment dictionaries are used in the current mainstream technique to add semantic information. Based on a recurrent neural network

(RNN), Liu et al. proposed a new model by combining lexicons with a bi-directional long short-term memory neural network (BiLSTM) for text classification [18]. To estimate the sentiment polarity, dictionary-based methods generally evaluate the semantic pointing of each sentiment word in the text span.

However, sentiment dictionaries often need manual involvement, resulting in sentiment dictionaries that do not keep up with the ever-changing social media. To overcome these problems, the researchers used knowledge bases to assist in sentiment analysis. Vizarra et al. proposed a knowledge-based methodology for sentiment analysis on social media [19]. Their research analyses the text and applies semantic processing to the comments' implicit knowledge. Agarwal employs knowledge bases to expand the vocabulary set, enhancing the system's performance by reducing the number of unseen terms [20]. His method efficiently translates the semantic representations of unseen words in social media to a domain-specific trained embedding space. The approaches mentioned above only use root domain single-source knowledge graphs to perform sentiment classification in social media, which frequently needs knowledge graphs from other data sources. Clustering the entities in different knowledge graphs is an urgent problem for them to solve. To get a more precise assessment of emotional polarity, researchers have employed fine-grained approaches for sentiment analysis. Sentic-LSTM was suggested by Ma et al. as an extension of the LSTM model for sentiment analysis based on aspects [21]. Their research focuses on integrating tasks, such as object-related aspect detection and aspect-based polarity classification. They also used commonsense knowledge to integrate it into a recurrent encoder for target sentiment analysis [22]. Their model is a hybrid of attention architecture and Sentic-LSTM. Attention mechanisms have been successfully used for a variety of NLP tasks.

Conventional attention mechanisms cannot distinguish target-sensitive sentiment expression. Lin et al. presented the Multi-Head Self-Attention Transformation (MSAT) network, which uses target-specific self-attention and dynamic target representation to perform more effective sentiment analysis [23]. Even though researchers have made great progress in fine-grained sentiment analysis, the current problem of a lack of huge data that is appropriate for fine-grained sentiment analysis remains, making it challenging to translate experimental results into widespread practical applications.

Considering that aspect-based sentiment analysis research needs specialized datasets, current sentiment analysis approaches seek to pre-train models using a combination of large data. Wang et al. proposed a deep learning approach based on pre-trained language representation models. They combined the results of robustly optimized BERT with the generalized autoregressive language model XLNET for language understanding to obtain better results for sentiment analysis [24–26]. Imran et al. took the embeddings of the pre-training model fastText, which trains on big data of tweets, to analyze the sentiment of people in different countries under the new crown epidemic [27,28]. AlBadani et al. suggest a novel effective technique of sentiment analysis based on deep learning architectures that combine fine-tuning the universal language model and SVM to improve sentiment analysis [29]. The method provides a novel, deep learning approach to Twitter sentiment analysis that uses reviews to discover people's opinions about specific items.

3.2. Sentiment Analysis Methods Based on Graph

With the introduction of graph neural networks, new types of external information are introduced for sentiment analysis, reinforcing the sentiment features of texts. To capture corpus-level word co-occurrence information, Zhang and Qian created a global lexical graph, as well as a conceptual hierarchy based on syntactic and lexical graphs [30]. Ghosal et al. utilized seed concepts collected from text to create a subgraph with ConceptNet [31,32]. Chen et al. created a graph with edges connecting perceptual nodes to the corresponding aspect nodes for capturing intra-aspect consistency [33]. Moreover, they built a graph with edges connecting sentence nodes within the same document for capturing inter-aspect trends to capture document-level sentiment preference information.

These techniques update the graph's node features from a variety of angles, but they all have the same flaw: they only consider nearby nodes, while disregarding global features. To better incorporate the global properties of the nodes, Liao et al. use varying widths of node connection windows at different levels [34]. Furthermore, academics have attempted to model user connections using graph neural networks to improve the performance of sentiment analysis [35].

3.3. Sentiment Analysis Methods Based on Dependency Parsing

Dependency parsing was first applied to aspect-based sentiment analysis studies. Since CNN lacks a technique to account for important syntactical restrictions and long-range word dependencies, Zhang et al. advocated building a GCN over the DT of a phrase to leverage the syntactical information and long-range word dependencies [11]. Sun et al. describe a convolution over a dependency tree method, which incorporates a Bi-LSTM to train representations for sentence features, and then uses a GCN to improve the embeddings by operating directly on the DT of the sentence [15]. However, the above approaches are limited to the syntactic structure in the field of fine-grained text analysis, ignoring the sequence information in the original sentences. With the application of dependency parsing in sentiment analysis, some studies have started to use it in non-fine-grained text research. For example, Xiang et al. used dependency parsing in the sentiment analysis of financial texts on social media [36]. Furthermore, Wu et al. proposed an attention-dependent mechanism to combine contextual, lexical, and syntactic cues [16]. However, there is a lack of research that takes a unique attention mechanism to analyze the dependency parsing in social media in non-fine-grained text research. With this motivation, we propose DPG-LSTM and DPG-Attention to fill the current research gap.

4. Materials and Methods

4.1. Deep Learning Architecture

4.1.1. Sequential Neural Network Architecture

The essence of RNN is to make full use of the sequential information in the text. In traditional neural networks, all inputs and outputs are supposed to be directly independent of one another. For many projects, however, there are still points for improvement. It is critical to comprehend the words that precede the following word in a phrase before anticipating the next word. A typical RNN network unfolds at any t moment, as shown in Figure 2.

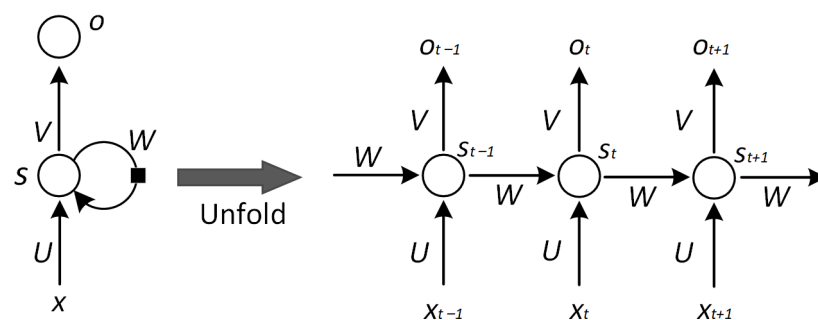


Figure 2. One Cell of RNN.

In Figure 2, x_t is the input of the input layer; s_t is the output of the hidden layer; s_0 is required to compute the first hidden layer, which is usually initialized to all zeros; and o_t is the output of the output layer. The weight matrices of the input, hidden, and output layers are U , V , and W .

LSTM is considered a special case of RNN that passes only the essential parts of the data to the next layer instead of the entire data. Gradient descent is a technique for minimizing errors in neural networks by adjusting the weights of each neuron. Backpropagation usually causes the gradient of the loss function to diminish exponentially in consecutive

stages in RNNs, which is known as the gradient vanishing problem. The performance of an RNN tends to decline as the distance between two related words grows. In the case of long-term dependencies, LSTM solves this problem and performs well. A BiLSTM is an extension of LSTM. The key idea behind the BiLSTM model is to use the future and past contexts, so it processes the sequences in two opposite orders.

4.1.2. Graph Neural Network Architecture

Social media text can be expressed in graphical structures instead of sequential structures. An illustration of the general design pipeline is shown in Figure 3. The typical schema of the GNN is shown in the middle of Figure 3, where convolution operators, recurrent operators, sampling operators, and skip connections are used to propagate information across each layer. Finally, advanced features are extracted using pooling operators.

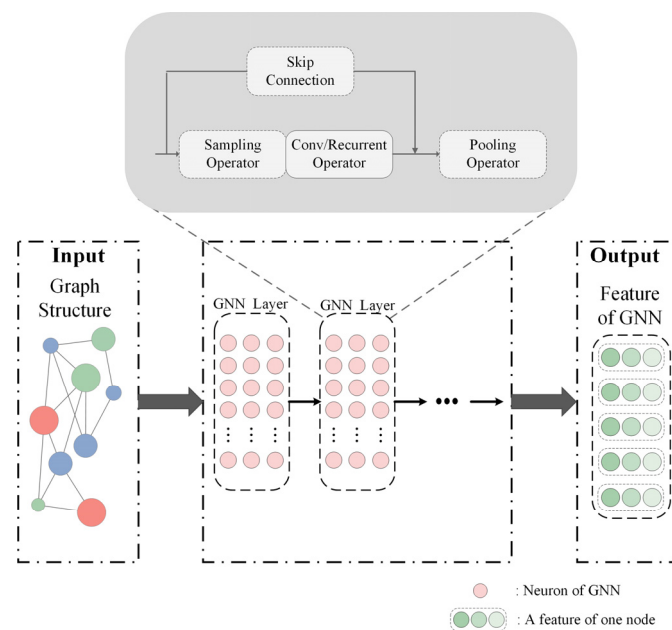


Figure 3. Pipeline of GNN.

A GNN obtains greater quality features by stacking neural network layers. A GCN is a traditional GNN model that is frequently used to process data with a graph structure directly. A GCN is similar to a CNN in that nodes and their neighboring nodes are aggregated by an aggregation function to obtain their feature information. However, there is no fixed sequence structure for graphs, so the GCN takes a more complex convolution kernel (i.e., first-order local approximation of spectral graph convolution).

4.2. The Proposed Method DPG-LSTM of Sentiment Analysis

The proposed DPG-LSTM combining semantic and syntactic information is designed to improve the performance of sentiment analysis. The DPG-LSTM is divided into five modules: I. Node Feature Extraction, II. Dependency Graph Construction, III. Semantic-Syntactic Dual-Channel Network, IV. Dependency Parsing Graph Attention Mechanism (DPG-Attention), and V. Feature Classification. The DPG-LSTM is shown in Figure 4.

First, to obtain accurate semantic features from social media text, the data are cleaned and extracted from the corresponding embedding in the text by the pre-training model. Second, for each social media text, a dependency graph is created, based on the syntactic structure of the sentence, with each edge representing a syntactic relationship between nodes. Third, the dependency graphs and the sequences of texts are fed into the semantic-syntactic dual-channel network, which is combined with DPG-Attention to extract the text's semantic and syntactic information. Finally, the retrieved social media text features are input into the pooling layer, and the polarity is determined using the fully connected layer.

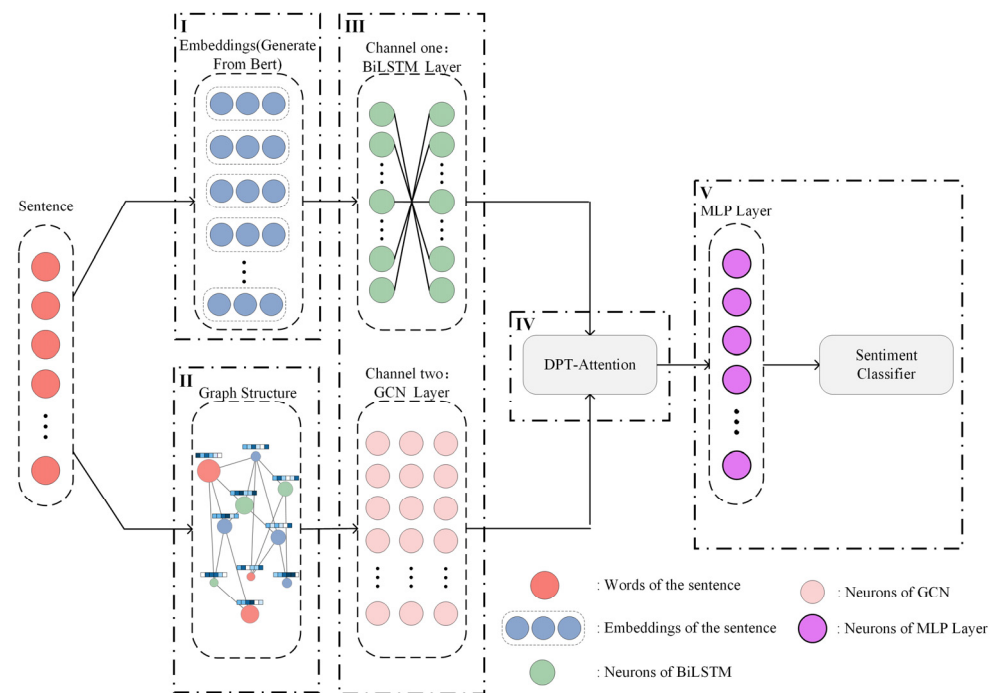


Figure 4. Framework diagram of DPG-LSTM.

4.2.1. Node Feature Extraction

To generate proper sentiment classification results, a pre-trained model is required to obtain the initial embeddings of the words. The current pre-trained models are more helpful for application tasks. Pre-trained models provide better initialization embeddings, which usually lead to better generalization performance and accelerated convergence to the target loss function [37]. In addition, the pre-trained model is a kind of regularization to avoid the over-fitting of small data. BERT, a pre-training result obtained after Google trained a large number of tweets, is used to learn the features of words as word embeddings. Given a short textbook sentence, $s = \{s_1, s_2, \dots, s_n\}$, consisting of n ordered words, the pre-trained word embeddings, $e = \{e_1, e_2, \dots, e_n\}$, are obtained by BERT for sentence s . All the texts are aggregated and marked to generate 768-dimensional embeddings for each word by BERT.

4.2.2. Dependency Graph Construction

A DT is represented as a graph, G , with n nodes (i.e., dependency graph), with the nodes representing words in sentences and the edges representing the syntactic dependency paths between them. G is generated by the Stanford Parser, which is most commonly used in DP [38]. G is shown in Formula (1):

$$G = (V, E) \quad (1)$$

where V denotes the words that make up the text, and E denotes the edge between words.

The node features of G are given by the real-valued embedding obtained by BERT, as described in Section 4.1. The structure of G allows the GCN to operate directly on the graph to model the dependencies between words [14]. G is allowed to have self-loops for the GCN to model the node embeddings. The GCN ensures that the syntax represented by G is efficiently encoded, allowing the node embeddings to be encoded based on the local position of words in dependencies concerning one another.

The dependencies between words are represented as an $n \times n$ adjacency matrix A . A_{ij} signals if node i is connected to node j through a single dependency path in G . Specifically, $A_{ij} = 1$ if node i is connected to node j , otherwise, $A_{ij} = 0$. In combination with the node embeddings generated by BERT, the GCN that operates directly on the graph is adopted.

4.2.3. Semantic–Syntactic Dual-Channel Network

Because its neural units memorize values at arbitrary time intervals, a BiLSTM precisely captures the connections between contextual feature words. Moreover, the BiLSTM utilizes the forgotten gate structure to filter useless information, which is beneficial to improving the total capture capability of the classifier. To improve the quality of the extracted text features, a GCN uses syntactic information in sequences. The GCN uses dependent paths to transform and propagate information between the paths and aggregates the propagated information to update the node embeddings.

Our dual-channel network mainly consists of BiLSTM and GCN networks. In channel one, pre-trained word embeddings are fed into the BiLSTM to enhance the semantic features of the BiLSTM. In channel two, the pre-trained word embeddings are treated as node features of G into the GCN to strengthen the syntactic features of the text.

The BiLSTM integrates contextual information into word embeddings by tracking dependencies in word chains. The hidden-state representation of the LSTM is $\vec{h}' = \{\vec{h}'_1, \vec{h}'_2, \dots, \vec{h}'_n\}$ in the positive direction of e . This allows the LSTM to capture contextual information in the forward direction. Similarly, the learning representation of the backward LSTM is $\overleftarrow{h}' = \{\overleftarrow{h}'_1, \overleftarrow{h}'_2, \dots, \overleftarrow{h}'_n\}$. The corresponding parallel representations of the forward and backward LSTM are connected to model the higher dimensional representation, $h' = \{h'_1, h'_2, \dots, h'_n\}$. Using the GCN that acts directly on the DT, the dependency information is integrated into the pre-trained embedding in channel two. The hidden representation of the i node of the GCN at the layer $k + 1$ is shown in Formula (2):

$$h_i^{(k+1)} = \phi\left(\sum_{j=1}^n c^i A_{ij} \left(W^{(k)} h_j^{(k)} + b^{(k)}\right)\right) \quad (2)$$

where $h_j^{(k)}$ is the hidden-state representation of the GCN layer k node j ; $b^{(k)}$ is the deviation term; $W^{(k)}$ is the parameter matrix; c^i is the normalization constant; and $c^i = 1/d^i$. d^i represents the order of node i in the diagram, calculated as $d^i = \sum_{j=1}^n A_{ij}$. Note that h_i^0 represents the initial embedding generated by BERT.

4.2.4. Dependency Parsing Graph Attention Mechanism

Generally, given a particular word, different words within the text may have different influences. Extracting text features with the help of different influences is the role of the attention mechanism. A DPG is employed to supplement the syntactic information that the BiLSTM lacks, to extract all of the text's hidden features. Combining the existing attention mechanisms, we propose a dependency parsing graph attention mechanism that simultaneously enhances the semantic and syntactic information.

To obtain the hybrid features, the syntactic features obtained from the last layer of the GCN are spliced with the semantic features obtained from the BiLSTM. Afterward, the self-attention mechanism is used to obtain the self-attention values of the hybrid features. In the next part, the self-attention matrix of the hybrid features is summed with matrix A to obtain the dependency graph attention matrix. Finally, the new dependency graph attention matrix and the hybrid features are passed through the Hadamard product to obtain the modified dependency graph attention features. The schematic diagram of DPG-Attention is shown in Figure 5.

Let the fusion feature $h^\sim = \{h_1^\sim, h_2^\sim, \dots, h_3^\sim\}$, where $h_i^\sim$ is shown in Formula (3):

$$h_i^\sim = [h'_i \parallel h_i^{(k+1)}] \quad (3)$$

In Equation (3), h'_i is the hidden-state representation of the BiLSTM; $h_i^{(k+1)}$ is the hidden-state representation of the $(k + 1)$ layer of GCN network; and the meaning of the formula is to splice h'_i with $h_i^{(k+1)}$ back and forth to get the fused feature $h_i^\sim$.

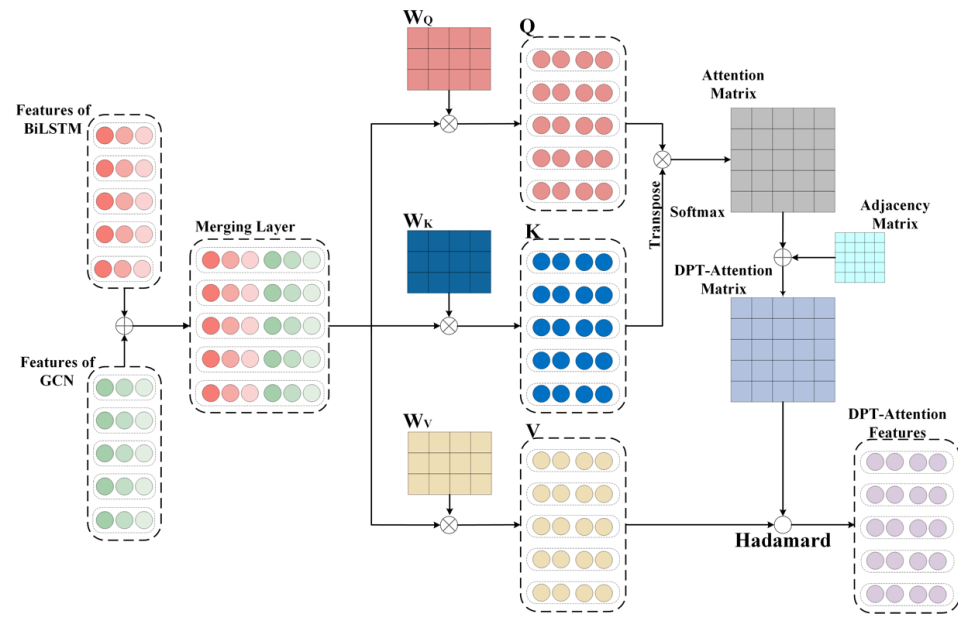


Figure 5. The schematic diagram of DPG-Attention.

The fused features are multiplied with the matrices W_Q , W_K , W_V to obtain the features Q , K , and V :

$$Q = h^{\sim} \times W_Q \quad (4)$$

$$K = h^{\sim} \times W_K \quad (5)$$

$$V = h^{\sim} \times W_V \quad (6)$$

The features Q are multiplied with the transcripts of the features K , and then the normalized results are added to the adjacency matrix A to obtain a new dependency graph attention score matrix, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]$. a and α_i are shown in Formulas (7) and (8).

$$a = \frac{(QK^T)}{\sqrt{d_k}} + A \quad (7)$$

$$\alpha_i = \frac{\exp(a_i)}{\sum_{a_i \in a_\tau} \exp(a_i)} \quad (8)$$

Multiplying α with the feature V , the final output feature $h^\alpha = [h_1^\alpha, h_2^\alpha, \dots, h_3^\alpha]$ is obtained.

$$h^\alpha = \alpha \times V \quad (9)$$

4.2.5. Feature Classification

The output of the DPG-Attention, h^α , is sorted and pooled in the sort-pooling layer. The sort-pooling layer obtains a tensor P of $1 \times D$ dimensions, where the value of D is the same as the dimensionality of h^α . The MLP layer uses the softmax function to generate the emotional polarity from P .

$$P = \text{GlobalAveragePool}(h^\alpha) \quad (10)$$

$$O_i = \text{Softmax}(P_i) \quad (11)$$

After obtaining the probability distribution, O_i , the binary cross-entropy is used as the loss function to calculate the difference between the actual and predicted sentiment:

$$\text{loss} = - \sum_{i=1}^k R(O_i) \times \log(O_i) \quad (12)$$

where k is the number of categories, and $R(O_i)$ is the actual sentiment associated with the text, which is taken as a discrete value from the set of labels. The loss function is similar to the likelihood function, which seeks to minimize the difference between the probability distribution in the training set and the probability distribution in the test data set predicted by DPG-LSTM.

4.3. Datasets

In this paper, the real-world short-text social media datasets sentiment140, airline reviews, and self-driving car reviews are tested in our experiments [39–41]. Even though all three datasets are social media text, they are twitters from distinct domains. Tweets are frequently expressed in different grammatical formats across domains. Table 1 summarizes the statistics of these datasets. The following are the datasets and their descriptions:

Table 1. Details of the datasets.

Dataset	Pos		Neu		Neg		Average Length
	Train	Test	Train	Test	Train	Test	
Sentiment140	720,000	80,000	0	0	720,000	80,000	12.8
Airline reviews	2000	200	2000	200	2000	200	16.7
Self-driving car reviews	700	70	700	70	700	70	18.5

Sentiment140: This is an NLP dataset for detecting Twitter sentiment, which originated from Stanford University. This dataset includes 1,600,000 annotated tweets.

Airline reviews: CrowdFlower’s data for everyone library provided the source for this information. It includes assessments of major American airlines and contains whether the sentiment of tweets in this set is positive, neutral, or negative toward U.S. airlines. The samples in this dataset were balanced so that the number of samples in the positive, negative, and neutral categories was equal or nearly equal.

Self-driving car reviews: This data was compiled from user reviews of self-driving cars on Twitter. It has three characteristics: A Twitter ID, a review text, and the polarity of the opinion. On a scale of 1 to 5, consumers’ sentiments about self-driving cars are represented in this dataset. We set 1 and 2 as negative classes, 3 as a neutral class, and 4 and 5 as positive classes.

4.4. Social Media Pre-Processing

Before training the model, we processed the content of the social media texts. Cleaning social media texts is required to obtain accurate sentiment classification results. We converted tweets to lowercase, removed user mentions, URLs, HTML tags, stop words (except “n’t”, “not”, and “no”), and any non-ASCII characters (including emoticons). We padded each input tweet to ensure a uniform size, which is the standard tweet maximum size.

4.5. Experiment Settings

The experiment was conducted on an RTX 3090 server, with Ubuntu18.04 and Python 3.6. We used BERT as the word embedding training tool, Stanford Core NLP as the dependency parsing toolkit, NLTK as the text preprocessing toolkit, and Tensorflow 1.15 as the deep learning toolkit.

The dimensionality of BERT was set to 768. The DPG-LSTM was trained with 20 epochs, a batch size of 256, a dropout of 0.2, and a text length of 50. We used the Adam optimizer with a learning rate of 10^{-5} .

4.6. Baseline Models

To show the effectiveness of our proposed model (DPG-LSTM), we compared three datasets using traditional sentiment analysis methods as a baseline. Furthermore, we selected sentiment140 for comparison with the results of the most advanced schemes

available. For succinctness, we do not introduce traditional classification methods such as SVM, RF, CNN, etc. Table 2 presents the existing state-of-the-art methods so that we can study the effectiveness of our model in social media texts for sentiment analysis.

Table 2. Introduction of advanced methods.

Method	Description
AC-BiLSTM [42]	An attention-based BiLSTM with convolution layers
FastText-BiLSTM [26]	A BiLSTM with fastText
ABCDM [43]	An attention-based bidirectional CNN-RNN deep model
Stacked DeBERT [44]	An improved BERT for sentiment classification
ACL [45]	An attention-based CNN-LSTM deep model

4.7. Evaluation Metrics

We chose the same assessment metrics as the baseline in the prior research to establish a fair comparison, including P (Precision), R (Recall), and $F1$ (F1-score). These three metrics are defined as:

$$P = \frac{TP}{TP + FP} \quad (13)$$

$$R = \frac{TP}{TP + FN} \quad (14)$$

$$F1 = \frac{2 * P * R}{P + R} \quad (15)$$

The variables in the metrics are defined as follows:

True positive (TP): Comments that were initially categorized as positive and were projected to be positive by the classifier.

False positive (FP): Comments that were initially categorized as positive but were projected by the classifier to be negative.

True negative (TN): Comments that were initially categorized as negative and were also predicted to be negative by the classifier.

False negative (FN): Comments that were categorized as negative but were predicted as positive by the classifier.

5. Results

5.1. Experimental Results

The experimental results of conventional approaches on the sentiment140, self-driving car, and airline data sets are presented in Tables 3–5. Compared with the advanced methods described in Table 2, Table 6 shows a more extensive performance of the proposed method on the sentiment140 dataset.

Table 3. Experimental results on sentiment140 dataset.

Model	R	P	$F1$
SVM	0.782	0.782	0.782
Random Forest	0.765	0.767	0.764
RNN	0.812	0.812	0.812
CNN	0.808	0.806	0.805
TextGCN	0.786	0.786	0.786
BiLSTM	0.824	0.825	0.825
BERT	0.805	0.807	0.806
DPG-LSTM	0.845	0.842	0.843

Table 4. Experimental results on airline review dataset.

Model	<i>R</i>	<i>P</i>	<i>F1</i>
SVM	0.917	0.919	0.917
Random Forest	0.891	0.889	0.881
RNN	0.895	0.900	0.898
CNN	0.887	0.912	0.899
TextGCN	0.853	0.854	0.854
BiLSTM	0.966	0.932	0.949
BERT	0.907	0.891	0.899
DPG-LSTM	0.971	0.940	0.955

Table 5. Experimental results on self-driving car review dataset.

Model	<i>R</i>	<i>P</i>	<i>F1</i>
SVM	0.770	0.782	0.765
Random Forest	0.787	0.785	0.753
RNN	0.802	0.854	0.827
CNN	0.778	0.824	0.800
TextGCN	0.726	0.879	0.795
BiLSTM	0.816	0.801	0.813
BERT	0.796	0.809	0.802
DPG-LSTM	0.812	0.845	0.828

Table 6. Experimental results of advanced methods.

Model	<i>R</i>	<i>P</i>	<i>F1</i>
AC-BiLSTM	0.816	0.823	0.815
FastText-BiLSTM	0.824	0.825	0.825
ABCDM	0.823	0.828	0.820
Stacked DeBERT	0.833	0.800	0.816
ACL	0.840	0.776	0.802
DPG-LSTM	0.845	0.842	0.843

Based on the *R*, *P*, and *F1*, Table 3 gives a comparative analysis of the sentiment140 dataset. It is worth noting that the RNN and BiLSTM models outperform the CNN model since they focus on the text's sequence features. In general, deep learning approaches perform better than regular machine learning methods, as deep learning eliminates the large and challenging feature engineering required for machine learning. Although BERT has made breakthroughs in many NLP tasks, it does not work well on the sentiment140 dataset. Even though pre-trained models have taken center stage in NLP development, task-specific method creation is still crucial. In comparison to previous conventional approaches, the DPG-LSTM suggested in this study achieved better *P*, *R*, and *F1* outcomes for the sentiment140 dataset.

Table 4 shows the *R*, *P*, and *F1* experimental results for the airline review dataset. Like the sentiment140 dataset, DPG-LSTM outperforms all the other classifiers in terms of the *R*, *P*, and *F1* for the airline review dataset. It can be seen that the performance results of DPG-LSTM and the BiLSTM are extremely similar. The DPG-LSTM and BiLSTM yield *F1* values of 0.955 and 0.949, respectively.

The performance results of the self-driving car dataset are presented in Table 5. The DPG-LSTM achieves a better performance in the *F1* values for the self-driving car dataset. Unlike the previous dataset, the BiLSTM algorithm obtained the highest *R* value of 0.816, while the DPG-LSTM had an *R* of 0.812. Meanwhile, the TextGCN is satisfactory for the self-driving car reviews in terms of *P*. Because the average length of text in the self-driving car dataset is longer, the BiLSTM and TextGCN may use their capabilities in the long text.

As shown in Table 6, DPG-LSTM surpassed other advanced approaches in terms of *R*, *P*, and *F1*, reaching 0.845, 0.842, and 0.843. In terms of the *R*, *P*, and *F1* outcomes, the

FastText-BiLSTM surpasses the AC-BiLSTM, which is based on the BiLSTM. It is self-evident that extensive pre-training in certain fields may increase sentiment analysis accuracy significantly. In that it employs an attention mechanism and somewhat enhances the experimental outcomes, the ACL is comparable to the ABCDM. The ACL trails other models in all other metrics, despite having a stronger R value than the ABCDM. Additionally, neither the ACL nor the ABCDM considers the syntactic structural data from the text itself. Stacked DeBERT is a modified version of BERT since, in comparison to other models, the pure BERT framework does not produce good metrics. Therefore, our DPG-LSTM, based on BERT's pre-training results, deservedly received the highest score.

5.2. Ablation Experiments

This section evaluates the improvement points of DPG-LSTM. Table 7 compares the performance of several DPG-LSTM versions on the sentiment140 dataset to demonstrate the efficacy of neural networks that incorporate semantic and syntactic information. Based on the BiLSTM, the BiLSTM-G, BiLSTM-GA, and DPG-LSTM are implemented. The BiLSTM-G combines a GCN based on the BiLSTM, while the BiLSTM-GA further combines a self-attention mechanism based on the BiLSTM-G.

Table 7. Experimental results of ablation experiments.

Model	R	P	$F1$
BiLSTM	0.824	0.825	0.825
BiLSTM-G	0.826	0.825	0.825
BiLSTM-GA	0.832	0.835	0.833
DPG-LSTM	0.845	0.842	0.843

The R , P , and $F1$ of the ablation experiments for the sentiment140 dataset are shown in Table 7. In terms of the P and $F1$ outcomes, the BiLSTM-G is identical to the BiLSTM, and just marginally better in terms of R . Simple feature combinations cannot play the role of text syntactic information. It is vital to have a mechanism that extracts both semantic and syntactic information. Combined with the usual attention mechanism, the BiLSTM-GA achieves better results in R , P , and $F1$. However, existing attention mechanisms ignore information about syntax. After implementing our suggested DPG-Attention, DPG-LSTM had the highest R , P , and $F1$ scores with 0.845, 0.842, and 0.843.

5.3. Impact of Social Media Text Length

The text lengths in various domains of social media are usually different. The BiLSTM-GA, which is presented in Section 5.2, is adopted as a baseline to examine the influence of the DPG-Attention on various text lengths. The sentiment140 test sets are separated into different average text lengths (15, 20, 25, 30, and 35) in our studies.

Figure 6 indicates that, as sentence length increases, both models' performance degrades. This is in line with the fact that lengthier expressions include more words that have an impact on sentiment analysis. Furthermore, when the length increases, the sentence may contain more phrases, posing a bigger barrier for sentiment analysis. In most situations, DPG-LSTM outperforms the BiLSTM-GA, and the improvement is more evident for long sentences than for short sentences. This demonstrates that the DPG-Attention is successful in analyzing the syntax of the sentences and assisting the model in analyzing the sentiment polarity of social media.

5.4. Case Study

The visualization case of the attention mechanism is used to demonstrate the difference between DPG-LSTM and the baseline in this section. Similar to Section 5.3, we employ the BiLSTM-GA, which applies the self-attention mechanism as a baseline. Specially, we chose basic and sophisticated examples for each emotion. The visualization results for numerous

test examples are presented in Table 8 (Darker colors in the samples indicate higher attention scores). In addition, both models' prediction results and labels are displayed.

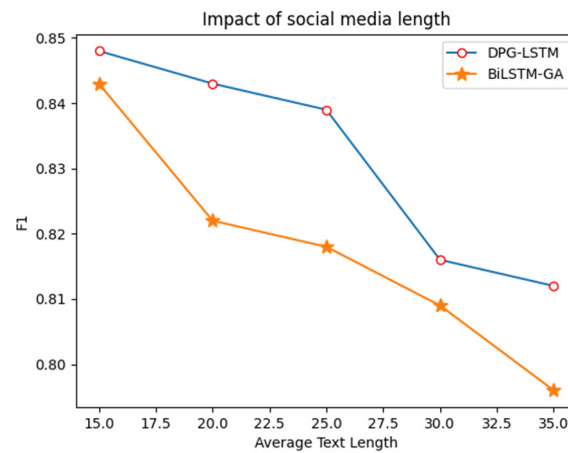


Figure 6. Results at different text lengths.

Table 8. Visualization Case.

Model	Visualization of Sentence	Prediction	Label
BiLSTM-GA	I hate when I have to call and wake people up	negative	negative
	I love write but I sure hate editing and proofing	positive	negative
	What a good night of just hanging out and not really drinking	positive	positive
	My exams are about to end finally, thank god just one exam left, I was so sick of them.	negative	positive
DPG-LSTM	I hate when I have to call and wake people up	negative	negative
	I love write but I sure hate editing and proofing	negative	negative
	What a good night of just hanging out and not really drinking	positive	positive
	My exams are about to end finally, thank god just one exam left, I was so sick of them.	positive	positive

Both models correctly predicted the first and third samples. However, the BiLSTM-GA failed in predicting the second and fourth samples, which have complex sentence structures. Compared to the first and third samples, the other two samples have more complex sentence components, and multiple emotions are present in both sentences. As the sentiment components of sentences become complex, the visual distinction of attentional weights of different models becomes pronounced. In complex sentences, our proposed method focuses on strengthening the attentional weights of words that are critical to the sentence-level sentiment. In contrast, the BiLSTM-GA focuses on all the sentiment words that occur in the sentence. Our proposed approach focuses on weights from a syntactic perspective, thus capturing more accurately the correct effective tendency of the sentence.

6. Discussion

The paper proposes an enhanced LSTM framework (DPG-LSTM) for sentiment analysis in social media text based on DP and GCN. Due to the complexities of data in social media, there is a large diversity of semantic information, as well as specialized syntactic information. For DPG-LSTM, we employ the BiLSTM as the semantic information analysis module. Although LSTM overcomes the problem of RNN gradient explosion and captures semantic dependencies over greater distances, LSTM can better capture the proximity semantics in social media from the reverse. Because the syntactic relations derived via DP are essentially non-Euclidean data structures, conventional neural networks based on Euclidean data structures are unable to parse the syntactic information adequately.

Such non-Euclidean data structures are properly represented by graphs, and the syntactic properties of the nodes are well constructed using the GCN, which can analyze graphs directly. The current mainstream attention mechanism focuses on exploring the effect of each word in a sentence on the sentence's semantic level, assigning attention weights to each word, and capturing the relevant components of the sentence semantics. Our suggested DPG-Attention, which is based on DP theory, captures not only the significant components of phrase semantics but also those of sentence syntax. The combination of these methods makes it possible to improve the classification ability of DPG-LSTM. Experiments have shown that GCN and DPG-Attention have a significant influence on DPG-LSTM performance. DPG-Attention, in particular, has a larger influence on categorization accuracy than GCN. The experiments also reveal that the length of the text has an impact on DPG-LSTM performance.

Previous studies simply combined semantic and syntactic data in social media, with the majority of them employing step-by-step feature extraction or feature concatenation techniques. Compared with earlier research, the study we propose varies in two main aspects. First, we present a dual-channel neural network that can collect both semantic and syntactic information in a social media text. Second, the upgraded DPG-Attention is adopted with DP to completely incorporate the semantic and syntactic information. The sentence feature is augmented with additional syntactic information by the DPG-Attention to produce superior sentiment analysis results.

In future work, there are still two points that can be improved. The existence or absence of dependence relations is used to generate the dependency graph, although this dependency graph does not adequately capture the whole sentence's syntactic information. Furthermore, our present studies are focused on English tweet data, but social media is made up of a variety of national languages.

7. Conclusions

Since the existing state-of-the-art approaches rely solely on the semantic information of social media text, this research proposes a novel method for classifying sentiment on social media that incorporates the structural information of the sentences themselves. Combining semantic and syntactic information is an effective way to distinguish emotional polarity. However, if a reliable approach to extracting both semantic and syntactic information characteristics is lacking, the model's performance may deteriorate.

In this study, we propose a new model, DPG-LSTM, based on the BiLSTM and GCN. DPG-LSTM analyzes the syntax of the user-posted text and employs DPG-Attention to strengthen the semantic and syntactic features. Experiments on real-world datasets show that our approach outperforms state-of-the-art methods in sentiment classification, achieving *R*, *P*, and *F1* scores of 0.845, 0.842, and 0.843.

Our ultimate aim is to evaluate how useful the newly introduced syntactic information is for sentiment analysis by feeding it into a deep learning model, which should help us build a high-quality sentiment classifier that handles a wider range of emotions. The use of DPG-LSTM improves performance in numerous domains without the need for domain-specific feature engineering.

Author Contributions: Conceptualization, L.W. and Z.Y.; methodology, L.W., Z.Y. and M.J.H.; software, Z.Y. and J.S.; validation, Z.Y., Y.H. and Y.C.; resources, Y.H. and Y.C.; data curation, Z.Y. and J.S.; writing—original draft preparation, L.W. and Z.Y.; writing—review and editing, M.J.H., Z.Y. and X.Z.; visualization, Z.Y., J.S. and M.J.H.; project administration, X.Z. and L.W.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by National Science Foundation of China [No. 81873915], Project of Nantong Health Commission [MB2020045], Science and Technology Project of Nantong City [MS22021027].

Data Availability Statement: The datasets supporting this paper are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Acknowledgments: We gratefully acknowledge the assistance of Jian Zhao and Ying Yu in funding the experiment.

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

References

1. Cengiz, A.B.; Kalem, G.; Boluk, P.S. The Effect of Social Media User Behaviors on Security and Privacy Threats. *IEEE Access* **2022**, *10*, 57674–57684. [\[CrossRef\]](#)
2. Hidekazu, T.; El-Sayed, A.; Masao, F.; Kazuhiro, M.; Kazuhiko, T.; Aoe, J.I. Estimating sentence types in computer related new product bulletins using a decision tree. *Inf. Sci.* **2004**, *168*, 185–200. [\[CrossRef\]](#)
3. National Research Council. *Frontiers in Massive Data Analysis*; National Academies Press: Washington, DC, USA, 2013.
4. Ju, S.; Li, S. Active learning on sentiment classification by selecting both words and documents. In *Workshop on Chinese Lexical Semantics*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 49–57.
5. Liu, L.; Nie, X.; Wang, H. Toward a fuzzy domain sentiment ontology tree for sentiment analysis. In Proceedings of the International Congress on Image and Signal Processing, Chongqing, China, 16–18 October 2012; pp. 1620–1624.
6. Sánchez-Rada, J.F.; Iglesias, C.A. Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison. *Inf. Fusion* **2019**, *52*, 344–356. [\[CrossRef\]](#)
7. Birjali, M.; Kasri, M.; Beni-Hssane, A. A comprehensive survey on sentiment analysis: Approaches, challenges, and trends. *Knowl.-Based Syst.* **2021**, *226*, 107134. [\[CrossRef\]](#)
8. Zhao, Z.; Rao, G.; Feng, Z. DFDS: A Domain-Independent Framework for Document-Level Sentiment Analysis Based on RST. In Proceedings of the Asia-Pacific Web and Web-Age Information Management Joint Conference on Web and Big Data, Beijing, China, 7–9 July 2017; pp. 297–310.
9. Chen, T.; Xu, R.; He, Y.; Wang, X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst. Appl.* **2017**, *72*, 221–230. [\[CrossRef\]](#)
10. Tang, D.; Zhang, M. Deep Learning in Sentiment Analysis. In *Deep Learning in Natural Language Processing*; Springer: Singapore, 2008; pp. 219–253.
11. Zhang, C.; Li, Q.; Song, D. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 4568–4578.
12. Mukherjee, S.; Malu, A.; Ar, B.; Bhattacharyya, P. Feature specific sentiment analysis for product reviews. In Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, Maui, HI, USA, 29 October–2 November 2012; pp. 475–487.
13. Kponyo, J.J.; Kuang, Y.; Zhang, E.; Domenic, K. VANET cluster-on-demand minimum spanning tree (MST) prim clustering algorithm. In Proceedings of the International Conference on Computational Problem-Solving, Jiuzhai, China, 26–28 October 2013; pp. 101–104.
14. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2019**, arXiv:1609.0290.
15. Sun, K.; Zhang, R.; Mensah, S.; Mao, Y.; Liu, X. Aspect-level sentiment analysis via convolution over dependency tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Online, 3–7 November 2019; pp. 5679–5688.
16. Yang, T.; Yin, Q.; Yang, L.; Wu, O. Aspect-based sentiment analysis with new target representation and dependency attention. *IEEE Trans. Affect. Comput.* **2019**, *13*, 640–650. [\[CrossRef\]](#)
17. Li, Y.; Zhou, H.; Lin, Z.; Wang, Y.; Chen, S.; Liu, C.; Wang, Z.; Gifu, D.; Xia, J. Investigation in the influences of public opinion indicators on vegetable prices by corpora construction and WeChat article analysis. *Future Gener. Comput. Syst.* **2020**, *102*, 876–888. [\[CrossRef\]](#)
18. Liu, H.; Xu, Y.; Zhang, Z.; Wang, N.; Huang, Y.; Hu, Y.; Yang, Z.; Rui, J.; Chen, H. A natural language processing pipeline of chinese free-text radiology reports for liver cancer diagnosis. *IEEE Access* **2020**, *8*, 159110–159119. [\[CrossRef\]](#)
19. Vizcarra, J.; Kozaki, K.; Torres Ruiz, M.; Quintero, R. Knowledge-based sentiment analysis and visualization on social networks. *New Gener. Comput.* **2021**, *39*, 199–229. [\[CrossRef\]](#)
20. Agarwal, B. Financial sentiment analysis model utilizing knowledge-base and domain-specific representation. *Multimed. Tools Appl.* **2022**, 1–22. [\[CrossRef\]](#)
21. Ma, Y.; Peng, H.; Khan, T.; Cambria, E.; Hussain, A. Sentic LSTM: A Hybrid Network for Targeted Aspect-Based Sentiment Analysis. *Cogn. Comput.* **2018**, *10*, 639–650. [\[CrossRef\]](#)
22. Ma, Y.; Peng, H.; Cambria, E. Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5876–5883.

23. Lin, Y.; Wang, C.; Song, H.; Li, Y. Multi-head self-attention transformation networks for aspect-based sentiment analysis. *IEEE Access* **2021**, *9*, 8762–8770. [\[CrossRef\]](#)
24. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
25. Wang, X.; Chen, S.; Li, T.; Li, W.; Zhou, Y.; Zheng, J.; Chen, Q.; Yan, J.; Tang, B. Depression risk prediction for chinese microblogs via deep-learning methods: Content analysis. *JMIR Med. Inform.* **2020**, *8*, 17958. [\[CrossRef\]](#)
26. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Processing Syst.* **2019**, *32*, 5754–5764.
27. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
28. Imran, A.S.; Daudpota, S.M.; Kastrati, Z.; Batra, R. Cross-Cultural Polarity and Emotion Detection Using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets. *IEEE Access* **2020**, *8*, 181074–181090. [\[CrossRef\]](#)
29. AlBadani, B.; Shi, R.; Dong, J. A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Appl. Syst. Innov.* **2022**, *5*, 13. [\[CrossRef\]](#)
30. Zhang, M.; Qian, T. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 3540–3549.
31. Ghosal, D.; Hazarika, D.; Roy, A.; Majumder, N.; Mihalcea, R.; Poria, S. KinGDOM: Knowledge-Guided DOMain Adaptation for Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3198–3210.
32. Speer, R.; Havasi, C. ConceptNet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP*; Association for Computational Linguistics: Berlin, Germany, 2013; pp. 161–176.
33. Chen, X.; Sun, C.; Wang, J.; Li, S.; Si, L.; Zhang, M.; Zhou, G. Aspect sentiment classification with document-level sentiment preference modeling. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3667–3677.
34. Liao, W.; Zeng, B.; Liu, J.; Wei, P.; Cheng, X.; Zhang, W. Multi-level graph neural network for text sentiment analysis. *Comput. Electr. Eng.* **2021**, *92*, 107096. [\[CrossRef\]](#)
35. Wu, H.; Zhang, Z.; Shi, S.; Wu, Q.; Song, H. Phrase dependency relational graph attention network for Aspect-based Sentiment Analysis. *Knowl. Based Syst.* **2022**, *236*, 107736. [\[CrossRef\]](#)
36. Xiang, C.; Zhang, J.; Li, F.; Fei, H.; Ji, D. A semantic and syntactic enhanced neural model for financial sentiment analysis. *Inf. Process. Manag.* **2022**, *59*, 102943. [\[CrossRef\]](#)
37. Rezaeina, S.M.; Rahmani, R.; Ghodsi, A.; Veisi, H. Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.* **2019**, *117*, 139–147. [\[CrossRef\]](#)
38. Zou, H.; Tang, X.; Xie, B.; Liu, B. Sentiment Classification Using Machine Learning Techniques with Syntax Features. In Proceedings of the International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, 7–9 December 2015; pp. 175–179.
39. Hammou, B.A.; Lahcen, A.A.; Mouline, S. Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics. *Inf. Process. Manag.* **2020**, *57*, 102122. [\[CrossRef\]](#)
40. Bifet, A.; Frank, E. Sentiment Knowledge Discovery in Twitter Streaming Data. In *International Conference on Discovery Science*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–15.
41. Chen, L.C.; Barron, J.T.; Papandreou, G.; Murphy, K.; Yuille, A.L. Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4545–4554.
42. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [\[CrossRef\]](#)
43. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharya, U.R. ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [\[CrossRef\]](#)
44. Sergio, G.C.; Lee, M. Stacked DeBERT: All attention in incomplete data for text classification. *Neural Netw.* **2021**, *136*, 87–96. [\[CrossRef\]](#)
45. Kamyab, M.; Liu, G.; Adjeisah, M. Attention-based CNN and Bi-LSTM model based on TF-IDF and glove word embedding for sentiment analysis. *Appl. Sci.* **2021**, *11*, 11255. [\[CrossRef\]](#)

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.