



Article Transformer-Based Graph Convolutional Network for Sentiment Analysis

Barakat AlBadani 🔍, Ronghua Shi, Jian Dong 🐿, Raeed Al-Sabri 🕒 and Oloulade Babatounde Moctard

School of Computer Science and Engineering, Central South University, Changsha 410083, China; barakat_lt@hotmail.com (B.A.); shirh@csu.edu.cn (R.S.); alsabriraeed@tu.edu.ye (R.A.-S.); b.oloulade@whut.edu.cn (O.B.M.)

* Correspondence: dongjian@csu.edu.cn

Abstract: Sentiment Analysis is an essential research topic in the field of natural language processing (NLP) and has attracted the attention of many researchers in the last few years. Recently, deep neural network (DNN) models have been used for sentiment analysis tasks, achieving promising results. Although these models can analyze sequences of arbitrary length, utilizing them in the feature extraction layer of a DNN increases the dimensionality of the feature space. More recently, graph neural networks (GNNs) have achieved a promising performance in different NLP tasks. However, previous models cannot be transferred to a large corpus and neglect the heterogeneity of textual graphs. To overcome these difficulties, we propose a new Transformer-based graph convolutional network for heterogeneous graphs called Sentiment Transformer Graph Convolutional Network (ST-GCN). To the best of our knowledge, this is the first study to model the sentiment corpus as a heterogeneous graph and learn document and word embeddings using the proposed sentiment graph transformer neural network. In addition, our model offers an easy mechanism to fuse node positional information for graph datasets using Laplacian eigenvectors. Extensive experiments on four standard datasets show that our model outperforms the existing state-of-the-art models.



1. Introduction

With the rapid growth of textual content on the Internet such as social networks and e-commerce websites, the need for contextual processing and mining of the subjective information that text holds is increasing [1]. Sentiment analysis, also called opinion mining, is an automatic technology to extract, process, judge, and summarize opinions, attitudes, and emotions from opinionative data. Nowadays, text sentiment analysis has become essential for many fields such as movie recommendation, e-commerce, and public opinion analysis [2]. For example, sentiment analysis aims to obtain the sentiment tendency of the person's opinions towards products, hot events, or any specific topic, which helps human decision-making [3]. Generally, researchers have explored three types of sentiment analysis approaches dictionary-based sentiment methods, machine learning-based sentiment methods.

Sentiment dictionary-based methods utilize dictionaries to determine the sentiment words in the given text and obtain the sentiment values. Then, using the sentiment calculation rules, the sentiment tendency is calculated [4,5]. The implementation of this approach is easy and does not require labeling samples. However, the quality of sentiment analysis depends on the sentiment dictionaries, which are insufficient to cover the sentiment words and lack the domain words, leading to the low quality of the sentiment analysis.

Later, to address the problem of dictionary dependency, machine learning approaches were proposed; such approaches utilize support vector machine SVM algorithm, naive Bayesian algorithm, graph-based semi-supervised classification algorithms to analyze the



Citation: AlBadani, B.; Shi, R.; Dong, J.; Al-Sabri, R.; Moctard, O.B. Transformer-Based Graph Convolutional Network for Sentiment Analysis. *Appl. Sci.* 2022, *12*, 1316. https://doi.org/ 10.3390/app12031316

Academic Editor: Evgeny Nikulchev

Received: 21 November 2021 Accepted: 20 January 2022 Published: 26 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



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/). text sentiment [6–8]. Despite the improvement of sentiment analysis that machine learning made, it strongly relies on corpus quality labeled with polarity.

In recent years, deep learning models have attracted the attention of many researchers to address the problem of feature extraction. They propose various deep learning-based methods for sentiment analysis, which achieved promising results compared to machine learning methods in sentiment association and sentiment classification [3,9–11]. However, deep learning models face the difficulty of extracting more comprehensive sentimental and emotional features since a large amount of emotional information is not utilized. As a result, more researchers try to integrate emotional information [12] and language knowledge [13] into the models [14–16]. Despite the great success of these models, they face the problem of extracting more comprehensive text emotional features since such models heavily rely on emotional resources and text information.

More recently, graph neural networks [17], or graph representation learning is a new research field that has received much attention form researchers. The entire corpus is represented as a graph in graph-based methods [18]. In graph embeddings, graph convolutional networks have proven to be effective at tasks involving knowledge representation and can retain the global structure information of a graph. However, most of the existing GNNs are built to learn node representations on fixed and homogeneous graphs. When learning representations on a misspecified graph or a heterogeneous graph with multiple types of nodes and edges, the restrictions become increasingly severe. In this work, we present a novel text graph transformer networks to address the GNNs issues. The text graph transformer network contains a new graph structure that can determine the useful connections between not directly connected nodes and learn the soft selection of edge types and complex relations.

To summarize, our contributions are as follows:

- 1. We propose a novel Sentiment Transformer Graph Convolutional Network (ST-GCN) that learns a new graph structure on a heterogeneous graph, including determining the useful connections between nodes that are not directly connected, and learning the soft selection of edge types and complex relations for learning node representation for sentiment classification. To the best of our knowledge, this is the first study to model the sentiment corpus as a heterogeneous graph and learn document and word embeddings using the proposed text graph transformer network;
- Inspired by the widespread use of positional encoding in NLP transformer models and current research on node positional features in GNNs, our model offers an easy mechanism to fuse node positional information for graph datasets using Laplacian eigenvectors;
- 3. Results on several sentiment benchmark datasets demonstrate that our model outperforms the state-of-the-art sentiment classification methods.

2. Related Work

2.1. Sentiment Analysis

The origin of sentiment analysis refers to the sciences of psychology, sociology and anthropology which focus on human emotions [19–21]. Scholars have conducted extensive related research because of its usefulness in online review monitoring and business competitive intelligence. To date, several methods have been used for such analysis. They can be classified into two broad groups: the traditional methods based on feature engineering, which essentially use dictionaries and machine learning approaches, and modern methods based on deep learning methods.

Early models performed sentiment analysis based on a set of rules, relying on sets of emotion dictionaries, and a large amount of labeled data was required for feature engineering. Liu et al. [22] defined emotion as a tuple of (holder, target, polarity, time) where holder represents the opinion's author, target refers to the related subject, polarity is the category of the expressed emotion, and time means the time of the evaluation. Another method by [23] classifies sentiments by combining the individual word-level sentiment. Ref. [24] introduced a generative model that jointly models emotion words, subject words and emotion polarity in a sentence as a triple. The main drawback of this method is the resulting high dimensional feature space. For addressing this problem, many works have used feature selection techniques [25,26] applying various machine learning approaches. Of the various machine learning classification methods used to classify users' sentiments from a text, decision tree, LDA, Naive Bayes, Support Vector Machine (SVM), and artificial neural networks are the most common and have achieved a higher performance [9,22,27,28]. However, these methods need massive training data and are often slow. To approach these problems, unsupervised lexicon-based methods were proposed, making use of both supervised and lexicon-based approaches [29,30]. Following this idea, many other methods [31,32] have been introduced.

In recent years, many researchers have applied deep neural networks for classifying sentiment. Unlike traditional machine learning methods, they can automatically complete the feature generation step and learn more extensive representation. Ref. [33] used a convolutional neural network (CNN) based model and connected a max-pooling layer after each convolution to extract features from the text. The emotion polarity is determined after inputting the fully connected layer. Ref. [34] adopted a dynamic max-pooling to capture fine-gained features. The authors learn the embedding of text regions by applying CNNs to high-dimensional text data. Later, Ref. [35] used the CNN model based on letter-level features, combining six convolutional layers and three fully connected layers for largescale text classification datasets. Although CNN models are faster than RNNs because of parallelization, they can only extract the local features in the filter region. A memory unit is introduced with recurrent neural networks (RNNs) to make the network have memory ability. Hence, RNN can consider the long-distance dependency within texts. However, original RNNs suffer from gradient dispersion and gradient disappearance, which affect the learning process [3]. To solve this problem, the long short term memory (LSTM) model has been used [36]. LSTMs use a gate mechanism which can keep the connection within instances and capture the relationship between words. Recently, attention-based sentiment analysis models have been used and outperform previous methods. Yang et al. [37] propose an attention-based model that mirrors the hierarchical structure of documents before applying two attention mechanism layers at the sentence and word level.

More recently, graph neural networks (GNNs) have become a powerful approach for industries and academies. GNNs have been widely used in NLP tasks [38–40]. Ref. [18] proposed Text-GCN, which uses a heterogeneous graph where nodes are documents and words appear in documents. An edge between two words means the words appear in the same text and an edge between a text and a word means the word appears in the text. Edge weight is calculated using TF-IDF for words-text edge and positive point-wise mutual information (PPMI) for a word-word edge. Next, the data graph representation is learned using a convolutional graph network. The task, which can be seen as node classification, suffers from memory problems because they have to build a single graph for a whole dataset. Moreover, the graph is built ignoring the order information of words. To overcome the former drawback, Huang et al. [41] proposed another GNN-based method for text classification using a text-level graph for each input text. Thus, they perform graph classification instead of node classification. However, they ignore the rich word positional information, which is critical in sentiment analysis. To address the problems above, we propose a transformer-based Graph Convolutional Network, following up on [18] and adding word positional information encoding to word features, and propose a new batching mechanism to alleviate the memory problem.

2.2. Transformer Convolutional Networks

NLP problems, such as language modeling and machine translation, have been solved by recurrent neural networks (RNNs). RNN factor computation along with the positions of elements in the input and output sequences to keep the order of the sentence in place. This intrinsically sequential nature prevents parallel computation inside the training set and is non-trivial for extended length sequences computation because the memory constraints limit batch processing between samples. To overcome this limitation, Refs. [42,43] proposed factorization tricks, and conditional computation, respectively, notably increase the computational efficiency. However, they still make use of sequential computation. To mitigate the effect of the sequential computation, many researchers have used attention mechanisms [44,45] as they allow the modeling of dependencies regardless of their distance in the input or output sequence. Attention mechanisms break the memory constraint problem and have become an indispensable part of sequence modeling, but such attention was used in conjunction with RNNs. Ref. [46] proposed a transformer model architecture, which avoids recurrence and alternatively relies entirely on an attention mechanism to describe global dependencies between input and output. Unlike RNNs, transformers do not necessarily process data in order. Instead, the attention mechanism provides context for any position in the input sequence, which can be passed in parallel. This feature allows greater parallelization than RNNs and therefore reduces training times. Thus, only attention mechanisms without any RNN can match the performance of RNNs with attention. In this work, we propose a sentiment transformer graph convolutional network to predict sentiment.

3. Method

In this section, we describe the framework of the proposed model as shown in Figure 1. First, we describe the data preprocessing step. Next, we introduce textual graph building. We introduce the word embedding representation. Then, we introduce the transformer convolutional networks. Finally, we present the text graph transformer convolutional network.



Figure 1. The methodology flowchart.

3.1. Data Preprocessing

In this section, we describe the data preprocessing step. First, we remove the irrelevant data from reviews. For example, punctuation, URLs, mentions, numbers, and non-English words have been removed from the reviews using the regular expression library in Python. Secondly, we define our stop words list, which contains words that do not hold emotional and systematical feelings, such as the articles and determiners, because the commonly used stop word lists (e.g., NLTK stop words (https://www.nltk.org/nltk_data/ accessed on 14 October 2021)) contain words that have a sentiment role. Then, we remove the defined stop words from the long review datasets. We use the white space to tokenize text into words.

All upper words are changed into lowercase. The output tokenized words will be used to build the text graph.

3.2. Textual Graph Building

In this section, we construct the text graph from the corpus. Let G = (N, E), be a graph where *N* is the node-set and *E* is the edge set. We represent the textual graph as follows:

3.2.1. Node Assignment

Each review and a unique keyword are represented as nodes in the text graph. The number of nodes in the textual graph is the number of reviews *D* plus the number of the unique keywords in *V* in the entire corpus.

3.2.2. Edging

Two types of edges are built between nodes. Term frequency-inverse document frequency (TF-IDF) is used to build the edges between a review node r_i and keyword nodes r_j , and point-wise mutual information (PMI) is used to build the edges between two keyword node pairs within a fixed window. We build an adjacency matrix that represents the edge weights. Those weights determine the relationship strength between two nodes.

We build the adjacency matrix *A* (the edge weights) as follows:

$$A_{i}(i,j) = \begin{cases} PMI(i,j) & i,j \text{ are keywords; PMI}(i,j) > 0\\ TF-IDF(i,j) & i \text{ is a review and } j \text{ is a keyword}\\ 1 & i = j\\ 0 & \text{otherwise.} \end{cases}$$
(1)

The PMI for keyword pair is calculated as follows:

$$PMI(i,j) = \log \frac{p(i,j)}{p(i)p(j)}.$$
(2)

Given a sliding window #W for the entire review corpus, the sliding windows in which keyword *i* and *j* appear together #W(i, j), and the sliding window in which the keyword *i* occur #W(i) the p(i, j) and p(i) is calculated as:

$$P(i,j) = \frac{\#W(i,j)}{\#W}$$
(3)

$$P(i) = \frac{\#W(i)}{\#W}.$$
(4)

3.3. Embedding (Word Representation)

In most natural language processing applications, words are used as features. The most popular word vector representations are distributed representation and one-hot representation [27,47]. However, the one-hot representation has various problems, such as the too-large vector dimension, the sparsity of the word vector, and ignoring the word semantic association. Although the distributed representation has addressed the problem of one-hot representation, the need to improve the accuracy of the word vector and the training speed is still crucial [48]. Recently, different word vectors have been applied to sentiment analysis [49–51]. However, the current word representation used in sentiment analysis does not take into account the sentiment information contained in words. In our work, we address the above problems by using a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model [52], which makes use of the transformers to learn the contextual information from the corpora, to obtain the review node embedding in the textual graph. To the best of our knowledge, this is the first study that utilizes the Bert model for document node embeddings in sentiment analysis tasks.

3.4. Graph Transformer Convolutional Networks

The Figure 2 shows the architecture of the proposed model. The model consists of a stack of functions of operator including positional encoding, feature transform, sampling, message computing, multi-head, and aggregation.



Figure 2. The ST-GCN architecture.

3.4.1. The Positional Encoding

The positional encoding consists of encoding positional information for each word of a sentence, which is difficult to apply to a graph because the presence of symmetries in the graph makes it non-trivial to get the canonical position of nodes. Meanwhile, words in the text need disambiguation, that is, words with the same spelling, but different meanings need to be differentiated. Ideally, each node of a graph should have unique PE, and nodes that are close in the graph should have similar PE whereas nodes that are far from each other should have different positional encoding. Node position embedding has been explored in recent GNN works [53–56] to learn both positional and structural features of nodes in graphs. We leverage the success of the recent works on positional information in GNNs [54,56] and use pre-computed Laplacian eigenvectors as Positional Encodings, which allow us to differentiate isomorphic nodes. Eigenvectors are defined using the factorization of the graph Laplacian matrix:

$$\Delta = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = \nu^T \Lambda \nu, \tag{5}$$

where *A* is $n \times n$ adjacency matrix, D is the degree matrix, and ν , Λ are the eigenvectors and eigenvalues, respectively. We use pre-computed Laplacian eigenvectors to add into the feature of the nodes, which are used as input for the first layer.

3.4.2. Feature Transform Operator

We input the node and edge features described above into the graph transformer. The input node and edge feature are d-dimensional hidden feature h_i^0 and e_i^0 , respectively. Then we embed the pre-computed node PE of dimension k using a linear projection. It should be noticed that we add the Laplacian positional encoding only to the node feature

for the first layer uniquely. Basically, for a graph *G* with node feature $X_u \in \mathbf{R}^{1 \times \mathbf{d}}$ for each node v_i and edge feature $X_{ij}^e \in \mathbf{R}^{1 \times \mathbf{d}_e}$ for each node between nodes v_i et v_j where *d* and d_e denote the node feature size and edge feature size respectively, the input node features x_u and edge features e_{ij} are passed via a linear projection to embed these to *d* dimensional hidden features h_i^0 and e_{ij}^0 .

$$\hat{h}_{i}^{0} = A^{0}X_{i} + a^{0} + \lambda_{i}^{0}; e_{ij}^{0} = B^{0}e_{ij} + b^{0},$$
(6)

where $A^0 \in \mathbf{R}^{d \times d_n}$, $B^0 \in \mathbf{R}^{d \times d_e}$ and $a^0, b^0 \in \mathbf{R}^d$ are parameters of the linear projection layer. λ_i^0 represents the pre-computed node positional encoding of dimension *k*.

3.4.3. Message Computation Operator

Based on attention mechanisms, the message computation operator makes it possible to focus on the most relevant neighboring nodes to improve information aggregation. Our message computation operator aims to learn an importance weight w_{ij} for each edge relationship e_{ij} between the two corresponding nodes v_i and v_j . We better utilize edge attributes information by designing an attention layer with edge feature (see Figure 2). We maintain a node-symmetric edge feature representation pipeline for propagating edge features. The update equation for a layer l is defined as follows:

$$\hat{h}_{i}^{l+1} = O_{h}^{l} H_{k=1} (\sum_{j \in N_{i}} w_{ij}^{k,l} V^{k,l} h_{j}^{l})$$
(7)

$$\hat{e}_{i}^{l+1} = O_{e}^{l} H_{k=1}(\hat{w}_{i,j}^{k,l}), where$$
(8)

$$w_{ij}^{k,l} = softmax_j(\hat{w}_{i,j}^{k,l}) \tag{9}$$

$$\hat{w}_{i,j}^{k,l} = \left(\frac{Q^{k,l}h_i^l \cdot K^{k,l}h_j^l}{\sqrt{d_k}}\right) \cdot E^{k,l}e_{i,j}^l, \tag{10}$$

with $Q^{k,l}$, $K^{k,l}$, $V^{k,l}$, $E^{k,l} \in \mathbf{R}d_k$, O_h^l , $O_e^l \in \mathbf{R}^{d \times d}$, $k \in \{1, 2, ..., H\}$ represents the number of attention head, and where $O_h^l \in \mathbf{R}^{d \times d}$, $V^{k,l} \in \mathbf{R}^{d_k \times d}$, H denotes the number of heads, L the number of layers, d is the hidden dimension and d_k is the dimension of a head $\frac{d}{H} = d_k$. Note that h_i^l is the i - th node's feature at the l - th layer.

3.4.4. Multi-Head Operator

For stabilizing the learning process, we follow up on [46] and perform multiple attentions independently. The multiple representation outputs by multi-head attention for each node v_i are then concatenated or averaged to generate the final representation h_i

3.4.5. Aggregation Operator

For combining features from multiple neighbors to obtain the representation h_i , an aggregation function is required. We use *max* formulated as :

$$a_v^k = MAX(RELU(W.h_u^{k-1}), \forall u \in N(v)).$$
(11)

4. Experiments

4.1. Baselines

The proposed model is compared with multiple state-of-the-art sentiment analysis models as follows:

RGWE: Unsupervised methods, in particular neural network-based approaches, exploit unstructured data to generate and retrieve hidden sentiment information by identifying the constraints of conjunctions on the positive or negative semantic orientations [57];

- Seninfo+TF-IDF: an improved word representation method, which integrated the contribution of sentiment information into the traditional TF-IDF algorithm and generated weighted word vectors [58];
- Re(Glove): a word vector refinement model to refine pre-trained word vectors using sentiment intensity scores provided by sentiment lexicons, which improved each word vector and performed better in Sentiment Analysis [59];
- CHIM: a model in which the author represents attributes as chunk-wise important weight metrics. The authors consider four locations to inject attributes (i.e., encoding, embedding, classifier, and attention) with simple BiLSTM [60]. In our comparison, we compare with the embedding location inject since it achieved the highest accuracy score;
- HCSC: a model that combines BiLSTM and CNN as the base model and incorporates attributes by the bias-attention method, and considers the existence of cold start entities [61];
- CMA: a model that incorporates attributes using the bias-attention method with the baseline LSTM and hierarchical attention classifier [62];
- Single-layered BiLSTM: a single-layered BiLSTM model with a global pooling mechanism in which the number of parameters is reduced, leading to looser computation [63];
- LSTM/BiLSTM: Long short-term memory network and Bidirectional long short-term memory network;
- SAMF-BiLSTM: a bidirectional model with the self-attention technique and multichannel features for sentiment classification [64];
- SMART: a robust computation framework that fine-tunes large-scale pre-trained natural language models in a principled manner. we report the usage of *BERT*_{BASE} and the *RoBERTa*_{LARGE} as the pre-trained models [65];
- RCNN: a model that combines RNN and CNN for text sentiment classification [66];
- BERT-pair-TextCNN: a representation framework called Bert-pair-Networks (p-BERTs) in which BERT is used to encode sentences for sentiment classification to classify a single sentence utilizing, on the top, the auxiliary sentence and feature extraction [67].

4.2. Datasets

We select four classical public datasets to evaluate the proposed TGTCN model. The statistics of the datasets are shown in Table 1. For the datasets that have standard train/valid/test such as SemEval [68] and SST-B [36], we have conducted our experiments according to the standard split. For those datasets that do not have a standard split, we split the datasets with 7:1:2 to obtain the corresponding train/valid/test. We also made sure that the intersection of the training and test sets was not empty to avoid technical terms influencing Sentiment Analysis.

| Dataset | Train | Valid | Test | Total | #Labels | Labels | Balance |
|------------------------|--------|-------|------|--------|---------|---|---------|
| SemEval | 9684 | 1654 | 3813 | 15,151 | 3 | positive/negative/neutral | No |
| SST2 | 6920 | 872 | 1821 | 9613 | 2 | positive/negative | No |
| IMDB | 40,000 | 5000 | 5000 | 5000 | 2 | positive/negative | Yes |
| Yelp 2014 (Restuarant) | 3072 | 384 | 384 | 3840 | 5 | very positive/positive/ neutral/negative/very negative | Yes |

Table 1. Detailed statistics of the evaluation datasets.

4.3. Experiments Settings

SGTN is implemented using PyTorch and is optimized with an Adam optimizer. Training and experiments are done using an NVIDIA GeForce GTX 1080 Ti graphics card. We select the optimal values of learning parameters when the model achieves the highest accuracy for the validation samples. The optimal value of the learning rate α is

set to 0.0005. L2 regularization is set to 10^{-6} , and the dropout rate is set to 0.3 for the best performance. For learning SGTN, the model is trained for 100 epochs with the early-stop strategy. For baseline models, we either run the codes provided by the authors using the same parameters described in the papers or the results reported in the previous work [57].

4.4. Evaluation Criteria

To evaluate the performance of the proposed SGTN model, we use the two main evaluation criteria, namely Accuracy (*Acc*) and *F*1 measure (*F*1). These criteria have been used extensively in text classification, and sentiment analysis tasks [69], which are computed as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}.$$
(12)

To calculate the *F*1 measure, we first compute the Precision (*Pr*) and Recall (*Re*) as follows.

$$Pr = \frac{IP}{TP + FP},\tag{13}$$

$$Re = \frac{TP}{TP + FN}.$$
(14)

Then the *F*1 is calculated as follows:

$$F1 = \frac{2 \times Pr \times Re}{Pr + Re},\tag{15}$$

where *TN*, *TP*, *FN* and *FP* are true negative, true positive, false negative, and false positive, respectively [69].

4.5. Comparison Results

The optimal parameters that achieved the best results in our model are shown in Table 2. The proposed model is compared with 12 models on four public datasets. The main results are reported in Tables 3 and 4.

| Table 2. The optimal h | per-parameters | on datasets. |
|-------------------------------|----------------|--------------|
|-------------------------------|----------------|--------------|

| Hyperparameter | SST-B | IMDB | SemEval | Yelp |
|-----------------------|-----------------------------|---------------|----------------|---------|
| Epochs | 200 | 200 | 200 | 200 |
| learning rate | 0.2 | 0.05 | 0.001 | 0.05 |
| Optimization function | | Mini-Batch Gi | radient Desent | |
| loss function | Cross Entropy Loss function | | | |
| Dropout | 0.6 | 0.5 | 0.6 | 0.5 |
| Batch Size | 512 | 512 | 512 | 512 |
| Weight Decay | 0.0005 | 0.00005 | 0.00001 | 0.00001 |
| Hidden layer unit | 32 | 32 | 64 | 16 |

From the result in Table 3, we noticed that the proposed model has achieved better classification accuracy than the baseline state-of-the-art models over all datasets. For example, the classification performance is improved by 2.63%, 0.43% over *SMART*_{RoBERTa} and *BERT_pair_RCNN*. On SST-B, the classification accuracy rate of the proposed model reached 95.43%, on the IMDB, the accuracy rate reached 94.95%, on the Yelp dataset, the accuracy rate reached 72.7%.

We also report the F1-score of the proposed model compared with five state-of-the-art models. From the results in Table 4, it is noticed that our model outperforms the baseline model over the four datasets. For example, our model achieved 74.12% on the Semeval dataset, 95.11 on SST-B, 93.52 on IMDB, and 50.2 on the Yelp dataset. The F1-score is improved by 1.23% and 3.95% over RCNN and BiLSTM on SST-B, respectively.

For more in-depth analysis, the Bert-based models have achieved better classification results than the conventional deep learning models. We can also see that the neural network models have better results compared with the machine learning methods.

Table 3. The sentiment classification accuracy of different models over datasets. The best score on each task produced by a single model is in bold and "–" denotes the missed result.

| Method | SST-B | IMDB | Yelp 2014 |
|-----------------------|-------|--------|-----------|
| CHIM | - | 54.2 | - |
| HCSC | _ | 56.4 | 69.2 |
| СМА | - | 54.0 | 67.6 |
| Single-layered BiLSTM | 85.78 | 90.585 | _ |
| SAMF-BiLSTM-D | 89.7 | 48.9 | _ |
| LSTM | 84.9 | 37.8 | 53.9 |
| BiLSTM | 91.24 | 83.02 | _ |
| RCNN | 93.96 | 84.70 | _ |
| BERT_pair_RCNN | 95.00 | - | - |
| Bert | 90.9 | - | - |
| SMART_BERT | 90.0 | - | _ |
| SMART_RoBERTa | 92.8 | _ | _ |
| SGTN (ours) | 95.43 | 94.94 | 72.7 |

Table 4. The F1-score of different models over datasets. The best score on each task produced by a single model is in bold and "–" denotes the missed result.

| Method | SemEval | SST-B | IMDB | Yelp 2014 |
|----------------|---------|-------|-------|-----------|
| Re(Glove) | 68.2 | 89.5 | 89.6 | 46.1 |
| Seninfo+TF-IDF | 66.7 | 88.8 | 89.0 | 45.4 |
| RGWE | 69.1 | 89.68 | 90.1 | 46.9 |
| BiLSTM | - | 91.16 | 83.05 | - |
| RCNN | - | 93.88 | 84.72 | - |
| SGTN (ours) | 74.12 | 95.11 | 93.52 | 50.2 |

4.6. Ablation Study

4.6.1. Impact of Removing Less Frequent Words

Removing the less frequent words from tweets may affect the performance of sentiment analysis. We conduct an ablation study to test the impact of removing the less-frequency words. We delete the words with frequency less than five times in the entire corpus. The result from Tables 5 and 6 show that removing the less frequent words have slightly degraded the performance. For example, the sentiment accuracy performance decreases by 0.21% on the SST-B dataset and by 0.42%. We also test the influence of our predefined stop words. From the results in Tables 5 and 6, it shows that using NLTK stop words has affected the accuracy sentiment performance.

Table 5. The impact of removing less frequent words on the accuracy performance of SGTN.

| Method | SST-B | IMDB | Yelp 2014 |
|------------------------------|-------|-------|-----------|
| SGTN with less Freq words | 95.24 | 94.53 | 72.3 |
| SGTN without less Freq words | 95.43 | 94.95 | 72.7 |
| SGTN with NLTK Stopwords | 93.43 | 93.01 | 71.10 |

| Method | SemEval | SST-B | IMDB | Yelp 2014 |
|------------------------------|---------|-------|-------|-----------|
| SGTN with less Freq words | 74.01 | 94.91 | 93.72 | 49.50 |
| SGTN without less Freq words | 74.12 | 95.11 | 93.52 | 50.2 |
| SGTN with NLTK Stopwords | 72.46 | 93.09 | 92.84 | 48.23 |

Table 6. The impact of removing less frequent words on the F1 score performance of SGTN.

4.6.2. Epoch

The number of iterations in the training set are called epochs. The model's generalization ability improves as the number of Epochs increases. However, if the number of epochs is too great, the over-fitting problem can easily arise, reducing the model's generalization capabilities. As a result, selecting the appropriate Epochs is critical. Figure 3 depicts the model's classification effect with different epochs.



Figure 3. Relation between Epochs and accuracy score.

It is noticed from Figure 3 that with the increasing of the epoch, the classification performance (accuracy score) of the model is gradually increasing. It tends to be stable when epochs are 60.

4.6.3. Learning Rate

When it comes to optimizing weights and offsets, identifying the appropriate learning rate is critical. It is easy to overshoot the extreme point if the learning rate is too high, causing the system to become unstable. The training duration will be excessive if the learning rate is too slow. The model's classification impact at various learning rates is depicted in Figure 4.



Figure 4. Relation between learning rate and accuracy score.

5. Conclusions and Future Work

In this research, we propose a convolutional network of transformer-based graphs for sentiment analysis. We represented the problem as a node classification task and learned the representation of nodes on a heterogeneous graph through the message passing. We show that using a transformer to aggregate local substructures with appropriate position encoding is a very efficient node representation strategy, and the multi-head attention allows a simple interpretation of the model. The learned graph structure leads to a more efficient node representation, resulting in peak performance without any predefined metapath from domain knowledge. Comprehensive experiments illustrate the effectiveness of the proposed model. ST-GCN outperforms previous cutting-edge models on four real-world datasets: *SemEval, SST-B, IMDB,* and *Yelp 2014*. In addition to generalizing the ST-GCN design to inductive parameters, some interesting future directions include using Dynamic neighborhood aggregation operators to improve classification performance. As several heterogeneous graph datasets have been recently studied for other network analysis tasks, such as link prediction and graph classification, applying ST-GCN to the other tasks can be interesting future directions.

Author Contributions: Conceptualization, B.A.; Funding acquisition, R.S. and J.D.; Methodology, B.A.; Project administration, J.D.; Supervision, R.S. and J.D.; investigation, B.A. and J.D.; resources, R.S.; data curation, B.A. and R.S.; writing—original draft preparation, B.A.; Writing—review & editing, B.A., J.D., R.A.-S. and O.B.M.; visualization, B.A., R.A.-S. and O.B.M. software, B.A., R.A.-S. and O.B.M.; validation, B.A., O.B.M. and R.A.-S. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China: 61971450 and 61801521; Natural Science Foundation of Hunan Province: 2018JJ2533; Fundamental Research Funds for Central Universities of the Central South University: 2018gczd014 and 20190038020050.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

| NLP | natural language processing |
|--------|---|
| DNN | deep neural network |
| GNN | graph neural network |
| ST-GCN | Sentiment Transformer Graph Convolutional Network |
| CNN | convolutional neural network |
| TF-IDF | Term frequency-inverse document frequency |
| PMI | point-wise mutual information |
| SVM | support vector machine |
| RNN | ecurrent neural network |
| LSTM | long short term memory |
| | |

References

- Habimana, O.; Li, Y.; Li, R.; Gu, X.; Yu, G. Sentiment analysis using deep learning approaches: An overview. *Sci. China Inf. Sci.* 2020, *63*, 111102. [CrossRef]
- 2. Anbazhagu, U.; Anandan, R. Emotional interpretation using chaotic cuckoo public sentiment variations on textual data from Twitter. *Int. J. Speech Technol.* **2021**, *24*, 281–290. [CrossRef]
- 3. Cheng, Y.; Sun, H.; Chen, H.; Li, M.; Cai, Y.; Cai, Z.; Huang, J. Sentiment Analysis Using Multi-Head Attention Capsules With Multi-Channel CNN and Bidirectional GRU. *IEEE Access* 2021, *9*, 60383–60395. [CrossRef]
- 4. Lee, S.H.; Cui, J.; Kim, J.W. Sentiment analysis on movie review through building modified sentiment dictionary by movie genre. *J. Intell. Inf. Syst.* **2016**, *22*, 97–113.
- 5. Li, Z.; Li, R.; Jin, G. Sentiment analysis of danmaku videos based on naïve bayes and sentiment dictionary. *IEEE Access* 2020, *8*, 75073–75084. [CrossRef]
- 6. Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine learning-based sentiment analysis for twitter accounts. *Math. Comput. Appl.* **2018**, *23*, 11. [CrossRef]
- 7. Hew, K.F.; Hu, X.; Qiao, C.; Tang, Y. What predicts student satisfaction with MOOCs: A gradient boosting trees supervised machine learning and sentiment analysis approach. *Comput. Educ.* **2020**, *145*, 103724. [CrossRef]
- Jagdale, R.S.; Shirsat, V.S.; Deshmukh, S.N. Sentiment analysis on product reviews using machine learning techniques. In Cognitive Informatics and Soft Computing; Springer: Berlin/Heidelberg, Germany, 2019; pp. 639–647.
- 9. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2018, *8*, e1253. [CrossRef]
- 10. Yadav, A.; Vishwakarma, D.K. Sentiment analysis using deep learning architectures: A review. *Artif. Intell. Rev.* 2020, 53, 4335–4385. [CrossRef]
- 11. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [CrossRef]
- 12. Kang, K.; Tian, S.; Yu, L. Drug Adverse Reaction Discovery Based on Attention Mechanism and Fusion of Emotional Information. *Autom. Control Comput. Sci.* 2020, 54, 391–402. [CrossRef]
- Peng, Z.; Song, H.; Kang, B.; Moctard, O.; He, M.; Zheng, X. Automatic textual Knowledge Extraction based on Paragraph Constitutive Relations. In Proceedings of the 6th International Conference On Systems And Informatics, ICSAI 2019, Shanghai, China, 2–4 November 2019; pp. 527–532.
- 14. Luo, L.X. Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Pers. Ubiquitous Comput.* **2019**, 23, 405–412. [CrossRef]
- 15. Chen, K.; Liang, B.; Ke, W.D. Sentiment analysis of Chinese Weibo based on multi-channel convolutional neural network. *Comput. Res. Develop.* **2018**, *55*, 945–957.
- Teng, Z.; Vo, D.T.; Zhang, Y. Context-sensitive lexicon features for neural sentiment analysis. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1629–1638.
- 17. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
- Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
- Lee, C.M.; Narayanan, S.S.; Pieraccini, R. Classifying emotions in human-machine spoken dialogs. In Proceedings of the 2002 IEEE International Conference on Multimedia and Expo, ICME 2002, Lausanne, Switzerland, 26–29 August 2002; Volume I, pp. 737–740. [CrossRef]

- 20. Lisetti, C.L.; Nasoz, F. Using Noninvasive Wearable Computers to Recognize Human Emotions from Physiological Signals. *EURASIP J. Adv. Signal Process.* 2004, 2004, 1672–1687. [CrossRef]
- Sharaf Al-deen, H.S.; Zeng, Z.; Al-sabri, R.; Hekmat, A. An Improved Model for Analyzing Textual Sentiment Based on a Deep Neural Network Using Multi-Head Attention Mechanism. *Appl. Syst. Innov.* 2021, *4*, 85. [CrossRef]
- Liu, B. Sentiment Analysis and Opinion Mining. Synth. Lect. Hum. Lang. Technol. 2012, 5, 1–167. HLT016. [CrossRef]
- Kim, S.; Hovy, E.H. Determining the Sentiment of Opinions. In Proceedings of the COLING 2004, 20th International Conference on Computational Linguistics, Geneva, Switzerland, 23–27 August 2004.
- Eguchi, K.; Lavrenko, V. Sentiment Retrieval using Generative Models. In Proceedings of the EMNLP 2006, 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; Jurafsky, D., Gaussier, É., Eds.; ACL: Stroudsburg, PA, USA, 2006; pp. 345–354.
- Duric, A.; Song, F. Feature selection for sentiment analysis based on content and syntax models. *Decis. Support Syst.* 2012, 53, 704–711. [CrossRef]
- Abbasi, A.; France, S.L.; Zhang, Z.; Chen, H. Selecting Attributes for Sentiment Classification Using Feature Relation Networks. *IEEE Trans. Knowl. Data Eng.* 2011, 23, 447–462. [CrossRef]
- Naseem, U.; Razzak, I.; Khan, S.K.; Prasad, M. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Trans. Asian Low-Resour. Lang. Inf. Process.* 2021, 20, 1–35. [CrossRef]
- 28. Singh, G. Decision Tree J48 at SemEval-2020 Task 9: Sentiment Analysis for Code-Mixed Social Media Text (Hinglish). *arXiv* 2020, arXiv:2008.11398.
- 29. Liu, B. Sentiment Analysis-Mining Opinions, Sentiments, and Emotions; Cambridge University Press: Cambridge, UK, 2015.
- Ren, S.; Liu, S.; Zhou, M.; Ma, S. A Graph-based Coarse-to-fine Method for Unsupervised Bilingual Lexicon Induction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3476–3485. [CrossRef]
- 31. Ghiassi, M.; Lee, S. A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. *Expert Syst. Appl.* 2018, 106, 197–216. [CrossRef]
- 32. Chikersal, P.; Poria, S.; Cambria, E.; Gelbukh, A.F.; Siong, C.E. Modelling Public Sentiment in Twitter: Using Linguistic Patterns to Enhance Supervised Learning. In *Computational Linguistics and Intelligent Text Processing, Proceedings of the 16th International Conference, CICLing 2015, Cairo, Egypt, 14–20 April 2015;* Lecture Notes in Computer Science; Part II; Gelbukh, A.F., Ed.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9042, pp. 49–65._4. [CrossRef]
- 33. Zhang, Y.; Wallace, B.C. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *arXiv* **2015**, arXiv:1510.03820.
- Liu, J.; Chang, W.; Wu, Y.; Yang, Y. Deep Learning for Extreme Multi-label Text Classification. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017; Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A.P., White, R.W., Eds.; ACM: New York, NY, USA, 2017; pp. 115–124. [CrossRef]
- Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst. 2012, 25, 1097–1105. [CrossRef]
- 36. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, Grand Hyatt Seattle, Seattle, WA, USA, , 18–21 October 2013; A Meeting of SIGDAT, a Special Interest Group of the ACL; ACL: Stroudsburg, PA, USA, 2013; pp. 1631–1642.
- 37. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.J.; Hovy, E.H. Hierarchical Attention Networks for Document Classification. In Proceedings of the NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Knight, K., Nenkova, A., Rambow, O., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1480–1489. [CrossRef]
- Ma, Q.; Yuan, C.; Zhou, W.; Hu, S. Label-Specific Dual Graph Neural Network for Multi-Label Text Classification. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 3855–3864. [CrossRef]
- 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]
- 40. Xu, S.; Xiang, Y. Frog-GNN: Multi-perspective aggregation based graph neural network for few-shot text classification. *Expert Syst. Appl.* **2021**, *176*, 114795. [CrossRef]
- Huang, L.; Ma, D.; Li, S.; Zhang, X.; Wang, H. Text Level Graph Neural Network for Text Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 3442–3448. [CrossRef]

- Kuchaiev, O.; Ginsburg, B. Factorization tricks for LSTM networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
- Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.V.; Hinton, G.E.; Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
- 44. Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured Attention Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
- Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
- Ma, T.; Al-Sabri, R.; Zhang, L.; Marah, B.; Al-Nabhan, N. The Impact of Weighting Schemes and Stemming Process on Topic Modeling of Arabic Long and Short Texts. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 2020, 19, 1–23. [CrossRef]
- Alon, U.; Zilberstein, M.; Levy, O.; Yahav, E. code2vec: Learning distributed representations of code. *Proc. ACM Program. Lang.* 2019, 3, 1–29. [CrossRef]
- Jin, N.; Wu, J.; Ma, X.; Yan, K.; Mo, Y. Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification. *IEEE Access* 2020, *8*, 77060–77072. [CrossRef]
- 50. Picasso, A.; Merello, S.; Ma, Y.; Oneto, L.; Cambria, E. Technical analysis and sentiment embeddings for market trend prediction. *Expert Syst. Appl.* **2019**, 135, 60–70. [CrossRef]
- Wu, Z.; Dai, X.Y.; Yin, C.; Huang, S.; Chen, J. Improving review representations with user attention and product attention for sentiment classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
- 52. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Volume 1 (Long and Short Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
- 53. Murphy, R.L.; Srinivasan, B.; Rao, V.A.; Ribeiro, B. Relational Pooling for Graph Representations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 9–15 June 2019*; Volume 97, pp. 4663–4673.
- You, J.; Ying, R.; Leskovec, J. Position-aware Graph Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 7134–7143.
- 55. Dwivedi, V.P.; Joshi, C.K.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking Graph Neural Networks. arXiv 2020, arXiv:2003.00982.
- Srinivasan, B.; Ribeiro, B. On the Equivalence between Positional Node Embeddings and Structural Graph Representations. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
- Wang, Y.; Huang, G.; Li, J.; Li, H.; Zhou, Y.; Jiang, H. Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis. *IEEE Access* 2021, 9, 37075–37085. [CrossRef]
- Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* 2019, 7, 51522–51532. [CrossRef]
- Gu, S.; Zhang, L.; Hou, Y.; Song, Y. A position-aware bidirectional attention network for aspect-level sentiment analysis. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 774–784.
- 60. Amplayo, R.K. Rethinking attribute representation and injection for sentiment classification. arXiv 2019, arXiv:1908.09590.
- Amplayo, R.K.; Kim, J.; Sung, S.; Hwang, S.W. Cold-start aware user and product attention for sentiment classification. *arXiv* 2018, arXiv:1806.05507.
- Ma, D.; Li, S.; Zhang, X.; Wang, H.; Sun, X. Cascading multiway attentions for document-level sentiment classification. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Taipei, Taiwan, 27 November–1 December 2017; pp. 634–643.
- Hameed, Z.; Garcia-Zapirain, B. Sentiment classification using a single-layered BiLSTM model. *IEEE Access* 2020, *8*, 73992–74001. [CrossRef]
- 64. Li, W.; Qi, F.; Tang, M.; Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [CrossRef]
- Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; Zhao, T. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv* 2019, arXiv:1911.03437.
- 66. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
- 67. Wang, Z.; Wu, H.; Liu, H.; Cai, Q.H. Bert-Pair-Networks for Sentiment Classification. In Proceedings of the 2020 International Conference on Machine Learning and Cybernetics (ICMLC), Adelaide, Australia, 2 December 2020; pp. 273–278.

- Poursepanj, H.; Weissbock, J.; Inkpen, D. uottawa: System description for semeval 2013 task 2 sentiment analysis in twitter. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, 14–15 June 2013; Association for Computational Linguistics: Stroudsburg, PA, USA, 2013; pp. 380–383.
- 69. Jamadi Khiabani, P.; Basiri, M.E.; Rastegari, H. An improved evidence-based aggregation method for sentiment analysis. *J. Inf. Sci.* 2020, *46*, 340–360. [CrossRef]