electronics

MDPI

Article

Knowledge-Guided Heterogeneous Graph Convolutional Network for Aspect-Based Sentiment Analysis

Xiangxiang Song, Guang Ling *, Wenhui Tu and Yu Chen

School of Science, Wuhan University of Technology, Wuhan 430070, China; songxiang@whut.edu.cn (X.S.); tuwenhui@whut.edu.cn (W.T.); 261376@whut.edu.cn (Y.C.)
* Correspondence: guangling@whut.edu.cn

Abstract: The purpose of aspect-based sentiment analysis (ABSA) is to determine the sentiment polarity of aspects in a given sentence. Most historical works on sentiment analysis used complex and inefficient methods to integrate external knowledge. Furthermore, they fell short of completely utilizing BERT's potential because when trying to generate word embeddings, they merely averaged the BERT subword vectors. To overcome these limitations, we propose a knowledge-guided heterogeneous graph convolutional network for aspect-based sentiment analysis (KHGCN). Specifically, we consider merging subword vectors utilizing a dynamic weight mechanism in the BERT embedding layer. Additionally, heterogeneous graphs are constructed to fuse different feature associations between words, and graph convolutional networks are utilized to identify context-specific syntactic features. Furthermore, by embedding a knowledge graph, the model can learn additional features from sources other than the corpus. Based on this knowledge, it is consequently possible to obtain more knowledge representation for a particular aspect by utilizing the attention mechanism. Last but not least, semantic features, syntactic features, and knowledge are dynamically combined using feature fusion. Experiments on three public datasets demonstrate that our model achieves accuracy rates of 80.87%, 85.42%, and 91.07%, which is an improvement of more than 2% compared to other benchmark models based on HGCNs and BERT.

Keywords: ABSA; HGCN; knowledge graph; feature fusion

check for updates

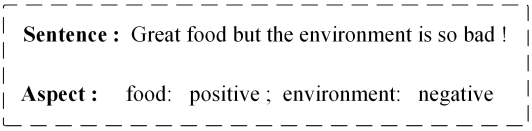
Citation: Song, X.; Ling, G.; Tu, W.; Chen, Y. Knowledge-Guided Heterogeneous Graph Convolutional Network for Aspect-Based Sentiment Analysis. *Electronics* **2024**, *13*, 517. https://doi.org/10.3390/electronics13030517

Academic Editor: Cecilio Angulo

Received: 29 December 2023
Revised: 21 January 2024
Accepted: 25 January 2024
Published: 26 January 2024

1. Introduction

Aspect-based sentiment analysis (ABSA) is a crucial factor in sentiment analysis and has become an increasingly popular subject in natural language processing research [1,2]. ABSA explores the aspect vocabulary's sentiment polarity (positive, neutral, or negative), given the sentence and the aspect vocabulary. As an example, when someone states "Great food, but the environment is so bad!", the sentiment polarities for the aspects of food and environment are opposite, as seen in Figure 1. In this regard, ABSA outperforms sentence-level sentiment analysis in determining a certain aspect's polarity [3].



Sentence : Great food but the environment is so bad !

Aspect : food: positive ; environment: negative

Figure 1. ABSA of the statement "Great food, but the environment is so bad!".

Neural networks have been utilized in a majority of the initial ABSA investigations to extract sentiment information associated with specific aspects within a textual context [4–7]. Subsequently, ABSA model architectures using an attention-based mechanism [8] and a pre-trained model [9] have become popular approaches [10,11]. BERT is instrumental in transforming input text into a more nuanced semantic representation. Simultaneously, the attention mechanism is crucial for highlighting a sentence's pertinent viewpoint that pertains to a particular aspect, focusing on contextual relationships. Despite the fact that models founded on pre-training and attention mechanisms have demonstrated commendable classification accuracies in ABSA tasks, a notable limitation arises in the simplistic methodology of averaging subword vectors to create word-level embeddings in the application of BERT [12,13]. This approach potentially restricts the semantic representation capability of BERT.

Numerous studies have underscored that in the context of ABSA, it is important to consider both the syntactic dependencies and semantic interactions between aspect and context words [14,15]. These studies highlight the significance of the syntactic dependency tree, which encapsulates syntactic information and is encoded using a graph convolutional network (GCN). This encoding effectively bridges aspect words with corresponding opinion words in a syntactically coherent manner. However, the use of a solitary dependency tree graph presents a limitation. It does not fully harness the latent information embedded within a sentence, nor does it capitalize on the robust feature fusion capacity inherent in GCNs. Modeling the enhanced syntactic dependency tree by adding additional nodes (such as sentences or knowledge) through a heterogeneous graph convolutional network (HGCN) can effectively improve the problem of a single structured dependency tree [16]. Despite these advancements, most models employing HGCNs encounter a critical limitation: the vectors for aspect words and context words generated by the intermediate hidden layers are typically averaged to form explicit representations of the additional nodes [17,18]. This procedure unintentionally results in an absence of potentially significant sentiment features, thereby impinging upon the efficacy and performance of the ABSA model.

Knowledge graphs have emerged as a powerful tool for infusing external knowledge into neural network models, significantly augmenting the ability of these models to comprehend semantic textual information. This integration of external knowledge is particularly beneficial in enhancing semantic features within ABSA tasks. Researchers have employed external knowledge to enrich the semantic dimensions of these tasks, primarily by utilizing words (identified as aspect nodes within sentences) in the knowledge graph as foundational seed nodes. These seed nodes establish connections with context nodes within the graph. Despite the substantial performance improvements achieved through these methods [19–21], it is posited that they do not entirely leverage the full spectrum of features offered by external knowledge. A critical concern is that potential features may be lost during the process of transposing them into the graph as nodes. Additionally, the construction of a knowledge subgraph for each sentence tends to be a complex and intricate task.

To address the issues stated above, we propose a new model: the KHGCN. The encoding layer involves feeding a concatenation of the context and aspect vocabulary into the coding layer. The model obtains semantic and syntactic knowledge through BiLSTM and HGCN layers, respectively. To further enrich the constructed graph features, sentiment external knowledge is introduced during the graph construction phase. The knowledge embedding matrices for context and aspect are obtained through low-dimensional continuous embedding. These embeddings are then integrated with the semantic representations derived from the BiLSTM layer. Subsequently, aspect-oriented knowledge representation is obtained through an attention mechanism. The semantic, syntactic, and knowledge feature representations thus captured are subsequently fused through a feature fusion layer. This fusion facilitates the prediction of sentiment classification, leveraging the combined strengths of semantic understanding, syntactic structure, and external knowledge integration. The primary contributions of our work can be summarized as follows:

(1) We propose a new knowledge-guided heterogeneous graph convolutional network for aspect-based sentiment analysis. Through the utilization of BiLSTM, HGCN, and external knowledge, the model incorporates multifaceted features of semantics, syntax, and additional knowledge.

(2) A dynamic weighting mechanism is proposed to address the underutilization of BERT and the inconsistency between BERT and GCN disambiguation in previous ABSA tasks. Sentence and aspect nodes, as well as their connection weights, are explicitly

defined and enhanced with external sentiment knowledge when constructing the heterogeneous graph.

(3) We also introduce external affective knowledge in a different manner, obtaining knowledge embeddings for both the aspect and context to individually capture affective information corresponding to specific aspects.

The remaining portions of this paper are structured as follows. Section 2 summarizes some previous research that is relevant to our work. Section 3 outlines the structure of our model: the KHGCN. Section 4 presents and explains the experimental results. Section 5 summarizes our findings.

2. Related Works

2.1. Aspect-Based Sentiment Analysis

In the realm of ABSA, current methodologies primarily employing neural networks begin by analyzing the contextual information within a text. These methods concentrate on identifying crucial emotional cues to ascertain the polarity corresponding to particular aspect concepts [4–7,22,23]. With the goal of providing an accurate aspect representation, Majumder et al. [4] utilized the memory network to add specific information close to the aspect words. Gandhi et al. [6] used conditional random fields and bi-directional LSTM to extract aspectual terms from text and model their sentiment. Utilizing an attention-based approach, a multi-granular attention mechanism was implemented by Zhu et al. [24] with the aim of enhancing the dependence among aspects and words of opinion. Additionally, Xue and Li [22] utilized a gating mechanism in a gated CNN model to output sentiment information exclusively, in accordance with a specified aspect. To further restrict the adverse effects of words unrelated to aspect and constrain the propagation of information, Zhao et al. [25] constructed an aspect-oriented weighting mechanism and proposed a GCN model incorporating multiple weighting techniques.

2.2. Graph Convolutional Network

There is a growing recognition of the limitations inherent in sequential models, particularly their neglect of syntactic relationships between sentences. The significance of syntactic relationships in comprehending and assessing sentiments in ABSA makes this error noteworthy. Graph network-based ABSA models have been developed at an impressive rate in the past few years, showing heightened potential for extracting and interpreting syntactic relationships [26–29]. A heterogeneous graph neural network was proposed by Zhang et al. [30], which enriches the created graph characteristics with information from various node types and connectivity interactions. The model has shown good application in link prediction [31], node classification [32], and personalized recommendation [33]. A heterogeneous graph convolutional network sentiment classification model was presented by Zhang et al. [18]. It prunes the dependency tree and lessens the effects of plume knowledge regarding the outcomes. Therefore, we believe that it is very meaningful to utilize an HGCN for ABSA.

2.3. Considering External Knowledge

Deep learning models are progressively incorporating outside knowledge, especially in the realm of natural language processing [34–37]. This trend underscores the significant role that both linguistic and general knowledge play in enhancing the comprehension of natural language. In the context of ABSA, where the primary objective is to analyze and interpret sentiment, the incorporation of external sentiment knowledge into models is especially advantageous. SenticNet7 [37] is an excellent public resource for categorizing sentiment and mining opinions and has performed very well in sentiment analysis tasks [38]. Liang et al. [39] employed a SenticNet-enhanced dependency graph and utilized contextual sentiment knowledge to enhance sentiment categorization efficiency. In constructing the graph, Xu et al. [40] incorporated sentiment knowledge and amalgamated information from the latent semantic graph and the enhanced dependency graph dynamically by employing a gating mechanism. By incorporating additional sentiment nodes into the heterogeneous graph, Zhang et al. [18] calculated the similarity among nodes of aspect and nodes of knowledge to determine the connection weights. When introducing sentiment knowledge, the above model simply calculates the sentiment score values of related words to enhance the dependency graph and does not fully utilize the sentiment vector representations in SenticNet. Thus, based on the rich sentiment knowledge in SenticNet, for aspect words and context, we extract an embedded matrix of the sentiment information by utilizing the 300,000-concept affective knowledge space.

2.4. Limitations

The related works mentioned above simply average the subword vectors to form word-level embeddings when implementing BERT, which limits the powerful semantic representation capability of BERT and also affects the performance of the ABSA model. Furthermore, the previous works do not comprehensively consider semantic, syntactic, and external knowledge features. There are very few studies that consider all three features at the same time. When conducting syntactic feature extraction, isomorphic graphs alone do not fully utilize the powerful feature fusion capabilities of the GCN. It is also necessary to account for the impact of additional nodes on the syntactic structure. Additionally, the incorporation of external knowledge is too simple to fully exploit the rich sentiment information in the affective space.

3. Methodology

We present a detailed illustration of the construction of heterogeneous graphs and the KHGCN in this section. In Figure 2, an overview of the proposed model is shown.

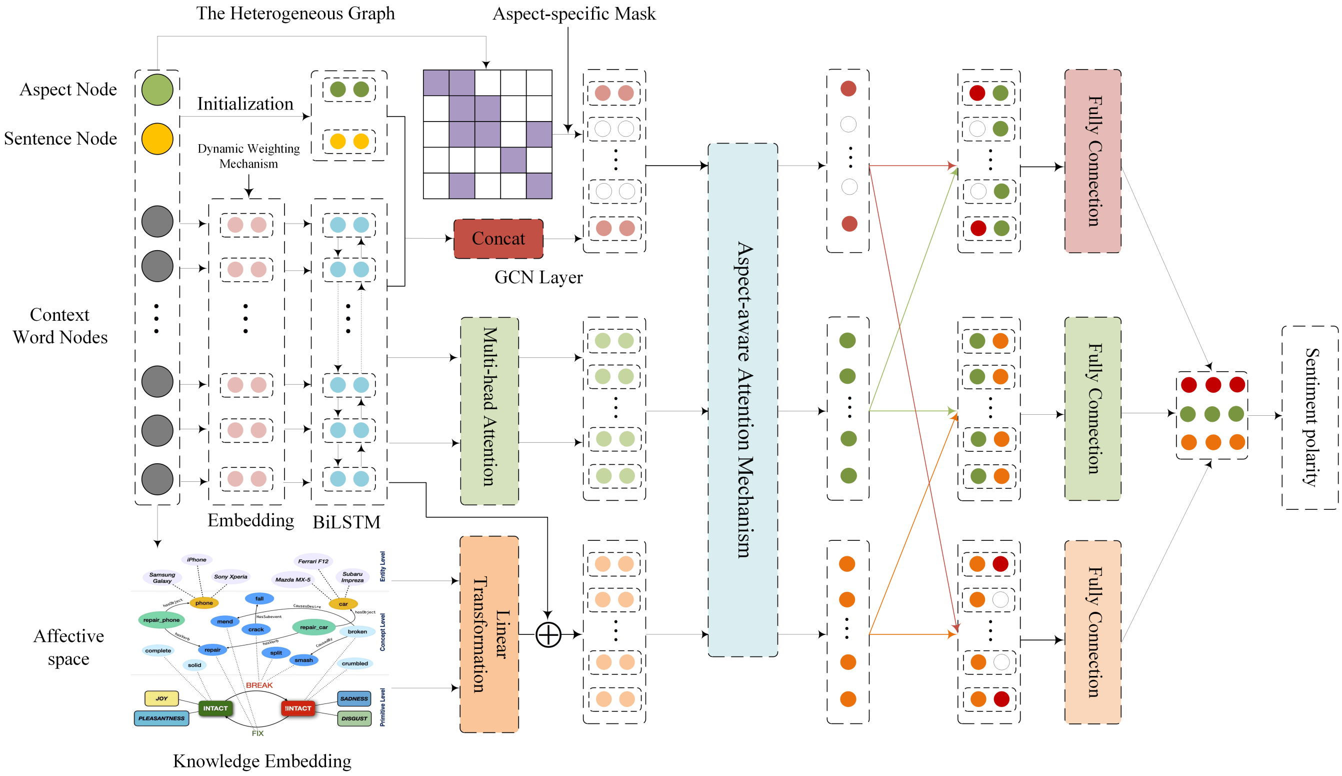


Figure 2. The detailed architecture of the KHGCN.

3.1. Problem Description

The purpose of ABSA is to determine the sentiment polarity of aspects in a given sentence. Assuming a sentence containing n words, $S = \{w_1, w_2, \ldots, w_{n-1}, w_n\}$, and its subsequence of $A = \{w_\tau, w_{\tau+1}, \ldots, w_{\tau+m-1}\}$ represents an aspect in S. In this task, the aspect terms are labeled as Y = {0, 1, 2}, representing negative, neutral, and positive aspects, respectively.

3.2. Embedding Based on BERT

Word embeddings are acquired by applying BERT, which encodes each component word into a high-dimensional vector. BERT [9] has demonstrated notable efficacy in the domain of contextual representation learning. The sequence of inputs is constructed in the form of $[CLS] + S + [SEP] + A + [SEP]$. BERT's internal tokenizer is used to segment input sequences into subword sequences, $\tilde{S} = \{[CLS], w_1^1, w_1^2, w_2, \ldots, w_n^j, [SEP]\}$, where $\{w_t^1, \ldots, w_t^j\}$ represents the subword sequence of the word w_t. Therefore, the representation of the input sequence is $\tilde{E} = \{e_{[CLS]}, \tilde{e}_1^1, \tilde{e}_1^2, \tilde{e}_2, \ldots, \tilde{e}_n^j, e_{[SEP]}\}$, where $e \in {}^{d_{bert}}$, d_{bert} represents the size of the hidden dimensions.

Dynamic Weighting Mechanism

Inconsistent tokens arise from the differences in word segmentation methods between BERT and the GCN [41,42]. Take, for example, the following sentence: "He hates playing games". BERT generates tokens such as "[CLS]", "He", "hates", "play", "##ing", and "games", ".", "[SEP]". However, the GCN generates tokens such as "He", "hates", "playing", "games", and ".". This is due to the WordPiece word segmentation strategy in BERT that splits "playing" into two subwords: "play" and "##ing". The importance of different subwords within a text surely fluctuates. For the purpose of addressing the problem of inconsistent word segmentation and improving the effectiveness of the BERT pre-trained model, we introduce a dynamic weighting mechanism to amalgamate subwords, as delineated in Algorithm 1. For an input sequence, we first obtain the index sequence of subwords divided into $[[idx(e_1^1), idx(e_1^2)], \ldots, [idx(e_n^1), \ldots, idx(e_n^j)]]$. The corresponding embedding vector should not change if the word is divided consistently in both BERT and the GCN. When the subwords are divided inconsistently between the two, the subword weight within the original word is determined utilizing the exponential function. Next, the subword weights are normalized and weighted to obtain the embedding vectors of the words.

Algorithm 1 Dynamic weighting mechanism

```
Input:
Subword index list of each word in a sentence:
$L = [[idx(e_1^1), idx(e_1^2)], \ldots, [idx(e_n^1), \ldots, idx(e_n^j)]]$
embedding vector sequence: $\tilde{E} = \{e_{[CLS]}, \tilde{e}_1^1, \tilde{e}_1^2, \tilde{e}_2, \ldots, \tilde{e}_n^j, e_{[SEP]}\}$
Output:
Weighted embedding vector sequence: $E = [e_{[\mathrm{CLS}]}, e_1, e_2, \ldots, e_n, e_{[\mathrm{SEP}]}]$
for $i = 0$ to $n - 1$ do
    if $\mathrm{len}(L(i)) = 1$ then
        $e_i = \tilde{e}_i$
    else
        $weight = []$
        for $j = 0$ to $\mathrm{len}(L(i)) - 1$ do
            $weight[j] = e^{-\alpha j}$    where $e$ is a natural constant and $\alpha$ is a decay factor
        end for
        $e_i = \tilde{e}_i \times \mathrm{softmax}(weight)$
    end if
    $i = i + 1$
end for
```

3.3. BiLSTM Layer

BiLSTM is commonly applied to text and time-series data. It can be used to capture token context-dependent information, including forward and backward LSTM [6].

After the embedding layer, we obtain the context's embedding vector matrix $E^s = \{e_1^s, e_2^s, \ldots, e_n^s\}$, as well as the aspect word's embedding vector matrix $E^a = \{e_\tau^a, e_{\tau+1}^a, \ldots, e_{\tau+m-1}^a\}$. These matrices are then input into independent BiLSTM layers. The context feature representation corresponds to the hidden state $H^s = \{h_1^s, h_2^s, \ldots, h_n^s\}$, where $H^s \in {}^{n \times 2d_h}$, and the representation of the corresponding aspect is $H^a = \{h_\tau^a, \ldots, h_{\tau+m-1}^a\}$, where $H^a \in {}^{m \times 2d_h}$, and d_n is the hidden layer dimension of BiLSTM.

3.4. Detailed Description of the Heterogeneous Graph

$G(V, E, T_v, T_e, \phi, \psi)$ is the representation of a heterogeneous graph, where V represents a node set and E represents an edge set. The mapping of each node to its appropriate node type is represented by ϕ: $\phi_v : V \to T_v$. ψ represents mapping each node to its corresponding edge type: $\psi_e : E \to T_e$. In our work, each word in a sentence $\{w_1, w_2, \ldots, w_{\tau+m-1}, \ldots, w_n\}$ is a node in the graph, and $|T_v| = 3$, $|T_e| = 4$. Apart from word nodes, our framework incorporates two supplementary node categories: aspect nodes, denoted as t, and sentence nodes, denoted as s. Below, we sequentially delineate the pertinent node types alongside the inter-nodal relational structures:

(1) DT_{ij}: Employing the spaCy toolkit, we ascertain the syntactic dependencies inherent within each proffered sentence. The syntactic dependency tree's interrelations serve as the edges, whereas the associated lexical tokens constitute the nodes, thereby forging the connective topology.

$$T_{ij} = \begin{cases} 1 & \text{if } w_i, w_j \text{ have a syntactic dependency} \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We incorporate knowledge from SenticNet and integrate emotional knowledge into syntactic dependencies using the sentiment scores from SenticNet by making use of the affective common sense information between aspect words and context. Initially, we determine each lexical group's sentiment score based on syntactic dependencies:

$$Score_{i,j} = Sentic(w_i) + Sentic(w_j) \tag{2}$$

where $Sentic(w_i) \in [-1, 1]$ represents the sentiment score in SenticNet. Conforming to the processing methodology delineated in [39], we derive the final syntactic matrix as follows:

$$DT_{ij} = \begin{cases} T_{ij} \times (\mathrm{Score}_{ij} + 2) & \text{if } w_i \text{ or } w_j \text{ in } A \\ T_{ij} \times (\mathrm{Score}_{ij} + 1) & \text{else} \end{cases} \tag{3}$$

(2) $TF\text{-}IDF$: The $TF\text{-}IDF$ metric is utilized to determine the relative importance of every word within a document or corpus. The model has the capacity to decrease the impact of words that contribute only slightly to the meaning of the sentence by employing the TF-IDF metric to calculate the connective weight connecting a word with a sentence. Consequently, this facilitates a model-centric emphasis on terms that furnish more substantive contributions.

(3) The mutual indication relationship between aspects and sentences, which has been shown to perform well in the ABSA task, is represented by an edge across the aspect node t and the sentence node s.

(4) $DWIN_{ij_{aspect}}$: This illustrates how word nodes affect aspect nodes' sentiment polarity and how aspect nodes affect word nodes. Typically, a word node's influence on an aspect increases with its closeness to the aspect. Thus, by adjusting the window size, you can determine whether and to what degree the word node affects the aspect node. The window is dynamically set in this article according to the length of the input text. The

precise implementation approach, along with its magnitude and degree of influence, is as follows:

$$DWIN_{ij_{\text{aspect}}} = \begin{cases} e^{-\beta|j_{\text{aspect}}-i|} & \text{if } i \in (j_{\text{aspect}} - kn, j_{\text{aspect}} + kn) \\ 0 & \text{else} \end{cases} \tag{4}$$

where n is the text length, k controls the window size, β is the attenuation factor that controls the degree of contribution, and k and β are hyperparameters, which can be set separately according to different datasets.

We represent the relationships in this heterogeneous graph utilizing the adjacency matrix $A \in {}^{(n+2)(n+2)}$. In mathematical terms, the weight of the edge is determined as follows:

$$A_{ij} = \begin{cases} DT_{ij} & i, j \text{ are words} \\ TF - IDF & i \text{ is } s, j \text{ is word} \\ DWIN_{ij} & i \text{ is word}, j \text{ is t} \\ 1 & i \text{ is } s, j \text{ is } t \\ 0 & \text{others} \end{cases} \tag{5}$$

3.5. Graph Convolutional Networks

Through the multiple relationships mentioned above, given a sentence, we can construct a heterogeneous graph $G(V, E, T_v, T_e, \phi, \psi, A)$, where A represents the adjacency matrix. To create a new node representation vector, $H' = \{h_1^s, h_2^s, \ldots, h_n^s, s, t\}$, the representation vectors for the additional nodes, s and t, are combined with the context node vector. Subsequently, the heterogeneous graph adjacency matrix A alongside the nodal vectors H' are assimilated as inputs to the GCN, wherein the graph convolution operation is harnessed to encode the locoregional nodal information. Through the graph convolution process, each node incrementally accrues efficacious information from its immediate neighbors with each convolutional iteration. By augmenting the number of graph convolution layers, a node's capacity to garner a broader informational ambit from its neighboring nodes is enhanced, thereby enabling the efficacious learning of node-specific feature representations. Drawing upon the insights of extant scholarship [43,44], we posit that a monolithic layer in a GCN is insufficient for the acquisition of a comprehensive informational spectrum from neighboring nodes. Conversely, a multi-layered GCN approach inflates the model's complexity. Consequently, our work advocates for a two-layer GCN.

Graph convolution is utilized to update each node's representation in a GCN layer. The detailed equations are as follows:

$$h_i^l = \text{Re}LU(\sum_{j=1}^{n} c_{ij} A_{ij} (W^l h_j^{l-1} + b^l)) \tag{6}$$

$$c^i = \frac{1}{D_{ii}} \tag{7}$$

$$D_{ii} = \sum_{j=1}^{n} A_{ij} \tag{8}$$

where D represents the degree matrix, c^i represents the normalization constant, and W^l and b^l represent learnable parameters.

3.6. Aspect-Specific Mask

The enhanced aspect feature vector can be acquired by applying the zero-mask layer following the GCN layer. Only the aspect nodes' feature representation remains after the mask adjusts the representation of non-aspect nodes to 0.

$$H^{mask} = \{0, \ldots, h^s_\tau, \ldots, h^s_{\tau+m-1}, \ldots, 0, t\} \tag{9}$$

3.7. Affective Knowledge Graph

The smallest significant units of language are referred to as words. However, many words or phrases themselves have profound personal significance. For example, the word "disastrously" typically conveys a negative sentiment in a sentence. We present a new approach that utilizes the embedding of knowledge and introduces the affective knowledge space from SenticNet7 [37] to better leverage the rich sentimentality of the words themselves.

SenticNet7 constructs a three-level concept primitive knowledge representation, as shown in Figure 3, by first extracting concept primitives from the text and then connecting the concept primitives, the public knowledge concept layer, and the command entity layer. The 300,000 concepts inside SenticNet7. In addition to providing conceptual-level representation, assigning semantics and sentiment provides semantic information and sentiment for individual words.

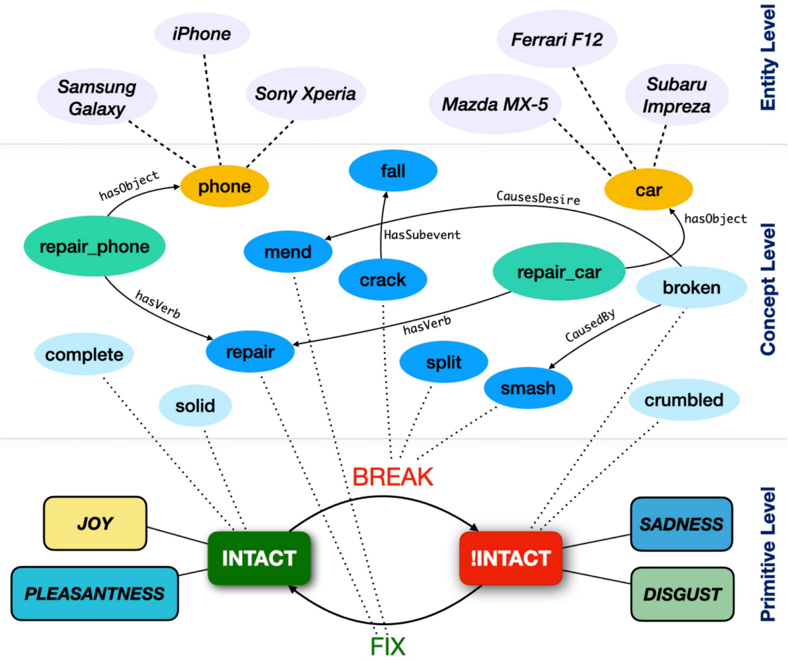


Figure 3. A sketch of the SenticNet 7 semantic network of three-level knowledge representation.

The affective knowledge space maintains the relationship between semantics and sentiment by mapping the SenticNet7 concepts to the continuous low-dimensional embeddings. Consequently, we are able to apply the matrix E_{Aff} to obtain the affective knowledge embeddings of the input text S and aspect words A.

$$H^s_{Aff} = \{h_1^{s_{aff}}, h_2^{s_{aff}}, \ldots, h_n^{s_{aff}}\} \tag{10}$$

$$H^a_{Aff} = \{h_\tau^{a_{aff}}, h_{\tau+1}^{a_{aff}}, \ldots, h_{\tau+m-1}^{a_{aff}}\} \tag{11}$$

Since H_{Aff} is a 100-dimensional embedding vector, we first transform it into the dimensions of the BiLSTM layer's hidden state utilizing a linear transformation. Then, we integrate affective knowledge into a feature representation vector matrix as the final knowledge graph's embedding through matrix addition:

$$H^s_{Aff} = Linear(H^s_{Aff}) + H^s \tag{12}$$

$$H^a_{Aff} = Linear(H^a_{Aff}) + H^a \tag{13}$$

We obtain the external knowledge representation of the text utilizing this approach. With this foundation, we can employ external affective knowledge to efficiently determine the sentiment polarity of aspect words in a phrase.

3.8. Attention Layer

In this layer, to extract long-term dependencies from context and capture the interactivity of aspect words and context, we utilize a multi-head attention. Aspect-specific contextual information is updated through the aspect-aware attention mechanism.

3.8.1. Multi-Headed Attention

The self-attention mechanism is responsible for learning long-term dependencies from the input sequence. We first project the input sequence to the representations, Q , K , V, and then calculate the attention. The formula is as follows:

$$Q, K, V = Linear_Q(H^s), Linear_K(H^s), Linear_V(H^s) \tag{14}$$

$$head_i = Attention(Q_i, K_i, V_i) = soft\max(\frac{Q_i {K_i}^T}{\sqrt{d_k}})V_i \tag{15}$$

$$MultiHead(H^s) = Concat(head_i, \ldots, head_h)W^O \tag{16}$$

where Q_i, K_i, V_i represent the i-th attention head, W^O is the learnable parameter of the multi-head attention output, and the remaining linear transformations are parameters that can be learned.

3.8.2. Aspect-Aware Attention Mechanism

To extract meaningful sentiment features from the representations of particular aspects, we implement an aspect-aware attention mechanism, which is consistent with the work in [45]. In this research, we apply this mechanism sequentially to semantic, syntactic, and knowledge representations. The calculation procedure is detailed below. The calculations for semantic representation are as follows:

$$\theta_t = \sum_{i=1}^{n} (multihead(h^s_t))^T h^a_i \tag{17}$$

$$Z^c = \sum_{t=1}^{n} softmax(\theta_t) h^s_t \tag{18}$$

where h^s_t and h^a_i are obtained through the BiLSTM layer. The calculations for syntactic representation are as follows:

$$\theta_t = \sum_{i=1}^{n} h^s_t h^{mask}_i \tag{19}$$

$$Z^s = \sum_{t=1}^{n} softmax(\theta_t) h^s_t \tag{20}$$

where h^{mask}_i is the aspect-specific output obtained through the zero-mask layer. The calculations for knowledge representation are as follows:

$$\theta_t = \sum_{i=1}^{n} h_{aff,t}^{s} h_{aff,i}^{a} \tag{21}$$

$$Z^k = \sum_{t=1}^{n} soft\max(\theta_t) h_{aff,t}^{s} \tag{22}$$

where $h_{aff,t}^{s}$ and $h_{aff,i}^{a}$ are the knowledge embedding representations of S and A, respectively.

3.9. Feature Fusion

It would be difficult to take advantage of the complementarity between the representations if they were immediately fused by gathering the corresponding representation from the embedding of semantics, syntactic, and affective knowledge. As a result, we use the fusion strategy described in [45]. During the local fusion process, three types of features are first connected in pairs: $[Z^c; Z^s]$, $[Z^c; Z^k]$, and $[Z^s; Z^k]$. Then, the connected representations are submitted to independent fully connected layers to yield the expected sentiment features: Z^{cs}, Z^{ck}, and Z^{sk}. In order to achieve the sentiment prediction probability, we merge them globally after the first splicing by feeding them into a 3 × 3 convolutional layer.

3.10. Sentiment Classification

Ultimately, as the last sentiment prediction, we take the output of the feature fusion layer, $\hat{y}$, and utilize a cross-entropy loss as guidance for training:

$$Loss = -\sum_{i} y_i \log(\hat{y}) + \lambda \|\Theta\| \tag{23}$$

where y_i is the real label, and $\hat{y}$ is the label of the prediction.

4. Experiments

4.1. Datasets and Experimental Details

Three public benchmark datasets—Lap14 [46], Rest15 [47], and Rest16 [48]—were utilized for our research. The statistics for each polarity of the dataset are shown in Table 1. To show the datasets in more detail, the statistics on the sentence lengths are shown in Table 2. Most of the sentence lengths are distributed in the range of 10–30, and there are fewer sentences greater than a length of 50. A more intuitive distribution of the sentence lengths can be seen in Figure 4. In our experiments, the depth of the GCN layer was set to 2, the coefficient of the L2 regularization term was set to 0.00001, and α in Algorithm 1 was chosen to be 0.3. The hidden state vectors had a dimensionality of 768. The dimension of the word embeddings was adjusted to 768 based on the pre-trained BERT-base-uncased model, and the model parameters were optimized and updated using the Adam optimizer. We fixed the learning rate to 0.00002 and the batch size to 32, utilizing Xavier [49] to initialize the parameters. Dropout was then applied to the word embeddings, and the dropout rate was chosen to be 0.5 in order to prevent overfitting. The KHGCN model was trained on an RTX 3080Ti, 12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40 GHz, and the model was built using PyTorch 2.0.0.

Table 1. Statistics of the datasets.

Dataset		Positive	Neutral	Negative	Total
Lap14	Train	994	464	870	2328
	Test	341	169	128	638
Rest15	Train	912	36	256	1204
	Test	326	34	182	542
Rest16	Train	1240	69	439	1748
	Test	469	30	117	616

Table 2. The length distributions of the datasets.

Length		0–10	10–20	20–30	30–40	40–50	$\geq$50
Lap14	Train	233	901	676	322	116	80
	Test	109	316	135	38	27	13
Rest15	Train	248	583	260	75	26	12
	Test	121	228	111	45	29	8
Rest16	Train	372	811	370	120	55	20
	Test	134	274	107	54	16	31

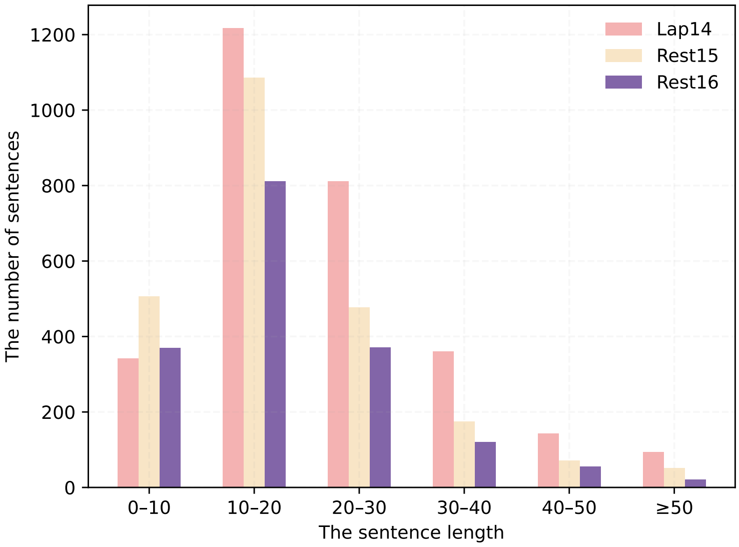


Figure 4. Histogram of sentence length statistics on the three datasets.

4.2. Evaluation Metrics

In the present study, we employed the same accuracy (*Acc.*) and $F1$ score metrics from the previous ABSA model. Accuracy can be utilized to evaluate how effectively the model performs overall in terms of classification by quantifying the percentage of correctly categorized samples to all samples. In statistics, the accuracy of a two-classification model is shown by the $F1$ value. It considers the classification model's recall rate as well as accuracy. When processing data that are unbalanced by category, the *Macro-*$F1$ score is a highly beneficial metric because it automatically averages the $F1$ score for each category. TP represents the number of true positives, FP represents the number of false positives, FN represents the number of false negatives, and TN represents the number of true negatives. The formulas for these metrics are as follows:

$$Acc. = \frac{TP + TN}{TP + TN + FN + FP} \tag{24}$$

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

$$Recall = \frac{TP}{TP + FN} \tag{26}$$

$$Macro\text{-}F1 = \frac{1}{|Y|} \sum \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{27}$$

4.3. Baseline

We contrasted our model with other baseline models used for the ABSA task, which are detailed below:

- MemNet [50] employs a multi-hop architecture, context processing, and external memory.
- AOA [51]: Adapts the attention-over-attention technique from machine translation to the realm of ABSA.
- IAN [52]: Constructs representations for aspects and contexts independently after interactively learning attentions in aspects and contexts.
- ASGCN [14]: Applies a GCN to sentence dependency trees to take advantage of syntactic information.
- AEGCN [53]: Utilizes an improved GCN that combines phrase dependency trees with multi-head attention.
- SK-GCN [19]: Utilizes a mixed syntax and external knowledge model that successfully integrates external knowledge with syntax information.
- AHGCN [16]: Employs a new GCN-based model using heterogeneous graphs.
- ADHGCN [18]: Utilizes a new heterogeneous graph construction method, which adds post-pruning on top of the traditional construction method.
- Sentic LSTM [54]: Employs a common-sense knowledge solution for directed sentiment analysis of aspect words.
- Sentic-GCN [39]: Incorporates SenticNet into a model that makes full use of syntactic relationships and sentiment common sense to enhance dependency trees.
- DM + GCN + BERT [55]: Performs dynamic and multi-channel GCN modeling of syntactic and semantic information in sentences.
- SGGCN + BERT [56]: Alters the graph-based model's hidden vectors to make the most of information from the aspects.
- AIEN + BERT [57]: Constructs an interaction encoder using a GCN and attention mechanisms for extracting interaction features.
- KHGCN: Utilizes a dynamic weighting mechanism to acquire word-level embeddings during the encoding phase. It employs BiLSTM, an HGCN, and affective space to obtain semantic, syntactic, and external sentiment features, respectively. Additionally, it utilizes an attentional mechanism to extract features for sentiment prediction.

4.4. Performance Comparison

The experimental results, as indicated in Table 3, demonstrate that our proposed KHGCN performed better overall compared to all other models examined on the three benchmark datasets.

In particular, the proposed KHGCN performed noticeably better than earlier homomorphic and heteromorphic graph models based on GCNs (ASGCN, AEGCN, SK-GCN, AHGCN, andADHGCN), verifying the effectiveness and feasibility of utilizing emotional knowledge encoding as well as heteromorphic graphs for sentence dependency enhancement. Poorer results were achieved for the macro-F1 scores on the Rest16 dataset compared to the excellent GCN-based model (Sentic-GCN). In addition, for the BERT-based model, our proposed KHGCN outperformed the vanilla BERT model on the three public datasets. Although it was not as competitive as the SGGCN + BERT model on the Lap14 dataset, both the Acc. and macro-F1 values exhibited significant improvements over this model on the Rest15 and Rest16 datasets, demonstrating the effectiveness of the dynamic weighting mechanism we proposed in the BERT encoding stage. The results show that our proposed knowledge-guided heterogeneous graph convolutional network approach is effective.

Table 3. Performance comparison. (The best result for each model is shown in bold, whereas the second-best result is underlined).

Model	**Lap14 (%)**		**Rest15 (%)**		**Rest16 (%)**	
	Acc.	**Macro-F1**	**Acc.**	**Macro-F1**	**Acc.**	**Macro-F1**
MemNet [50]	70.64	65.17	77.31	58.28	85.44	65.99
AOA [51]	72.62	67.52	78.17	57.02	87.5	66.21
IAN [52]	72.05	67.38	78.54	52.65	84.74	55.21
ASGCN [14]	74.14	69.24	79.34	60.78	88.69	66.64
AEGCN [53]	75.91	71.63	79.95	60.879	87.39	68.22
SK-GCN [19]	73.20	69.18	80.12	60.70	85.17	68.08
AHGCN [16]	76.80	73.00	79.94	62.79	88.53	72.18
ADHGCN [18]	78.52	76.21	85.16	63.77	88.53	71.94
Sentic-LSTM [54]	70.88	67.19	79.55	60.56	83.01	68.22
Sentic-GCN [39]	77.90	74.71	82.84	67.32	90.88	**75.91**
BERT [9]	77.59	73.28	83.48	66.18	90.10	74.16
DM + GCN + BERT [55]	80.22	77.28	N/A	N/A	N/A	N/A
SK-GCN + BERT [19]	79.00	75.57	83.20	66.78	87.19	72.02
SGGCN + BERT [56]	**82.80**	**80.20**	82.72	65.86	90.52	74.53
AIEN + BERT [57]	78.21	73.39	83.58	64.67	90.58	74.49
KHGCN	80.87	77.90	**85.42**	**68.90**	**91.07**	74.65

4.5. Ablation Study

The detailed results of the ablation experiments are shown in Table 4. The performance of the model without sentiment knowledge encoding (w/o Z^k) was unsatisfactory on all datasets, which indicates that our way of encoding sentiment knowledge provided the corresponding sentiment information for the contexts and aspects. Meanwhile, after removing the semantic and syntactic branches (w/o Z^c, w/o Z^s), the model's performance decreased by varying degrees. We can conclude that the BiLSTM and HGCN layers were effective components for extracting semantic and syntactic features. In addition, the experiment without DWM yielded worse experimental results overall compared to the benchmark tests, again confirming that the integration of encoded information using the dynamic weighting mechanism in the BERT encoding phase was effective.

Table 4. Ablation study results (%). "DWM" represents the dynamic weighting mechanism, "Z^c" represents semantic features, "Z^s" represents syntactic features, and "Z^k" represents affective knowledge).

Model	**Lap14 (%)**		**Rest15 (%)**		**Rest16 (%)**	
	Acc.	**Macro-F1**	**Acc.**	**Macro-F1**	**Acc.**	**Macro-F1**
KHGCN w/o DWM	80.09	76.93	85.23	67.72	90.75	73.93
KHGCN w/o Z^c	79.31	75.78	84.69	65.20	90.42	72.14
KHGCN w/o Z^s	80.25	76.84	84.31	65.80	90.09	71.75
KHGCN w/o Z^k	80.25	76.90	85.24	66.96	90.42	73.45
KHGCN	80.87	77.90	85.42	68.90	91.07	74.65

4.6. Parameter Experiment

Since the KHGCN model uses a dynamic window mechanism to construct the heterogeneous graphs, we investigated how this mechanism's hyperparameters, k and β, impacted the model's performance, as illustrated in Table 5. The optimal Acc. and macro-F1 values on the Lap14 and Rest16 datasets were achieved with $k = 0.3$ and $\beta = 0.3$, the optimal Acc. value on Rest15 was achieved with $k = 0.3$ and $\beta = 0.4$, and the optimal macro-F1 value on Rest15 was achieved with $k = 0.2$ and $\beta = 0.6$. The reason for this result may be that in the Rest15 dataset, there are large differences in the data distribution models of different categories.

4.7. Complexity Analysis

In this subsection, we have chosen AHGCN, Sentic-GCN, and AIEN+BERT to compare the number of parameters and the runtimes of the models, and the results are shown in Table 6. Since the data pre-processing, heterogeneous graph generation, and external knowledge embedding only need to be performed once, the generated embedding matrix can be recycled. Therefore, this part was ignored when counting the number of parameters and the runtimes of the models. As we can observe from Table 6, the number of parameters in our model was significantly higher than those in AHGCN and Sentic-GCN, and slightly lower than those in AIEN+BERT. In terms of time complexity, we were able to reach convergence faster on the Rest16 dataset, and the runtimes of the different models for the remaining two datasets were not significantly different. Thus, the model in this paper sacrifices space complexity while performing better in terms of the Acc. and macro-F1 metrics.

Table 5. Performance using different values k and β.

k	β	Lap14 (%)		Rest15 (%)		Rest16 (%)	
		Acc.	Macro-F1	Acc.	Macro-F1	Acc.	Macro-F1
0.2	0.3	80.56	77.62	85.24	68.98	90.91	74.51
	0.4	80.41	77.57	85.06	68.43	90.75	74.24
	0.5	80.41	77.03	84.87	67.03	90.26	74.05
	0.6	80.72	77.77	84.87	69.19	90.56	73.58
0.3	0.3	80.87	77.90	85.05	67.72	91.07	74.65
	0.4	80.40	77.74	85.42	68.90	90.26	74.09
	0.5	80.56	77.15	85.23	65.14	91.07	73.66
	0.6	80.25	77.04	85.06	69.08	90.58	72.30

Table 6. Runtime and number of parameters of the model (time represents the time for each epoch to achieve convergence).

Model	Params	Dataset	Time/s
AHGCN	44.09 M	Lap14	197.37
		Rest15	169.61
		Rest16	321.98
Sentic-GCN	44.09 M	Lap14	274.78
		Rest15	204.82
		Rest16	264.57
AIEN + BERT	132.36 M	Lap14	212.21
		Rest15	133.93
		Rest16	215.92
KHGCN	103.16 M	Lap14	227.04
		Rest15	273.08
		Rest16	205.90

4.8. Discussion

The proposed model performs well, as can be seen from the comparison of the experimental results above. Table 4 shows that feature extraction modules, whether semantic, syntactic, or external information, play an invaluable role in text feature extraction. This demonstrates that the ABSA model can incorporate multiple simultaneous feature extraction modules for text, which is beneficial for enhancing classification accuracy. This paper proposes a dynamic weighting mechanism for using BERT to embed words to address the problem of inconsistency between the GCN model and the BERT model for word segmentation. The experimental results also verify the effectiveness of this method, which provides a solvable idea for subsequent research. Although external knowledge embedding introduces additional complexity, it is a one-time process, and the resulting embeddings can be reused. Furthermore, the aforementioned modules can be utilized independently as branches of the ABSA model for feature extraction, which is highly scalable, providing greater convenience for subsequent researchers. The model proposed in this paper can be beneficial in domains that require sentiment analysis, such as online shopping, where sellers can analyze the likes and dislikes of various types of products through customer reviews and better adjust the shelving strategy of the products. Another area is social media monitoring, which can help companies track user sentiment on social platforms and understand the public's views on specific issues, products, or events.

4.9. Case Study

In this section, to better analyze how decisions are made within the KHGCN, the specific decision-making process is presented using the test example "The staff should be a bit more friendly". The results are shown in Figure 5. In the text semantic extraction stage, the model notices the key negative expression "should be". In the semantic structure stage, the model focuses more on the connection between the sentence nodes and the aspect word "staff". Simultaneously, the model learns the words with obvious emotional information, such as "bit" and "friendly", through external knowledge embedding. By combining the above three stages of sentiment extraction, the KHGCN accurately predicts the sentiment polarity of the aspect "staff".

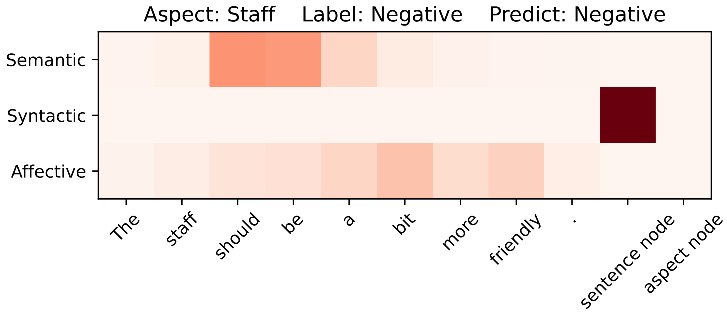


Figure 5. Attention visualization for sentiment analysis example.(Darker color represents higher attention scores).

5. Conclusions

In response to the complexity and inefficiency of the traditional literature on integrating external knowledge and generating word embeddings, which often merely average BERT subword vectors, we propose a knowledge-guided heterogeneous graph convolutional network for aspect-based sentiment analysis to address these limitations. In concrete terms, we propose a dynamic weight mechanism for merging subword vectors in the BERT embedding layer. In addition, the model can acquire additional information by embedding the knowledge graph, signifying its ability to utilize the attention mechanism to generate aspect-oriented knowledge representations. And finally, feature fusion is utilized to dynamically combine the feature representations of semantic, syntactic, and knowledge aspects. The experimental results demonstrate that our proposed KHGCN performs better overall on three benchmark datasets compared to all other models examined.

Author Contributions: Conceptualization, X.S.; methodology, X.S. and W.T.; software, X.S.; validation, X.S. and Y.C.; formal analysis, T.W.; investigation, Y.C. and X.S.; resources, X.S.; data curation, X.S. and W.T.; writing—original draft preparation, X.S.; writing—review and editing, G.L.; visualization, X.S. and W.T.; supervision, G.L.; project administration, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article. Lap14: https://alt.qcri.org/semeval2014/task4/ (accessed on 24 January 2024); Rest15: http://alt.qcri.org/semeval2015/task12/ (accessed on 24 January 2024); Rest16: http://alt.qcri.org/semeval2016/task5/ (accessed on 24 January 2024).

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

References

1. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
2. Schouten, K.; Frasincar, F. Survey on aspect-level sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 813–830. [CrossRef]
3. Arya, B.; Bassi, B.; Phiyega, R. Transformation charters in contemporary South Africa: The case of the ABSA group limited. *Bus. Soc. Rev.* **2008**, *113*,227–251. [CrossRef]
4. Majumder, N.; Poria, S.; Gelbukh, A.; Akhtar, M.; Cambria, E.; Ekbal, A. AIARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3402–3411.
5. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
6. Gandhi, H.; Attar, V. Extracting aspect terms using CRF and bi-LSTM models. *Procedia Comput. Sci.* **2020**, *167*, 2486–2495. [CrossRef]
7. Yang, J.; Yang, J. Aspect based sentiment analysis with self-attention and gated convolutional networks. In Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 16–18 October 2020; pp. 146–149.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; p. 30.
9. Devlin, J.; Chang, MW.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
10. Hoang, M.; Bihorac, O.; Rouces, J. Aspect-based sentiment analysis using bert. In Proceedings of the 22nd Nordic Conference on Computational Linguistics, Turku, Finland, 30 September–2 October 2019; pp. 187–196.
11. Zhu, X.; Zhu, Y.; Zhang, L.; Chen, Y. A BERT-based multi-semantic learning model with aspect-aware enhancement for aspect polarity classification. *Appl. Intell.* **2023**, *53*, 4609–4623. [CrossRef]
12. Li, X.; Fu, X.; Xu, G.; Yang, Y.; Wang, J.; Jin, L.; Liu, Q.; Xiang, T. Enhancing BERT representation with context-aware embedding for aspect-based sentiment analysis. *IEEE Access* **2020**, *8*, 46868–46876. [CrossRef]
13. Park, H.; Shin, K. Aspect-Based Sentiment Analysis Using BERT: Developing Aspect Category Sentiment Classification Models. *J. Intell. Inf. Syst.* **2020**, *26*, 1–25.
14. Zhang, C.; Li, Q.; Song, D. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv* **2019**, arXiv:1909.03477.
15. Hou, X.; Qi, P.; Wang, G.; Ying, R.; Huang, J.; He, X.; Zhou, B. Graph ensemble learning over multiple dependency trees for aspect-level sentiment classification. *arXiv* **2021**, arXiv:2103.11794.
16. Xu, K.; Zhao, H.; Liu, T. Aspect-specific heterogeneous graph convolutional network for aspect-based sentiment classification. *IEEE Access* **2020**, *8*, 139346–139355. [CrossRef]
17. Zeng, Y.; Li, Z.; Chen, Z.; Ma, H. Aspect-level sentiment analysis based on semantic heterogeneous graph convolutional network. *Front. Comput. Sci.* **2023**, *17*, 176340. [CrossRef]
18. Zhang, Z.; Hu, C.; Pan, H.; Wang, Y.; Xu, Y. Aspect-Dependent Heterogeneous Graph Convolutional Network for Aspect-Level Sentiment Analysis. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
19. Zhou, J.; Huang, J.; Hu, Q.; He, L. Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. *Knowl.-Based Syst.* **2020**, *205*, 106292. [CrossRef]
20. Zhao, A.; Yu, Y. Knowledge-enabled BERT for aspect-based sentiment analysis. *Knowl.-Based Syst.* **2021**, *227*, 107220. [CrossRef]
21. Zhang, K.; Liu, Q.; Qian, H.; Xiang, B.; Cui, Q.; Zhou, J.; Chen, E. Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 377–389. [CrossRef]
22. Xue, W.; Li, T. Aspect based sentiment analysis with gated convolutional networks. *arXiv* **2018**, arXiv:1805.07043.
23. He, R.; Lee, W.; Ng, H.; Dahlmeier, D. Exploiting document knowledge for aspect-level sentiment classification. *arXiv* **2018**, arXiv:1806.04346.
24. Zhu, Z.; Zhang, D.; Li, L.; Li, K.; Qi, J.; Wang, W.; Zhang, G.; Liu, P. Knowledge-guided multi-granularity GCN for ABSA. *Inf. Process. Manag.* **2023**, *60*, 103223. [CrossRef]
25. Zhao, Z.; Tang, M.; Tang, W.; Wang, C.; Chen, X. Graph convolutional network with multiple weight mechanisms for aspect-based sentiment analysis. *Neurocomputing* **2022**, *500*, 124–134. [CrossRef]

26. Huang, B.; Carley, K. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv* **2019**, arXiv:1909.02606.
27. 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 (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5679–5688.
28. Pan, Y.; Li, D.; Dai, Z.; Cui, P. Aspect-Based Sentiment Analysis Using Dual Probability Graph Convolutional Networks (DP-GCN) Integrating Multi-scale Information. In Proceedings of the International Conference on Neural Information Processing, Changsha, China, 20–23 November 2023; Springer: Singapore, 2023; pp. 495–512.
29. Zhang, Z.; Zhou, Z.; Wang, Y. SSEGCN: Syntactic and Semantic Enhanced Graph Convolutional Network for Aspect-based Sentiment Analysis. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2022, Seattle, WA, USA, 10–15 July 2022; pp. 4916–4925.
30. Zhang, C.; Song, D.; Huang, C.; Swami, A.; Chawla, N. Heterogeneous graph neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 793–803.
31. Wang, X.; Chai, Y.; Li, H.; Wu, D. Link prediction in heterogeneous information networks: An improved deep graph convolution approach. *Decis. Support Syst.* **2021**, *141*, 113448. [CrossRef]
32. Zhao, J.; Wang, X.; Shi, C.; Hu, B.; Song, G.; Ye, Y. Heterogeneous Graph Structure Learning for Graph Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 4697–4705.
33. Qin, H.; Han, X.; Ma, X.; Yan, W. Personalized literature recommendation based on heterogeneous entity academic network. *J. King Saud-Univ.-Comput. Inf. Sci.* **2023**, *35*, 101649. [CrossRef]
34. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced language representation with informative entities. *arXiv* **2019**, arXiv:1905.07129.
35. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-BERT: Enabling Language Representation with Knowledge Graph. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 2901–2908.
36. Xing, Y.; Shi, Z.; Meng, Z.; Lakemeyer, G.; Ma, Y.; Wattenhofer, R. Km-bart: Knowledge enhanced multimodal bart for visual commonsense generation. *arXiv* **2021**, arXiv:2101.00419.
37. Cambria, E.; Liu, Q.; Decherchi, S.; Xing, F.; Kwok, K. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 3829–3839.
38. Xing, F.; Pallucchini, F.; Cambria, E. Cognitive-inspired domain adaptation of sentiment lexicons. *Inf. Process. Manag.* **2019**, *56*, 554–564. [CrossRef]
39. Liang, B.; Su, H.; Gui, L.; Cambria, E.; Xu, R. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowl.-Based Syst.* **2022**, *235*, 107643. [CrossRef]
40. Xu, J.; Yang, S.; Xiao, L.; Fu, Z.; Wu, X.; Ma, T.; He, L. Graph Convolution over the Semantic-syntactic Hybrid Graph Enhanced by Affective Knowledge for Aspect-level Sentiment Classification. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
41. Xu, H.; Shu, L.; Yu, P.; Liu, B. Understanding pre-trained bert for aspect-based sentiment analysis. *arXiv* **2020**, arXiv:2011.00169.
42. Wu, Z.; Ong, D. Context-guided bert for targeted aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence 2021, Virtually, 2–9 February 2021; pp. 14094–14102.
43. Kipf, T.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
44. 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; pp. 7370–7377.
45. Zhong, Q.; Ding, L.; Liu, J.; Du, B.; Jin, H.; Tao, D. Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 10098–10111. [CrossRef]
46. Manandhar, S. Semeval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014.
47. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. Semeval-2015 task 12: Aspect based sentiment analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 486–495.
48. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
49. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
50. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *arXiv* **2016**, arXiv:1605.08900.
51. Huang, B.; Ou, Y.; Carley, K. M. Aspect level sentiment classification with attention-over-attention neural networks. In Proceedings of the Social, Cultural, and Behavioral Modeling: 11th International Conference, Washington, DC, USA, 10–13 July 2018; pp. 197–206.
52. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. *arXiv* **2017**, arXiv:1709.00893.

53. Xiao, L.; Hu, X.; Chen, Y.; Xue, Y.; Gu, D.; Chen, B.; Zhang, T. Targeted sentiment classification based on attentional encoding and graph convolutional networks. *Appl. Sci.* **2020**, *10*, 957. [CrossRef]
54. 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]
55. Pang, S.; Xue, Y.; Yan, Z.; Huang, W.; Feng, J. Dynamic and multichannel graph convolutional networks for aspect-based sentiment analysis. In Proceedings of the ACM Conference Findings Association for Computational Linguistics Joint Conference Natural Language Processing , Online, 1–6 August 2021; pp. 2627–2636.
56. Veyseh, A.; Nour, N.; Dernoncourt, F.; Tran, Q.; Dou, D.; Nguyen, T. Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. *arXiv* **2020**, arXiv:2010.13389.
57. Yang, B.; Li, H.; Teng, S.; Sun, Y.; Xing, Y. Attentional Interactive Encoder Network Focused on Aspect for Sentiment Classification. *Electronics* **2023**, *12*, 1329. [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.