A Multiple Graphical Convolution Networks Approach for Aspect-Based Sentiment Classification

Diana Kazakova, Flavius Frasincar(✉)[0000−0002−8031−758X], and Jasmijn Klinkhamer

Erasmus University Rotterdam, PO Box 1738, NL-3000 DR, Rotterdam, the Netherlands
kazakovadiana432@gmail.com, frasincar@ese.eur.nl, 584059jk@eur.nl

Abstract. Aspect-Based Sentiment Classification (ABSC) has attracted a lot of attention in recent years. Some of the best performing models for ABSC are the hybrid ones and the ones based on deep learning. This work presents a model for ABSC dubbed HAABSA4GCN, as an extension to the state-of-the-art hybrid model HAABSA++. The HAABSA4GCN model combines a domain ontology from HAABSA++ with a novel backup model based on the Graph Convolution Neural (GCN) network, dubbed 4GCN. This backup model uses four graphs: syntactic, semantic, lexical, and ontological. For the evaluation, the SemEval 2015 and SemEval 2016 datasets are used. The evaluation results show that HAABSA4GCN outperforms HAABSA++, suggesting that the combination of different graphs within the model increases the prediction accuracy. In addition, 4GCN performs better than HAABSA4GCN, implying that 4GCN is better in dealing with cases that the domain ontology can decide upon.

Keywords: Aspect-based sentiment classification · Graph convolution neural network · Domain ontology

1 Introduction

In the information age, individuals freely express their opinions about various topics, including products, media, and people. Approximately 47% of consumers actively share their thoughts and experiences through reviews on websites like Google, Yelp, Facebook, and TripAdvisor, enriching the digital landscape with user-generated content [9]. This surge in online feedback provides businesses with valuable insights into customer opinions, aiding in informed decision-making for product development, leading to better product quality, and market competitiveness. By monitoring brand positioning and managing customer satisfaction, businesses can better understand market perceptions and make necessary adjustments to enhance overall customer satisfaction.

To gain insights from textual data, the most valuable field is sentiment analysis. This involves a natural language processing task to categorise text with a sentiment score that reflects the author's opinion, emotion, or attitude, typically as positive, neutral, or negative. Sentiment can be computed at three levels: document, sentence, or aspect. Document level analysis assigns a single sentiment to the entire text, sentence level evaluates each sentence separately, and Aspect-Based Sentiment Analysis (ABSA) assesses sentiment towards specific entities or aspects within the text [20].

ABSA encompasses two main phases: Aspect Detection (AD), which identifies aspects, and Aspect-Based Sentiment Classification (ABSC), which determines the sentiment related to these aspects [20]. The primary goal of ABSC is to associate aspects with their corresponding opinions, a challenging task requiring contextual understanding. Recent efforts [7] use attention mechanisms to prioritise crucial text sections for sentiment analysis, capturing subtle word relationships. However, these mechanisms do not inherently distinguish between aspects. Hierarchical attention mechanisms have been introduced to address this [11]. [23] proposed the HAABSA++ model that significantly enhanced sentiment prediction. Despite this, the complexity of language can still pose challenges for accurate ABSC.

Graph Neural Networks (GNNs) are proven to effectively handle the complexity of assigning opinions to specific targets, considering the challenges presented by language [28]. However, GNNs have limitations, such as ignoring dependency types and varying encoding techniques that can affect model stability [30]. To address these issues, different graphs are used in GNNs: syntactic (Syn), semantic (Sem), lexical (Lex), and knowledge (Kno) graphs.

Recent studies have developed several models that make use of GNNs for ABSA. The Relational Graph Attention Network (R-GAT) model uses a syntactic graph [26]. The Dual Graph Convolutional Network (DualGCN) leverages syntactic and semantic graphs [12]. The dependency connections and the usage of dependency types are combined in the T-GCN model [22]. Lexical and semantic graphs are covered in the Bi-level interactive Graph Convolution Network (BiGCN) [29]. Last, an ontological graph is used in [13] where the relationship between an opinion/aspect word and words from the domain ontology is analysed and exploited for ABSA.

This study aims to enhance the effectiveness of the HAABSA++ model in addressing ABSC by combining the syntactic, semantic, lexical, and ontological graphs in a GCN model. We replace the LCR-Rot-hop++ backup model of the HAABSA++ model with 4GCN, a novel graph-based neural model using the four GCN modules operating in parallel, resulting in the new hybrid model HAABSA4GCN. The Python source code and data are available on GitHub at https://github.com/DianaKazakova/HAABSA4GCN.

The rest of the paper is organised as follows. Section 2 discusses existing related research. Then, Sect. 3 presents the used data. Section 4 explains the methods used in this work, including our newly proposed method. The evaluation is presented in Sect. 5. Last, Sect. 6 provides the conclusion with suggestions for future research.

2 Related Works

ABSC is a subtask of ABSA addressing the determination of the sentiment polarity of an aspect target word in a given sentence. [2] classifies the present ABSC models into knowledge-based, machine learning, and hybrid. Hybrid models combine the knowledge-based approach with machine learning methods. HAABSA++ [23] is such a model and has achieved state-of-the-art results in the class of hybrid models. First, we present graph neural networks, as a special class of machine learning methods that have been recently successfully used for ABSA. Then, we describe hybrid models.

2.1 Graph Neural Networks

Neural networks designed to process graph structures, and pioneered by [19], were described as a variant of RNNs. [27] divides the GNNs into four distinct types: recurrent GNNs, convolutional GNNs, graph autoencoders, and spatial-temporal GNNs. Among these diverse categories, two particular types are popular and effective: GCNs and GATs.

GCN, introduced by [10], builds on spectral graph convolutional neural networks [5]. [10] addresses the lack of self-loops and the use of unnormalised adjacency matrices, by modifying the adjacency matrix to include self-loops and applying normalisation to the adjacency matrix to preserve feature vectors scales. However, GCN approaches depend on the graph structure, hence a trained model can not be directly applied to another graph. [24] introduced GATs, which incorporates an attention mechanism to give different importance levels to neighbouring features, improving feature aggregation [8]. In ABSC tasks, the attention mechanism is also used for the GCN models [4]. However, ABSC tasks employing GCN or GAT models, often face limitations like, a lack of utilisation of semantic relationships and sentiment knowledge among words or phrases. These limitations can be handled with the help of graph modules.

Several works have used a text classification model based on GCN, which constructs the word relationship graph using word co-occurrence and word-text relationships [14]. BiGCN [29] combines syntactic and lexical graph modules to group lexical word pairs based on their co-occurrence frequency. [14] constructs a text graph tensor describing semantics, grammar, context, and other information. [31] used a semantic module in the GCN model to capture the semantic dependence between multiple aspects and used an attention mechanism to encode aspects and context.

[26] presents R-GAT to handle sentences containing multiple aspects, acknowledging the significance of the dependency relation type for the preservation of syntactical information. Similarly a Type-Aware Graph Convolution Network (T-GCN) addresses the oversight of dependency types, targeting a more comprehensive integration of dependency structures [22]. Moreover, the DualGCN model addresses parsing inaccuracies and informal expressions using a syntactic module and identifies semantic relationships using a semantic module [12]. In addition, BiCGN leverages lexical and semantic information [29].

However, most models based on graph networks overlook the incorporation of domain ontology information during graph construction [3]. External affective commonsense knowledge can enrich sentiment feature representations in sentiment analysis tasks [17]. An ontological graph is used in [13] where the relationships between an opinion/aspect words and words from the domain ontology are analysed.

2.2 Hybrid Models

Hybrid approaches for ABSC aim to combine the strengths of knowledge-based and machine-learning algorithms. [21] creates a specialised ontology for the restaurant sector, utilising it to compute the sentiment, with an SVM as a backup model. In more recent studies, neural networks have been utilised as a backup model. Recently, the domain-specific ontology of [21] was used with an attention-based model using hierarchical attention and deep contextualised word embeddings (BERT). This combination led to the development of HAABSA++, which includes the LCR-Rot-Hop++ neural network as the backup model [23].

HAABSA++ has achieved state-of-the-art results for the sentiment classification task. The process unfolds in distinct phases. Firstly, it uses the ontology from [21] to identify aspect-related sentiment. If the ontology provides inconclusive results due to neutral sentiments (not modelled in the ontology), conflicting sentiments, or lack of coverage, the LCR-Rot-hop++ neural network is activated. Combining the neural network with the ontology improves accuracy over using the neural network alone and enhances the interpretability of the results. HAABSA++ represents a significant advancement in the field of ABSC, skillfully detecting nuanced expressions of sentiment through semantic information and structural dependencies.

3 Data

For our research, the datasets SemEval 2015, Task 12, Subtask 2 [16] and SemEval 2016, Task 5, Subtask 1 [15] are used. Previous ABSA works have chosen SemEval datasets, indicating their reliability and relevance, and facilitating meaningful comparisons to our research. Besides, we use the same evaluation measures proposed by SemEval to maintain a fair comparison with the existing models. The datasets consist of reviews from various domains. Since our approach relies on an ontology designed for the restaurant domain, we focus only on the restaurant datasets. These consist of online reviews in English, annotated by human linguists, and formatted in an XML structure.

Each review comprises multiple sentences that each may contain aspects. In cases where a word in the sentence does not explicitly identify the aspect category, the target is marked as NULL.

As in this work we are doing a sentiment classification task, we consider only sentiment polarity datasets, where each observation consists of a sentence, the target aspect within the sentence, and its corresponding polarity—negative (-1), neutral (0), or positive (1). The opinions with implicit targets were excluded for this dataset.

Table 1 summarises the distribution of sentiment polarities in our datasets, highlighting the prevalence of positive sentiment and the rarity of neutral sentiment. Furthermore, both training sets show class imbalance. The SemEval 2016 dataset exhibits a slightly more similar distribution between training and test data than the SemEval 2015 dataset. However, the 2016 test set displays more imbalance than the 2015 test set, featuring fewer negative examples but more positive examples. Also, we can note that the 2016 dataset is larger than the 2015 dataset.

Table 1. The number of observations and the distribution of polarity (for the SemEval 2015 and SemEval 2016 datasets).

	Positive		Neutral		Negative		Total
	Frequency	%	Frequency	%	Frequency	%	Frequency
SemEval 2015 train	962	75.3	36	2.8	280	21.9	1278
SemEval 2015 test	353	59.1	37	6.2	207	34.7	597
Total (Av. for %)	**1315**	**67.2**	**73**	**4.5**	**487**	**28.3**	**1875**
SemEval 2016 train	1319	70.2	72	3.8	489	26.0	1880
SemEval 2016 test	483	74.3	32	4.9	135	20.8	650
Total (Av. for %)	**1802**	**72.25**	**104**	**4.35**	**624**	**23.4**	**2530**

4 Methodology

In this section, our methodology is presented. First, we introduce the reference HAABSA++ model. Then, we present our novel 4GCN model.

4.1 HAABSA++

The HAABSA++ model, developed by [23], first uses a rule-based method based on a domain sentiment ontology [21] to determine the sentiment of a given aspect. Then, the LCR-Rot-hop++ neural network model is employed for handling sentences for which the ontology is inconclusive.

The domain sentiment ontology predicts sentiment using predefined classes, class relations, and axioms. It has three main classes: SentimentValue, AspectMention, and SentimentMention [21]. SentimentValue assigns Positive and Negative subclasses to words or phrases expressing sentiments, excluding Neutral due to its intrinsic ambiguity. AspectMention represents mentions of specific features or attributes. SentimentMention categorises sentiment expressions into three types: Type-1 for general positive or negative sentiment, Type-2 for a particular category of aspect mentions, and Type-3 for aspect category dependent mentions. Furthermore, the classifier checks for negation and inverts the sentiment polarity if a certain concept is negated.

Nevertheless, the ontology encounters inconclusiveness in three scenarios: prediction of conflicting sentiments (both positive and negative) for a target, instances with no hits due to the limited coverage of the ontology, or the sentiment is neutral. To address these limitations, the LCR-Rot-hop++ model is used as a backup model.

The LCR-Rot-hop++ model was enhanced by [25] with multiple iterations of rotatory attention to better distinguish sentiment words. [23] further improved it using hierarchical attention and contextual word embeddings (BERT), leading to the refined LCR-Rot-hop++ model.

4.2 4GCN

4GCN is a GNN-based model using four GCN graphs: SynGCN, SemGCN, LexGCN, and KnoGCN. These graphs are used in parallel, each taking a BERT-encoded sentence as input and applying graph convolutions to produce updated hidden representations, incorporating syntactic, semantic, lexical, and domain-specific ontology information. The results of every graph, the modules' last hidden representations of the aspect terms, are then concatenated to obtain the final output representation. The resulting vector is reduced to three dimensions through a fully connected layer. Last, a softmax function is applied to compute the probabilities associated with each sentiment class. The class with the highest probability is selected as the final label.

Encoding. The initial phase of the neural network classifier involves transforming each sentence s to understand its hidden contextual meaning. This transformation is achieved by leveraging a pre-trained BERT model to generate encodings of sentences, as BERT has been proven to excel in these kinds of tasks [12, 23]. BERT uses the [CLS] and [SEP] tokens to derive aspect-aware hidden sentence representations. Each sentence is first tokenised into words or subwords. Then, BERT combines these tokens with two unique tokens: [CLS] at the beginning and [SEP] at the end of the sentence. Another [SEP] token is used to construct a sentence-aspect pair (s, t) as an input for our model. This process allows BERT to distinguish between the aspect of a sentence and the context where the polarity is expressed. The optimal encoding is shown in Eq. (1).

$$\text{[CLS] sentence [SEP] aspect [SEP]} \tag{1}$$

For instance, given the sentence "The selection of desserts was small", we input it into BERT-large to generate aspect-aware hidden representations $h_i^{(0)}$ of 1024 dimensions for each word i in sentence s (word embeddings are averages of token embeddings initially produced by BERT). The encoded input format is "[CLS] The *selection* of desserts is small [SEP] *selection* [SEP]", where the word *selection* was encoded as an aspect of the sentence. Moreover, sentences are extended with padding to achieve a uniform length of 100 tokens, ensuring consistent dimensions across all inputs.

Graph Convolutional Network. The fundamental structure of each network layer is expressed in Eq. (2). Each element A_{ij} in matrix A signifies the connection between the i-th and j-th nodes (nodes denote words). Specifically, A_{ij} equals 1 if the i-th node is linked to the j-th node, and 0 otherwise. Moreover, the adjacency matrix A, consisting of binary values (0s and 1s), can be the discrete output from a dependency parser. For the i-th node at the l-th layer, its hidden state representation, formally denoted as h_i^l, undergoes an update according to the following equation:

$$h_i^{(l)} = \sigma\left(\sum_{j=1}^{n} A_{ij} W^{(l)} h_j^{(l-1)} + b^{(l)}\right) \tag{2}$$

where n is the number of words in a sentence, $W^{(l)}$ is the weight matrix, $b^{(l)}$ is the bias term, and σ is an activation function.

This research employs the $ReLU$ activation function for the graph convolution layers. Each module creates a graph over s using a distinct approach: SynGraph for syntactic information, SemGraph for semantic relations, LexGraph for lexical dependencies, and KnoGraph for ontology relations. The associated adjacency matrices A are utilised to perform graph convolutions, following Eq. (2), for multiple layers.

SynGCN Module. To exploit syntactic relations in a sentence and construct the graph for the SynGCN module, the dependency parser is first used to obtain the dependency tree of every sentence [22]. In a dependency tree, each word is represented as a node, and the edges between nodes signify the syntactic dependencies. The root of the dependency tree, often an artificial symbol, represents the main governing element in the sentence.

The general results of a dependency tree can be represented by three elements $(x_i, x_j, r_{i,j})$, where $r_{i,j}$ is a dependency type between x_i word and x_j word. Moreover, the adjacency matrix $A = \{a_{ij}\}_{n\times n}$ is utilised to record the existence of an edge between x_i word and x_j word and n is the number of words in a sentence. Consequently, A is a 0-1 matrix where a_{ij} equals 1 if there is an edge between x_i word and x_j word and a_{ij} equals 0, otherwise. Additionally, the relation type matrix $R = {r_{ij}}_{n\times n}$ is created to obtain the dependency type between x_i and x_j words if there is an edge between them. Given the set of dependency relations R, the corresponding type embeddings e_{ij}^r are generated.

In evaluating the importance of an edge for sentiment polarity within a sentence, the weighting of edges by their contribution is proposed. In detail, for every edge connecting words x_i and x_j, the l-th GCN layer leverages the hidden vectors $h_i^{(l-1)}$ and $h_j^{(l-1)}$ derived from the (l-1)-th GCN layer for x_i and x_j words. The initial hidden vectors, $h_i^{(0)}$ and $h_j^{(0)}$, are obtained from the BERT encoder. Subsequently, hidden vectors are combined with the corresponding type embeddings e_{ij}^r. Equations (3) and (4) represent the concatenation process for the words x_i and x_j:

$$s_i^{(l)} = h_i^{(l-1)} \oplus e_{ij}^r \tag{3}$$

and

$$s_j^{(l)} = h_j^{(l-1)} \oplus e_{ij}^r \tag{4}$$

where $\oplus$ denotes the concatenation operator.

Next, a weight $p_{i,j}^{(l)}$ (contribution) of an edge for the sentiment polarity is computed with the following formula:

$$p_{ij}^{(l)} = \frac{a_{ij} \cdot exp\left(s_i^{(l)} \cdot s_j^{(l)}\right)}{\sum_{j=1}^{n} a_{ij} \cdot exp\left(s_i^{(l)} \cdot s_j^{(l)}\right)} \tag{5}$$

To transform the dimension of e_{ij}^r to $h_i^{(l-1)}$ we used trainable matrix $W_{\mathcal{R}}^{(l)}$ of the l-th GCN layer. Therefore, the type embeddings are transformed into the hidden representations by the following equation:

$$h_i^{(l-1)'} = h_i^{(l-1)} + W_{\mathcal{R}}^{(l)} \cdot e_{ij}^r \tag{6}$$

Lastly, the output of x_i at the l-th layer from the SynGCN module is computed by applying $p_{ij}^{(l)}$ to Eq. (2):

$$h_i^{(l)(sym)} = \sigma\left(\sum_{j=1}^{n} p_{ij}^{(l)} W^{(l)} h_j^{(l-1)'} + b^{(l)}\right) \tag{7}$$

where $W^{(l)}$ and $b^{(l)}$ denote the trainable parameters in the l-th GCN layer, and σ refers to the *ReLU* activation function.

SemGCN Module. The SemGCN module is constructed via a self-attention mechanism to analyse semantic associations. Self-attention can capture semantically related terms of each word in a sentence [12]. Computing the attention score of each pair of elements in parallel, the attention score matrix $A^{(l)(sem)} \in R^{n \times n}$ is constructed as follows:

$$A^{(l)(sem)} = softmax\left(\frac{Q^{(l)} W^Q \times \left(K^{(l)} W^K\right)^T}{\sqrt{d}}\right) \tag{8}$$

where $Q^{(l)}$ and $K^{(l)}$ are both equal to the hidden representation of the previous layer $H^{(l-1)}$. W^Q and W^K are learnable weight matrices, and d is the number of features for each node (word).

The semantic graph representation $H^{(l)(sem)}$ is obtained from the SemGCN module using Eq. (2), where $h_i^{(l)(sem)} \in R^d$ is a hidden representation of the i^{th} node at the l-th layer.

$$h_i^{(l)(sem)} = \sigma\left(\sum_{j=1}^{n} a_{ij}^{(l)(sem)} W^{(l)} h_j^{(l-1)} + b^{(l)}\right) \tag{9}$$

where $W^{(l)}$ and $b^{(l)}$ denote the trainable parameters in the l-th GCN layer, and σ refers to the *ReLU* activation function.

LexGCN Module. To employ a LexGraph to ABSC and represent the relationships between words quantitatively, a global co-occurrence matrix A^{gc} is constructed [29]. This global co-occurrence matrix A^{gc} is generated using a corpus combining the SemEval training sets with the WebText corpus. The WebText corpus, containing informal language texts, is sourced from NLTK [1] and is intentionally employed to augment the information in the SemEval training corpus. Equation (10) presents the global co-occurrence matrix:

$$A^{gc} = \{a_{ij}^{gc}\}_{N \times N} \tag{10}$$

where i and j denote indexes for x_i and x_j words in the corpus, respectively, and a_{ij}^{gc} denotes the co-occurrence frequency between x_i and x_j words in the corpus whose vocabulary size is N. The weights of a_{ij}^{gc} are scaled by the frequency of i and j in the training corpus. The scaling provides a more balanced representation of co-occurrence relationships, considering the importance of each word in the context of the entire training corpus.

The local co-occurrence matrix A^{lc} for each sentence s is constructed to delve into sentence-level analysis [29]. To reduce overemphasising specific sentence structures and avoid redundancy, up to one co-occurrence per x_i word is tallied within each sentence when constructing the local co-occurrence matrix. This matrix, acting as the adjacency matrix for the LexGCN module, is formulated using Eq. (11),

$$A_s^{lc} = \{a_{ij}^{lc}\}_{N \times N} \tag{11}$$

where i and j denote indexes for x_i and x_j words in the sentence, respectively, and a_{ij}^{lc} denotes the co-occurrence frequency between x_i and x_j words in the sentence.

To integrate global word distribution information from A^{gc} into the local lexical co-occurrence matrix A_s^{lc}, the co-occurrence weights of existing words in A^{gc} are maintained. Specifically, if the sentence co-occurring word x_i and word x_j are present in the training corpus of A^{gc}, a_{ij}^{lc} is set equal to scaled co-occurrence of word x_i with word x_j in the global co-occurrence matrix A^{gc}, otherwise, a_{ij}^{lc} $= 0$.

The output of the LexGraph module for x_i at the l-th layer is then obtained using Eq. (2). The lexical graph representation is presented as $H^{(l)(lex)}$, where $h_i^{(l)(lex)} \in R^d$ is a hidden representation of the i^{th} node at the l-th layer.

$$h_i^{(l)(lex)} = \sigma\left(\sum_{j=1}^{n} a_{ij}^{lc} W^{(l)} h_j^{(l-1)} + b^{(l)}\right) \tag{12}$$

where $W^{(l)}$ and $b^{(l)}$ denote the trainable parameters in the l-th GCN layer, and σ refers to the $ReLU$ activation function.

KnoGCN Module. In this work, the domain ontology of [25] serves as structured representations of knowledge, capturing relationships and dependencies regarding the restaurant domain. This ontology is employed to form the KnoGCN graph. Developing this graph involves the creation of a domain knowledge vocabulary $V = \{v_1, v_2, ..., v_s\}$, where s is the vocabulary size.

The process begins with identifying words recognised by the ontology, which are then utilised to construct a KnoGCN graph within the 4GCN model. Steps analogous to those outlined for the SynGCN module have been followed to construct the KnoGCN graph. These steps involve creating an adjacency matrix to represent semantic relations between words recognised by the ontology. Once the KnoGCN graph is built, we apply a similar approach to weigh the importance of edges for sentiment polarity. The weighting involves computing importance $p_{ij}^{(l)}$ of each edge, where l represents the GCN layer. The computation of $p_{ij}^{(l)}$ involves utilising the hidden vectors derived from the GCN layers, as well as the type embeddings associated with each edge within the KnoGCN graph.

Lastly, the output of each word at the l-th layer from the KnoGCN graph is computed using an adapted version of Eq. (6) utilised in the SynGCN graph. This output encapsulates the enhanced representation of each word, incorporating a domain-specific ontology to enrich sentiment analysis capabilities.

Decoding. A final output vector $\mathbf{o}$ is created by concatenating the last hidden layer of every graph constructed within the 4GCN model for aspect term a.

$$o_a = \underset{1024\times1}{h_a^{(L)(SynGCN)}} \oplus \underset{1024\times1}{h_a^{(L)(SemGCN)}} \oplus \underset{1024\times1}{h_a^{(L)}(LexGCN)} \oplus \underset{1024\times1}{h_a^{(L)(KnoGCN)}} \tag{13}$$

where L represents the last layer for the graph g, and g stands for SynGCN, SemGCN, LexGCN, or KnoGCN. $h_a^{(L)(g)}$ is a layer output with dimension 1024. a is the aspect term used within a sentence.

To decode the results and obtain a C-dimensional output, the vector $\mathbf{o}$ undergoes processing through a fully connected layer, yielding a three-component vector.

$$\underset{|C|\times1}{u_a} = \underset{|C|\times4096}{W} \cdot \underset{4096\times1}{o_a} + \underset{|C|\times1}{b} \tag{14}$$

where $C = 3$ for (positive, negative, neutral), a is the aspect term, W and b are the trainable matrix and the bias, respectively.

To make a sentiment polarity prediction for aspect term a, a softmax function to $\mathbf{u_a}$ is applied. It maps the $\mathbf{u_a}$ for each dimension to a probability.

$$\underset{|C|\times1}{u'_a} = softmax(\underset{|C|\times1}{u_a}) \tag{15}$$

Then, a sentiment prediction $\hat{y_a}$ is made as follows:

$$\underset{1\times1}{\hat{y_a}} = \arg\max \underset{|C|\times1}{u'_a} \tag{16}$$

To update model weights through backpropagation, the standard cross-entropy loss function is used:

$$\underset{1\times1}{L} = -\sum_a \underset{|C|\times1}{y_a} \times \log(\underset{|C|\times1}{u'_a}) + \lambda||\Theta||^2 \tag{17}$$

where y_a represents a vector containing the true sentiment value for the a-th aspect, u'_a denotes a softmax probability comprising the predicted sentiment for the a-th aspect, λ corresponds to the weight assigned to the L_2-regularization term, and Θ represents all trainable parameters.

5 Results

This section presents the obtained results for the proposed HAABSA4GCN hybrid model and existing models. First, we present a hyperparameter tuning process applied to all the models. Second, we consider the results obtained from HAABSA++ and LCR-Rot-hop++ and compare them with HAABSA4GCN and 4GCN results, respectively. In addition, we compare the results obtained from DualGCN, R-GAT, BiGCN, and T-GCN models with 4GCN model.

5.1 Implementation Details

The Stanza toolkit, as described by [18], implements the dependency parser of [6]. This work uses this dependency parser due to its rapid processing and high accuracy.

We randomly sample 20% of the training data to create a validation set. The validation set is used to select the model hyperparameters using a Tree-structured Parzen Estimator (TPE). The highest accuracy on the validation set highlights the best hyperparameters and relates to the final model.

The word embeddings are initialised using a pre-trained BERT-large uncased model with 1024 dimensions. To address overfitting, dropout rates of 0.2286 are applied to each graph g and the concatenated output o. Each graph is configured with two layers. The Adam optimiser has a learning rate of 0.000011 and a weight decay of 0.027. The 4GCN model undergoes 15 epochs with a batch size of 4.

5.2 Evaluation

The novel model has been evaluated using the test sets from the SemEval 2015 and SemEval 2016 restaurant datasets. In Table 2 the following comparisons are made. The HAABSA4GCN model was compared against the HAABSA++ model [23]. Furthermore, the backup models, 4GCN (HAABSA4GCN) and LCR-Rot-hop++ (HAABSA++), were compared against each other. Accuracy results for HAABSA++ and LCR-Rot-hop++ were reported from [23].

Interestingly, 4GCN achieves an even higher accuracy when used on the entire testing sample, outperforming HAABSA4GCN by 0.3 and 1.4 percentage points for 2015 and 2016, respectively. The ability of the 4GCN model to capture underlying relationships and predict sentiment surpasses that of LCR-Rot-hop++. This superiority is attributed to the unique combination of various modules within 4GCN, which enables it to capture a broader spectrum of relationships within the dataset. While 4GCN achieves a higher classification accuracy, the interpretability of the hybrid model's results remains valuable.

Table 2. Comparison of HAABSA4GCN and HAABSA++ and their respective backup models using classification accuracy.

	SemEval 2015	SemEval 2016
HAABSA++	81.7%	87.0%
HAABSA4GCN	**83.1**%	**87.7**%
LCR-Rot-hop++	81.1%	86.7%
4GCN	**83.4**%	**89.1**%

The novel hybrid model and proposed neural network model outperform previous results of HAABSA++ and LCR-Rot-hop++, respectively. HAABSA-4GCN outperforms HAABSA++ by 1.4 percentage points for the SemEval 2015 dataset and 0.7 percentage points for the SemEval 2016 dataset. 4GCN outperforms LCR-Rot-hop++ by 2.3 percentage points for the SemEval 2015 dataset and 2.4 percentage points for the SemEval 2016 dataset. The better classification accuracy of 4GCN compared to HAABSA4GCN indicates that the GCN model based on four different graphs is more effective at predicting examples classified by the ontology across the entire testing sample.

In addition to the analysis provided in Table 2, Table 3 presents a comparison of the prediction accuracy of the 4GCN model with the DualGCN, R-GAT, BiGCN, and T-GCN models from the literature (all GCN models).

Table 3. Comparison of 4GCN and DualGCN, R-GAT, BiGCN, and T-GCN models using classification accuracy.

	SemEval 2015	SemEval 2016
4GCN	**83.4**%	89.1%
DualGCN	81.6%	86.9%
R-GAT	81.2%	85.7%
BiGCN	81.1%	88.9%
T-GCN	83.0%	**90.3%**

In the SemEval 2015 dataset, 4GCN shows strong performance with an accuracy of 83.4%, proving that the introduced extensions have boosted the result. T-GCN follows closely with marginally lower accuracy of 83.0%, illustrating its strong ability to handle complex relationships within sentiment analysis using dependency relations. The DualGCN, R-GAT, and BiGCN models also perform well, achieving accuracy of 81.6%, 81.2%, and 81.1%, respectively. Although the HAABSA4GCN is better than HAABSA++, the 4GCN model outperforms HAABSA4GCN on the SemEval 2015 dataset.

Moving to the SemEval 2016 dataset, 4GCN reaches an accuracy of 89.1%. Except for the T-GCN model, this performance is significantly higher than its peers. The leading performance of 4GCN and T-GCN in this dataset is connected to the usage of the SynGCN graph. This graph presents the dependency relations in the adjacency matrix for prediction purposes. BiGCN, DualGCN, and R-GAT have lower accuracy, i.e., 88.9%, 86.9%, and 85.7%, respectively. The reasons for the higher accuracy of DualGCN and BiGCN are based on using two graphs for the prediction. DualGCN uses syntactic and semantic information, BiGCN uses syntactic and lexical information, and R-GAT uses only syntactic information.

Interestingly, every model achieves significantly higher accuracy for the SemEval 2016 dataset than the SemEval 2015 dataset. This disparity can be attributed to the more data available in the training set of the SemEval 2016 dataset, i.e., almost 1.5 times larger (Table 1). This consequently offers more robust data for model training purposes. In addition, it is notable that even though HAABSA4GCN is more accurate than HAABSA++ on the SemEval 2016 dataset, the 4GCN model surpasses HAABSA4GCN.

6 Conclusion

This research proposes two models for ABSC: HAABSA4GCN and 4GCN. The first is a hybrid, two-stage model with a domain-specific ontological classifier and a backup model, 4GCN, a novel approach based on GNN, employing four different graphs, specifically, a syntactic graph, a semantic graph, a lexical graph, and an ontological graph. Additionally, DualGCN, R-GAT, BiGCN, and T-GCN, four core GCN models, are used for evaluation purposes.

We compare HAABSA4GCN with HAABSA++ [23] to assess whether a GNN-based backup model could enhance performance over the LCR-Rot-hop++ model. Our findings indicate that the 4GCN and HAABSA4GCN models outperform the LCR-Rot-hop++ and HAABSA++ models, respectively. The prediction accuracy of HAABSA4GCN on the SemEval 2015 and SemEval 2016 datasets are 83.1% and 87.7%, respectively. In addition, 4GCN is better than HAABSA4GCN, with prediction accuracy of 83.4% and 89.1%, respectively. 4GCN also beats DualGCN, R-GAT, BiGCN, and T-GCN on the SemEval 2015 dataset and DualGCN, R-GAT, and BiGCN on the SemEval 2016 dataset.

As future work we suggest experimenting with alternative dependency parsers for the SynGCN module, as only the Stanza parser was used in this study. Still, others might provide better results. Additionally, constructing a global co-occurrence matrix from a large corpus of restaurant reviews could improve the reliability of the LexGCN module's adjacency matrix.

References

1. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: Analyzing text with the natural language toolkit. O'Reilly Media (2009)
2. Brauwers, G., Frasincar, F.: A survey on aspect-based sentiment classification. Computing Surveys **55**(4), 1–37 (2023)
3. Cambria, E., Hussain, A., Havasi, C., Eckl, C.: Common sense computing: From the society of mind to digital intuition and beyond. In: 2nd European Workshop on Biometrics and Identity Management (BioID 2009). LNIP, vol. 5707, pp. 252–259. Springer (2009)

4. Chen, J., Hou, H., Ji, Y., Gao, J.: Graph convolutional networks with structural attention model for aspect based sentiment analysis. In: 2019 International Joint Conference on Neural Networks (IJCNN 2019). pp. 1–7. IEEE (2019)
5. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: 30th International Conference on Neural Information Processing Systems. pp. 3844–3852. Curran Associates (2016)
6. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. In: 5th International Conference on Learning Representations (ICLR 2017). OpenReview.net (2017)
7. Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 3433–3442 (2018)
8. Gehring, J., Auli, M., Grangier, D., Dauphin, Y.: A convolutional encoder model for neural machine translation. In: 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017). pp. 123–135. ACL (2017)
9. Global Web Index: What percent of customers write online reviews? (2019), https://www.gwi.com/reports/product-reviews
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations (ICLR 2017). OpneReview.net (2017)
11. Li, L., Liu, Y., Zhou, A.: Hierarchical attention based position-aware network for aspect-level sentiment analysis. In: 22nd Conference on Computational Natural Language Learning (CoNLL 2018). pp. 181–189. ACL (2018)
12. Li, R., Chen, H., Feng, F., Ma, Z., Wang, X., Hovy, E.H.: Dual graph convolutional networks for aspect-based sentiment analysis. In: 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021). pp. 6319–6329. ACL (2021)
13. Liang, B., Su, H., Gui, L., Cambria, E., Xu, R.: Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. Knowledge-Based Systems **235**, 107643 (2022)
14. Liu, X., You, X., Zhang, X., Wu, J., Lv, P.: Tensor graph convolutional networks for text classification. In: 34th AAAI Conference on Artificial Intelligence (AAAI 2020). pp. 8409–8416. AAAI Press (04 2020)
15. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., Clercq, O.D., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N.V., Kotelnikov, E.V., Bel, N., Zafra, S.M.J., Eryigit, G.: SemEval-2016 task 5: Aspect based sentiment analysis. In: 10th International Workshop on Semantic Evaluation (SemEval 2016). pp. 19–30. ACL (2016)
16. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: SemEval-2015 task 12: Aspect based sentiment analysis. In: 9th International Workshop on Semantic Evaluation (SemEval 2015). pp. 486–495. ACL (2015)
17. Poria, S., Chaturvedi, I., Cambria, E., Bisio, F.: Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In: 2016 International Joint Conference on Neural Networks (IJCNN 2016). pp. 4465–4473. IEEE (2016)
18. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2020). pp. 101–108. ACL (2020)

19. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**, 61–80 (2008)
20. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **28**(3), 813–830 (2016)
21. Schouten, K., Frasincar, F.: Ontology-driven sentiment analysis of product and service aspects. In: 15th Extended Semantic Web Conference (ESWC 2018). LNCS, vol. 10843, pp. 608–623. Springer (2018)
22. Tian, Y., Chen, G., Song, Y.: Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In: 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), 2021. pp. 2910–2922. ACL (2021)
23. Trusca, M.M., Wassenberg, D., Frasincar, F., Dekker, R.: A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In: 20th International Conference Web Engineering (ICWE 2020). LNCS, vol. 12128, pp. 365–380. Springer (2020)
24. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations (ICLR 2018). OpenReview.net (2018)
25. Wallaart, O., Frasincar, F.: A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models. In: 16th Extended Semantic Web Conference (ESWC 2019). LNCS, vol. 11503, pp. 363–378. Springer (2019)
26. Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R.: Relational graph attention network for aspect-based sentiment analysis. In: 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). pp. 3229–3238. ACL (2020)
27. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**, 4–24 (2019)
28. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019). pp. 4567–4577. ACL (2019)
29. Zhang, M., Qian, T.: Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020). pp. 3540–3549. ACL (2020)
30. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). pp. 2205–2215. ACL (2018)
31. Zhao, P., Hou, L., Wu, O.: Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. Knowledge-Based Systems **193**, 105443 (2020)