From Graphs to Hypergraphs: Enhancing Aspect-Based Sentiment Analysis via Multi-Level Relational Modeling

Anonymous EMNLP submission

Abstract

Aspect-Based Sentiment Analysis (ABSA) demands nuanced modeling of complex aspect-sentiment interactions, a challenge amplified by the limited context in short texts. While graph-based methods have shown promise, they often fall short in capturing higher-order, multi-node relationships, leading them to construct multiple graphs that model fine-grained relationships inherent in language. However, such approaches suffer from poor generalization and increased parameter overhead. To overcome these limitations, we introduce HyperABSA, the first hypergraph-based approach to ABSA, which uniquely leverages a novel hypergraph construction method based on hierarchical clustering with a variance-sensitive threshold. This enables dynamic control over relational granularity via a acceleration based elbow criterion. This single hypergraph framework efficiently captures varying granularities of aspect-sentiment dependencies, while reducing parameter overhead, thereby simplifying prior approaches. Extensive experiments conducted on three public datasets (Lap14, Rest14 and MAMS) demonstrate the effectiveness of our proposed method.

1 Introduction

ABSA is a popular task within Natural Language Processing (NLP) that focuses on predicting the sentiment polarity of aspect terms within sentences. For instance, in the sentence *"Service is good although a bit in your face , we were asked every five mins if food was ok, but better that than being ignored"*, the aspects "service" and "food" reflect positive and neutral sentiments, respectively. This nuanced opinions in text is essential in domains like product reviews, customer feedback, and social media monitoring.

One of the key innovations in ABSA has been the integration of dependency trees (Poria et al., 2014; Chen et al., 2022), which capture syntactic relationships between aspect and opinions in text in a hierarchical manner. To further enhance these contextual dependencies, graph-based methods have emerged as a powerful paradigm (Kipf and Welling, 2016; Liang et al., 2020; Li et al., 2021; Zhang et al., 2022b; Tian et al., 2021; Bai et al., 2020). However, a fundamental limitation of these techniques lies in their inherent focus on pairwise relationships potentially overlooking more intricate, higher-order dependencies that are crucial for nuanced sentiment understanding (Battaglia et al., 2018). They also falter in managing varying granularities of relationships, resulting in reduced sensitivity between local and global dependencies, leading to suboptimal performance.

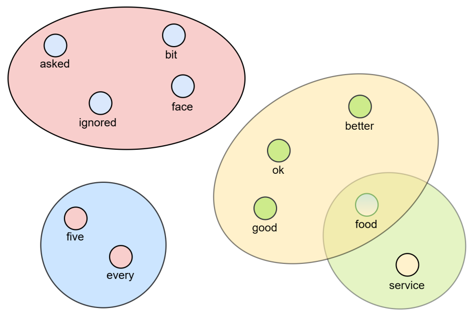

Figure 1: A hypergraph of word interactions showing several semantic clusters based on aspect and sentiment polarity. This illustrates how words are grouped according to meaning and sentiment.

To partially address these limitations, recent approaches have explored multi-graph architectures (Aziz et al., 2024; Zheng and Li, 2024), which capture different facets of text, such as syntactic dependencies and semantic relationships, and then attempt to fuse information from these disparate graph sources. While they represent an advancement, they introduce significant model complexity, increasing the number of parameters and often requiring sophisticated fusion mechanisms. This complexity can potentially hinder model performance, efficiency and generalization, especially when data is limited, as is often the case with short text in ABSA.

To address these challenges, we utilize hypergraphs which help capture varying granularities (Zhang et al., 2022a) between aspects and sentiments as seen in Figure 1. We also propose a novel hypergraph construction methodology that leverages hierarchical clustering to dynamically form hyperedges. This involves employing an adaptive thresholding technique to identify the optimal cutoff distance for hyperedge formation by analyzing the change in merge distances and determining the point of diminishing returns. This flexibility allows the model to accommodate variations in node sizes and ensures a more robust construction of hyperedges. By optimizing the distance parameter, our method ensures that the constructed hypergraph accurately captures the underlying structure of complex and densely packed data.

This paper makes the following contributions to the field of ABSA:

- We introduce HyperABSA, the first hypergraph-based framework for ABSA, demonstrating its effectiveness in capturing intricate aspect–sentiment interactions, particularly in small datasets.
- We propose a novel hypergraph construction strategy that uses hierarchical clustering with an acceleration-based thresholding criterion to dynamically form hyperedges.
- Our method achieves state-of-the-art performance. We also conduct a thorough ablation study on various graph and hypergraph construction methodologies.

2 Related Work

Over the years, ABSA has been widely explored using various methodologies.

2.1 Graph Based Methods

Graph-based approaches model syntactic and semantic word relationships using GCNs. Early works integrated dependency tags (Chen et al., 2019), as well as syntactic and semantic features from dependency trees (Zhang et al., 2022b, 2024; Gu et al., 2024) into GCNs to enrich the learning of word correlations and improve contextual understanding. Other works employed relational graph attention networks and type-aware GCNs to capture aspect-specific and inter-aspect dependencies (Wang et al., 2020; Tian et al., 2021; Ansari et al., 2020; Huang and Carley, 2019; Bao et al., 2023).

Attention mechanisms have also been pivotal, as seen in (Xu et al., 2021; Pan et al., 2023; Cui et al., 2023; Yuan et al., 2020), which combined multi-head attention with graph convolutional networks to capture semantic and syntactic dependencies effectively. Furthermore, heterogeneous graphs (Zeng et al., 2023; Niu et al., 2022) represent these different relationships explicitly, ensuring that sentiment propagation respects their distinct roles. Multi-graph models (Aziz et al., 2024; Zheng and Li, 2024) have been proposed to capture both local aspect-specific dependencies and global shared contextual information within a sentence.

2.2 Hypergraph Construction Methods

While prior works have successfully leveraged hypergraphs in other fields, their potential remains unexplored in ABSA. Much of the focus has been on developing advanced hypergraph neural network architectures (Feng et al., 2019; Zhi, 2024), with less emphasis on the original construction of hypergraph from text.

Recent hypergraph construction methods often use techniques like the Nearest-neighbor methods (Yu et al., 2012; Gao et al., 2022; Nguyen et al., 2020; Dai and Gao, 2023) that connect tokens based on proximity in feature space but often include irrelevant tokens. Latent Dirichlet Allocation (LDA) (Ding et al., 2020; Turnbull et al., 2024) improves word relationship modeling by grouping similar words into predefined topics. Clustering methods (Han et al., 1997, 1998; Chang et al., 2008; Leordeanu and Sminchisescu, 2012; Saito, 2022) like K-Means enhance hyperedge coherence by grouping tokens into clusters.

Despite their potential, hypergraphs have yet to be applied to short text data, particularly for ABSA. This is mainly due to the challenge posed by sparse feature representations in short text, which make it difficult for existing hypergraph algorithms to effectively capture semantic and syntactic relationships. due to their fixed constraints on cluster size and hyperedge count. Along with this, the computational overhead associated with hypergraph modeling further limits it's ability to such tasks.

3 Methodology

In ABSA tasks, a p-word input sentence is represented as $T = \{v_1, v_2, \ldots, v_p\}$, where v_i de-

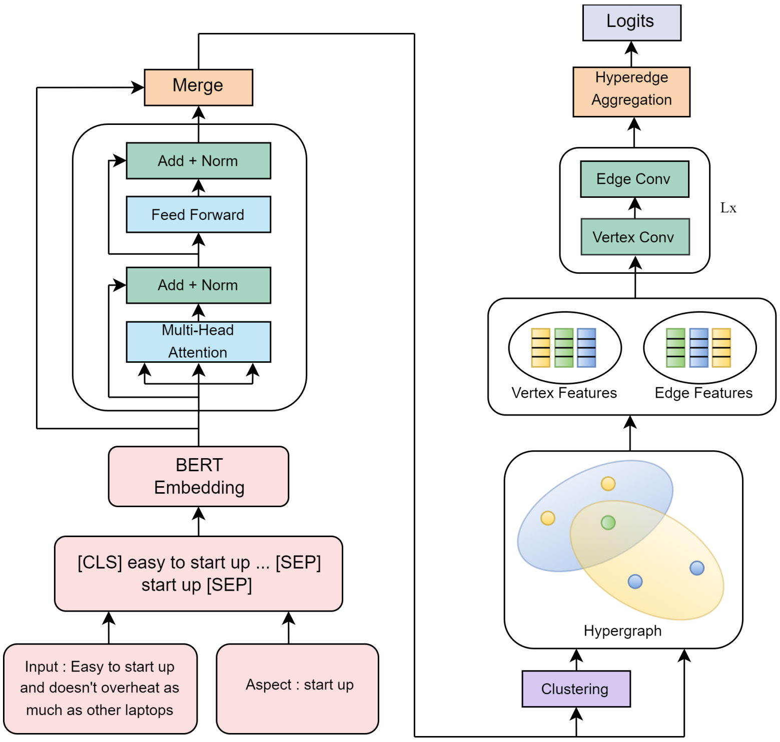

Figure 2: Architecture of HyperABSA.

notes the i-th word in the sequence. The task involves r distinct aspects, represented as $B = \{b_1^1, b_2^1, \ldots, b_q^1, b_1^2, \ldots, b_q^r\}$, where b_q^r denotes the q-th word of the r-th aspect.

The objective is to predict a mapping function, $g_r : (T, b_r) \mapsto z$, which takes as input the pair of the sentence T and aspect-specific features b_r, and outputs the sentiment polarity z for the respective aspects.

3.1 Hypergraph Definition

A hypergraph is a generalization of a standard graph, where an edge, called a hyperedge, can connect more than two nodes. Formally, a hypergraph is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (vertices), and $\mathcal{E}$ is the set of hyperedges, with each hyperedge $e \in \mathcal{E}$ being a subset of $\mathcal{V}$ (i.e., $e \subseteq \mathcal{V}$).

To mathematically represent a hypergraph, we use an incidence matrix $\mathbf{I} \in R^{|\mathcal{V}|\times|\mathcal{E}|}$, which is a binary matrix where each entry $I_{i,j}$ is defined as:

$$I_{i,j} = \begin{cases} 1 & \text{if node } v_i \text{ is part of hyperedge } e_j \\ 0 & \text{otherwise} \end{cases}$$

Given the sentence $T = \{v_1, v_2, \ldots, v_p\}$, each token v_i is represented as a node $v_i \in \mathcal{V}$. A hyperedge $e_i \in \mathcal{E}$ will be formed if a subset of nodes $\{v_1, v_2, \ldots, v_n\} \subseteq \mathcal{V}$ share semantic information.

3.2 Representation Learning

Similar to Zheng and Li (2024), we choose BERT as the text encoder. Based on the approach of Zeng et al. (2019), we format the input as "[CLS] + sentence + [SEP] + aspect + [SEP]", where [CLS] is used to represent the sentence, and [SEP] separates the sentence and aspect, as illustrated in Figure 2. Since sentences may contain multiple aspects, each aspect is treated independently. For the input sentence T, the output of BERT would be the hidden states of the last layer, $h = \{h_1, h_2, \ldots, h_n\}$ where $h \in R^{n\times d}$, with n being the sequence length and d the dimensionality of the hidden state.

To further refine the representations, we pass the hidden states through multiple layers of the transformer encoder which consists of two main components: Multi-Head Self-Attention and a Position-Wise Feed-Forward Network as implemented by (Vaswani, 2017)

3.3 Hypergraph Construction

In this section, we construct a hypergraph based on clusters derived from hierarchical clustering. The key step in this process is determining an optimal cutoff distance for partitioning the hierarchical linkage matrix $\mathbf{Z}$. To achieve this, we employ a modified version of the elbow method that dynamically adjusts the cutoff threshold based on dataset size and variability.

3.3.1 Hierarchical Clustering and Linkage Matrix

Given a dataset with n samples, each represented by BERT hidden states h, hierarchical clustering generates a linkage matrix $\mathbf{Z} \in R^{(n-1)\times 4}$, where each row $\mathbf{z}_i = [c_1, c_2, \delta_i, s_i]$. Here, c_1 and c_2 are the indices of the merged clusters at step i, δ_i represents the inter-cluster dissimilarity, and s_i is the size of the resulting cluster. The dissimilarity values $\{\delta_i\}_{i=1}^{n-1}$ quantify the hierarchical structure of the data.

3.3.2 Optimal Cutoff Distance Using the Elbow Method

To determine the optimal cutoff threshold for clustering, we employ an acceleration-based elbow method that dynamically adapts to the dataset's size and structure. Traditional elbow methods often minimize the total within-cluster sum of squared errors (WSS) to estimate the optimal number of clusters (Nainggolan et al., 2019; Humaira and Rasyidah, 2020). In contrast, our approach directly analyzes the hierarchical linkage distances and uses acceleration (second-order differences) to detect the "elbow point," where the rate of change in dissimilarity exhibits a significant shift. Additionally, we introduce a fallback mechanism to handle datasets with limited hierarchical depth.

Let $\rho \in (0, 1]$ denote the proportion parameter that controls the fraction of merges considered in the hierarchical linkage matrix, balancing local and global cluster structures.

The number of recent merges m is computed as:

$$m = \max(1, \lfloor \rho \cdot (n-1) \rfloor), \tag{1}$$

where $n-1$ is the total number of merges in the hierarchical clustering dendrogram. The corresponding dissimilarities of the recent merges are:

$$\boldsymbol{\delta}_{\text{recent}} = [\delta_{n-m}, \delta_{n-m+1}, \dots, \delta_{n-1}]. \tag{2}$$

When $|\boldsymbol{\delta}_{\text{recent}}| > 3$, we analyze the second-order differences of the recent dissimilarities $\boldsymbol{\delta}_{\text{recent}}$:

$$\Delta^2 \boldsymbol{\delta}_{\text{recent}} = [\Delta\delta_{i+1} - \Delta\delta_i \mid i = n-m, \dots, n-3], \tag{3}$$

or equivalently:

$$\boldsymbol{\alpha} = [\delta_{i+2} - 2\delta_{i+1} + \delta_i \mid i = n-m, \dots, n-3]. \tag{4}$$

A large positive value of α_i indicates a sharp increase in the dissimilarities, corresponding to a transition from compact clusters to larger, less cohesive groups.

The maximum acceleration is determined as:

$$k = \arg\max(\boldsymbol{\alpha}), \tag{5}$$

where k is the index of the largest value in $\boldsymbol{\alpha}$.

When $|\boldsymbol{\delta}_{\text{recent}}| \leq 3$, there are too few values to compute meaningful accelerations. In such cases, a fallback threshold is calculated using the mean and standard deviation of the recent dissimilarities:

$$\delta_{\text{fallback}} = \bar{\delta}_{\text{recent}} + \lambda \cdot \sigma_{\text{recent}}, \tag{6}$$

where $\bar{\delta}_{\text{recent}}$ and σ_{recent} are the mean and standard deviation of $\boldsymbol{\delta}_{\text{recent}}$, respectively, and $\lambda > 0$ is a scaling factor. This fallback mechanism provides a robust baseline cutoff threshold for small datasets by accounting the variabilities in dissimilarities of the recent merges, ensuring better cohesion.

We thus define the elbow dissimilarity δ_{elbow} as:

$$\delta_{\text{elbow}} = \begin{cases} min(\delta_{n-m+k}, \delta_{\text{fallback}}), & \text{if } |\boldsymbol{\delta}_{\text{recent}}| > 3, \\ \delta_{\text{fallback}}, & \text{otherwise.} \end{cases} \tag{7}$$

This approach ensures that the cutoff distance adapts to both the structure of the dataset and the variability in the distances.

Once the cutoff distance δ_{elbow} is determined, the dataset is partitioned into clusters $\{\mathcal{C}\}_{i=1}^{k}$, where each cluster C_i is a set of points such that the intra-cluster distances are less than δ_{elbow}. The hypergraph $\mathcal{H}$ is then constructed, where the vertex set $\mathcal{V}$ corresponds to the data points and the hyperedge set $\mathcal{E}$ is defined as $\mathcal{E} = \{e_i \mid e_i = C_i,\ C_i \in \mathcal{C}\}$

3.4 Hypergraph Neural Network

We adopt a classic approach to a hypergraph neural network (Feng et al., 2019), which involves multiple layers of vertex and edge convolution. The network ends with a final layer of aggregation which combines the vertex and edge features.

3.4.1 Vertex Convolution

At layer l, we define the vertex feature matrix as $\mathbf{V}^l \in R^{|\mathcal{V}|\times d}$ and the edge feature matrix as $\mathbf{E}^l \in R^{|\mathcal{E}|\times d}$ where d is the feature dimension.

We perform convolution on the vertex set $\mathcal{V}$ using $\mathbf{I}$ and $\mathbf{V}^{(l)}$. We compute the hidden states for each edge $e \in \mathcal{E}$ by aggregating the features of all the vertices in its set.

To compute the edge weights for the current layer's hidden states, we perform the following operations:

$$\mathbf{A}_e = \text{softmax}\left(\mathbf{I}^T \cdot \left(\mathbf{W}_a \cdot \mathbf{E}^{l-1}\right)\right) \tag{8}$$

$$\mathbf{M_v} = \mathbf{V}^l \odot \left(\mathbf{I} \cdot \mathbf{A}_e^l\right) \tag{9}$$

$$\mathbf{E}^l = \alpha \cdot \mathbf{E}^{l-1} + (1-\alpha) \cdot (\mathbf{W}_p \cdot \mathbf{M_v}) \tag{10}$$

Here, $\mathbf{W}_a, \mathbf{W}_p \in R^{d \times d}$ are learnable weight matrices for the edge and node operations, respectively, $\alpha \in R$ is a learnable parameter which adaptively controls the contribution from the prior layer's hidden states and the vertex aggregation at each step and $\odot$ denotes element-wise multiplication.

3.4.2 Edge Convolution

Similar to vertex convolution, edge convolution involves aggregating information across all hyperedges associated with each vertex, updating the vertex feature matrix based on the edge features.

$$\mathbf{A}_v^l = \text{softmax}\left(\mathbf{I} \cdot \left(\mathbf{W}_a \cdot \mathbf{V}^{l-1}\right)\right) \tag{11}$$

$$\mathbf{M_e} = \mathbf{E}^l \odot \left(\mathbf{I}^T \cdot \mathbf{A}_v^l\right) \tag{12}$$

$$\mathbf{V}^l = \alpha \cdot \mathbf{V}^{l-1} + (1-\alpha) \cdot (\mathbf{W}_p \cdot \mathbf{M_e}) \tag{13}$$

3.4.3 Aggregation

After performing vertex and edge convolution for multiple layers, we merge the refined vertex and edge feature matrices to get the final logits.

$$\mathbf{E} = \mathbf{W_e}^L \cdot \mathbf{E}^L \odot \text{softmax}(\mathbf{W_e}^L \cdot \mathbf{E}^L) \tag{14}$$

$$\mathbf{V} = \mathbf{W_v}^L \cdot \mathbf{V}^L \odot \text{softmax}(\mathbf{W_v}^L \cdot \mathbf{V}^L) \tag{15}$$

$$\text{Logits} = \mathcal{F}(\mathbf{V}, \mathbf{E}) \tag{16}$$

Here, $\mathbf{W}_v, \mathbf{W}_e \in \mathbf{R}^{d \times d}$ are trainable weights and $\mathcal{F}$ is a mapping function designed to effectively combine $\mathbf{V}$ and $\mathbf{E}$:

$$\begin{aligned} \mathcal{F}(\mathbf{V}, \mathbf{E}) = \sigma\left(\mathbf{W}_g \cdot [\mathbf{V}; \mathbf{E}]\right) \odot \mathbf{V} \\ + \left(1 - \sigma\left(\mathbf{W}_g \cdot [\mathbf{V}; \mathbf{E}]\right)\right) \odot \mathbf{E} \end{aligned} \tag{17}$$

where $\sigma(\cdot)$ is the sigmoid function, $[\mathbf{V}; \mathbf{E}]$ represents concatenation, and $\odot$ denotes element-wise multiplication.

4 Experiments

4.1 Datasets

We evaluate our proposed model using three benchmark datasets: the Multi-Aspect Multi-Sentiment (MAMS) dataset (Jiang et al., 2019), the SemEval 2014 datasets for Restaurants (Rest14) and Laptops (Lap14) (Pontiki et al., 2014). The split and the statistics of the data is adopted from (Bai et al., 2020)

4.2 Baselines

HyperABSA, as the first approach to introduce hypergraphs to ABSA, is initially compared against several baseline methods, including IARM (Majumder et al., 2018), MIAD (Hazarika et al., 2018), StageI+StageII (Ma et al., 2019), CDT (Sun et al., 2019) and RepWalk (Zheng et al., 2020), BERT-SPC (Song et al., 2019) and CapsNet (Jiang et al., 2019) to showcase it's effectiveness. We then evaluate HyperABSA's performance against multiple state-of-the-art methods which utilise dependency trees or graphs that employ GCNs including InterGCN (Liang et al., 2020), R-GAT (Wang et al., 2020), DGEDT (Tang et al., 2020), RGAT (Bai et al., 2020), RMN (Zeng et al., 2022), CHGMAN (Niu et al., 2022), DMGLT (Fang, 2022), MWGCN (Yu and Zhang, 2023), YORO (Zheng and Li, 2024).

4.3 Implementation details

For the encoder, we utilize the BERT architecture, specifically the *bert-base-uncased* variant. To mitigate overfitting, we apply dropout with a rate selected from the range [0.2,0.3] to both the BERT encoder and the hypergraph convolution layers and an L2 regularization of $\lambda = 2 * 10^{-5}$. Model optimization is performed using the Adam optimizer (Kingma, 2014) with a learning rate of 10^{-2}, and a batch size of 16 is used during training. We conduct experiments on a single NVIDIA 4090 GPU.

4.4 Results

Table 1 presents a comparative analysis of HyperABSA against both baseline and recent state-of-the-art models. On the MAMS dataset, our method achieves the highest accuracy, outperforming existing approaches by a margin of 0.3%, while also maintaining a competitive F1 score. For the Rest14 dataset, HyperABSA demonstrates superior performance in both accuracy and F1 score, with an average improvement of 0.4% over prior methods.

Model	MAMS		Rest14		Lap14	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
BERT-SPC† (Song et al., 2019)	82.82	81.9	84.46	76.98	78.99	75.03
CapsNet† (Jiang et al., 2019)	83.46	82.89	84.91	76.59	77.12	71.84
InterGCN† (Liang et al., 2020)	82.49	81.95	85.45	77.64	78.06	73.83
R-GAT* (Wang et al., 2020)	83.16	82.42	84.64	77.14	78.21	74.07
DGEDT* (Tang et al., 2020)	-	-	86.30	80.00	79.80	75.60
RGAT* (Bai et al., 2020)	82.96	82.12	85.77	79.81	80.31	76.38
DMGLT (Fang, 2022)	-	-	86.25	79.04	78.82	75.56
RMN (Zeng et al., 2022)	79.97	78.79	84.56	79.05	77.95	70.83
CHGMAN* (Niu et al., 2022)	83.23	82.66	85.98	79.31	78.04	74.46
MWGCN (Yu and Zhang, 2023)	-	-	86.36	80.54	79.78	76.68
HGCN (Xu et al., 2023)	-	-	86.45	80.60	79.59	76.24
LLaMa2-13b‡ (Su et al., 2024)	-	-	78.00	67.00	73.00	65.00
ChatGPT (zero-shot)‡	-	-	82.39	73.64	77.64	72.30
ChatGPT (few-shot)‡	-	-	84.62	76.08	78.15	75.79
YORO*(Zheng and Li, 2024)	84.21	**83.78**	83.69	76.22	77.45	73.21
HyperABSA	**84.56**	83.74	**86.762**	**80.641**	**80.46**	**77.42**

Table 1: Performance of Accuracy and F1 score of HyperABSA with other models. † denotes implementation from (Zheng and Li, 2024), ‡ denotes implementation from (Chen et al., 2024) and * denotes our implementation.

Similarly, on the Laptop dataset, our model attains the highest accuracy as well as F1 score, with an average margin of 2% over competitive baselines. This highlights HyperABSA's ability to effectively handle short, multi-aspect, multi-sentiment textual complexities.

5 Discussion

5.1 Effects of Adaptive tuning

We evaluate the effect of adaptive tuning in hypergraph construction against a fixed, non-adaptive variant that uses a static fallback distance (Equation 6), with varying α values, while the adaptive method dynamically adjusts this parameter based on local structure, enabling more flexible hyperedge formation. To ensure a fair comparison, both methods are evaluated using the same sentence as in Figure 1

As shown in Figure 3, the non-adaptive method is highly sensitive to α, producing fragmented clusters at lower values (e.g., $\alpha = 0.3$) and overly coarse groupings at higher ones ($\alpha = 0.5, 0.7$), which dilute semantic distinctions. This instability reveals the limitations of fixed thresholds. In contrast, the adaptive method consistently forms semantically coherent hyperedges by balancing local context and global structure. It effectively separates concepts, like grouping "service" and "food" as core subjects, while isolating sentiment-bearing words like "good", "ok", and "better", enabling more precise representation of contextual relationships.

Model	Silhouette Score			Davis-Bouldin Score		
	Min	Mean	Max	Min	Mean	Max
Random	-0.24	-0.23	-0.22	1.51	1.59	1.64
KNN-KMeans	0.31	0.33	0.40	1.05	1.17	1.32
HyperABSA	**0.36**	**0.42**	**0.62**	**0.56**	**0.99**	**1.10**

Table 2: Comparison of cluster quality across different hypergraph construction methods.

5.2 Cluster Quality Analysis

We evaluate the effectiveness of our hypergraph construction method by comparing it against (i) a Random hypergraph, in which nodes and hyperedges are generated without structural priors, and (ii) a KNN-KMeans hybrid hypergraph, where local and global structural cues are captured by integrating K-Nearest Neighbors and K-Means clustering. The quality of the resulting cluster structures is quantified using standard clustering validation metrics, namely the Silhouette Score (Rousseeuw, 1987), which evaluates cluster compactness and separation, where higher values indicate well-formed and distinct clusters, and the Davis-Bouldin Score (Davies and Bouldin, 1979), which measures the average similarity between clusters, where lower values indicate better clustering, across different training epochs.

As shown in Table 2, HyperABSA consistently outperforms these baseline methods. The Random hypergraph fails to form meaningful clusters due

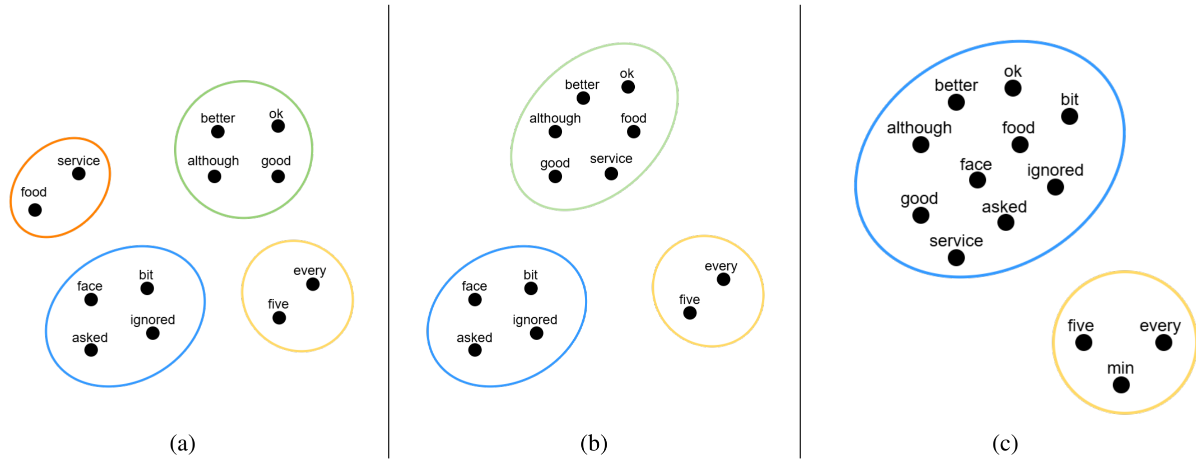

Figure 3: Hypergraphs formed by a) Adaptive tuning, b) $\lambda = 0.3$, c) λ = 0.5 and 0.7 as in Equation 6

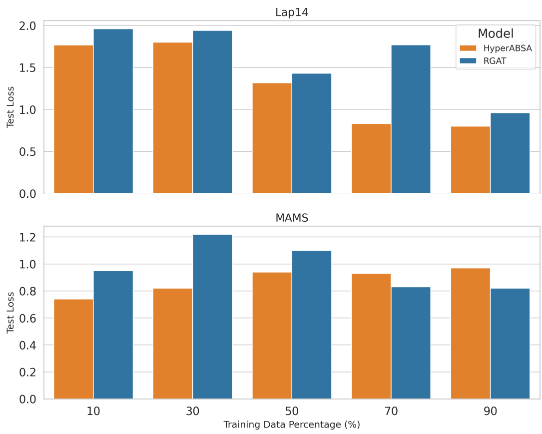

Figure 4: Comparison of test loss between HyperABSA and a graph-based model RGAT on the Lap14 and MAMS datasets

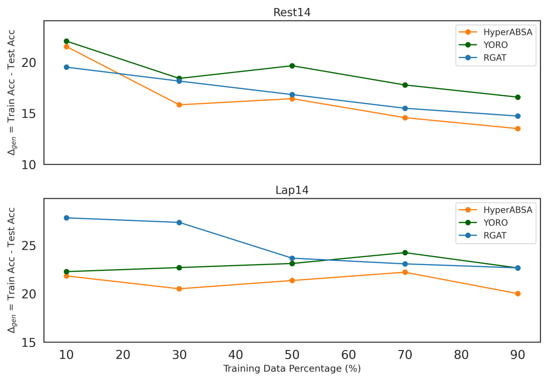

Figure 5: Evaluation of HyperABSA against multi-graph-based models on the Rest14 and MAMS datasets in terms of generalization gap.

to it's stochastic nature, often yielding negative silhouette scores. While the KNN-KMeans hybrid introduces some structural priors, it still underperforms in terms of clustering quality. These results highlight the effectiveness of HyperABSA in preserving structure and semantic coherence across training epochs.

5.3 Generalization Gap

Prior works often relied on constructing multiple graphs, each capturing a distinct semantic view or level of granularity to enrich representation learning. While effective, this approach introduces significant overhead in graph construction and fusion mechanisms. To evaluate the generalization ability of our proposed model, we measured the generalization gap, defined as the difference between training and test accuracy, as well as loss, across varying amounts of training data. Each configuration was repeated across multiple random seeds, and we report the average values to ensure robustness. We conducted this evaluation on both the Lap14 and Rest14 datasets, comparing HyperABSA with two strong baselines: YORO, a multi-graph model, and RGAT, a single-graph model. As shown in Figure 4 and Figure 5, HyperABSA consistently achieves smaller generalization gaps across most training sizes. Our model exhibits strong generalization even in cases with less data, whereas the other models require at least 50-70% of the training data to achieve a comparable amount of generalization. Notably, while our primary aim was to serve as an alternative to multi-graph models, HyperABSA also consistently outperforms the single-graph baseline across both datasets.

These results suggest that the dynamic and sample-sensitive structure of HyperABSA enables

Method Variant (with Formula)	ρ	Rest14		Lap14		MAMS	
		Accuracy (%)	F1 Score (%)	Accuracy (%)	F1 Score (%)	Accuracy (%)	F1 Score (%)
HyperABSA (Equation 7)	Dynamic	**86.76**	**80.64**	**80.46**	**77.42**	**84.56**	**83.74**
$\delta_{\text{elbow}} = \delta_{fallback}$	-	84.07	76.89	79.06	75.84	84.00	83.51
$\delta_{\text{elbow}} = \delta_{n-m+k}$	0.2	80.59	71.61	79.68	76.30	83.48	82.82
	0.5	83.11	74.75	78.13	74.88	83.48	82.87
	0.8	82.12	74.60	77.03	73.18	83.55	82.90
$\delta_{\text{elbow}} = min(\delta_{n-m+k}, \delta_{fallback})$	0.2	84.78	77.35	79.22	77.14	84.22	83.46
	0.5	80.95	72.06	78.75	75.95	84.07	83.24
	0.8	84.98	78.24	79.53	76.03	83.70	83.09

Table 3: Ablation study on Rest14, Lap14, and MAMS showing the impact of acceleration formula and proportion (p) on HyperABSA's performance. Formula types are indicated in parentheses within the method name.

Model	MAMS		Rest14		Lap14	
	Params(100M)	Acc/P	Params(100M)	Acc/P	Params(100M)	Acc/P
RGAT	1.10	75.41	1.10	77.97	**1.10**	73.00
YORO	1.15	73.22	1.15	72.77	1.15	67.37
HyperABSA	**1.10**	**76.87**	**1.10**	**78.87**	1.11	**73.14**

Table 4: Model efficiency comparison based on parameter count and accuracy-per-parameter (Acc/P).

it to better model context-specific relationships while avoiding overfitting, particularly in low-data regimes. In addition to generalization performance, we assessed model efficiency by computing accuracy-to-parameter ratios for all models across datasets. As shown in Table 4, HyperABSA achieves consistently better ratios compared to both YORO and RGAT, indicating higher performance per parameter. This demonstrates that our approach not only generalizes better but also incurs less overhead in terms of model size. Together, these findings reinforce our claim that HyperABSA is a robust, efficient, and generalizable alternative to multi-graph models in ABSA.

5.4 Geometric Interpretation of Acceleration

To better understand the role of acceleration in detecting the elbow point in hierarchical clustering, we treat the sequence of recent dissimilarities $\mathbf{d}_{\text{recent}}$ as a discrete signal capturing hierarchical merge distances (Equation 2). The first-order differences, $\Delta\mathbf{d}_{\text{recent}}$, describes the slope of this sequence, while the second-order differences, $\Delta^2\mathbf{d}_{\text{recent}}$, describes the curvature, $\mathbf{d}_{\text{recent}}$, quantifying how much the sequence deviates from linearity. High curvature values indicate regions where the dissimilarity values exhibit sharp increases, corresponding to structural shifts in the dendrogram. This curvature-based acceleration serves as a reliable indicator for detecting the elbow and as described in Equation 5, the index of the maximum acceleration is selected to identify this point.

5.5 Multi granular approach of hypergraph

To explore whether a dynamically constructed hypergraph can serve as a viable alternative to manually designed multi-graph architectures for multi-granular reasoning, we conduct a series of comparative experiments. We compare our dynamic hypergraph approach with several fixed-granularity baselines, including models with only fallback connections (coarse granularity), and acceleration paths with static thresholds (ρ = 0.2, 0.5, 0.8). As seen in Table 3, across datasets, these fixed strategies yield lower or inconsistent performance, indicating their inability to capture the optimal granularity across samples. In contrast, our model dynamically selects both the threshold and the graph construction strategy per instance, effectively adapting to sample-specific views. These findings support our broader claim, that automatically identifying an appropriate granularity per instance can offer a strong alternative to using multiple graphs for capturing the different granularities.

6 Conclusion

In this paper, we introduce HyperABSA, a novel hypergraph construction methodology for ABSA that dynamically forms hyperedges via adaptive hierarchical clustering. Our approach addresses the challenge of overfitting in short-text scenarios by leveraging an acceleration-based thresholding mechanism, ensuring that hyperedges capture meaningful multi-node interactions while preventing excessive fragmentation or over-merging. Comprehensive evaluations on Lap14, Rest14, and MAMS datasets demonstrate that HyperABSA achieves state-of-the-art performance among graph-based approaches, highlighting its effectiveness in capturing nuanced multi-node interactions for fine-grained sentiment reasoning.

7 Limitations

Multi-graph models offer interpretable edge semantics grounded in syntactic or semantic roles, while hypergraphs, though rich in context, lack this clarity, posing challenges for interpretability and fine-grained error analysis. Our approach is computationally complex compared to conventional single-graph baselines, making it susceptible to overfitting, particularly on low-resource datasets such as Lap14, where aspect-opinion annotations are sparse and domain-specific vocabularies limit generalization. Although we introduced minor architectural adjustments to the base HGNN framework, it was not designed for ABSA. This mismatch added to the modeling complexity and may have hindered performance in ABSA-specific scenarios.

References

Gunjan Ansari, Chandni Saxena, Tanvir Ahmad, and MN Doja. 2020. Aspect term extraction using graph-based semi-supervised learning. *Procedia Computer Science*, 167:2080–2090.

Kamran Aziz, Donghong Ji, Prasun Chakrabarti, Tulika Chakrabarti, Muhammad Shahid Iqbal, and Rashid Abbasi. 2024. Unifying aspect-based sentiment analysis bert and multi-layered graph convolutional networks for comprehensive sentiment dissection. *Scientific Reports*, 14(1):14646.

Xuefeng Bai, Pengbo Liu, and Yue Zhang. 2020. Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:503–514.

Xiaoyi Bao, Zhongqing Wang, and Guodong Zhou. 2023. Exploring graph pre-training for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3623–3634.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Yuchou Chang, Dah-Jye Lee, James Archibald, and Yi Hong. 2008. Unsupervised clustering using hyperclique pattern constraints. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE.

Chenhua Chen, Zhiyang Teng, Zhongqing Wang, and Yue Zhang. 2022. Discrete opinion tree induction for aspect-based sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2064.

Junjie Chen, Hongxu Hou, Yatu Ji, Jing Gao, and Tiangang Bai. 2019. Graph-based attention networks for aspect level sentiment analysis. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1188–1194. IEEE.

Yanjiang Chen, Kai Zhang, Feng Hu, Xianquan Wang, Ruikang Li, and Qi Liu. 2024. Dynamic multi-granularity attribution network for aspect-based sentiment analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10920–10931.

Xiaodong Cui, Wenbiao Tao, and Xiaohui Cui. 2023. Affective-knowledge-enhanced graph convolutional networks for aspect-based sentiment analysis with multi-head attention. *Applied Sciences*, 13(7):4458.

Qionghai Dai and Yue Gao. 2023. *Hypergraph Computation*. Springer Nature.

David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387*.

Chuan Fang. 2022. Dependencymerge guided latent tree structure for aspect-based sentiment analysis. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565.

Xiang Gao, Fan Zhou, Kedi Xu, Xiang Tian, and Yaowu Chen. 2022. A parallel algorithm for maximal cliques enumeration to improve hypergraph construction. *Journal of Computational Science*, 65:101905.

Qun Gu, Zhidong Wang, Hai Zhang, Siyi Sui, and Rui Wang. 2024. Aspect-level sentiment analysis based on syntax-aware and graph convolutional networks. *Applied Sciences*, 14(2):729.

Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. 1998. Hypergraph based clustering in high-dimensional data sets: A summary of results. *IEEE Data Eng. Bull.*, 21(1):15–22.

Euihong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. 1997. Clustering based on association rule hypergraphs. *IEEE Transactions on Knowledge and Data Engineering*, 9(2):335–348.

Devamanyu Hazarika, Soujanya Poria, Prateek Vij, Gangeshwar Krishnamurthy, Erik Cambria, and Roger Zimmermann. 2018. Modeling inter-aspect dependencies for aspect-based sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 266–270.

Binxuan Huang and Kathleen M Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv preprint arXiv:1909.02606*.

Hestry Humaira and R Rasyidah. 2020. Determining the appropiate cluster number using elbow method for k-means algorithm. In *Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA) 2018, 24-25 January 2018, Padang, Indonesia*.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. 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)*, pages 6280–6285.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Marius Leordeanu and Cristian Sminchisescu. 2012. Efficient hypergraph clustering. In *Artificial Intelligence and Statistics*, pages 676–684. PMLR.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6319–6329.

Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, and Ruifeng Xu. 2020. Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis. In *Proceedings of the 28th international conference on computational linguistics*, pages 150–161.

Xiao Ma, Jiangfeng Zeng, Limei Peng, Giancarlo Fortino, and Yin Zhang. 2019. Modeling multi-aspects within one opinionated sentence simultaneously for aspect-level sentiment analysis. *Future Generation Computer Systems*, 93:304–311.

Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md Shad Akhtar, Erik Cambria, and Asif Ekbal. 2018. Iarm: 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*, pages 3402–3411.

Rena Nainggolan, Resianta Perangin-angin, Emma Simarmata, and Astuti Feriani Tarigan. 2019. Improved the performance of the k-means cluster using the sum of squared error (sse) optimized by using the elbow method. In *Journal of Physics: Conference Series*, volume 1361, page 012015. IOP Publishing.

Dong Quan Ngoc Nguyen, Lin Xing, and Lizhen Lin. 2020. Community detection, pattern recognition, and hypergraph-based learning: approaches using metric geometry and persistent homology. In *Fuzzy Systems and Data Mining VI*, pages 457–473. IOS Press.

Hao Niu, Yun Xiong, Jian Gao, Zhongchen Miao, Xiaosu Wang, Hongrun Ren, Yao Zhang, and Yangyong Zhu. 2022. Composition-based heterogeneous graph multi-channel attention network for multi-aspect multi-sentiment classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6827–6836.

Yunhui Pan, Dongyao Li, Zhouhao Dai, and Peng Cui. 2023. Aspect-based sentiment analysis using dual probability graph convolutional networks (dp-gcn) integrating multi-scale information. In *International Conference on Neural Information Processing*, pages 495–512. Springer.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Soujanya Poria, Nir Ofek, Alexander Gelbukh, Amir Hussain, and Lior Rokach. 2014. Dependency tree-based rules for concept-level aspect-based sentiment analysis. In *Semantic Web Evaluation Challenge: SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 41–47. Springer.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Shota Saito. 2022. Hypergraph modeling via spectral embedding connection: Hypergraph cut, weighted kernel k-means, and heat kernel. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8141–8149.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

Huizhe Su, Xinzhi Wang, Jinpeng Li, Shaorong Xie, and Xiangfeng Luo. 2024. Enhanced implicit sentiment understanding with prototype learning and demonstration for aspect-based sentiment analysis. *IEEE Transactions on Computational Social Systems*.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. 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)*, pages 5679–5688.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6578–6588.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021. Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2910–2922.

Kathryn Turnbull, Simón Lunagómez, Christopher Nemeth, and Edoardo Airoldi. 2024. Latent space modeling of hypergraph data. *Journal of the American Statistical Association*, 119(548):2634–2646.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.12362*.

Guangtao Xu, Peiyu Liu, Zhenfang Zhu, Jie Liu, and Fuyong Xu. 2021. Attention-enhanced graph convolutional networks for aspect-based sentiment classification with multi-head attention. *Applied Sciences*, 11(8):3640.

Lvxiaowei Xu, Xiaoxuan Pang, Jianwang Wu, Ming Cai, and Jiawei Peng. 2023. Learn from structural scope: Improving aspect-level sentiment analysis with hybrid graph convolutional networks. *Neurocomputing*, 518:373–383.

Bengong Yu and Shuwen Zhang. 2023. A novel weight-oriented graph convolutional network for aspect-based sentiment analysis. *The Journal of Supercomputing*, 79(1):947–972.

Jun Yu, Dacheng Tao, and Meng Wang. 2012. Adaptive hypergraph learning and its application in image classification. *IEEE Transactions on Image Processing*, 21(7):3262–3272.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2020. Graph attention network with memory fusion for aspect-level sentiment analysis. In *Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing*, pages 27–36.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16):3389.

Jiandian Zeng, Tianyi Liu, Weijia Jia, and Jiantao Zhou. 2022. Relation construction for aspect-level sentiment classification. *Information Sciences*, 586:209–223.

Yufei Zeng, Zhixin Li, Zhenbin Chen, and Huifang Ma. 2023. Aspect-level sentiment analysis based on semantic heterogeneous graph convolutional network. *Frontiers of Computer Science*, 17(6):176340.

Fan Zhang, Wenbin Zheng, and Yujie Yang. 2024. Graph convolutional network with syntactic dependency for aspect-based sentiment analysis. *International Journal of Computational Intelligence Systems*, 17(1):37.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2022a. Hegel: Hypergraph transformer for long document summarization. *arXiv preprint arXiv:2210.04126*.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022b. 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*, pages 4916–4925.

Yaowei Zheng, Richong Zhang, Samuel Mensah, and Yongyi Mao. 2020. Replicate, walk, and stop on syntax: an effective neural network model for aspect-level sentiment classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9685–9692.

Yongqiang Zheng and Xia Li. 2024. You only read once: constituency-oriented relational graph convolutional network for multi-aspect multi-sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19715–19723.

Xinke Zhi. 2024. A review of hypergraph neural networks. *EAI Endorsed Transactions on e-Learning*, 10.