Composition-based Heterogeneous Graph Multi-channel Attention Network for Multi-aspect Multi-sentiment Classification

Hao Niu[1], **Yun Xiong**[1]*, **Jian Gao**[2], **Zhongchen Miao**[2], **Xiaosu Wang**[1], **Hongrun Ren**[1], **Yao Zhang**[1], **Yangyong Zhu**[1]
[1]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[2]Shanghai Financial Futures Information Technology Co., Ltd.
[1]{hniu18, yunx, xswang19, renhr20, yaozhang, yyzhu}@fudan.edu.cn
[2]{gaojian, miaozc}@cffex.com.cn

Abstract

Aspect-based sentiment analysis (ABSA) has drawn more and more attention because of its extensive applications. However, towards the sentence carried with more than one aspect, most existing works generate an aspect-specific sentence representation for each aspect term to predict sentiment polarity, which neglects the sentiment relationship among aspect terms. Besides, most current ABSA methods focus on sentences containing only one aspect term or multiple aspect terms with the same sentiment polarity, which makes ABSA degenerate into sentence-level sentiment analysis. In this paper, to deal with this problem, we construct a heterogeneous graph to model inter-aspect relationships and aspect-context relationships simultaneously and propose a novel Composition-based Heterogeneous Graph Multi-channel Attention Network (CHGMAN) to encode the constructed heterogeneous graph. Meanwhile, we conduct extensive experiments on three datasets: MAMS-ATSA, Rest14, and Laptop14, experimental results show the effectiveness of our method.

1 Introduction

Aspect-based sentiment analysis (ABSA) is a sentiment analysis task, which aims to predict the sentiment polarity (e.g., POSITIVE, NEGATIVE, NEUTRAL) towards the given aspect term in a sentence. For example, there are two aspect terms *decor* and *food* in the sentence: *The decor is not a special at all but their amazing food makes up for it*, and these two aspect terms will be assigned with NEGATIVE and POSITIVE sentiment polarities respectively. Since the sentiments expressed by these two aspect terms are opposite in polarity, it is unreasonable to assign a sentence-level sentiment polarity. In this regard, ABSA can provide more detailed sentimental predictions compared with sentence-level sentiment analysis.

In recent years, neural network-based methods (Tang et al., 2016; Wang et al., 2016; Ma et al., 2017; Chen et al., 2017) are used to tackle ABSA task and achieve promising performance. Subsequently, with the rise of contextual embedding, contextualized language models, such as BERT (Devlin et al., 2019), are introduced into ABSA and boost performance. Additionally, in order to utilize the dependency tree information, a lot of graph-based methods (Zhang et al., 2019; Sun et al., 2019; Huang and Carley, 2019; Wang et al., 2020) are proposed to tackle this task by encoding both contextual and dependency information simultaneously. However, most of these methods focus on the datasets, in which most sentences consist of only one aspect or multiple aspects with the same sentiment polarity. Under these circumstances, aspect-based sentiment analysis degenerates into sentence-level sentiment analysis. Existing ABSA methods based on these datasets can hardly adapt to the situation, where multiple aspect terms in a sentence with multiple different sentiment polarities. Thus, the fine-grained sentiment analysis task derived from ABSA towards the multi-aspect multi-sentiment situation, Multi-aspect Multi-sentiment Classification (MAMSC), is worthy of exploration(Jiang et al., 2019). MAMSC is more challenging than ABSA, and each sentence contains at least two aspects with different sentiment polarities in MAMSC.

Besides, most existing methods neglect to consider inter-aspect relationships when predicting sentiment polarity. Especially in MAMSC, the main challenge is that the sentiment polarities of some aspects in a given sentence can not be judged solely relied on the context and may be guided by the sentiments of other aspects. Inter-aspect relationships will play an essential role in identifying sentiment polarity in this case. Though InterGCN (Liang et al., 2020) claims to have considered inter-aspect relationships, what it considered is merely the distance relationship among aspects and neglecting

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 6827–6836
October 12–17, 2022.

the semantic relationship between every two aspects, which is essential when indicating the relationship between two aspects. Furthermore, InterGCN ignores the heterogeneity of aspect terms and context words and the different importance of surrounding nodes when aggregating information.

To sum up, ABSA faces the following challenges: (1) do not pay special attention to the multi-aspect multi-sentiment situation; (2) do not incorporate semantic relationships among aspects into predicting sentiment polarity; (3) do not introduce heterogeneity and attention mechanism when using graph-based models to predict sentiment polarity.

To deal with the challenges mentioned above, we first construct a heterogeneous graph with distinct node and edge types to represent inter-aspect relationships, aspect-context relationships, and node heterogeneity simultaneously in a concrete way. Unlike InterGCN, we treat aspect terms and context words as different types of nodes, and meanwhile, the edges in our heterogeneous graph are attributed. We utilize dependency tags as the attributes of aspect-context edges. The shortest dependency path between each two aspect terms is the representation of each inter-aspect edge attribute, which signifies the semantic relationship between these two aspect terms. Then we propose a novel model called *Composition-based Heterogeneous Graph Multi-channel Attention Network (CHGMAN)* to capture important information conveyed by constructed heterogeneous graph. Our proposed CHGMAN equipped with a multi-channel attention mechanism can take edge features, node types, and node features into account concurrently and aggregate composition information of nodes and node types in terms of importance, which is conducive to capturing the information expressed by our heterogeneous graph and making the correct predictions. The main contributions of this paper can be summarized as follows:

Heterogeneous Graph: to the best of our knowledge, this is the first attempt to model multi-aspect multi-sentiment classification with heterogeneous graph networks to capture both inter-aspect relationships and context-aspect relationships.

Multi-channel Attention: we propose a novel model *Composition-based Heterogeneous Graph Multi-channel Attention Network (CHGMAN)* to incorporate inter-aspect relationship information, node and node type information simultaneously for predicting sentiment polarity.

Extensive Experiments: we conduct extensive experiments over multi-aspect multi-sentiment datasets: MAMS-ATSA, Rest14, and Laptop14, and the results show the effectiveness of our method. [1]

2 Related Work

2.1 Graph-based Models

Recently, graph neural networks (GNNs) have been introduced into ABSA and achieved promising performance. ASGCN (Zhang et al., 2019) employs graph convolutional network (GCN) to capture dependency syntactical information. CDT (Sun et al., 2019) incorporates GCN and BiLSTM over the dependency information and contextual information of the sentence. InterGCN (Liang et al., 2020) exploits GCN to extract both aspect-focused and inter-aspect information for specific aspects, but it only considers distance relations among aspect terms. RGAT (Wang et al., 2020) designs a graph attention networks (GAT) derivative to utilize dependency tag information by taking them as gates to control information flow. However, all these models do not utilize inter-aspect semantic relationships and neglect the different roles of aspect terms and context words.

2.2 Multi-aspect Multi-sentiment

There exist several works that tend to model inter-aspect relationships to boost the performance of ABSA. IARM (Majumder et al., 2018) utilizes memory networks to incorporate the neighboring aspects-related information to predict the sentiment polarity of the given aspect term. InterGCN models inter-aspect distance relationships to help predict sentiment polarity in the presence of other aspects. SDGCN (Zhao et al., 2020) utilizes GCNs to capture sentiment dependencies between aspects. Besides, MIAD (Hazarika et al., 2018), STAGE (Ma et al., 2019) and Joint+PERT (Zhou et al., 2020) also tend to model relationships among aspects by using LSTM based model and attention mechanism. CapsNet (Jiang et al., 2019) releases a new dataset dedicated to the multi-aspect multi-sentiment situation and proposes a model to employ capsule network and capture the intricate relationship between aspect terms and context words. Nonetheless, all of these existing works do not take edge information, such as dependency syntactic and semantic

[1] The code is available at https://github.com/hankniu01/chgman

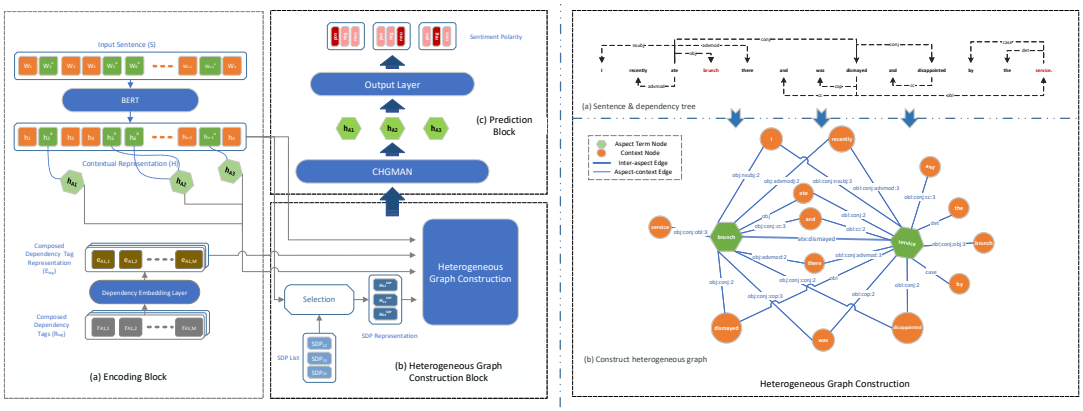

Figure 1: The overall architecture of our proposed method (left). The heterogeneous graph construction and composed dependency tags is shown in right.

relationship among aspects into consideration, and heterogeneity between aspect terms and context words is also ignored.

3 Methodology

Multi-aspect multi-sentiment classification can be formulated as follows: given a sentence S with n words $\{w_1, w_2, ..., w_n\}$ and a given aspect term set $\mathcal{A} = \{A_1, A_2, ..., A_\kappa\}$ containing κ aspect terms. And $A_k = \{w^a_\varphi, w^a_{\varphi+1}, ..., w^a_{\varphi+\tau}\}$ with $\tau \in [1, n-\varphi]$ is a sub-string of sentence S, which denotes one element of the given aspect term set $\mathcal{A}$. The annotation a represents aspect. This task aims at predicting the sentiment polarity y {POSITIVE, NEUTRAL, NEGATIVE} expressed on each aspect term included in the given aspect term set $\mathcal{A}$ of the sentence S. As shown in Figure 1(left), our proposed methods consist of three blocks: (1) Encoding Block; (2) Heterogeneous Graph Construction Block; (3) Prediction Block. In Section 3.1 to 3.3, we detail each block.

3.1 Encoding Block

As shown in Figure 1(left), we encode context words, aspect terms, and composed dependency tags, preparing for constructing heterogeneous graphs.

3.1.1 Contextual Encoder

For a given input sentence $S = \{w_1, w_2, ..., w_n\}$ with a aspect term set $\mathcal{A} = \{A_1, A_2, ..., A_\kappa\}$, we first employ pre-trained BERT (Devlin et al., 2019) as the contextual encoder to obtain hidden contextual representation H of the given sentence S. The embeddings of context words $H_w = \{h_1, h_2, ..., h_{n_w}\}$ and aspect term embeddings $H_A = \{h_{A_1}, h_{A_2}, ..., h_{A_\kappa}\}$ are derived from H. Here, n_w is the number of context words in the input sentence S. Specifically, the embedding of a specific aspect term A_k is $h_{A_k} = \{h^a_\varphi, h^a_{\varphi+1}, ..., h^a_{\varphi+\tau}\}$, which is the collection of all the embeddings of words belonging to A_k.

3.1.2 Composed Dependency Tags

Additionally, we design composed dependency tags to fully utilize dependency relation tag information based on the aspect-oriented dependency tree (Wang et al., 2020). Concretely, as shown in Figure 1(right), the form of composed dependency tag representation is $r_{A_k,i} = dep_1 : dep_2 : ... : dep_m : m$, where m represents the hops between aspect term node A_k and neighbor node i, and dep_m represents the original dependency tag between two nodes generated by the Biaffine Parser (Dozat and Manning, 2017). Then, we employ a trainable embedding lookup table as a dependency embedding layer. We feed a set of composed dependency tags $R_{tag} = \{r_{A_k,1}, r_{A_k,2}, ..., r_{A_k,M}\}$ into dependency embedding layer to initialize composed dependency tag representations $E_{tag} = \{e_{A_k,1}, e_{A_k,2}, ..., e_{A_k,M}\}$, where M is the number of dependency relation tags. We have different sets of composed dependency tags and their representations for different aspect terms.

3.2 Heterogeneous Graph Construction Block

In this section, we construct a heterogeneous graph to encode inter-aspect relationships and aspect-context relationships effectively.

3.2.1 Shortest Dependency Path

To discover the semantic relationship between aspect terms, we extract the shortest dependency path

(SDP) between aspect terms to represent the relationship. SDP can indicate semantic relationships effectively between two words and ignore irrelevant information, which has been widely used in relation extraction (Bunescu and Mooney, 2005). All tokens on SDP between aspect terms A_ξ and A_η are represented as $SDP_{\xi\eta}$. Furthermore, the features of all tokens belonging to $SDP_{\xi\eta}$ are extracted from H derived from pre-trained BERT through selection operating. The selection operating is based on the index of the token belonging to SDP in the sentence. The $SDP_{\xi\eta}$ representation $H^{SDP}_{\xi\eta} = \sum [h^{SDP}_1, h^{SDP}_2, ..., h^{SDP}_t]$, where t is the number of tokens in the SDP.

3.2.2 Heterogeneous Graph Construction

The heterogeneous graph contains two types of nodes: aspect term node and context node. The feature of each aspect term node is h_{A_k}, which is the element of aspect embedding set H_A. The feature of each context node is derived from word embedding $H_w = \{h_1, h_2, ..., h_{n_w}\}$. Besides, there exist two types of edges in our constructed heterogeneous graph: composed dependency edge and SDP edge. Both of these edges are feature . The feature of each composed dependency edge is derived from composed dependency tag representations $E_{tag} = \{e_{A_k,1}, e_{A_k,2}, ..., e_{A_k,M}\}$, and the feature of each SDP edge between aspect terms A_ξ and A_η is $H^{SDP}_{\xi\eta}$. A simple case of heterogeneous graph construction is shown in Figure 1(right).

3.3 Prediction Block

In the prediction block, we utilize the constructed heterogeneous graph to make predictions. First, we propose a Composition-based Heterogeneous Graph Multi-channel Attention Network (CHGMAN) and feed the constructed heterogeneous graph with its features into CHGMAN. Then, the output layer receives features of aspect terms output by CHGMAN and predicts sentiment polarities.

3.3.1 Composition-based Heterogeneous Graph Multi-channel Attention Network (CHGMAN)

To encode our constructed heterogeneous graph effectively, we propose a novel multi-channel attention network for heterogeneous graphs, which concerns the information of edges, nodes, and node types simultaneously. Figure 2(left) presents the whole view of CHGMAN. Concretely, the goal of CHGMAN is to aggregate information from each neighbor node i to update the representation of the target node t. And the process can be divided into three phases: (1) Multi-channel Attention Calculation, (2) Heterogeneous Massage Passing, and (3) Composed Aggregation.

Multi-channel Attention Calculation. In first phase, CHGMAN tend to calculate the attention score matrices between target node t and each neighbor node $i \in \mathcal{N}(t)$. Due to heterogeneity, based on considering neighbor node features, we further tend to incorporate the information of node type and edge feature into our model. Thus, we design a multi-channel attention mechanism to take node types and edge features into consideration in the phase of attention calculation. Concretely, this phase of CHGMAN is decomposed into three channels: *target-to-neighbor* channel, *target-to-edge* channel and *target-to-type* channel to calculate attention score matrices from different sides. Then, CHGMAN exploits a multi-channel fusion layer to fuse attention score matrices from each channel and get a weighted combination of attention score matrices. Plus, we also employ a multi-head attention mechanism (Vaswani et al., 2017) to enhance the representation ability of CHGMAN.

Firstly, we map the feature of target node t into a query vector $Q^u_{A_k}$ at u-head attention, and h^l_t is the representation of target node t at l-th layer.

$$Q^u_t = Linear^{Q,u}_t \left(h^l_t\right), \tag{1}$$

where $Linear^{Q,u}_t$ is the linear projection for h^l_t. In order to exploit information from neighbor node features, neighbor node types, and edges features, we map the feature of them into key vectors for each channel. Specifically,

$$K^u_{neigh} = Linear^{K,u}_{neigh} \left(h^{neigh,l}_i\right), \tag{2}$$

$$K^u_{edge} = Linear^{K,u}_{edge} \left(h^{edge,l}_{t,i}\right), \tag{3}$$

$$K^u_{type} = Linear^{K,u}_{type} \left(h^{type,l}_i\right), \tag{4}$$

where $Linear^{K,u}_{neigh}$, $Linear^{K,u}_{edge}$ and $Linear^{K,u}_{type}$ are linear projections for each representations, and h^{neigh}_i represents the feature of neighbor node i. Here, neighbor nodes consist of aspect term nodes and context nodes. Besides, $h^{edge}_{t,i}$ is the representation of edge feature between the target node t and neighbor node i. There are two kinds of edges: inter-aspect edges and aspect-context edges. For inter-aspect edges, we utilize SDP representation $H^{SDP}_{\xi\eta}$ as features of

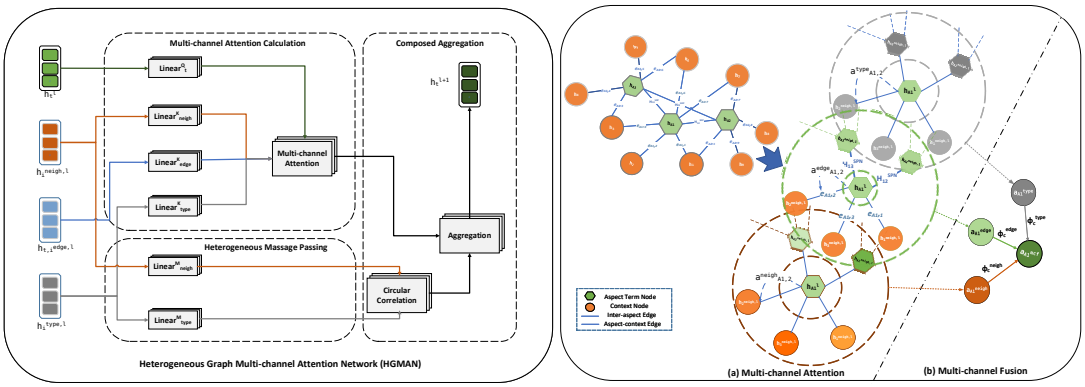

Figure 2: Overview of the proposed CHGMAN (left). Multi-channel attention calculation (right).

inter-aspect edges, and for aspect-context edges, we employ composed dependency tag representations. Additionally, the type representation of neighbor node i is h_i^{type}, and there exist two kinds of type representations corresponding to context node and aspect term node. We introduce a trainable embedding lookup table to initialize type representations.

Next, CHGMAN needs to calculate the correlations between query vectors and key vectors by dot product in each channel. The final multi-channel attention score $Att^u(t,i)$ is computed as follows:

$$Att^u(t,i) = \underset{\forall i \in \mathcal{N}(t)}{Softmax}\left(\alpha_{t,i}^{mcf,u}\right), \tag{5}$$

$$\begin{aligned}\alpha_{t,i}^{mcf,u} &= Mcf\left(\alpha_{t,i}^{neigh,u}, \alpha_{t,i}^{edge,u}, \alpha_{t,i}^{type,u}\right) \\ &= \Phi_c\left(\left[\alpha_{t,i}^{neigh,u}, \alpha_{t,i}^{edge,u}, \alpha_{t,i}^{type,u}\right]\right) \\ &= \Phi_c\left(\left[\frac{(Q_t^u)^\top K_{neigh}^u}{\sqrt{d}}, \frac{(Q_t^u)^\top K_{edge}^u}{\sqrt{d}}, \frac{(Q_t^u)^\top K_{edge}^u}{\sqrt{d}}\right]\right),\end{aligned} \tag{6}$$

where $[\alpha_{Ak,i}^{neigh,u}, \alpha_{Ak,i}^{edge,u}, \alpha_{Ak,i}^{type,u}]$ is the stacked attention matrices of different channels, and Mcf denotes a multi-channel fusion layer with parameters Φ_c, which is a learnable parameter to adjust contributions of input attention matrices automatically. We apply a scaling factor $\frac{1}{\sqrt{d}}$ on $\alpha_{t,i}^u$, which is beneficial to stabilize model training (Vaswani et al., 2017), and d is the dimension of hidden state.

Heterogeneous Massage Passing. Parallel to the computation of multi-channel attention, CHGMAN also passes information from neighbor nodes to the given target node. Taking into account the heterogeneity of nodes, we would like to incorporate the representations of node types attached with node features into the message passing process. Specifically, for target node t and its neighbor nodes $i \in \mathcal{N}(t)$, the multi-heads message is calculate as follows:

$$Msg_{neigh}^u(t,i) = Linear_{neigh}^{M,u}\left(h_i^{neigh,l}\right), \tag{7}$$

$$Msg_{type}^u(t,i) = Linear_{type}^{M,u}\left(h_i^{type,l}\right). \tag{8}$$

To obtain the u-th message head $Msg_{neigh}^u(t,i)$ and $Msg_{type}^u(t,i)$ from neighbor node features and the type representations of neighbor nodes, CHGMAN employs linear projections $Linear_{neigh}^{M,u}$ and $Linear_{type}^{M,u}$ to project them into message vectors.

Composed Aggregation. After multi-channel attention calculation and heterogeneous message passing, CHGMAN needs to aggregate messages from neighbor nodes to the given target node t. Firstly, we employ a circular correlation as a composition operation (Vashishth et al., 2020; Nickel et al., 2016) to combine the messages from node features and node type representations. The operation is computed as follows:

$$\begin{aligned}Msg_{corr}^u(t,i) &= \psi(Msg_{neigh}^u(t,i), Msg_{type}^u(t,i)) \\ &= \mathcal{F}^{-1}\left(\overline{\mathcal{F}(Msg_{neigh}^u(t,i))} \odot \mathcal{F}(Msg_{type}^u(t,i))\right),\end{aligned} \tag{9}$$

where $Msg_{corr}^u(t,i)$ is the composed message representation from neighbor node i, ψ is the circular correlation operation, and $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the Fast Fourier Transform (FFT) and its inverse. The $\overline{\mathcal{F}(\cdot)}$ is the complex conjugate of $\mathcal{F}(\cdot)$, and $\odot$ denotes the Hadamard product.

Then, we aggregate messages from neighbor nodes and update the target node representation The aggregate process at l-th layer is

$$h_t^{l+1} = ||_{u=1}^{U} \sum_{i \in \mathcal{N}(t)} \left(Att^u(t,i) \cdot Msg_{corr}^u(t,i)\right). \tag{10}$$

Dataset	Positive		Neutral		Negative		MM Statistics					
	Train	Test	Train	Test	Train	Test	Train Size	Train MM	Test Size	Test MM	Total Size	Total MM
MAMS-ATSA	3380	400	5042	607	2764	329	11186	11186	1336	1336	12522	12522
Rest14	2164	728	807	196	637	196	3608	2594	1120	835	4728	3429
Laptop14	994	341	870	128	464	169	2328	1396	638	379	2965	1775

Table 1: Statistics of MAMS-ATSA, Rest14, and Laptop14 datasets. (Train|Test|Total) Size and (Train|Test|Total) MM denote the number of instances and multi-aspect multi-sentiment instances in the training, testing, and overall (sum of training and testing) datasets, respectively.

We also incorporate multi-heads attention mechanism into CHGMAN, and $||_{u=1}^{U} x_u$ denotes the concatenation of vectors from x_1 to x_U. The aggregate process at l-th layer is calculated as follows:

3.3.2 Output Layer

In the end, we obtain the output feature of the given target aspect term node A_k from CHGMAN as the final representation h_{final}. Then, we feed h_{final} into a fully connected softmax layer and map it to probabilities over the different sentiment polarities.

$$P(y = c) = softmax(W_P h_{final} + b_P), \quad (11)$$

where W_P and b_P are the weight matrix and bias, respectively. $P \in \mathbb{R}^C$ is the probability distribution for the sentiment polarity of a specific aspect term, where C is the set of sentiment classes. The training objective is to minimize the standard cross-entropy loss with L2-regularization:

$$L(\Theta) = -\sum_{(S,A)\in D} log(y = c) + \Lambda||\Theta||_2, \quad (12)$$

where D is the set of training data, Θ represents all trainable parameters, and Λ is the coefficient of the L2-regularization term.

4 Experiments

4.1 Datasets and Experiment Settings

To verify the effectiveness of our proposed model, we conduct experiments on MAMS-ATSA, Rest14, and Laptop14. MAMS-ATSA is released by (Jiang et al., 2019), all sentences in MAMS-ATSA contain multiple aspect terms, and at least two of them with different sentiment polarities. Also, the start and end positions of each aspect term in a sentence are provided. Rest14 and Laptop14 (Pontiki et al., 2014) have been widely used, but not all sentences have multiple aspect terms in these two datasets. Statistics of these datasets are displayed in Table 1, where (Train|Test|Total) Size and (Train|Test|Total) MM denote the number of instances and multi-aspect multi-sentiment instances on the training, testing, and overall (sum of training and testing) datasets, respectively. Each multi-aspect multi-sentiment instance contains multiple aspects and at least two aspects with different sentiment polarities. Additionally, we count the distribution of instances with the different number of aspects in the three datasets, which is shown in Figure 3. It can be seen that in the case of multiple aspects, instances with two or three aspects account for the majority. And, in MAMS-ATSA, there is no instance with only one aspect.

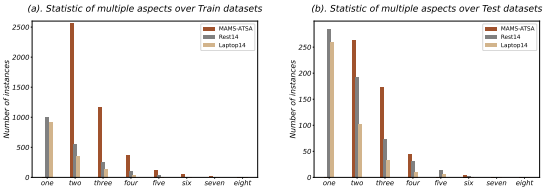

Figure 3: Statistics of multiple aspects over MAMS-ATSA, Rest14, and Laptop14 datasets. The number on the x-axis indicates the number of aspects in each instance in the dataset.

We utilize the last hidden states of the pre-trained BERT-base model for word representations (Devlin et al., 2019), the BERT containing 12 hidden layers, and 768 hidden dimensions for each layer. Moreover, the dimensions of the composed dependency tag embedding and type embedding are also initialized as 768. The hidden dimension of CHGMAN is 128. The dropout rate is 0.2. The number of the epoch is 30. We use Adam optimizer (Kingma and Ba, 2015) while training with the learning rate initialized by 0.00005. Our code will be released later.

4.2 Compared Methods

To evaluate our proposed model CHGMAN, we compare it with the following a series of baselines. **TD-LSTM** (Tang et al., 2016) utilizes LSTMs to model bidirectional contextual information. **ATAE-LSTM** (Wang et al., 2016) tends to combine learned attention embeddings with LSTM. **IAN**(Ma et al., 2017) is introduced to learn the coarse-grained attention for prediction. **MIAD** (Hazarika et al., 2018) simultaneously classifies

Model	MAMS-ATSA		Rest14		Laptop14	
	Accuracy	*Macro-F1*	*Accuracy*	*Macro-F1*	*Accuracy*	*Macro-F1*
TD-LSTM	74.60♭	-	78.00♮	66.73♮	71.83♮	68.43♮
ATAE-LSTM	77.05♭	-	77.20♯	-	68.70♯	-
IAN	76.60♭	-	78.60♯	-	72.10♯	-
RAM	-	-	80.23♯	70.80♯	74.49♯	71.35♯
GCAE	77.59♭	-	77.28♯	-	69.14♯	-
IARM	74.48*	73.66*	80.00	-	73.80	-
MIAD	-	-	79.00	-	72.50	-
ASGCN	44.06*	50.39*	80.77♯	72.02♯	75.55♯	71.05♯
CDT	76.77*	75.77*	82.30♯	74.02♯	77.19♯	72.99♯
Joint+PRET	-	-	81.96	71.80	73.04	69.16
STAGE	-	-	80.10	-	73.10	-
BERT	82.22♭	-	85.62♯	78.28♯	77.58♯	72.38♯
§SDGCN	-	-	83.57	76.47	81.35	**78.34**
§InterGCN	82.49*	81.95*	85.45*	77.64*	78.06*	73.83*
§CapsNet	83.39♭	-	85.93	-	-	-
§RGAT	83.16*	82.42*	86.25*	79.95*	78.21	74.07
§**CHGMAN (Ours)**	**85.05**	**84.29**	**86.88**	**81.62**	**81.52**	77.68

Table 2: Overall performance of different methods on MAMS-ATSA, Rest14, and Laptop14. The results indicated by an asterisk(*) are reproduced by running the released code of the published paper.The results with ♮, ♯ and ♭ are retrieved from (Liang et al., 2020), (Wang et al., 2020) and (Jiang et al., 2019) respectively. The other results except for our model are from the results reported by other baseline papers. The results with § are fine-tuned based on BERT(base).

all aspect terms in a sentence in pace with temporal dependency processing of corresponding sentences by utilizing LSTM. **RAM** (Chen et al., 2017) proposes to learn multi-hop attention on BiLSTM. **GCAE** (Xue and Li, 2018) proposes a convolution network combined with gating mechanisms to control sentiment flow. **IARM** (Majumder et al., 2018) utilizes aspect-aware sentence representation and memory network to fuse neighboring aspect information. **CapsNet** (Jiang et al., 2019) proposes a capsule network-based model to capture relationships between aspects and contextual words. **ASGCN** (Zhang et al., 2019) combines BiLSTM to capture contextual information regarding word orders with a multi-layered GCNs. **CDT** (Sun et al., 2019) encodes both dependency and contextual information by utilizing GCNs and BiLSTM. **BERT** (Devlin et al., 2019) fine-tunes BERT model to predict the sentiment polarity. **SDGCN** (Zhao et al., 2020) proposes a model based on GCNs to capture sentiment dependencies among aspects. **STAGE** (Ma et al., 2019) develops a two stage paradigm to model multi-aspects by using attention mechanism. **Joint+PRET** (Zhou et al., 2020) proposes a LSTM based model and converts sentiment classification to a sequence labeling problem to model relationships among aspects. **RGAT** (Wang et al., 2020) feeds reshaped syntactic dependency graph into RGAT to capture long dependency information.

4.3 Overall Performance

Table 2 shows the overall performance of our model and compared methods on three datasets, and the main evaluation matrices are Accuracy and Macro-averaged F1-score. The results demonstrate our model outperforms all compared methods except for the Macro-F1 of SDGCN on Laptop14. The performance of CHGMAN is affected since MM size in Laptop14 is small relatively (shown in Table 1). Our model exceeds all graph-based methods on evaluation matrices, indicating our proposed model is more effective than other graph-based models. Compared with RGAT, our model takes inter-aspect relationships into consideration. The performance gain towards RGAT indicates that inter-aspect relationships can help predict the sentiment polarity of the given aspect. Moreover, in comparison with InterGCN, CapsNet, and SDGCN, introducing attention mechanism, inter-aspect relationship, aspect-context relationship, and heterogeneity can help our model to distinguish more helpful information and enhance performance. All these results denote that exploiting inter-aspect relationships, aspect-context dependency relationships, and heterogeneity can improve performance.

4.4 Ablation Study

We conduct an ablation study to further analyze the effectiveness of our model. The result of the

Model	MAMS-ATSA		Rest14		Laptop14	
	Accuracy	*Macro-F1*	*Accuracy*	*Macro-F1*	*Accuracy*	*Macro-F1*
CHGMAN	**85.05**	**84.29**	**86.88**	**81.62**	**81.52**	**77.68**
CHGMAN w/o comp	84.37	83.89	86.61	80.45	80.25	76.29
CHGMAN w/o comp&edge	83.53	82.86	86.33	80.19	80.41	75.73
CHGMAN w/o comp&type	84.29	83.34	86.07	79.72	80.09	76.38
CHGMAN w/o hetero	83.69	83.29	86.16	79.03	80.09	76.17

Table 3: Results of ablation study. CHGMAN w/o X denotes the performance of our method after eliminating the corresponding modules, and Y&Z represents deactivating modules Y and Z simultaneously. Additionally, comp, edge, type, and hetero indicate circular correlation operation, target-to-edge channel, target-to-type channel, and adopting homogeneous graphs instead of heterogeneous graphs constructed. These results are fine-tuned based on BERT(base).

ablation study is shown in Table 3. **CHGMAN w/o comp** denotes that CHGMAN removes the circular correlation operation in the composed aggregation process. The performance of **CHGMAN w/o comp** is worse than **CHGMAN**, which indicates conducting the circular correlation operation to combine node and its type features in the aggregation process is effective and conducive to performance improvement. Furthermore, on the basis of **CHGMAN w/o comp**, we removes the attention channel *target-to-edge* and *target-to-type* respectively, which are written as **CHGMAN w/o comp&edge** and **CHGMAN w/o comp&type**. The performance of **CHGMAN w/o comp&edge** and **CHGMAN w/o comp&type** are both inferior than **CHGMAN w/o comp**, demonstrating adding two additional channel *target-to-edge* and *target-to-type* to *target-to-neighbor* is helpful and beneficial to enhancing the handling ability of heterogeneous information. The node types consist of contexts and aspect terms, the performance difference between **CHGMAN w/o comp** and **CHGMAN w/o comp&type** also shows the significance of identifying the information of other aspect terms and contexts in the attention calculation phase, which is conducive to the utilization of both information concurrently. In the meantime, edge features are also essential in the attention calculation phase, *target-to-edge* channel can facilitate the usage of inter-aspect relationships and aspect-context relationships by putting into consideration the features on the edge. The performance of Laptop14 is not clear enough in contrast to the other datasets since the MM size in Laptop14 is less than the other two, as seen in Table 1. **CHGMAN w/o hetero** indicates that we eliminate the impact of heterogeneous graphs constructed and substitute them with homogeneous graphs with edge features. That is to say, we disregard inter-aspect relationships and construct a homogeneous graph for each aspect term, with just one type of node, but the architecture of CHGMAN stays unaffected. The performance of **CHGMAN w/o hetero** has declined in comparison to **CHGMAN**, revealing the heterogeneous graph our constructed is favorable to increasing the efficacy of CHGMAN for MAMSC.

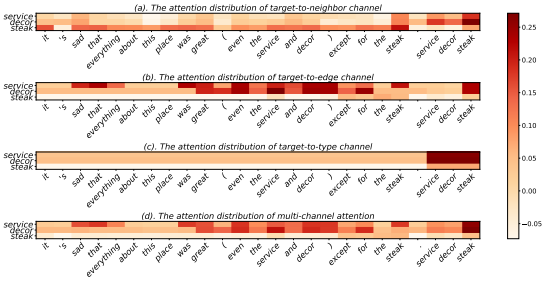

Figure 4: The attention distribution visualization of a specific testing sample.

4.5 Attention Distribution Exploration

To qualitatively illustrate how CHGMAN improves the performance in MAMSC. We post the attention distribution visualization of a testing sample, which is demonstrated in Figure 4. The input sentence is *It's sad that everything about this place was great (even the service and decor) except for the steak.*, there exist three aspect terms: *service*, *decor* and *steak*, and sentiment polarity labels of these three aspect terms are POSITIVE, POSITIVE and NEGATIVE respectively. Figure 4(a) is the attention distribution of *target-to-neighbor*, which shows that for aspect term *steak*, *target-to-neighbor* pays more attention to the context word *sad*, and it is helpful to judge the sentiment polarity to a certain degree. Whereas for *service* and *decor*, *target-to-neighbor* also notices context words *sad* and *great*, although it is ambiguous. Thus for the sake of introducing aspect term nodes, *target-to-*

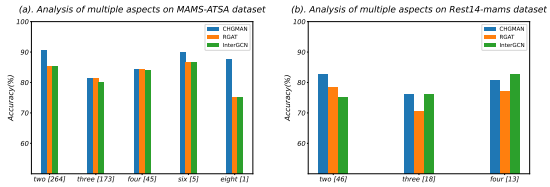

Figure 5: The comparison results of different aspect numbers on MAMS-ATSA and Rest14. The notation M $[N]$ on the horizontal axis represent the number of aspects (M) and the sample size of such a category (N). These results are fine-tuned based on BERT(base).

neighbor pays more attention to *steak* to get more guidance from inter-aspect relationships. As a powerful supplement to *target-to-neighbor*, the attention distribution of *target-to-edge* is displayed in Figure 4(b). For *service* and *decor*, *target-to-edge* notices context words *great* and *even*, and aspect term service is misled by *sad*, thus both of these aspect terms pay attention to aspect term *steak* to seek inter-aspect relationship guidance. In addition, for the *steak*, *target-to-edge* pays attention to *except for the steak*, which is helpful to indicate the NEGATIVE polarity what *steak* conveys. Figure 4(c) shows the attention distribution of *target-to-type*. In this case, for node type, our model tends to focus more on aspect term node to seek more help from inter-aspect relationships, which is reasonable. The attention distribution of overall disentangled attention is shown in Figure 4(d), which is a combination of these three attention terms. We can recognize our model focus on those valuable context words such as *except* when it tends to judge the sentiment polarity of *steak*. For *service* and *decor* that are not easy to distinguish, our model can combine the information both from context words and other aspect terms to make the right judgment.

4.6 Analysis of Multiple Aspects

To further analyze the performance of our model in the multiple aspects situation, we separate each test dataset into different subsets according to the number of aspects in each sentence. Figure 5 demonstrates the test accuracy of CHGMAN and the comparison models RGAT and InterGCN for different subsets on MAMS-ATSA and Rest14 (For Rest14, we only consider sentences containing multiple aspects and at least two aspect terms with different sentiment polarities.). Figure 5(a) shows the analysis of multiple aspects on MAMS-ATSA, and Figure 5(b) shows Rest14. The horizontal axis represents the number of aspects, and the number in square brackets is the sample size of such a category in the dataset. The vertical axis represents the accuracy score. We can observe that CHGMAN is superior to RGAT and InterGCN on most subsets. In the case of four aspects, InterGCN slightly outperforms CHGMAN since Rest14 has fewer multi-aspect multi-sentiment instances (see MM size in Table 1), making it hard to fully demonstrate the effect of the model. Overall, results indicate our model is adept at capturing inter-aspect relationship information when there exist multiple aspects.

5 Conclusion

In this paper, we construct a heterogeneous graph for MAMSC and propose a novel CHGMAN to tackle the heterogeneous graph. Our model predicts sentiment polarities by incorporating inter-aspect relationships, aspect-context relationships, and node heterogeneity. Moreover, our experiments prove the effectiveness of our method.

Acknowledgements

This work is funded in part by the National Natural Science Foundation of China Project (No.U1936213), and the Major Key Project of PCL (PCL2021A06).

References

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT/EMNLP*, pages 724–731. The Association for Computational Linguistics.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *EMNLP*, pages 452–461. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR (Poster)*. OpenReview.net.

Devamanyu Hazarika, Soujanya Poria, Prateek Vij, Gangeshwar Krishnamurthy, Erik Cambria, and Roger Zimmermann. 2018. Modeling inter-aspect dependencies for aspect-based sentiment analysis. In *NAACL-HLT (2)*, pages 266–270. Association for Computational Linguistics.

Binxuan Huang and Kathleen M. Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *EMNLP/IJCNLP (1)*, pages 5468–5476. Association for Computational Linguistics.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *EMNLP/IJCNLP (1)*, pages 6279–6284. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

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 *COLING*, pages 150–161. International Committee on Computational Linguistics.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *IJCAI*, pages 4068–4074. ijcai.org.

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 Gener. Comput. Syst.*, 93:304–311.

Navonil Majumder, Soujanya Poria, Alexander F. Gelbukh, Md. Shad Akhtar, Erik Cambria, and Asif Ekbal. 2018. IARM: inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In *EMNLP*, pages 3402–3411. Association for Computational Linguistics.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic embeddings of knowledge graphs. In *AAAI*, pages 1955–1961. AAAI Press.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval@COLING*, pages 27–35. The Association for Computer Linguistics.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *EMNLP/IJCNLP (1)*, pages 5678–5687. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification. In *COLING*, pages 3298–3307. ACL.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *ICLR*. OpenReview.net.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *ACL*, pages 3229–3238. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *EMNLP*, pages 606–615. The Association for Computational Linguistics.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *ACL (1)*, pages 2514–2523. Association for Computational Linguistics.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *EMNLP/IJCNLP (1)*, pages 4567–4577. Association for Computational Linguistics.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2020. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowl. Based Syst.*, 193:105443.

Jie Zhou, Jimmy Xiangji Huang, Qinmin Vivian Hu, and Liang He. 2020. Modeling multi-aspect relationship with joint learning for aspect-level sentiment classification. In *DASFAA (1)*, volume 12112 of *Lecture Notes in Computer Science*, pages 786–802. Springer.