Expert Systems With Applications 195 (2022) 116560

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Semantic Relatedness Enhanced Graph Network for aspect category sentiment analysis

Tao Zhou[a], Kris M.Y. Law[a,b,*]

[a] School of Engineering, Faculty of Science Engineering and Built Environment, Deakin University, Geelong, VIC 3217, Australia
[b] Department of Industrial Engineering Management, University of Oulu, Finland

ARTICLE INFO

Keywords:
Aspect category sentiment analysis
Semantic relatedness
Edge-Gated Graph Convolutional Network
Aspect–context attention

ABSTRACT

As a variant problem of aspect-based sentiment analysis (ABSA), aspect category sentiment analysis (ACSA) aims to identify the aspect categories discussed in sentences and predict their sentiment polarities. However, most aspect-based sentiment analysis (ABSA) research focuses on predicting the sentiment polarities of given aspect categories or aspect terms explicitly discussed in sentences. In contrast, aspect categories are often discussed implicitly. Additionally, most of the research does not consider the relations between contextual words and aspect categories. This paper proposes a novel Semantic Relatedness-enhanced Graph Network (SRGN) model which integrates the semantic relatedness information through an Edge-gated Graph Convolutional Network (EGCN). We introduce an ontology-based approach and a distributional approach to calculate the semantic relatedness values between contextual words and aspect categories. EGCN with the capability to aggregate multi-channel edge features, is then applied to model the semantic relatedness values in a graphical structure. We also employ an aspect–context attention module to generate aspect-specific representations. The proposed SRGN is evaluated on five datasets constructed based on SemEval 2015, SemEval 2016 and MAMC-ACSA datasets. Experimental results indicate that our proposed model outperforms the baseline models in both accuracy and F1 score.

1. Introduction

Customer needs analysis has been considered a critical part of the product development and improvement process. Nowadays, customer needs have become more dynamic and complicated due to the rapid development of technology and constantly optimised product life cycles (Jeong, Yoon, & Lee, 2019). Hence, commercial companies need to identify these needs timely and improve their products accordingly to gain more market share. With social media and e-commerce platforms, customer online comments and reviews are becoming a more valuable source of analysing the satisfaction of a product or service (Moghaddam, 2015). The online reviews' analysis is to identify product features by assessing the sentiment strengths of user-generated content. The traditional sentiment analysis aims to identify the sentiment polarity as a whole, which is not enough to evaluate a product's performance. Therefore, aspect-based sentiment analysis (ABSA) that targets on inferring the fine-grained sentiment polarity of aspect categories has drawn much attention from researchers and enterprises.

ABSA contains a few subtasks, which are aspect term extraction (ATE), aspect category detection (ACD) and sentiment classification (SC) (Do, Prasad, Maag, & Alsadoon, 2019). ATE aims to identify the aspect terms presenting in the sentence while ACD aims to detect aspect categories mentioned in the sentence. Fig. 1 shows an example sentence from dataset SemEval 2016 Laptop which explains these three subtasks. For ATE, three words "*use*", "*quality*" and "*price*" are identified as aspect terms. ACD detects three aspect categories indicated by these three aspect terms: "*Usability*", "*Quality*" and "*Price*". For each aspect, SC identifies the sentiment polarity based on the related sentiment words, such as "*easy*" for "*use*" and "*Usability*". These three tasks are strongly interrelated. Based on these three subtasks, two types of ABSA problems have been widely studied: aspect term sentiment analysis (ATSA) that classifies the sentiments towards identified aspect terms and aspect category sentiment analysis (ACSA) that aims to solve ACD and SC simultaneously. However, ATSA focuses on the explicit aspect terms presenting in the sentence while aspects are frequently discussed implicitly. Therefore, this paper focuses on the ACSA problem for detecting multiple aspect categories and sentiments discussed in one sentence.

In recent years, deep learning-based approaches have achieved great outcomes for the ABSA tasks due to the capability of extracting feature

[*] Corresponding author at: School of Engineering, Faculty of Science Engineering and Built Environment, Deakin University, Geelong, VIC 3217, Australia.
E-mail address: kris.law@deakin.edu.au (K.M.Y. Law).

https://doi.org/10.1016/j.eswa.2022.116560
Received 17 May 2021; Received in revised form 26 August 2021; Accepted 16 January 2022
Available online 5 February 2022

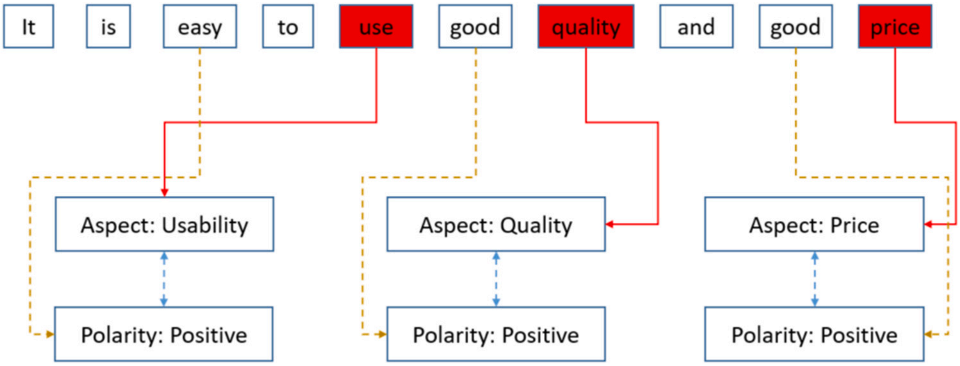


Fig. 1. Sample text from SemEval2016 laptop.

information and representing this information as low-dimensional vectors (Li, Fu et al., 2020). To capture the interaction between the aspects and the context, attention-based models have been developed to allocate greater attention to the relevant information in the context. Wang, Huang, Zhu, and Zhao (2016) applied an attention layer with a bidirectional LSTM model (ATAE-LSTM) to generate aspect focused representation for aspect category sentiment classification. Instead of using LSTMs for modelling contextual representations, an attentional encoder network (AEN) proposed by Song, Wang, Jiang, Liu, and Rao (2019) employed a multi-head attention layer. Xu, Zhu, Dai, and Yan (2020) proposed a multi-attention network (MAN), which utilised an intra-level and an inter-level attention layer to extract interactive information between aspects and context. These attention-based models have shown good performance in learning aspect-specific representations (Zhao, Hou, & Wu, 2020). Furthermore, knowledge-based approaches have been demonstrated effective to identify the semantic relationship of words concerning aspects and sentiments (Meškelė & Frasincar, 2020). For example, Tubishat, Idris, and Abushariah (2020) proposed a supervised aspect extraction algorithm based on pattern-based rules that has shown a great performance on customer review datasets. To exploit the dependency relations between words, graph neural networks (GNNs) have been applied to leverage the syntactic information in the dependency tree generated by dependency parsers. Sun, Zhang, Mensah, Mao, and Liu (2019) proposed a convolution over a dependency tree (CDT) to model the dependency relations between context words and aspect terms. Similarly, Zhang, Li, and Song (2019) utilised a multi-layer graph convolutional network (GCN) with an LSTM encoder to produce aspect-specific features based on the dependency tree.

Despite the success of these models, two disadvantages could potentially impede the improvement of the performance on the ACSA task. Firstly, unlike ATSA, the aspect categories and the associated aspect terms are often discussed implicitly in the sentences for the ACSA. Therefore, it is difficult to explicitly exploit the relations between context words and aspect categories through linguistic rules, such as dependency parsers. Besides, most of the models discussed above are initially designed for ATSA. Annotations for aspect terms are needed during the training process, which is very expensive in labour cost. Secondly, most of these models are built for each given aspect category individually without considering the relations between different aspect categories. This disadvantage could result in the mismatching of information between contextual words and aspect categories, especially when there are many pre-defined aspect categories.

This paper proposes a Semantic Relatedness-enhanced Graph Network (SRGN) model for ACSA to address these problems. SRGN aims to exploit the relation between aspects and context words over their semantic similarity and relatedness. Some studies have demonstrated the effectiveness of semantic similarity-based approaches in ATE and ACD (Araque, Zhu, & Iglesias, 2019). Priyantina and Sarno (2019) utilised semantic similarity-based approaches for matching the latent topics obtained from LDA with the pre-defined aspects, which improved aspect extraction performance. Rana, Cheah, and Rana (2020) extended their work by proposing a multi-level approach to extract and classify the implicit aspect clues based on co-occurrence and Normalised Google Distance. In this paper, to explore implicit relations, both ontology-based and distribution-based semantic similarity and relatedness are introduced. SRGN contains four main components: the contextual embedding module, semantic enhanced edge-gated GCN (S-EGCN) module, aspect–context attention module and classifier module. Our model utilises semantic similarity and relatedness in the S-EGCN module to build a semantic relatedness (SR) graph. An Edge-gated GCN (EGCN) is then used to model the SR graph and generate aspect category-specific representations. Based on the aspect category-specific representations, two layers of aspect–context attention are applied to generate weighted representations for sub-tasks ACD and SC respectively. The main contributions of our work can be summarised as follows:

- We propose a novel S-EGCN module to build an SR graph based on semantic similarity and relatedness to explore explicit and implicit connections between context and pre-defined aspect categories.
- We propose an improved version of GCN, i.e. EGCN, to model multiple edge features while the original GCN is limited to one-dimensional edge features.
- We propose a training scheme by combining loss functions for both ACD and SC.
- A series of experiments are conducted to compare our proposed model with the baseline models on five benchmark datasets, and the results demonstrate the effectiveness of SRGN.

The rest of this paper is structured as follows. Previous studies on aspect-based sentiment analysis, graph neural network and semantic similarity and relatedness are presented in Section 2. The detailed description of the proposed model's architecture is presented in Section 3, while the experiments' design and results are shown in Section 4. Section 6 presents the conclusions and future works.

2. Literature review

2.1. Aspect-based sentiment analysis

Over the past years, many neural network-based models have been developed to deal with ABSA tasks. Convolutional Neural Network (CNN) is a popular deep learning model with the ability to extract essential n-gram features (Do et al., 2019). Poria, Cambria, and Gelbukh (2016) applied CNN architecture for aspects extraction and integrated linguistic patterns for performance improvement. Gu, Gu, and Wu (2017) proposed a two-level model which contains multiple CNNs at level 1 to detect aspect categories and one CNN at level 2 for semantic polarity. Compared to CNN, Recurrent Neural Network (RNN) can explore the dependency in the context and deal with flexible length of input sequence (Do et al., 2019). Chen, Xu, He, and Wang (2017) introduced a bidirectional LSTM with CRF model in opinion target extraction. The attention mechanism has been widely studied to enhance the sentence representation by concentrating on the important part of the sentence (Zhao et al., 2020). Gan, Wang, Zhang, and Wang (2020) introduced a sparse attention based CNN for the sentiment classification on a given aspect. A separable dilated CNN was proposed to extract and integrate contextual semantic information based on diverse dilation rates in their model. A spare attention layer was utilised to force the model to focus more on sentiment oriented information (Gan et al., 2020).

Hybrid models combining neural networks and knowledge-based approaches have been widely studied in recent years. Meškelė and Frasincar (2020) utilised a lexicalised domain ontology to infer indirect relations among aspects and sentiment words. A neutral attention model was proposed to predict the sentiment polarity for the targeted aspect category. Chauhan, Meena, Gopalani, and Nahta (2020) proposed a two-step hybrid model for ATE. A Dependency parser was used to extract phrases as aspect terms and an attention-based bi-LSTM model was trained using the extracted aspects as labelled data. Instead of using only one type of linguistic knowledge, Li, Qi, Tang, and Yu (2020) proposed a bidirectional LSTM with a self-attention model to integrate multi-channel features including part-of-speech feature, position value and dependency parsing. In their model, linguistic features were concatenated with the pre-trained word vectors and integrated through a bi-LSTM layer and a self-attention layer.

2.2. Semantic similarity and relatedness

Semantic similarity and relatedness play an important role in understanding textual information by measuring the semantic resemblance between terms (Majumder, Pakray, Gelbukh, & Pinto, 2016). Strictly speaking, the definitions of semantic similarity (SS) and semantic relatedness (SR) are distinct. SS typically refers to how taxonomical alike for two terms while terms are classified into different concepts or types (Lofi, 2015). SR is a more general concept that relies not only on taxonomic relations but also on non-taxonomic relations, such as functionality and cause–effect (Zhu, Yang, Huang, Guo, & Zhang, 2019). For example, "*car*" is similar to "*bus*" as they are all vehicles, they would have high semantic similarity in this case. However, "*car*" is not similar to "*tyre*" and "*wheel*" but they are very related.

Approaches for measuring semantic similarity and semantic relatedness can be classified into ontology-based and corpus-based measurements. Ontology-based approaches measure the likeness of concepts based on the ontologies that define well structured and unambiguous knowledge representations (Lofi, 2015). One of the widely used domain ontologies is WordNet. WordNet is a manually complied conceptual thesaurus that structures English words into a semantic relation network (Zhu et al., 2019). Words in WordNet are linked and grouped into synonym sets, i.e. synsets. These synsets are connected lexically through relations such as hyponymy ("*is-a*" relations) and meronymy ("*part-of*" relations). Other ontologies are also proposed including Freebase (Wang, Song, Li, Zhang, & Han, 2015) and YAGO (Zhu & Iglesias, 2017). Based on the networks or graphs defined by these ontologies, semantic similarity can be estimated through edge-counting measures and information content-based measures. Edge-counting measures compute the path length of concepts in the network to estimate the degree of similarity and the shorter path length represents a closer semantic relationship (Priyantina & Sarno, 2019). Wu and Palmer similarity (Lofi, 2015) was proposed to measure the similarity considering semantic distance (number of edges) as well as the depth of least common subsummer (LCS) in the taxonomy:

$$Sim_{wup}(x_1, x_2) = 2 * \frac{depth(LCS(x_1, x_2))}{depth(x_1) + depth(x_2)} \tag{1}$$

Instead of merely considering the structure of ontologies, information content (IC) defined by the probability of appearance of a concept in a textual corpus is introduced (Lofi, 2015) and we can formulate $IC(x) = -\log p(x)$. Several IC-based measurements have been proposed, such as Resnik similarity, Lin similarity and Jian similarity. Resnik similarity measures the IC of the least common subsummer (LCS) for two concepts, while Lin and Jian similarity are improved versions of Resnik similarity by scaling and normalisation (Araque et al., 2019; Lofi, 2015). The Resnik, Lin and Jian similarity are formulated as Eqs. (2), (3) and (4), respectively.

$$Sim_{res}(x_1, x_2) = IC(LCS(x_1, x_2)) \tag{2}$$

$$Sim_{lin}(x_1, x_2) = \frac{2 * IC(LCS(x_1, x_2))}{IC(x_1) + IC(x_2)} \tag{3}$$

$$Sim_{jc}(x_1, x_2) = \frac{1}{IC(x_1) + IC(x_2) - 2 * IC(LCS(x_1, x_2))} \tag{4}$$

Apart from ontology-based measurements, distributional approaches aim to measure the semantic relatedness relying on the information distribution in a large corpus. The hypothesis is that if two concepts or terms are semantically related, they are more likely to have more common co-occurrences (Li, Zhou, & Li, 2015). Pointwise mutual information (PMI) was proposed to measure the ratio of the probability of co-occurrence for two concepts in the same corpus to the probability of each concept's appearance (Lofi, 2015). The PMI can be expressed as:

$$PMI(x_1, x_2) = \log \frac{p(x_1, x_2)}{p(x_1) * p(x_2)} = IC(x_1) + IC(x_2) - IC(x_1, x_2) \tag{5}$$

Web search engines, such as Google and Baidu, are used to compute the PMI based on the hit counts of search terms (Lofi, 2015). Cilibrasi and Vitanyi (2007) defined Normalised Google Distance (NGD) for evaluating the relatedness of search keywords. Besides, Wikipedia is also used as a database for calculating PMI values (Salahli, 2009). Another strategy of representing the relatedness is to construct each word as a high dimensional vector and compute the distance between two word vectors such as cosine similarity (Lofi, 2015).

Semantic similarity and relatedness have been widely applied for the ACD task (Rana & Cheah, 2016). Li et al. (2015) proposed a frequency-based extraction method augmented by PMI to measure the semantic similarity between aspects and target entities. Liu, Liu, Zhang, Kim, and Gao (2016) applied a double propagation method to extract initial aspects. They then utilised the average cosine similarity of word vectors to recommend a new aspect based on the extracted aspects. Kang and Zhou (2017) extracted the aspects by extending double propagation (DP) with dependency parser-based rules and utilised frequency-based and semantic similarity-based filters for pruning irrelevant aspects. Rana and Cheah (2017) proposed a two-fold rule-based model for aspect extraction. They utilised sequential patterns-based rules to extract explicit aspects first and applied a frequency-based approach with NGD to filter aspects that are not semantically related. Except for these rule-based approaches, semantic similarity and relatedness have also been used with topic modelling-based approaches, especially Latent Dirichlet Allocation (LDA). Miller, Dligach, and Savova (2016) and Shams and Baraani-Dastjerdi (2017) used LDA to generate clusters of aspect terms and adopted co-occurrences similarity relations to classify these aspect terms into pre-defined categories.

2.3. Graph neural network

Graph neural networks (GNNs) are deep learning-based models used for processing graph data, which have been proven effective in modelling and aggregating information from graph structure (Zhou et al., 2018). As one of the most studied variants of GNN, the graph convolutional network (GCN) proposed by Kipf and Welling (2016) applies graph convolution operation on the node representation by aggregating the information from its neighbouring nodes (Zhou et al., 2018). The general node updating process of GCN can be formulated as:

$$H^{l+1} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^l W^l + b^l) \tag{6}$$

where H^l is the hidden vector of nodes at the lth layer of GCN; σ is a non-linear activation function, i.e. the ReLU function; A is the adjacency matrix where $A_{ij} = 1$ indicates the ith node and jth node are connected, otherwise $A_{ij} = 0$; D is the degree matrix of A where $D_{ii} = \sum_{j \to i} A_{ij}$ and $j \to i$ mean that the jth node is connected to the ith node, and W^l and b^l are weights and bias of the linear transformation.

Because of its great ability to deal with graph data, GCN has achieved state-of-the-art ABSA results (Zhou et al., 2018). Veyseh et al. (2020) proposed a gated-GCN over the dependency tree for ABSA. In their model, gate vectors were used to generate customised representation vector towards aspect terms. To better integrate dependency relations, the importance scores defined by the dependency path were used for regulating the training process. Zhao et al. (2020) employed GCN over the attention mechanism to exploit the relations between different aspect categories. Chen, Tian, and Song (2020) proposed a directional GCN to perform ATE and SC simultaneously and leverage the syntactic information from the dependency tree. Except for the dependency tree, other types of relationships are also utilised. Zhou, Huang, Hu, and He (2020) proposed a syntax- and knowledge-based GCN (SK-GCN) model that integrates dependency tree and knowledge graph. The proposed SK-GCN built an integrated graph that merges both the dependency tree and knowledge graph. Xu, Zhao, and Liu (2020) proposed a heterogeneous GCN on a graph defined by different edge features, i.e. dependency, TF–IDF, and distance between nodes. Apart from GCN, other variants of GNN are also adopted for ABSA, such as graph attention neural network (GAT) (Bai, Liu, & Zhang, 2020) and graph LSTM (Zhang, Liu, & Song, 2018).

Nevertheless, these studies adopted GNNs that are limited to one-dimensional edge features. Even though more than one type of edge features was adopted in Zhou et al. (2020) and Xu, Zhao et al. (2020), they were integrated into a graph with one-dimensional edge matrix. However, merely merging different edge features into a single matrix could lead to the loss of important information as different edge features could have a diverse range of values. Therefore, in this paper, an Edge-gated GCN (EGCN) is proposed to deal with multi-dimensional edge features.

3. Methodology

The aspect category sentiment analysis (ACSA) problem can be formulated as follows. Given M pre-defined aspect categories, $A = \{a_1, a_2, \dots, a_m, \dots, a_M\}$, K sentiment polarity labels and a customer review sentence that consists of N words, $S = \{s_1, s_2, \dots, s_n, \dots, s_N\}$, the aim of ACSA is to identify C aspect categories that are mentioned in the sentence S, $A^S = \{a_1^S, \dots, a_c^S, \dots, a_C^S\} \subset A$ and predict the corresponding sentiment polarity for each identified aspect category, $K^S = \{K_1^S, \dots, K_C^S\}$ where $K_c^S \in \{1, 2, \dots, K\}$. For ACSA, each aspect category might include more than one aspect word. Given that there are $\hat{M}$ unique aspect words, $W_{asp} = \{w_1, \dots, w_{\hat{M}}\}$, the aspect category a_m consists of $\hat{m}$ aspect words, $a_m = \{w_1^m, \dots, w_{\hat{m}}^m\}$. Different aspect categories may share some identical aspect words, and aspects words are not necessarily contained in sentence S.

Instead of unifying ACD and SC as a multi-class classification problem as proposed in many studies (Sun, Huang, & Qiu, 2019; Wang et al., 2016), a hierarchical approach is adopted to deal with the ACSA problem. Fig. 2 shows the comparison of the multi-class classification and the hierarchy classification. For the multi-class classification, the aspect-sentiment pairs of each input S, $\{(a_1^S, K_1^S), \dots, (a_c^S, K_c^S), \dots, (a_C^S, K_C^S)\}$, are converted to binary vectors with $K + 1$ dimensions where the first K dimensions represent the sentiment label space and the last dimension represents the presence or absence of each aspect. For example, the ground-truth label of the aspect "*Usability*" is represented by $[1, 0, 0, 0]$ as the sentiment polarity for the aspect is positive. Similarly, the ground-truth of "*Price*" is labelled as $[0, 0, 0, 1]$ as this aspect is not mentioned. However, the multi-class classification approach fails to consider the inter-relationship between ACD and SC. It is not necessary to perform SC if an aspect is not mentioned. The hierarchical classification approach models the ACD as a multi-label classification problem for all pre-defined aspects. For each detected aspect, SC is performed as a multi-class classification problem. In Fig. 2, aspect "*Usability*" and "*Quality*" are first detected and their sentiment polarities are further predicted.

The flowchart and the detailed framework of the proposed Semantic Relatedness-enhanced Graph Network (SRGN) model are shown in Fig. 3 and Fig. 4 respectively. The overall structure consists of four main components:

- **Contextual Embedding Module** maps each word from both context and pre-defined aspect categories into a high dimensional embedding vector.
- **Semantic-Enhanced Edge-Gated GCN (S-EGCN) Module** captures the relationship of words through injecting the semantic relatedness scores and generates new feature representation vectors of aspect words.
- **Aspect–Context Attention Module** calculates the attention weights of contextual words with respect to each aspect category.
- **Classifier Module** identifies the aspect categories for each input context and predicts their corresponding sentiment polarity.

In the rest of this section, we will introduce these components in detail.

3.1. Contextual embedding module

In this section, we adopt the pre-trained Glove embedding matrix (Pennington, Socher, & Manning, 2014) and the pre-trained BERT model (Devlin, Chang, Lee, & Toutanova, 2018) to generate initial word embedding for each word from the context and the aspect categories. The input sentence is re-constructed by combining the context and all unique aspect words to obtain the contextual word embedding. The newly constructed sentence $S_{aux} = \{s_1, \dots, s_N, w_1, \dots, w_{\hat{M}}\}$ is then transformed to the embedding $H = \{h_1, h_2, \dots, h_n, \dots, h_{\hat{N}}\}$ by mapping each word s_n into a real-valued vector h_n, where $\hat{N} = N + \hat{M}$, $h_n \in \mathbb{R}^{d_h}$ and d_h is the embedding dimension.

3.1.1. Glove embedding

For applying Glove embedding, we conduct a look-up operation for each word s_n, $s_n \in S_{aux}$, over the pre-trained Glove word embedding matrix $E_{glove} \in \mathbb{R}^{d_h \times |V|}$ according to the token id of the word, where $|V|$ is the size of the vocabulary. L layers of multi-head self-attention (MHSA) are deployed on top of initial embedding vectors H_0 to generate contextual embedding. The f_p function in (7) is a projection function with trainable parameters $W_q \in \mathbb{R}^{d_h \times d_k}$, $W_k \in \mathbb{R}^{d_h \times d_k}$, $W_v \in \mathbb{R}^{d_h \times d_v}$, where d_k and d_v are the hidden dimensions of key K and value V. The scaled dot-product attention procedure is shown in (8), and (9) performs self-attention for h times with $head_i = Attention(Q, K, V)$ and integrates the learned results. d_k and d_v equal to d_h/h and h is the number of heads. By applying L layers of MHSA, we can obtain the contextual hidden states of input context words $H_L^c = \{h_1^c, h_2^c, \dots, h_N^c\}$ and the aspect words $H_L^a = \{h_1^a, h_2^a, \dots, h_{\hat{M}}^a\}$.

$$f_p = \begin{cases} Q = W_q \cdot H_0 \\ K = W_k \cdot H_0 \\ V = W_v \cdot H_0 \end{cases} \tag{7}$$

$$Attention(Q, K, V) = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}})V \tag{8}$$

$$H_l = Concat(head_1, \dots, head_h) \cdot W_o^l \tag{9}$$

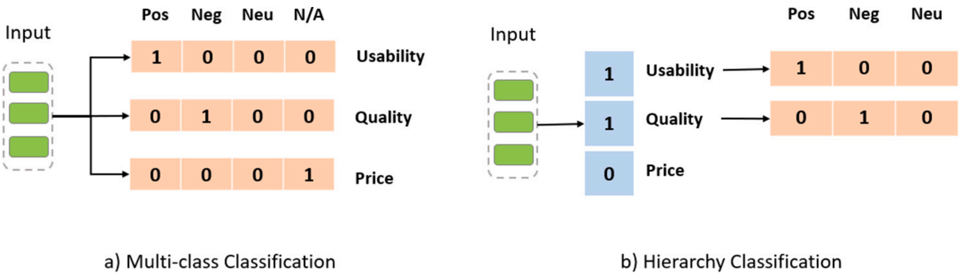


Fig. 2. The comparison between multi-class classification and hierarchy classification.

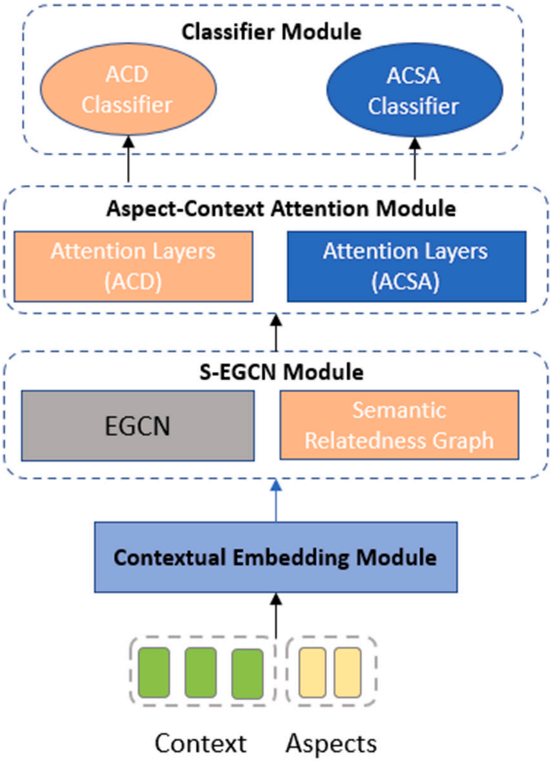


Fig. 3. The flowchart of the proposed model.

3.1.2. BERT embedding

BERT is a pre-trained language representation model based on the bi-directional transformers. The BERT model is initially trained on large scale corpora over several different pre-training tasks including masked language model (MLM) which masks 15% of input tokens (Devlin et al., 2018). In this paper, we adopt BERT Encoder to provide contextual embedding vectors.

The token embedding of an input sequence for BERT is constructed as $[\langle CLS\rangle, s_1, \dots, s_N, \langle SEP\rangle, w_1, \dots, w_{\hat{M}}, \langle SEP\rangle]$ where special token "CLS" is always defined as the first token of the input sentence and special token "SEP" is used to separate the context words and the aspect words. The BERT Encoder layer consists of L successive transformer encoder blocks. Each transformer encoder layer is constructed by a number of MHSA layers and fully connected feed-forward layers. The output of lth transformer block is represented as:

$$H_l = Transformer_l(H_{l-1}), l \in [1, L] \tag{10}$$

The BERT model uses WordPiece language model to split the words into characters and learns to merge these characters into sub-words. Training on large corpora could provide more contextual information and cover a wider range of Out-of-Vocabulary (OOV) words (Peters et al., 2018). Therefore, we use the average of the hidden states of all sub-words of the word s_n, $s_n \in S_{aux}$, to represent the hidden state of word s_n. As a result, we can obtain the hidden states of the context words H_c^L and the aspect words H_a^L.

3.2. Semantic-enhanced edge-gated GCN

In this section, we construct a semantic relatedness (SR) graph $G = (\hat{N}, M)$ that consists of $\hat{N}$ nodes where the ith node represents the ith word s_i, $s_i \in S_{aux}$. $X_i \in \mathbb{R}^{d_h}$, $i = 1, 2, \dots, \hat{N}$, is used to represent the node feature vector of the ith node. The original X_i equals to the contextual hidden state of h_i where $h_i \in H_L^c \cap H_L^a$. M is an $\hat{N} \times \hat{N} \times P$ matrix of edge features of the graph where P is the number of channels for edge features. $M_{i,j}^p$, $i = 1, 2, \dots, \hat{N}$, $j = 1, 2, \dots, \hat{N}$, $p = 1, 2, \dots, P$, represents the pth edge feature connecting the ith and the jth node.

3.2.1. Semantic relatedness graph

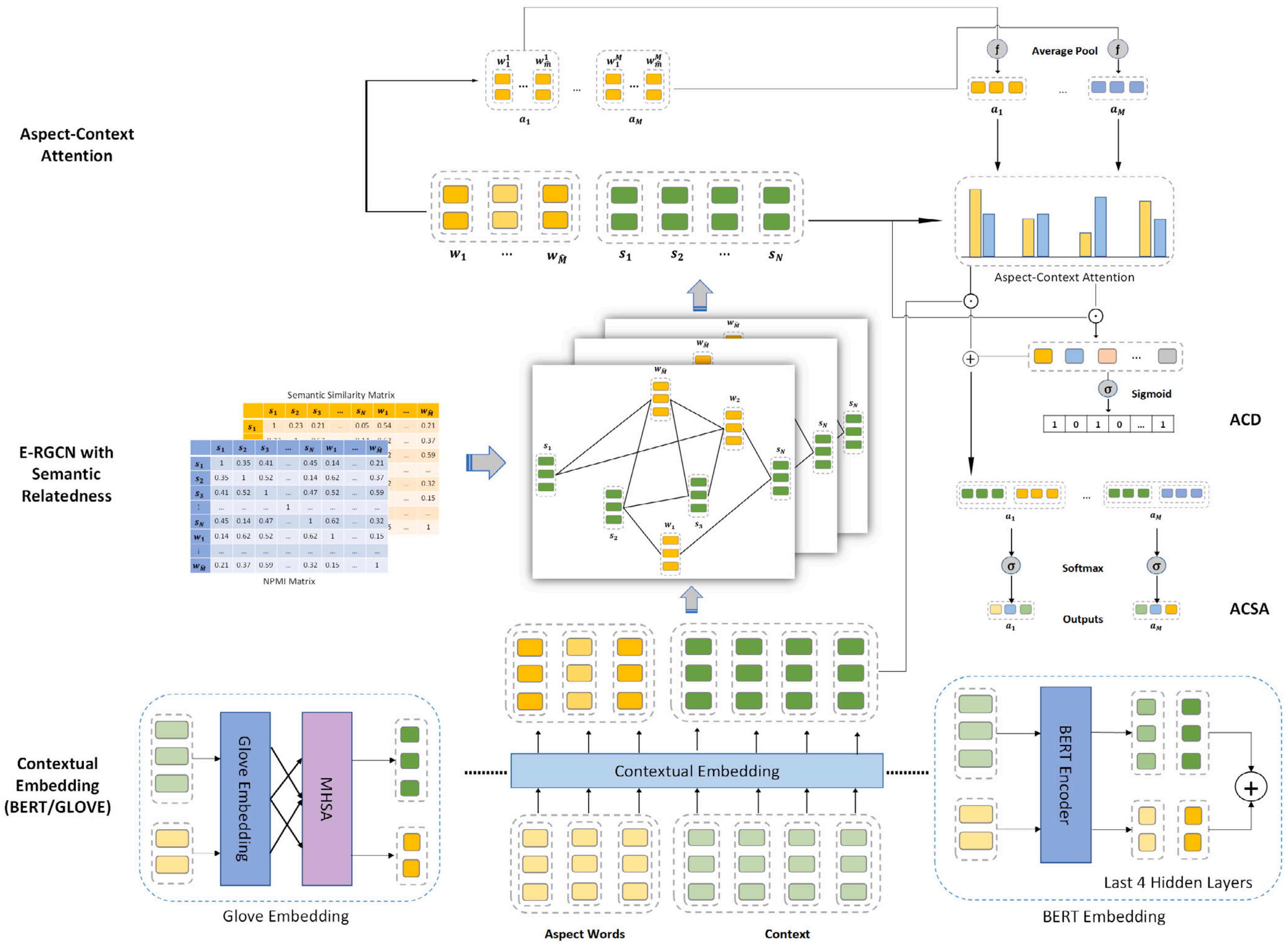
Following (Sun, Zhang et al., 2019), Zhang et al. (2019) and Veyseh et al. (2020), an adjacency matrix based on the dependency tree is constructed as the first channel edge feature to discover the inner relations between contextual words, denoted by $M^1 \in \mathbb{R}^{\hat{N} \times \hat{N}}$. If the dependency relation between node ω_i and ω_j, denoted by (ω_i, ω_j), is true, the corresponding edge feature $M_{ij}^1 = 1$, as shown by Eq. (11). In this paper, we set $M_{ij}^1 = 1$ when $\omega_i = \omega_j$ and $(M^1)^T = M^1$.

$$M_{ij}^1 = \begin{cases} 1 & \text{if } \omega_i = \omega_j \\ 1 & \text{if } (\omega_i, \omega_j) \text{ is True} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Apart from the adjacency matrix that defines the dependency of contextual words, semantic similarity and relatedness scores are used to capture the relations between contextual words and aspect words. As stated, to estimate the semantic similarity and relatedness, two types of methodology have been widely studied, namely ontology-based approaches and distributional approaches (Batet & Sánchez, 2016). The ontology-based approaches measure the similarity of words based on their taxonomic relationships (i.e. is-a, class inclusion, etc.) defined by the structured domain knowledge without exploring the non-taxonomic relationships (Zhu & Iglesias, 2017). The distributional approaches measure the relatedness by exploiting over a large text corpora (e.g. co-occurrence) without considering the ontological meaning of words (Priyantina & Sarno, 2019). We introduce Lin similarity and normalised pointwise mutual information (NPMI) as edge features of the proposed graph to leverage both approaches.

As an ontology-based measurement formulated based on information content (IC), Lin similarity has shown a good performance (Batet & Sánchez, 2016). Besides, Lin similarity provides measurement scores within the range of $[0, 1]$ which could avoid the possible dominance over the distributional approach. Hence, the edge feature on the second channel is defined by the Lin similarity score between context words and aspect words:

$$M_{ij}^2 = \begin{cases} 1 & \text{if } \omega_i = \omega_j \\ Sim_{lin}(\omega_i, \omega_j) & \text{if } \omega_i \in W_{asp} \\ Sim_{lin}(\omega_i, \omega_j) & \text{if } \omega_j \in W_{asp} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$



	s_1	s_2	s_3	...	s_N	w_1	...	$w_{\hat{M}}$
s_1	1	0.35	0.41	...	0.45	0.14	...	0.21
s_2	0.35	1	0.52	...	0.14	0.62	...	0.37
s_3	0.41	0.52	1	...	0.47	0.52	...	0.59
⋮	...	...	...	1	...	...	...	...
s_N	0.45	0.14	0.47	...	1	0.62	...	0.32
w_1	0.14	0.62	0.52	...	0.62	1	...	0.15
⋮	...	...	...	...	...	...	...	...
$w_{\hat{M}}$	0.21	0.37	0.59	...	0.32	0.15	...	1

Fig. 4. The detailed architecture of the proposed model.

where $Sim_{lin}(\cdot)$ refers to Eq. (3) and W_{asp} is the set of aspect words.

For complementing the limitation of Lin similarity, PMI measures the relatedness of two words by estimating the probability of their co-occurrence in the same corpus. To unify the value range of PMI with Lin similarity, a normalised PMI (NPMI) (Lofi, 2015) is introduced and used to define the third channel edge feature:

$$NPMI(\omega_i, \omega_j) = \frac{PMI(\omega_i, \omega_j)}{IC(\omega_i, \omega_j)} \tag{13}$$

$$M_{ij}^3 = \begin{cases} 1 & \text{if } \omega_i = \omega_j \\ NPMI(\omega_i, \omega_j) & \text{if } \omega_i \in W_{asp} \\ NPMI(\omega_i, \omega_j) & \text{if } \omega_j \in W_{asp} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where the calculation of PMI follows Eq. (5) and $IC(\omega_i, \omega_j)$ can be computed by counting the occurrence of words ω_i and ω_j in a textual corpus.

To better explain how the SR graph is constructed, an example of a similarity matrix and NPMI matrix is presented in Fig. 5. An example sentence "*small and light weight speakers, keyboards good sized and easy to use*" is used for demonstration. Assuming that 3 aspect categories, namely "*design features*", "*quality*" and "*price*", are pre-defined, the set of unique aspect words is { "*design*", "*features*", "*quality*", "*price*"}. In the data preprocessing stage, stop words including most common words that contain little meaningful information related to aspects, such as "*and*", "*is*", "*of*", are assigned with value of zeros. The values are normalised to range [0, 1].

3.2.2. Edge-gated GCN

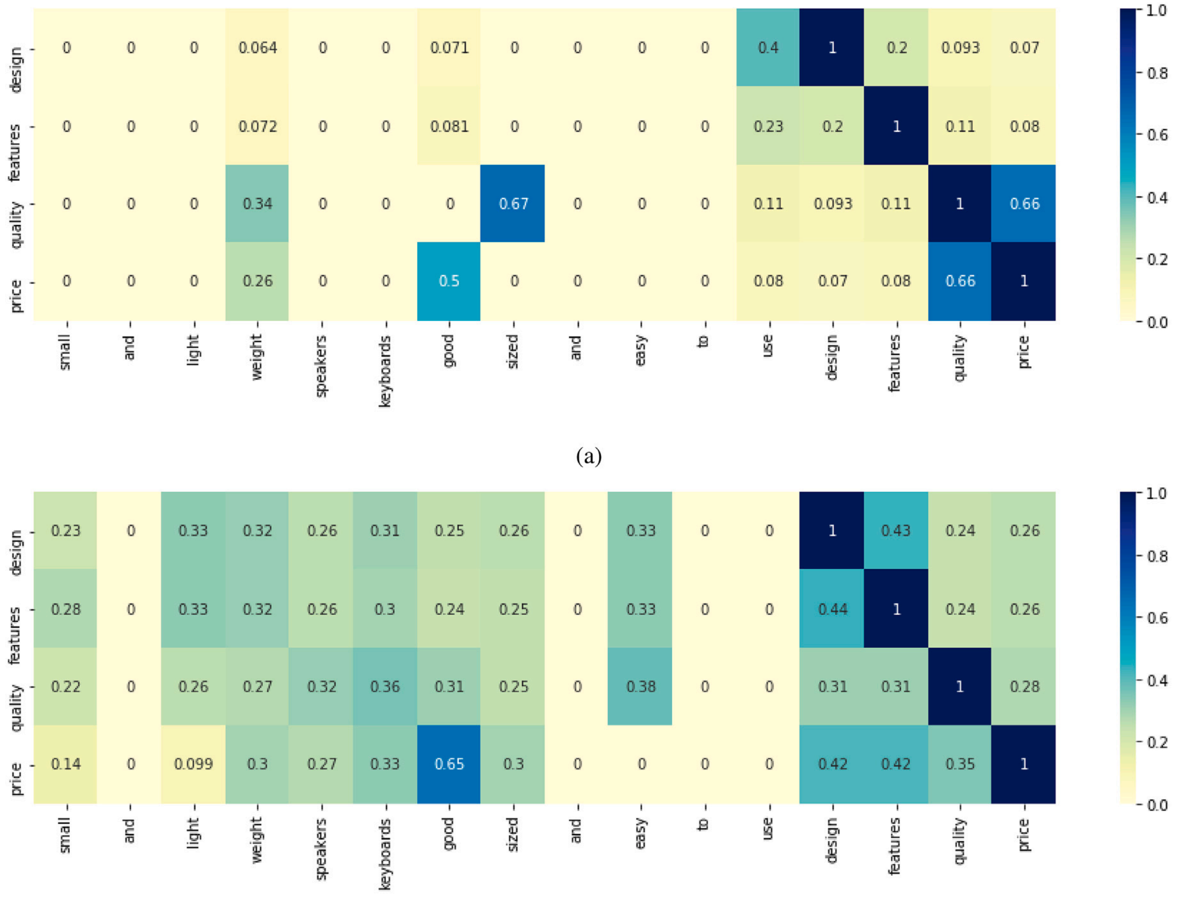
In this section, an edge-gated GCN (EGCN) is proposed to model and learn multiple channels of edge features defined in the semantic relatedness graph. The EGCN leverages the model architecture with edge gating mechanism proposed by Bresson and Laurent (2017) and the feature aggregation approach of Gong and Cheng (2019) to filter and combine important information from edge features.

For an $\hat{L}$-layers EGCN, the ith node representation of the lth layer is X_i^l, where $l = 1, 2, \ldots, \hat{L}$ and $X_i^0 = h_i$. Each node representation is updated based on its representation from the last layer of EGCN and its neighbours' features filtered by an edge gate. We define the updating procedure as following equations. For simplifying the expression, the bias of linear transformation is not included.

$$E_{ijp} = W_0^p M_{ij}^p \tag{15}$$

$$X_i^{l+1} = X_i^l + \sigma(BN[\|_{p=1}^P (W_1^l X_i^l + \sum_{j \to i} g(E_{ijp}^l) \odot W_2^l X_j^l)]) \tag{16}$$

Eq. (15) maps the edge feature matrix M to the edge vector representation E in the embedding space, where W_0^p is the weight matrix for edge feature on channel p and $W_0^p \in \mathbb{R}^{\frac{d_h}{P} \times 1}$. In Eq. (16), $\|$ is the concatenation operator that combines the node representations' updates on each channel of the edge features. W_1 and W_2 are the linear transformation weight matrices where $W_1 \in \mathbb{R}^{\frac{d_h}{P} \times d_h}$ and $W_2 \in \mathbb{R}^{\frac{d_h}{P} \times d_h}$. σ is a non-linear activation function, e.g. ReLU function, and BN represents the batch normalisation operation. Here, $g(\cdot)$ is an edge



(a)

(b)

Fig. 5. Example of semantic edge features: (a) Lin similarity matrix; (b) NPMI matrix.

gate function with the expression as:

$$g(E^{l}_{ijp}) = \frac{sigmoid(E^{l}_{ijp})}{\sum_{j'\to i} sigmoid(E^{l}_{ij'p}) + \epsilon} \tag{17}$$

$sigmoid$ is the sigmoid activation function and ϵ is a small constant. The vector representation of each edge is updated by combining the information of the edge and its connecting nodes from the previous layer:

$$E^{l+1}_{ijp} = E^{l}_{ijp} + \sigma(BN(W_3 E^{l}_{ijp} + W_4 X^{l}_{i} + W_5 X^{l}_{j})) \tag{18}$$

where $W_3 \in \mathbb{R}^{\frac{d_h}{P}\times\frac{d_h}{P}}$, $W_4 \in \mathbb{R}^{\frac{d_h}{P}\times d_h}$ and $W_5 \in \mathbb{R}^{\frac{d_h}{P}\times d_h}$. As we can observe, in Eqs. (16) and (18), we apply the residual mechanism during the updating process of both node and edge representations to prevent degradation and largely preserve the useful information (He, Zhang, Ren, & Sun, 2016).

After updating through $\hat{L}$-layers EGCN, we can obtain the final representations of nodes $X^{\hat{L}}$. To extract the vector representations of the aspect category $a_m \in A$, we choose an average pool that aggregates the nodes' representations for its corresponding aspect words $\{w^m_1, \dots, w^m_{\hat{m}}\}$. Assuming that the set of node indices of aspect words for a_m is Id_{a_m}, the average pool operation can be expressed as:

$$X^{\hat{L}}_{a_m} = \frac{\sum_{i\in Id_{a_m}} X^{\hat{L}}_{i}}{N_{a_m}} \tag{19}$$

where $X^{\hat{L}}_{a_m} \in \mathbb{R}^{d_h}$ is the vector representation of aspect a_m and N_{am} is the number of unique aspect words for aspect category a_m.

3.3. Aspect–context attention module

In this section, two successive aspect–context attention layers are proposed to extract the interactive and important information from the context with respect to each aspect category (Shen & Huang, 2016). The first attention layer aims to learn the attention weights of contextual representations of words $X^{\hat{L}}$ obtained from EGCN layers and produces the new aspect-specific context representation for the ACD task. The second attention layer further exploits and retrieves useful and relevant information from original embedding hidden states H_L for ACSA task based on the new representations of aspect categories.

Formally, for aspect category a_m, the first attention layer concatenates its representation $X^{\hat{L}}_{a_m}$ with ith word representation from EGCN layers $X^{\hat{L}}_{i}$ and feeds them to a multi-layer perceptron $f(\cdot)$, as shown in Eq. (20). Eq. (21) calculates the normalised weight score of ith word by applying a softmax function. Eq. (22) computes the new context representation about aspect a_m as a weighted sum of word representations from EGCN layers.

$$f(X^{\hat{L}}_{i}, X^{\hat{L}}_{a_m}) = W_7[tanh(W_6[X^{\hat{L}}_{i} \parallel X^{\hat{L}}_{a_m}])] \tag{20}$$

$$\eta^{i}_{a_m} = \frac{exp(f(X^{\hat{L}}_{i}, X^{\hat{L}}_{a_m}))}{\sum_i exp(f(X^{\hat{L}}_{i}, X^{\hat{L}}_{a_m}))} \tag{21}$$

$$S^{0}_{a_m} = \sum_{i}^{\hat{N}} \eta^{i}_{a_m} X^{\hat{L}}_{i} \tag{22}$$

Similarly, the second attention layer follows the same process of computation between the aspect representation $S^{0}_{a_m}$ and the original word representation H_i^L. The attention weights are calculated as below:

$$f(H_i^L, S_{a_m}^0) = W_9[tanh(W_8[H_i^L \parallel S_{a_m}^0])] \tag{23}$$

$$\hat{\eta}_{a_m}^i = \frac{exp(f(H_i^L, S_{a_m}^0))}{\sum_i exp(f(H_i^L, S_{a_m}^0))} \tag{24}$$

$$S_{a_m}^1 = \sum_i^{\hat{N}} \hat{\eta}_{a_m}^i H_i^L \tag{25}$$

In Eqs. (20) to (25), $W_6, W_8 \in \mathbb{R}^{d_h \times 2d_h}$, $W_7, W_9 \in \mathbb{R}^{d_h \times d_h}$ are the learnable weights of linear transformations. $\eta_{a_m}^i$ and $\hat{\eta}_{a_m}^i$ are attention weights obtained from the first and second attention layers, respectively.

3.4. Classifier module

In this section, we design a classifier module consisting of an aspect category detection (ACD) layer that identifies the aspect categories mentioned in the input sentence and an aspect category sentiment classification (ACSC) layer that predicts the sentiment polarity of each aspect category.

The ACD layer performs the multi-label classification where the output classes are not mutually exclusive. For aspect category a_m, the ACD layer takes $S_{a_m}^0$ as input and uses a fully-connected layer to map $S_{a_m}^0$ to a binary output, as shown in Eq. (26).

$$y_m^{ACD} = sigmoid(W_{10} S_{a_m}^0 + b_{ACD}) \tag{26}$$

where $W_{10} \in \mathbb{R}^{d_h \times 1}$ is the weights and $b_{ACD} \in \mathbb{R}$ is the bias. y_m^{ACD} is the probability that aspect a_m is detected. The ACSC layer takes the concatenation of $S_{a_m}^0$ and $S_{a_m}^1$ as the input to predict the sentiment polarity of aspect category a_m, which could be "positive", "negative" and "neutral". Following the idea of the attention mechanism discussed in Section 3.3, the ACSA layer is formulated as:

$$y_m^{ACSA} = softmax(W_{12} ReLU(W_{11}[S_{a_m}^0 \parallel S_{a_m}^1] + b_{ACSA}^0) + b_{ACSA}^1) \tag{27}$$

where $W_{11} \in \mathbb{R}^{2d_h \times d_h}$, $W_{12} \in \mathbb{R}^{d_h \times K}$, $b_{ACSA}^0 \in \mathbb{R}^{d_h}$ and $b_{ACSA}^1 \in \mathbb{R}^K$.

3.5. Model training

The training objective of the proposed model consists of three sub-objectives. Firstly, the ACD task aims to minimise the binary cross entropy loss which is defined by:

$$loss_{ACD} = -\sum_{m=1}^{M} [y_m^{ACD} log y_m^{ACD} + (1 - y_m^{ACD}) log(1 - y_m^{ACD})] \tag{28}$$

Secondly, for ACSC task, the loss function is defined as the cross entropy loss of mentioned aspect categories. For aspects $A^S = \{a_1^S, \dots, a_c^S, \dots, a_C^S\}$ that are detected in the input text S, the ACSC loss function can be expressed as:

$$loss_{ACSA} = -\sum_{c=1}^{C} \sum_{k=1}^{K} \hat{y}_{ck}^{ACSA} log y_{ck}^{ACSA} \tag{29}$$

where $\hat{y}_{ck}^{ACSA}$ is the ground-truth label of sentiment for aspect category a_c^S.

The third objective is to minimise the difference between the semantic relatedness scores and the aspect–context attention weight scores. This objective aims to incorporate the semantic relatedness information, i.e. semantic similarity and NPMI, into the proposed model through EGCN layers. This objective can be achieved by minimising the Jensen–Shannon (JS) Divergence of semantic relatedness and attention weight scores. JS Divergence is a smoothed, symmetric version of KL Divergence and it varies within the range of $[0, 1]$ (Lu, Zhu, Zhang, Wu, & Guo, 2020). The formulation of JS Divergence is presented in Eq. (30) and $d = \frac{1}{2}(a + b)$. As discussed in Section 3.2.1, the semantic similarity and NPMI can be obtained through Eqs. (12) and (13). For aspect category a_m, the third objective function can be formulated as presented in Eq. (31), where w_i^m is the ith aspect word of a_m and w_j is the jth word in the context.

$$D_{JS}(a \parallel b) = \frac{1}{2}[alog\frac{b}{d} + blog\frac{b}{d}] \tag{30}$$

$$\begin{aligned} loss_{SR} = & \sum_{i=1}^{\hat{m}} \sum_{j=1}^{N} D_{JS}(Sim_{lin}(w_i^m, w_j) \parallel S_{a_m}^1) \\ & + \sum_{i=1}^{\hat{m}} \sum_{j=1}^{N} D_{JS}(NPMI(w_i^m, w_j) \parallel S_{a_m}^1) \end{aligned} \tag{31}$$

The final loss function for model training is formulated as the summation of these three objective functions:

$$Loss = \alpha loss_{ACD} + \mu loss_{ACSA} + \rho loss_{SR} + \lambda \|\theta\|_2 \tag{32}$$

where α, μ and ρ are scalar weights of three objectives respectively and λ is the weight of L_2 regularisation term of $\|\theta\|_2$.

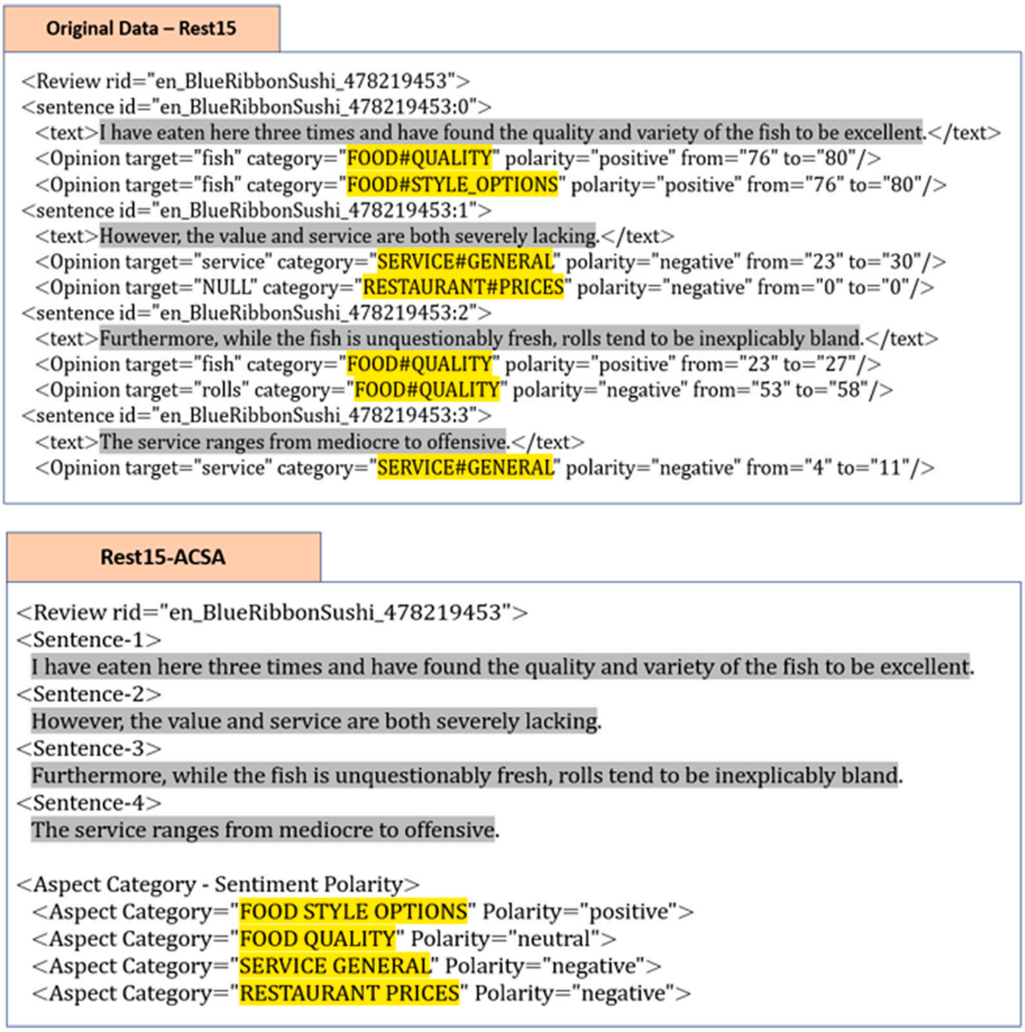
4. Experiments

In this section, the proposed model's effectiveness is evaluated on the benchmark datasets described in Section 4.1. The settings of model implementation are also addressed in Section 4.1. The state-of-the-art baseline models are introduced in Section 4.2 to be compared with our proposed model.

4.1. Datasets and experimental settings

The experiments are conducted on five benchmark datasets constructed based on the datasets from SemEval 2015 (Pontiki, Galanis, Papageorgiou, Manandhar, & Androutsopoulos, 2015), SemEval 2016 (Pontiki et al., 2016) and MAMS-ACSA (Jiang, Chen, Xu, Ao, & Yang, 2019). Since most sentences in SemEval 2015 and SemEval 2016 contain only one aspect category, to evaluate the ability of models detecting multiple sentiment polarities towards multiple aspect categories in a long sentence, we merge the sentences of each review and their corresponding aspect categories as one input sentence. Each aspect category is defined by a specific entity type E and its attribute A in SemEval datasets in the form of $E\#A$. In our experiment, we define E and A as the aspect words for an aspect category. We fix the conflict of sentiment polarity of the same aspect category by assigning it neutral polarity. For laptop datasets specifically, we merge the original 81 aspect categories into 23 most important and frequent ones by eliminating the aspects lower than 20. The newly constructed datasets are named as Rest15-ACSA, Laptop15-ACSA, Rest16-ACSA, Laptop16-ACSA respectively. Fig. 6 presents an example of merging the sentences of a restaurant review from SemEval 2015. For aspect category "Food Quality", two conflicting sentiment polarities are given in the original dataset. Therefore, we assign the polarity of "neutral" to this aspect. In dataset MAMS-ACSA, all sentences contain multiple aspect categories and each aspect category is defined by one aspect word. In our experiment, the classification labels are pre-defined as "none", "positive", "neutral" and "negative", where "none" means that the aspect category is not mentioned. We set aside 20% of each dataset as the test dataset. Table 1 presents the statistics of the datasets and Fig. 7 shows the occurrence of aspect categories for these five datasets.

In our implementation, we use the Glove (Pennington et al., 2014) pre-trained word embedding with a dimension of 300 and the pre-trained BERT word embedding with a dimension of 768 (Devlin et al., 2018). 8 multi-head attention layers and the uncased BERT-base encoder are used to generate contextual word vectors for Glove embedding and BERT embedding respectively. WordNet ontology database (Zhu & Iglesias, 2017) is used to calculate the Lin similarity scores



Original Data – Rest15

```
<Review rid="en_BlueRibbonSushi_478219453">
<sentence id="en_BlueRibbonSushi_478219453:0">
  <text>I have eaten here three times and have found the quality and variety of the fish to be excellent.</text>
  <Opinion target="fish" category="FOOD#QUALITY" polarity="positive" from="76" to="80"/>
  <Opinion target="fish" category="FOOD#STYLE_OPTIONS" polarity="positive" from="76" to="80"/>
<sentence id="en_BlueRibbonSushi_478219453:1">
  <text>However, the value and service are both severely lacking.</text>
  <Opinion target="service" category="SERVICE#GENERAL" polarity="negative" from="23" to="30"/>
  <Opinion target="NULL" category="RESTAURANT#PRICES" polarity="negative" from="0" to="0"/>
<sentence id="en_BlueRibbonSushi_478219453:2">
  <text>Furthermore, while the fish is unquestionably fresh, rolls tend to be inexplicably bland.</text>
  <Opinion target="fish" category="FOOD#QUALITY" polarity="positive" from="23" to="27"/>
  <Opinion target="rolls" category="FOOD#QUALITY" polarity="negative" from="53" to="58"/>
<sentence id="en_BlueRibbonSushi_478219453:3">
  <text>The service ranges from mediocre to offensive.</text>
  <Opinion target="service" category="SERVICE#GENERAL" polarity="negative" from="4" to="11"/>
```

Rest15-ACSA

```
<Review rid="en_BlueRibbonSushi_478219453">
<Sentence-1>
  I have eaten here three times and have found the quality and variety of the fish to be excellent.
<Sentence-2>
  However, the value and service are both severely lacking.
<Sentence-3>
  Furthermore, while the fish is unquestionably fresh, rolls tend to be inexplicably bland.
<Sentence-4>
  The service ranges from mediocre to offensive.

<Aspect Category - Sentiment Polarity>
  <Aspect Category="FOOD STYLE OPTIONS" Polarity="positive">
  <Aspect Category="FOOD QUALITY" Polarity="neutral">
  <Aspect Category="SERVICE GENERAL" Polarity="negative">
  <Aspect Category="RESTAURANT PRICES" Polarity="negative">
```

Fig. 6. Example of constructing Rest15-ACSA.

Table 1
Dataset statistics.

Dataset		Positive	Neutral	Negative	None
Rest15-ACSA	Train	789	73	223	2275
	Test	150	32	97	561
Laptop15-ACSA	Train	909	126	460	6762
	Test	227	37	125	1681
Rest16-ACSA	Train	758	71	196	2671
	Test	190	17	85	664
Laptop16-ACSA	Train	575	92	377	6500
	Test	253	19	101	1536
MAMC-ACSA	Train	1931	3086	2095	18160
	Test	484	772	511	4553

of words, and the SpaCy dependency parser is used to generate the input sentences' dependency tree. The max length of the input sentence is set as 128. The batch size is set as 16 for Glove-based models and 8 for BERT-based models for the training process. Adam optimiser is utilised with a learning rate of $1e{-}3$ for Glove-based models and $5e{-}5$ for BERT-based models (Zhao et al., 2020). We set the dropout rate as 0.3 for Glove embedding and 0.1 for BERT embedding. The default number of layers for EGCN is 2 and the hidden dimensions of all EGCN layers are set as the same. The weights of objectives are set as $\alpha = 1$, $\mu = 1$, $\rho = 1$ and $\lambda = 1e{-}5$. Early stopping is applied during the training process and the number of steps before stopping is set as 10. We run all models for 5 times and record the average results. Accuracy and Macro-Averaged F1 score are used as the metrics for evaluating the performance as Macro-Averaged F1 is more appropriate for unbalanced datasets. All the models are implemented using Pytorch and trained on a single NVIDIA GeForce GTX 1080 GPU device (8 GB RAM).

4.2. Baseline models

To comprehensively evaluate the performance of the proposed SRGN model, a number of state-of-art baseline models are selected for comparison. In the experiment, we aim to demonstrate the effectiveness of the main components of the SRGN model, namely S-EGCN module and aspect–context attention module. In this case, we divide the baseline models into Graph Network-based models, Attention-based models and other Neural Network models. More specifically, for Graph Network-based models, ASGCN (Zhang et al., 2019) is selected as it is one of the earliest GCN-based models for ABSA which exploits syntactical information and word dependencies. SDGCN (Zhao et al., 2020) is an extension of ASGCN which has a similar structure with our proposed SRGN model including the contextual attention mechanism and GCN for multi-aspects dependencies in one sentence. By comparing with ASGCN and SDGCN which merely considers one-dimensional dependency information, we can assess how the S-EGCN

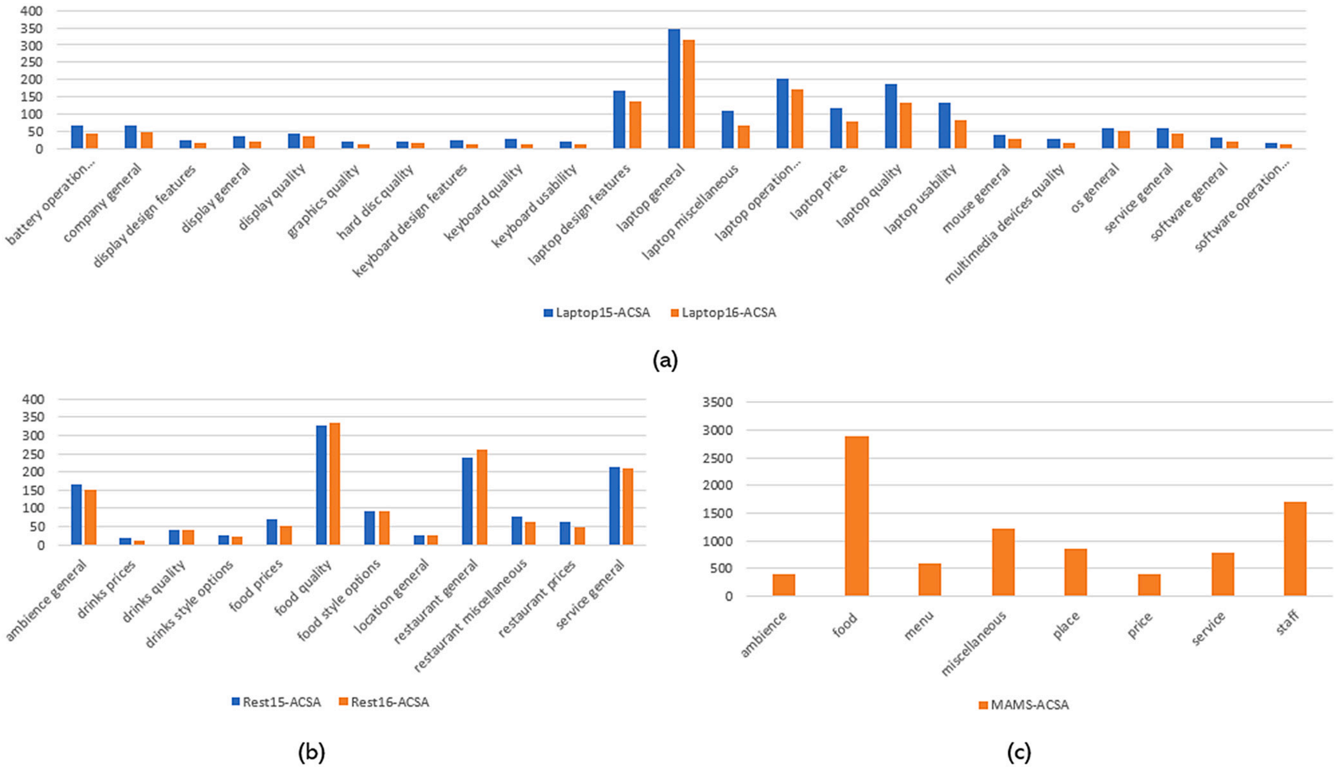


Fig. 7. The distribution of aspect categories: (a) Laptop datasets; (b) Restaurant datasets; (c) MAMS-ACSA.

module with sentiment relatedness information could improve our model's performance on ACSA. Attention-based models have proven to be effective in learning the semantic correlation between aspects and context (Zhao et al., 2020). However, we consider it is not sufficient to exploit the relations of multi-aspects. Hence, we select three attention-based models without considering semantic information as baselines. ATAE-LSTM (Wang et al., 2016) and AEN (Song et al., 2019) are two representative attention-based models, which have been used as baseline models in many studies (Zhao et al., 2020; Zhou et al., 2020). Recently proposed by Xu, Zhu et al. (2020), MAN achieves state-of-art results on ABSA compared to other attention-based models. Besides the Graph Network-based models and Attention-based models, we also compare the performance of our SRGN model with other popular Neural Network models that consist of pre-trained language models, a LSTM-based model and a CNN-based model. Through the comparison, the effectiveness of the S-EGCN module and aspect–context attention module can be evaluated. The detailed introduction of each model is presented as follow.

Graph Network-based Models:

- **ASGCN** (Zhang et al., 2019) is an Aspect-specific Graph Convolutional Network over the input sentences' dependency tree to exploit the syntactic information and dependency relations of words, and uses a bidirectional LSTM to capture contextual word information.
- **SDGCN** (Zhao et al., 2020) adopts bidirectional attention mechanism to generate the aspect-specific representations of aspects and context words and uses a multi-layer GCN to exploit the sentiment dependencies between different aspects.

Attention-based Models:

- **AEN** (Song et al., 2019) is an Attentional Encoder Network that first applies multiple multi-head attention layers on both context and the given aspect to obtain aspect-specific representations.
- **ATAE-LSTM** (Wang et al., 2016) integrates the embedding of the given aspect, the word embeddings and the hidden states of LSTM and uses attention mechanism to generate aspect-based representations.
- **MAN** (Xu, Zhu et al., 2020) is a multi-attention network that applies a global and a local attention layer to extract interactive information of the context and the given aspect.

Other Neural Network Models:

- **BERT-pair** (Sun, Huang et al., 2019) fine-tunes the pre-trained BERT model to train a classifier for each pre-defined aspect category.
- **BERT-pair-NLI-B** (Sun, Huang et al., 2019) converts the ACSA task to a sentence-pair binary classification task by constructing an auxiliary sentence with an aspect-polarity pair and fine-tunes the pre-trained BERT base model.
- **TC-LSTM** (Tang, Qin, Feng, & Liu, 2015) is a variant of TD-LSTM that first integrates the embedding of the given aspect with each context word embedding and utilises two LSTMs to obtain an aspect-dependent representation of context. TD-LSTM is designed for ATSA task while TC-LSTM can be applied for ACSA task.
- **GCAE** (Xue & Li, 2018) utilises two separate convolutional layers on the top of both the context embedding and the aspect embedding and then applies gated Tanh-ReLU units to extract aspect-specific sentiment information.

5. Results discussion and analysis

In this section, we present the experimental results and conduct various analysis to investigate the effectiveness of the proposed SRGN model. The Accuracy and Marco-Averaged F1 score of both baseline models and the proposed model on five datasets are reported in Section 5.1. In Section 5.2, the experimental results of ablated models of SRGN and the influence of the loss function design are reported. To visualise the effectiveness of our proposed model, the heatmaps of attention scores for a sample sentence are presented in Section 5.3.

5.1. Overall results discussion

Table 2 shows the results of average accuracy and macro-averaged F1 for all models on benchmark datasets. To mitigate the impact of the different word embedding methods, BERT-based embedding is applied to all models. As indicated by the experimental results, our proposed model achieves the best performance over the baseline models. Specifically, observations based on the experimental results in Table 2 are discussed as follows.

For the graph network-based models, we can observe that the SDGCN model performs better than ASGCN in accuracy and F1 score. For example, SDGCN achieves 0.8339 of accuracy and 0.3842 of F1 score on Laptop15-ACSA, which are slightly higher than the values of 0.8237 and 0.3679 obtained by ASGCN. The results indicate that SDGCN has a better ability to capture the dependency information than ASGCN. Comparing to ASGCN and SDGCN, the proposed SRGN model has achieved better results on all benchmark datasets. For example, ASGCN and SDGCN obtain the F1 scores of 0.4331 and 0.4736 on Rest15-ACSA, 14.3% and 6.25% lower than that of SRGN-BERT, respectively. Even though SDGCN also adopts the contextual attention mechanism, the results prove that the dependency information is insufficient to capture the relatedness between aspects and the context.

Generally speaking, attention-based models have a better performance than most of other baseline models. For example, even though ATAE-LSTM performs the worst among all attention-based models, it still outperforms the graph network-based baselines and other neural network models, except for BERT-pair-NLI-B, on both Rest15-ACSA and Laptop15-ACSA with higher F1 scores of 0.4874 and 0.3951. It indicates the effectiveness of attention mechanism in extracting more importance information related to aspects in ACSA. MAN has the best performance compared to AEN and ATAE-LSTM, and achieves the highest F1 scores of all baselines on Rest15-ACSA and Laptop15-ACSA with values of 0.4954 and 0.4287. However, the SRGN-BERT outperforms MAN on all datasets, which indicates that integrating the semantic relatedness information through the proposed S-EGCN module could improve the performance of ACSA.

Among all other neural network baseline models, BERT-pair-NLI-B performs the best on most benchmark datasets. For example, BERT-pair-NLI-B achieves the highest accuracy and F1 score on MAMC-ACSA compared to all baselines. The comparison between BERT-pair and BERT-pair-NLI-B shows that the construction of auxiliary sentences and the converting from a multi-class classification into a binary classification could improve the performance on ACSA. However, the training procedure would become more time consuming. Comparatively, the proposed SRGN model deals with ACSA hierarchically and achieves better results. In addition, benefit from BERT-based word embeddings, the SRGN-BERT model has shown a significant improvement compared to SRGN-Glove. SRGN-BERT achieves an increase of 0.0534 and 0.0487 in accuracy and F1 score on Rest15-ACSA and an increase of 0.0325 and 0.0507 in accuracy and F1 score on Laptop15-ACSA. To sum up, through the comparison with baseline models as shown in Table 2, the effectiveness of the SRGN model has been demonstrated.

5.2. Ablation study

To further investigate how different components of SRGN would impact the performance on ACSA, we evaluate the performance of ablated SRGN models. The ablated models exclude the components including dependency relations (SRGN/DT), semantic similarity (SRGN/Sim), pointwise mutual information (SRGN/NPMI), aspect–context attention module (SRGN/Atten) and both semantic similarity and NPMI (SRGN/SR). Additionally, we also assess the influence of different loss functions. SRGN/$loss_{ACD}$ means that ACD loss function is not considered during the training process. In this case, the SRGN model converts the ACSA problem into a multi-class classification instead of a hierarchical problem. SRGN/$loss_{SR}$ means that semantic relatedness (SR) loss function is not considered and SRGN/$loss_{ACD,SR}$ means that only ACSA loss function is considered during the training process. The results of comparison between the full SRGN model with its ablations are indicated in Table 2.

As we can observe, the performance of the SRGN model is impaired after removing the important components. SRGN/SR and SRGN/Atten perform most unsatisfactorily on all datasets. Comparing to the SRGN-BERT model, the F1 scores for these two ablated models drop significantly. For example, the F1 scores of SRGN/SR and SRGN/Atten for Rest15-ACSA reduced by 0.0679 and 0.0496 respectively. It demonstrates the effectiveness of the S-EGCN module and the Aspect–context attention module. To investigate the importance of each semantic relatedness information, we compare the results of SRGN/DT, SRGN/Sim and SRGN/NPMI. SRGN/Sim has the highest F1 scores on all datasets, which indicates that the removal of semantic similarity has the least impact on the performance of the SRGN model. In general, the difference of both accuracies and F1 scores between SRGN/NPMI and SRGN/DT is small. However, SRGN/NPMI obtains smaller F1 scores than SRGN/DT for most of the datasets. For example, the accuracy and F1 score of SRGN/DT are 0.8304 and 0.4311 on Laptop15-ACSA, which are slightly higher than 0.8204 and 0.4114 of SRGN/NPMI. For Rest16-ACSA, the F1 scores of SRGN/NPMI and SRGN/DT are very close. Therefore, the results indicate that NPMI has a more important impact on the performance of the SRGN model. By comparing the full SRGN model and SRGN with ablated loss functions, we can see that the performance of SRGN is significantly impaired if the ACSA problem is not tackled hierarchically, i.e. ACD loss function is not considered. SR loss function also positively impacts on the performance, which helps the SRGN model better integrate the semantic relatedness information.

To further exploit how different components and loss functions could affect the performance of SRGN, we report the accuracy and F1 score of the ablated models on subtask of ACD on Rest15-ACSA and MAMC-ACSA datasets in Table 3. Firstly, we can observe that the aspect–context attention module has an obvious impact on the SRGN model. The F1 scores of SRGN on ACD-Rest15 and ACD-MAMC drop by 0.0164 and 0.054 respectively after excluding the aspect–context module. Secondly, by comparing the results of SRGN/DT, SRGN/Sim and SRGN/NPMI, we can find that SRGN has the lowest accuracy and F1 scores on both datasets. Its accuracy decreases by 0.0231 on ACD-Rest15 compared to the full SRGN model. Thirdly, by removing the ACD loss function, SRGN/$loss_{ACD}$ declines in both aspect detection and sentiment classification. Similarly, the inclusion of the SR loss function could slightly improve the performance of our model on ACD task.

To summarise our discussion of the experimental results in Sections 5.1 and 5.2, we present several conclusions as follows:

- The integration of semantic relatedness information, which includes the semantic similarity and NPMI discussed in this paper, could complement the insufficiency of dependency information in exploiting the relations between aspects and the context.
- Even though dependency information, semantic similarity and NPMI have positive impact on improving the performance of SRGN model, NPMI is more influential on both ACD and SC.
- The design and implementation of aspect–context attention mechanism could improve the overall performance on ACSA.
- The hierarchical design of SRGN and the proposed loss function could further improve the overall performance on ACSA.

Table 2
Experimental results on comparison of SRGN with baseline models for ACSA.

	Model	Rest15-ACSA		Laptop15-ACSA		Rest16-ACSA		Laptop16-ACSA		MAMC-ACSA	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Baselines	ASGCN	0.7439	0.4331	0.8237	0.3679	0.7080	0.3126	0.8243	0.2945	0.7146	0.5004
	SDGCN	0.7663	0.4736	0.8339	0.3842	0.7323	0.3379	0.8406	0.3034	0.7437	0.5179
	AEN	0.7809	0.4837	0.8319	0.4237	0.7548	0.3469	0.8325	0.3061	0.7776	0.5423
	ATAE-LSTM	0.7657	0.4874	0.8361	0.3951	0.7129	0.3375	0.8267	0.3108	0.7032	0.5116
	MAN	0.7852	**0.4954**	**0.8433**	**0.4287**	0.7569	0.3545	0.8521	0.3192	0.7652	0.5470
	BERT-pair	0.7823	0.4775	0.8331	0.3926	0.7585	0.3539	0.8324	0.3045	0.7729	0.5490
	BERT-pair-NLI-B	**0.7881**	0.4941	0.8365	0.4108	**0.7626**	**0.3648**	**0.8499**	**0.3217**	**0.7831**	**0.5521**
	TC-LSTM	0.7714	0.4741	0.8348	0.3541	0.6951	0.3342	0.8194	0.3019	0.7159	0.5021
	GCAE	0.7692	0.4685	0.8323	0.3824	0.7529	0.3498	0.8439	0.3137	0.7209	0.5175
Ablated models	SRGN/SR	0.7457	0.4373	0.8057	0.3885	0.7345	0.3437	0.8350	0.3139	0.7405	0.5363
	SRGN/DT	0.7545	0.4647	0.8304	0.4311	0.7585	0.3472	0.8438	0.3224	0.7746	0.5459
	SRGN/Sim	0.7695	0.4808	0.8336	0.4258	0.7532	0.3537	0.8345	0.3262	0.7633	0.5575
	SRGN/NPMI	0.7572	0.4576	0.8204	0.4114	0.7469	0.3478	0.8248	0.3155	0.7494	0.5427
	SRGN/Atten	0.7623	0.4556	0.8113	0.3995	0.7406	0.3356	0.8319	0.3007	0.7534	0.5342
	SRGN/$loss_{ACD}$	0.7633	0.4724	0.8304	0.4212	0.7577	0.3594	0.8275	0.3228	0.7663	0.5384
	SRGN/$loss_{SR}$	0.7809	0.4795	0.8357	0.4385	0.7479	0.3614	0.8291	0.3331	0.7905	0.5562
	SRGN/$loss_{ACD,SR}$	0.7463	0.4559	0.8124	0.4073	0.7501	0.3537	0.8390	0.3217	0.7539	0.5336
Full models	SRGN-Glove	0.7465	0.4565	0.8212	0.3931	0.7551	0.3557	0.8222	0.3164	0.7403	0.5114
	SRGN-BERT	**0.7959**	**0.5052**	**0.8537**	**0.4437**	**0.7759**	**0.3798**	**0.8565**	**0.3407**	**0.7987**	**0.5689**

Table 3
Experimental results on comparison of ablated SRGN models for ACD.

	Model	ACD-Rest15		ACD-MAMC	
		Acc	F1	Acc	F1
Ablated models	SRGN/SR	0.7781	0.7594	0.7996	0.7732
	SRGN/DT	0.7914	0.7775	0.8355	0.7967
	SRGN/Sim	0.7883	0.7705	0.8223	0.7917
	SRGN/NPMI	0.7825	0.7615	0.8167	0.7873
	SRGN/Atten	0.7839	0.7673	0.7765	0.7505
	SRGN/$loss_{ACD}$	0.7540	0.7369	0.8228	0.7853
	SRGN/$loss_{SR}$	0.7945	0.7780	0.8357	0.7990
	SRGN/$loss_{ACD,SR}$	0.7503	0.7302	0.8307	0.7887
Full model	SRGN	0.8056	0.7837	0.8533	0.8045

5.3. Attention visualisation

To have an intuitive understanding and to illustrate the effectiveness of our proposed model, we visualise the attention weights obtained by the proposed SRGN model and AEN-BERT model, as shown in Fig. 8. In this example, four aspects words, "*price*", "*design*", "*features*" and "*quality*", are pre-defined. The deeper the colour, the higher attention weights of the word.

It can be observed that the proposed SRGN model could capture the most relevant words related to the targeted aspect word. For example, the SRGN model tend to pay more attention to the word "*use*" for aspect words "*design*" and "*features*". In contrast, AEN-BERT model only captures opinion words such as "*good*" and "*easy*". In this example, AEN-BERT model assigns a high attention weight to "*good*" for aspect word "*price*". However, the aspect "*price*" is not mentioned in the example sentence. This example demonstrates the effectiveness of SRGN in exploiting the relations between contextual and aspect words. Furthermore, as we can observe from Fig. 5, the Lin Similarity and NPMI for the word pair ("*design*", "*sized*") are 0 and 0.26 respectively, which are lower than other contextual words. However, besides "*use*" and "*good*", the SRGN model pays more attention to "*sized*" for aspect word "*design*" . This phenomenon indicates that the proposed SRGN model could not only learn from the SR graph but also adjust its attention to more relevant information during training process.

6. Conclusions and future works

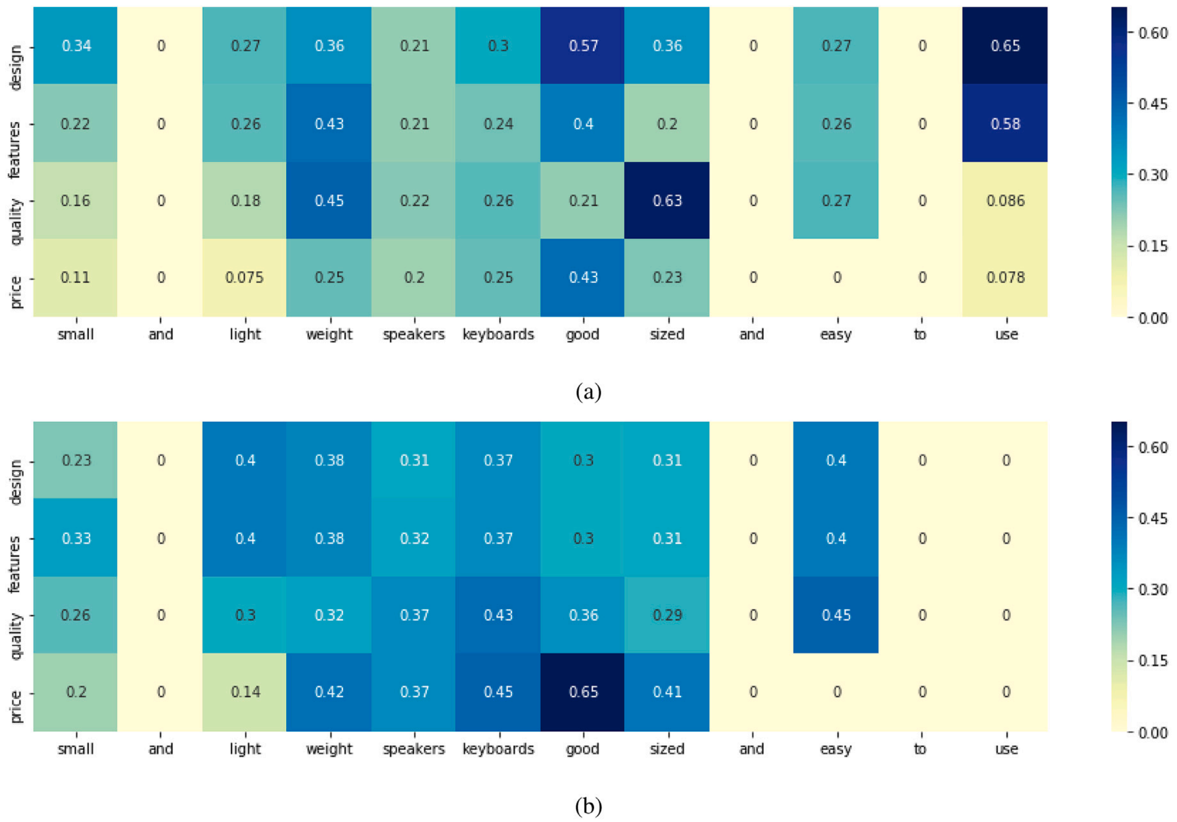
In this paper, a Semantic Relatedness-enhanced Graph Network (SRGN) model is proposed and presented for aspect category sentiment analysis (ACSA). Multi-channel edge information comprising of the ontology-based similarity (i.e. Lin Similarity) and the distribution-based relatedness (i.e. NPMI) has been considered to build a semantic relatedness (SR) graph. A S-EGCN module, which mainly consists of edge-gated GCN (EGCN) layers, is proposed to explore further contextual semantic information between aspects and the context. Additionally, an aspect–context attention module, which comprises two successive attention layers, is applied to impel the model to pay more attention to the features related to each specific aspect. Finally, a classifier consisting of an aspect category detection (ACD) layer and an aspect category sentiment classification layer is designed to perform the ACSA hierarchically.

To better incorporate and regulate the semantic relatedness (SR) information, we propose a JS divergence-based SR loss function to minimise the difference between attention scores and SR information. Experimental results on five benchmark datasets demonstrate the effectiveness of the proposed SRGN model. The ablation study implies that the semantic information helps identify the features related to multiple aspects in one sentence. Compared to word dependency and semantic similarity, the pointwise mutual information has a greater impact on the model performance.

The future work of this study can be continued in several directions. Firstly, the proposed model and the baselines discussed in this paper are supervised learning models. Considering data labelling is quite expensive in real life, a semi-supervised version of the proposed SRGN model would be more practical. Next, it is challenging to reserve as much information as possible when dealing with long textual documents. Hence, we would focus on designing the architecture and the training procedure of a large-scale graph neural network-based model considering the complexity and memory requirements. Furthermore, the experimental results of the proposed SRGN model imply the positive effects of injecting external semantic information through graph neural networks in understanding and identifying aspect-specific information. Hence, we would also explore the possibility of extending the proposed model on complex tasks, such as question answering and recommendation systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a)

(b)

Fig. 8. Visualisation of attention scores: (a) SRGN (b) AEN-BERT.

References

Araque, O., Zhu, G., & Iglesias, C. A. (2019). A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, *165*, 346–359.

Bai, X., Liu, P., & Zhang, Y. (2020). Exploiting typed syntactic dependencies for targeted sentiment classification using graph attention neural network. arXiv preprint arXiv:2002.09685.

Batet, M., & Sánchez, D. (2016). Improving semantic relatedness assessments: Ontologies meet textual corpora. In *KES* (pp. 365–374).

Bresson, X., & Laurent, T. (2017). Residual gated graph convnets. arXiv preprint arXiv:1711.07553.

Chauhan, G. S., Meena, Y. K., Gopalani, D., & Nahta, R. (2020). A two-step hybrid unsupervised model with attention mechanism for aspect extraction. *Expert Systems with Applications*, *161*, Article 113673.

Chen, G., Tian, Y., & Song, Y. (2020). Joint aspect extraction and sentiment analysis with directional graph convolutional networks. In *Proceedings of the 28th international conference on computational linguistics* (pp. 272–279).

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, *72*, 221–230.

Cilibrasi, R. L., & Vitanyi, P. M. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, *19*(3), 370–383.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Do, H. H., Prasad, P., Maag, A., & Alsadoon, A. (2019). Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Systems with Applications*, *118*, 272–299.

Gan, C., Wang, L., Zhang, Z., & Wang, Z. (2020). Sparse attention based separable dilated convolutional neural network for targeted sentiment analysis. *Knowledge-Based Systems*, *188*, Article 104827.

Gong, L., & Cheng, Q. (2019). Exploiting edge features for graph neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9211–9219).

Gu, X., Gu, Y., & Wu, H. (2017). Cascaded convolutional neural networks for aspect-based opinion summary. *Neural Processing Letters*, *46*(2), 581–594.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Jeong, B., Yoon, J., & Lee, J.-M. (2019). Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis. *International Journal Of Information Management*, *48*, 280–290.

Jiang, Q., Chen, L., Xu, R., Ao, X., & Yang, M. (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)* (pp. 6281–6286).

Kang, Y., & Zhou, L. (2017). RubE: RUle-based methods for extracting product features from online consumer reviews. *Information & Management*, *54*(2), 166–176.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Li, X., Fu, X., Xu, G., Yang, Y., Wang, J., Jin, L., et al. (2020). Enhancing BERT representation with context-aware embedding for aspect-based sentiment analysis. *IEEE Access*, *8*, 46868–46876.

Li, W., Qi, F., Tang, M., & Yu, Z. (2020). Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing*, *387*, 63–77.

Li, S., Zhou, L., & Li, Y. (2015). Improving aspect extraction by augmenting a frequency-based method with web-based similarity measures. *Information Processing & Management*, *51*(1), 58–67.

Liu, Q., Liu, B., Zhang, Y., Kim, D. S., & Gao, Z. (2016). Improving opinion aspect extraction using semantic similarity and aspect associations. In *Proceedings of the AAAI conference on artificial intelligence, vol. 30*.

Lofi, C. (2015). Measuring semantic similarity and relatedness with distributional and knowledge-based approaches. *Information and Media Technologies*, *10*(3), 493–501.

Lu, Q., Zhu, Z., Zhang, D., Wu, W., & Guo, Q. (2020). Interactive rule attention network for aspect-level sentiment analysis. *IEEE Access*, *8*, 52505–52516.

Majumder, G., Pakray, P., Gelbukh, A., & Pinto, D. (2016). Semantic textual similarity methods, tools, and applications: A survey. *Computación Y Sistemas*, *20*(4), 647–665.

Meškelė, D., & Frasincar, F. (2020). ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. *Information Processing & Management*, *57*(3), Article 102211.

Miller, T., Dligach, D., & Savova, G. (2016). Unsupervised document classification with informed topic models. In *Proceedings of the 15th workshop on biomedical natural language processing* (pp. 83–91).

Moghaddam, S. (2015). Beyond sentiment analysis: mining defects and improvements from customer feedback. In *European conference on information retrieval* (pp. 400–410). Springer.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., et al. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In *10th international workshop on semantic evaluation (SemEval 2016)*.

Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., & Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 486–495).

Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, *108*, 42–49.

Priyantina, R., & Sarno, R. (2019). Sentiment analysis of hotel reviews using latent Dirichlet allocation, semantic similarity and LSTM. *International Journal Of Intelligent Engineering and Systems*, *12*(4), 142–155.

Rana, T. A., & Cheah, Y.-N. (2016). Aspect extraction in sentiment analysis: comparative analysis and survey. *Artificial Intelligence Review*, *46*(4), 459–483.

Rana, T. A., & Cheah, Y.-N. (2017). A two-fold rule-based model for aspect extraction. *Expert Systems with Applications*, *89*, 273–285.

Rana, T. A., Cheah, Y.-N., & Rana, T. (2020). Multi-level knowledge-based approach for implicit aspect identification. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *50*(12), 4616–4630.

Salahli, M. A. (2009). An approach for measuring semantic relatedness between words via related terms. *Mathematical and Computational Applications*, *14*(1), 55–63.

Shams, M., & Baraani-Dastjerdi, A. (2017). Enriched LDA (ELDA): Combination of latent Dirichlet allocation with word co-occurrence analysis for aspect extraction. *Expert Systems with Applications*, *80*, 136–146.

Shen, Y., & Huang, X.-J. (2016). Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers* (pp. 2526–2536).

Song, Y., Wang, J., Jiang, T., Liu, Z., & Rao, Y. (2019). Attentional encoder network for targeted sentiment classification. arXiv preprint arXiv:1902.09314.

Sun, C., Huang, L., & Qiu, X. (2019). Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. arXiv preprint arXiv:1903.09588.

Sun, K., Zhang, R., Mensah, S., Mao, Y., & Liu, X. (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)* (pp. 5683–5692).

Tang, D., Qin, B., Feng, X., & Liu, T. (2015). Effective LSTMs for target-dependent sentiment classification. arXiv preprint arXiv:1512.01100.

Tubishat, M., Idris, N., & Abushariah, M. (2020). Explicit aspects extraction in sentiment analysis using optimal rules combination. *Future Generation Computer Systems*, *114*, 448–480.

Veyseh, A. P. B., Nour, N., Dernoncourt, F., Tran, Q. H., Dou, D., & Nguyen, T. H. (2020). Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. arXiv preprint arXiv:2010.13389.

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615).

Wang, C., Song, Y., Li, H., Zhang, M., & Han, J. (2015). Knowsim: A document similarity measure on structured heterogeneous information networks. In *2015 IEEE international conference on data mining* (pp. 1015–1020). IEEE.

Xu, K., Zhao, H., & Liu, T. (2020). Aspect-specific heterogeneous graph convolutional network for aspect-based sentiment classification. *IEEE Access*, *8*, 139346–139355.

Xu, Q., Zhu, L., Dai, T., & Yan, C. (2020). Aspect-based sentiment classification with multi-attention network. *Neurocomputing*, *388*, 135–143.

Xue, W., & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. arXiv preprint arXiv:1805.07043.

Zhang, C., Li, Q., & Song, D. (2019). Aspect-based sentiment classification with aspect-specific graph convolutional networks. arXiv preprint arXiv:1909.03477.

Zhang, Y., Liu, Q., & Song, L. (2018). Sentence-state lstm for text representation. arXiv preprint arXiv:1805.02474.

Zhao, P., Hou, L., & Wu, O. (2020). Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowledge-Based Systems*, *193*, Article 105443.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., et al. (2018). Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434.

Zhou, J., Huang, J. X., Hu, Q. V., & He, L. (2020). SK-GCN: Modeling Syntax and Knowledge via Graph Convolutional Network for aspect-level sentiment classification. *Knowledge-Based Systems*, *205*, Article 106292.

Zhu, G., & Iglesias, C. A. (2017). Sematch: Semantic similarity framework for knowledge graphs. *Knowledge-Based Systems*, *130*, 30–32.

Zhu, X., Yang, X., Huang, Y., Guo, Q., & Zhang, B. (2019). Measuring similarity and relatedness using multiple semantic relations in WordNet. *Knowledge and Information Systems*, 1–31.