Knowledge-Based Systems 260 (2023) 110150

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A disentangled linguistic graph model for explainable aspect-based sentiment analysis

Xiaoyong Mei [a], Yougen Zhou [a], Chenjing Zhu [a], Mengting Wu [a], Ming Li [a,*], Shirui Pan [b]

[a] Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua 321004, China
[b] School of Information and Communication Technology, Griffith University, QLD 4222, Australia

ARTICLE INFO

Article history:
Received 16 June 2022
Received in revised form 2 November 2022
Accepted 22 November 2022
Available online 25 November 2022

Keywords:
Aspect-based sentiment analysis (ABSA)
Explainability
Disentangled graph representation learning

ABSTRACT

Aspect-based sentiment analysis (ABSA) aims to use interactions between aspect terms and their contexts to predict sentiment polarity for given aspects in sentences. Current mainstream approaches use deep neural networks (DNNs) combined with additional linguistic information to improve performance. DNN-based methods, however, lack explanation and transparency to support predictions, and no existing model completely solves the trade-off between explainability and performance. In contrast, most previous studies explain the relationship between input and output by attribution; however, this approach is insufficient to mine hidden semantics from abstract features. To overcome the aforementioned limitations, we propose a disentangled linguistic graph model (DLGM) to enhance transparency and performance by guiding the signal flow. First, we propose a disentangled linguistic representation learning module that extracts a specific linguistic property via neurons to help capture finer feature representations. To further boost explainability, we propose a supervised disentangling module, in which labeled linguistic data help reduce information redundancy. Finally, a cross-linguistic routing mechanism is introduced into the signal propagation of linguistic chunks to overcome the defect of distilling information in an intralinguistic property. Quantitative and qualitative experiments verify the effectiveness and superiority of the proposed DLGM in sentiment polarity classification and explainability.

1. Introduction

In recent years, aspect-based sentiment analysis (ABSA) [1,2] has emerged as a significant focus in affective computing and sentiment analysis [3] regarding the ability of ABSA to predict the sentiment polarity of specific aspects in sentences. For example, a review about restaurants contains the statement *"I like the food here, but the service is dreadful"*.. The aspect term *"food"* has a positive sentiment polarity according to *"like"*; in contrast, the sentiment polarity of *"service"* is negative, as indicated by *"dreadful"*. The key idea behind ABSA is to model the information about interactions between aspect terms and their corresponding contexts. Along this line, a variety of deep neural network (DNN)-based methods have been proposed to learn a good representation with additional linguistic information.

DNNs, especially graph neural network (GNN) approaches [4–8], have performed very well in ABSA; however, they lack explanation and transparency to support predictions. Intuitively, detailing a reason for a prediction helps gain trust from users. To develop a human-interpretable approach to ABSA, Yadav et al. [9] proposed an explainable method based on a Tsetlin Machine to illustrate what drove the model to learn the corresponding context information for aspects. However, the model's performance still fell short of the performance of existing DNN-based ABSA models. Certain classic models, such as decision trees, are particularly easy to understand. However, these models do not offer optimal performance in terms of sentiment polarity prediction. Therefore, no existing model truly excels at both performance and explainability.

In addition, classification performance metrics, such as classification accuracy, are not sufficient to precisely reflect the capabilities of the model. For example, when different aspects express the same sentiment polarity, classification may be affected by other aspects or the entire sentence. As shown in Fig. 1, the model based on different regions can give the same prediction. Intuitively, the fusion of the multiple linguistic features can gradually mark the most helpful aspect terms and opinion words for sentiment polarity classification. We observe that the aspect-opinion pair tends to consist of a noun and an adjective, and the dependency types can help match the aspect-opinion pair. The distances can also boost the relevance of two words. To this

[*] Corresponding author.
E-mail addresses: cdmxy@zjnu.edu.cn (X. Mei), zyg@zjnu.edu.cn (Y. Zhou), zhuchenjing@zjnu.edu.cn (C. Zhu), wumengting@zjnu.edu.cn (M. Wu), mingli@zjnu.edu.cn (M. Li), s.pan@griffith.edu.au (S. Pan).

https://doi.org/10.1016/j.knosys.2022.110150
0950-7051/© 2022 Elsevier B.V. All rights reserved.

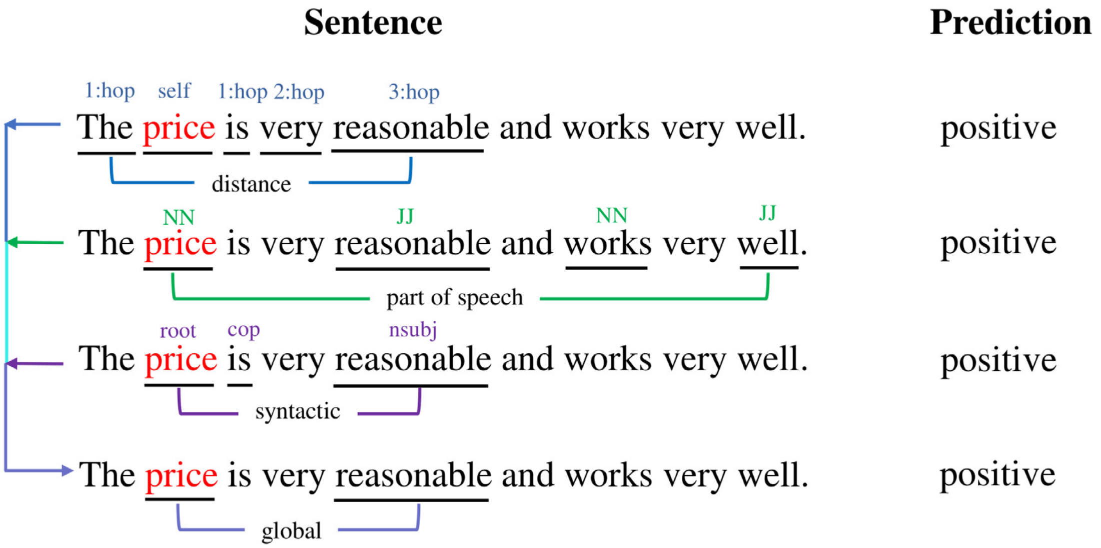

Sentence
Prediction
1:hop self 1:hop 2:hop 3:hop
The price is very reasonable and works very well.
distance
positive
NN JJ NN JJ
The price is very reasonable and works very well.
part of speech
positive
root cop nsubj
The price is very reasonable and works very well.
syntactic
positive
The price is very reasonable and works very well.
global
positive

Fig. 1. An example to illustrate that the activation regions between distance, parts of speech, and syntax in ABSA make the same sentiment polarity predictions, and all agree with sentiment labels. The aspect term is highlighted in red, and the words with high weights are underlined.

end, we propose an explainable ABSA task that can be easily understood and can explain the linguistic features the model has learned from the sentence to infer the sentiment result of the target aspect; achieving such a task requires a certain transparency of the model.

More importantly, we argue that existing explainable models for ABSA are inadequate for associating abstract semantic features with linguistic properties. Additionally, as shown in Fig. 1, the available explainable methods can infer a direct connection between the underlined words and sentiment polarity by attribution [10]; however, the hidden relation is not sufficient to mine out which linguistic feature is the most important. The key reason for this is that attribution-based models calculate the contribution of each word to the prediction according to the original word embedding vector, which ignores the fundamental fact that neural vector representations often contain much linguistic information. Disregarding this fact, the explanation is coarse and fails to reflect the impact of linguistic features on predictions.

To address the aforementioned issues, we propose a unified model to classify sentiment polarities and provide better explainability. We first formulate the explainable ABSA as an index generation task, which could enable sentiment polarity prediction by class index generation and generating textual or visual explanations by using sequence index generation. Specifically, we transform the raw text into a graph and introduce an ego node representing the aspect term, for which the proposed model can perform the index generation and leverage the powerful modeling capabilities of the GNN to achieve high performance. Simultaneously, we introduce a linguistic disentangling module in the GNN to provide explanations for the model by investigating the individual dimensions in the original word embedding vector as input signals to linguistic property neurons. Our goal is to encourage neurons to learn different linguistic information through the loss function. After the linguistic signals are extracted, a routing mechanism is employed to direct the information flow between the collections of chunks containing specific linguistic information for prediction.

In summary, our core research contributions are as follows:

1. A novel model (termed DLGM), which facilitates the multilinguistic properties in word embeddings for enhancing finer feature representation and provides graph-based model transparency, is proposed for explainable aspect-based sentiment analysis (ABSA).
2. An improved disentangling module under a supervised learning task is developed; labeled linguistic data help reduce information redundancy over various linguistic properties through independence regulation.
3. We develop a cross-linguistic routing mechanism in GNN, to overcome the exchange barriers among different linguistic information flows and to some extent offer explainable ABAS results from the perspective of linguistic properties.

The remainder of this article is organized as follows. Section 2 revisits prior works on ABSA and the background of explainability. Section 3 describes the details of our methods. Section 4 discusses the experimental results and analyzes the ablation studies. Finally, Section 5 presents the conclusion.

2. Related works

In this section, we primarily revisit prior works on ABSA; these works can be generally classified into two types: ABSA without linguistic information and ABSA with linguistic information. Additionally, we briefly review the existing works on explainability.

2.1. ABSA without linguistic information

Most works along this line aim to extract features from input sentences, often by splitting input sentences into aspect terms and their contexts. For example, Vo and Zhang [11] proposed extracting features from each section by using word2vec embeddings and combining the features for prediction by using pooling mechanisms. Pham and Le [12] initialized multiple convolutional neural networks (CNNs) to obtain diverse vector representations and concatenate all outputs to generate a comprehensive representation for classification. Tang et al. [13] proposed exploiting the interaction between aspects and context words to integrate features by using a long short-term memory (LSTM) network. To automatically mine relations between aspect terms and their contexts, Ma et al. [14] introduced attention mechanisms to model the relations between aspect terms and their context. Tay et al. [15] used a deep memory network for feature encoding and attention mechanisms to capture the importance of each context word for classification. Song et al. [16] used BERT as a word embedding module with attention mechanisms to extract the aspect-context interaction. Various neural network methods that do not rely on linguistic information have realized automatic feature encoding and achieved decent classification performance.

2.2. ABSA with linguistic information

To further improve the performance, researchers have attempted to introduce additional linguistic information into neural network models in ABSA. Li et al. [17] considered position information and proposed a hierarchical network based on position-aware attention to learn aspect-specific oriented representations. Phan and Ogunbona [18] proposed a part-of-speech (POS)-aware self-attention mechanism to model POS embeddings for subsequent prediction. Furthermore, recent research has shown that syntactic information is useful for prediction in ABSA. Dong et al. [19] achieved information propagation from text words to aspect terms by transforming the primitive dependency tree into a binary tree and applying an adaptive recurrent neural network (AdaRNN). Tian et al. [20] used key–value memory networks to encode the dependency label on arcs in the dependency tree to improve prediction accuracy. Recently, using graph neural networks (GNNs) to encode syntactic structures has achieved significant performance. Wang et al. [4] and Zheng et al. [21] attempted to reconstruct the original dependency tree into an aspect-oriented dependency tree. He et al. [22] and Zhang et al. [23] integrated the tree-based distance between words into an attention mechanism. Bai et al. [7] extended RGAT to separately encode dependency labels and aspect contextual information and then fused them for classification. In this work, our method continues to adopt a GNN as the feature encoder similar to the above models; however, a cross-linguistic routing mechanism is introduced to propagate distinct linguistic information on the graph and generate the explanation of ABSA with the help of the graph structure.

2.3. Explainable NLP

Driven by the need for transparency, deep learning explainability has been an important direction in the NLP community [24,25]. Current explainability methods are categorized into post hoc explainability and intrinsic explainability [26], which differ according to whether the generated explanations require post-processing after the model makes a prediction, or whether they arise as a part of the model. Post hoc explainability aims to provide explanations after the predictions are made for an existing model. A well-known example is LIME [25], which approximates model decisions by using a surrogate model applied following the predictors operation to produce explanations. A recent development in this line is GEF [27], which proposes a unified framework to explain a generic encoder–predictor architecture. Another group of post hoc methods are gradient-related [28–31]; these methods calculate the gradient of the output by using an input feature to identify the important features.

In contrast, intrinsic explainability approaches require the construction of self-explanatory models by using information emitted by the model to generate explanations along with the predictions. Decision trees [32] and rule-based models [33,34] are representative examples of intrinsic explainability models, while feature saliency approaches (such as attention mechanisms) outperform these classic models [35]. However, the explanations provided by attention weights are not always reliable [36–40], and the meaning of the weight distribution cannot be further inferred. A reliable explanation can be achieved by adding explainability constraints in model learning or by using an ensemble of neuron symbolic AI tools [41] to accurately represent the true reasoning behind the model prediction. Our work falls into this category by inducing labeled linguistic information in model learning to disentangle representations. Unlike previous works, we attempt to enhance model explainability by exploiting the given linguistic information in original word embeddings.

3. Methodology

For our proposed method, we first formalize the explainable ABSA under a unified framework to generate explanations and classify sentiment polarities classification. Then, we detail our proposed disentangled linguistic graph model (DLGM), which consists of three components to distangle linguistic graphs. The overall framework of the proposed DLGM is illustrated in Fig. 2.

3.1. Task formulation

For the explainable ABSA, there are two types of subtasks, namely, explanation generation and sentiment polarity classification, whose targets can be represented as sequence indices and class indices, respectively. Therefore, we can formulate these two types of subtasks in a unified framework. Each instance in the conventional ABSA consists of two components: an aspect and the corresponding sentence. Formally, we denote them as $T = \{w_i, w_{i+1}, \ldots, w_{i+m-1}\}$ and $S = \{w_1, w_2, \ldots, w_i, \ldots, w_m, \ldots, w_n\}$, where m and n are the lengths of T and S, respectively. For a given instance, we transform the raw text into a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes all words presented in the sentence and $\mathcal{E}$ contains all edges between nodes. In general, a graph includes an adjacency matrix $A \in \mathbb{R}^{n\times n}$ and a node feature matrix $X \in \mathbb{R}^{n\times c}$, where c denotes the dimension of the feature vector. The adjacency matrix satisfies $A_{ij} = 1$ if there is an edge between a pair of nodes, and the node feature matrix is obtained by a pre-trained language model, where each row contains much linguistic information. The purpose of explainable ABSA is to disentangle the input features X into chunks; map each chunk with the specific linguistic property (such as POS); and then introduce a linguistic information routing into a GNN to refine the linguistic representations, forming the final representation sequence H for classification. Simultaneously, a weighted adjacency matrix is obtained to generate explanations about prediction.

3.2. Overview of DLGM

The architecture of DLGM is depicted in Fig. 2. The model, consists of three components: aspect-oriented graph construction (AOGC), linguistic graph disentangling (LGD), and sentiment polarity prediction (SPP). The AOGC module is designed to preprocess the input raw sentence into the graph and simultaneously obtain the initial node feature matrix. The LGD module includes two parts: a mechanism for linguistic property neuron extraction (LPNE) and a mechanism for linguistic masking representation learning (LMRL). The LPNE mechanism is a linguistic-based encoder to disentangle the node features into multiple linguistic properties to learn finer linguistic-aware embeddings in the graph, and the LMRL mechanism is a graph convolutional layer with linguistic routing that is designed to encode the interactions between aspects and contexts to learn the fused disentangled linguistic embeddings for the aspect term. The SPP predicts the sentiment polarity according to the learned aspect embeddings.

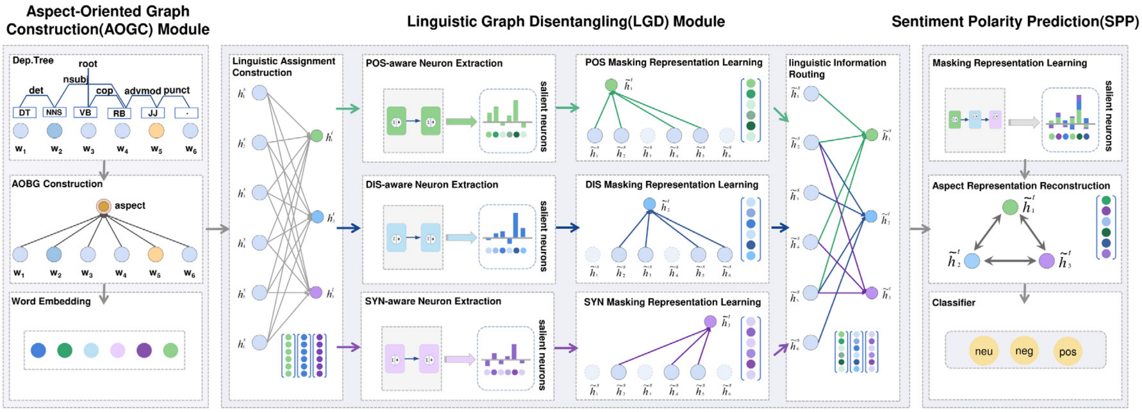

Aspect-Oriented Graph Construction(AOGC) Module
Linguistic Graph Disentangling(LGD) Module
Sentiment Polarity Prediction(SPP)
Dep.Tree
root
nsubj
det
cop
advmod
punct
DT
NNS
VB
RB
JJ
.
w1
w2
w3
w4
w5
w6
AOBG Construction
aspect
Word Embedding
Linguistic Assignment Construction
POS-aware Neuron Extraction
DIS-aware Neuron Extraction
SYN-aware Neuron Extraction
salient neurons
POS Masking Representation Learning
DIS Masking Representation Learning
SYN Masking Representation Learning
linguistic Information Routing
Masking Representation Learning
Aspect Representation Reconstruction
Classifier
neu
neg
pos

Fig. 2. Overall framework of DLGM. In the AOGC module, the given sentence is processed into an aspect-oriented bipartite graph with linguistic tags and an initial node feature matrix. In the LGD module, including linguistic property neuron extraction and linguistic masked representation learning, the node feature matrix is disentangled into specific linguistic chunks with the help of linguistic tags, and new representation learning is performed under the guidance of a linguistic routing mechanism. The final representation of the aspect is fused from linguistic features for prediction, and the information flow based on a graph is presented for explanation.

3.3. Aspect-oriented graph construction

Approaches applying GNNs to ABSA have shown that an aspect-oriented graph is more conducive to information transfer, allowing the aspect to update its representation by extracting useful information from the neighbors. More specifically, the connections between aspect and context nodes provide an available channel for guiding the propagation of information flow. Nevertheless, the focus of a dependency tree containing rich linguistic information is usually not an aspect term, as shown in Fig. 3; the dependency tree can also be regarded as a graph to help better encode the input sentence. For our explainable ABSA, a unified graph structure is required to integrate linguistic information from dependency trees and guide information flow based on the original aspect-oriented graph structure. Hence, we propose an aspect-oriented bipartite graph, termed AOBG, consisting of a new node representing the aspect term and the node in the raw sentence.

For each sample, we first introduce an ego node and link each node in the sentence to the ego node, for which there are edges between any nodes in S and the aspect node. Then, we employ the syntactic parsing tool to reveal its syntax rules and POS tags, where r_{ij} is the syntax relation from node i to j and p_i represents the POS of node i. To obtain linguistic labels, we traverse the generated dependency tree to record the shortest path from each node to the aspect term and put the nodes and edges in the shortest path into *path*, for which the number of passing nodes represents distance features, the typed syntactic dependencies corresponding to the edges contained in the *path* are denoted as syntactic features, and the original POS tags are retained as POS features. Moreover, we set the tree-based distance as distance tags, where the tree-based distance represents the length of the shortest path between aspect-context pairs. To avoid information confusion between different aspect terms, we construct a unique graph for each aspect contained in the sentence. Fig. 4 shows an example of an AOBG.[1] In contrast to existing approaches that provide aspect-oriented structures [4,22,42,43], our approach not only provides a unified bipartite graph structure to aggregate useful information from sentences through the directed connections with the ego nodes but also merges more linguistic information to facilitate disentangling the representations of each node in the sentence.

To obtain the node feature matrix with rich linguistic features, we use the pretrained language model based on Transformer [44] to generate word embedding representations as initial node features, including BERT and RoBERTa [45]. The input of the language model is "[CLS] + *sentence* + [SEP]", where "[CLS]" and "[SEP]" are two predefined special tokens, "[CLS]" is used to mark the beginning of the first sentence, while "[SEP]" is the separator appended to the end of the input sentence. Then, the same-length embedding representations are generated as the input sequence of the model. To obtain the same representations as [7], we adjust the input sequence as "[CLS] + *sentence* + [SEP] + *aspect* + [SEP]" to separate the aspect nodes from the sentence, while "[CLS] + *sentence* + [SEP]" represents only the sentence. After processing, the original word embedding x_i that fuses more linguistic information is obtained; this result has been verified in previous work [46,47].

3.4. Linguistic graph disentangling

Applying GNNs in ABSA has shown that the aspect node updates its representation by uniformly extracting salient information from the neighbors. Intuitively, the aspect node exchanges information between aspect terms and their corresponding opinion words according to linguistic rules that can filter out irrelevant information to significantly improve classification performance. Hence, we develop a disentangled linguistic graph network to leverage the hidden linguistic relations among words, which are already implicit in word embedding representations. We encode each dimension of the initialized embedding as input signals of individual neurons to extract specific linguistic property representation for the node, considering that different neurons are sensitive to different linguistic properties. Then, a linguistic routing mechanism is introduced into embedding propagation to guide the signal flow of the linguistic chunks.

3.4.1. Extracting linguistic property neurons

The LPNE encoder differs from the methods that operate directly on the original word embedding to calculate the contribution of each word toward the prediction. Inspired by the emotional recurrent unit [48], we encode the single dimension of embeddings into neurons and assign the neurons into chunks according to specific linguistic properties. The word embedding of each node can be set as:

$$x_i = \{x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{ic}\} \tag{1}$$

[1] If node i is the aspect itself, we set the syntactic label as self, and the distance is 0; if the distance is longer than 4, the syntactic label is set to outer, and the distance is −1.

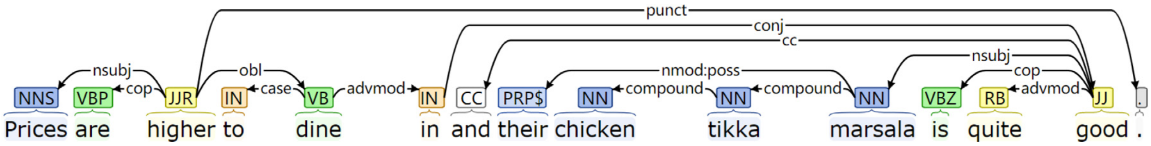


Fig. 3. An ordinary dependency tree obtained by the dependency parsing tool.

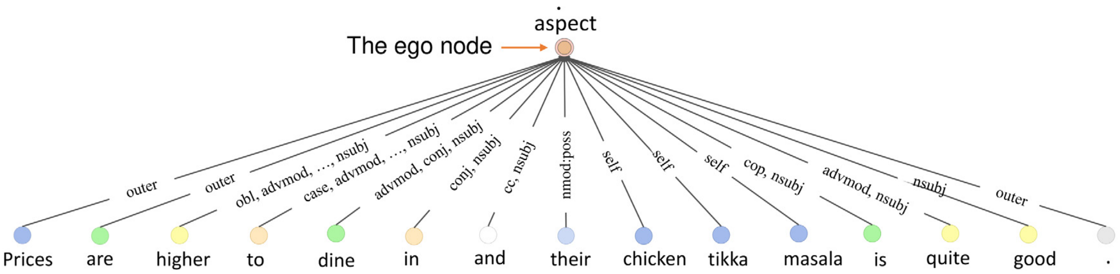


Fig. 4. An aspect-oriented bipartite graph constructed from an ordinary dependency tree. The ego node is the center of the graph and represents the aspect term labeled in the sentence. All edges are directed from context nodes to the ego node. The tags in the edges denote the syntactic relations in the shortest path between the aspect-context pairs in the dependency tree. The different colors on the nodes represent distinct POS tags.

where $x_i \in \mathbb{R}^c$ represents the intrinsic embedding feature of node i, and x_{ij} is the jth signal input to the neuron.

We use a linear model to extract the signals in x_i due to its explainability; the model can directly query the learned weights to measure the importance of each input signal. More formally, such extraction is represented as:

$$h_{i,k} = \sigma(W_k x_i + b_{i,k}) \tag{2}$$

where $k \in \{POS, DIS, SYN\}$ represent the linguistic properties of POS, distance, and syntactic dependency, respectively. $h_{i,k}$ represents the output of neurons associated with linguistic property k. To ensure the individual linguistic property neurons are compatible, we adjust the output of neurons to the same dimensional size, $h_{i,k} \in \mathbb{R}^{\frac{c}{3}}$. W_k and $b_{i,k}$ denote the parameter of weights and bias, respectively. σ is a nonlinear activation function to filter the signal of neurons.

We need to minimize the differences between each neuron and encourage the signal to represent the corresponding linguistic property to reduce information redundancy. Consequently, we use a supervised task with labeled data to constrain the extracted signals correlated with the labeled linguistic property. The sigmoid cross-entropy loss is embraced in the learning process of linguistic property neurons:

$$\mathcal{L}_E = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}(P_{i,k}\log\hat{P}_{i,k} + (1-P_{i,k})\log(1-\hat{P}_{i,k})) \tag{3}$$

$$\hat{P}_{i,k} = \frac{1}{1+\exp(-h_{i,k})} \tag{4}$$

where $\mathcal{L}_E$ is the loss of extracting the linguistic features from the original embedding, K is the number of specific linguistic properties we predefined, $P_{i,k}$ defines the ground-truth tag. Label $P_{i,k} = 1$ if the linguistic property k appears in the ith node; otherwise label $P_{i,k} = 0$. $\hat{P}_{i,k}$ is the predicted probability for linguistic property k in the ith node.

After the extraction, we need to make the learned linguistic signals independent by minimizing their similarity. Although the loss functions $\mathcal{L}_E$ ensure that the neurons select input signals that are sensitive to specific linguistic properties, it is still important to maximize the variability constraint between the output representations of linguistic property neurons, since research on explainability shows that the similarity between different features is too high to facilitate explainable analysis by using information flow. Hence, we utilize a simple and efficient loss function, such as the conicity similarity calculation method [49,50], as a regularization term. More formally, the compatibility regularization is calculated as follows:

$$\mathcal{L}_I = \frac{1}{K}\sum_{k=1}^{K} cosine(h_{i,k}, \frac{1}{K}\sum_{k=1}^{K} h_{i,k}) \tag{5}$$

where $\mathcal{L}_I$ is the loss of the difference of linguistic features. A low value of conicity implies that there is little alignment between the learned linguistic embedding to the mean of all linguistic vectors. In this way, the $\mathcal{L}_I$ could lead to a larger difference in the extracted signals of neurons, thereby enhancing the linguistic property of the disentangled representation.

3.4.2. Linguistic masking representation learning

The representation of each node is chunked into specific linguistic property representations. We aim to provide individual channels to guide the signal flow and distill valuable information from the nodes of a sentence to the aspect. To this end, rather than just using the original interaction, we introduce a linguistic routing mechanism into embedding propagation by masking irrelevant neuron signals to capture the linguistic property-aware relation, as shown in Fig. 5.

Specifically, we first estimate the difference score of each interaction (i, t) between nodes within a single linguistic property channel as the mask matrix for updating node representations. The value of t is fixed since the aspect node is unique in each AOBG. Hence, for a given linguistic property k, the relevant score of each node is given as follows:

$$q_{i,k} = \frac{\exp((h_{i,k})^\top \cdot h_{t,k})}{\sum_{i=1}^{n}\exp((h_{i,k})^\top \cdot h_{t,k})} \quad \forall k \in \{1,\dots,K\} \tag{6}$$

where $h_{t,k}$ is the representation of the aspect nodes kth linguistic property. It is assumed that each linguistic property has an equal probability among nodes when the initial model is learning. Thus, such a score matrix $Q \in \mathbb{R}^{n\times K}$ can be regarded as the adjacency matrix of individual nodes in intralinguistic property channels.

Moreover, the influence of linguistic properties on sentiment polarity is not always identical. For example, in the dependency tree, adjectives that are closer to aspect terms are more likely to be opinion words for sentiment polarity classification than for just considering POSs; consequently, all linguistic signals captured by the neurons must be considered. Hence, cross-linguistic

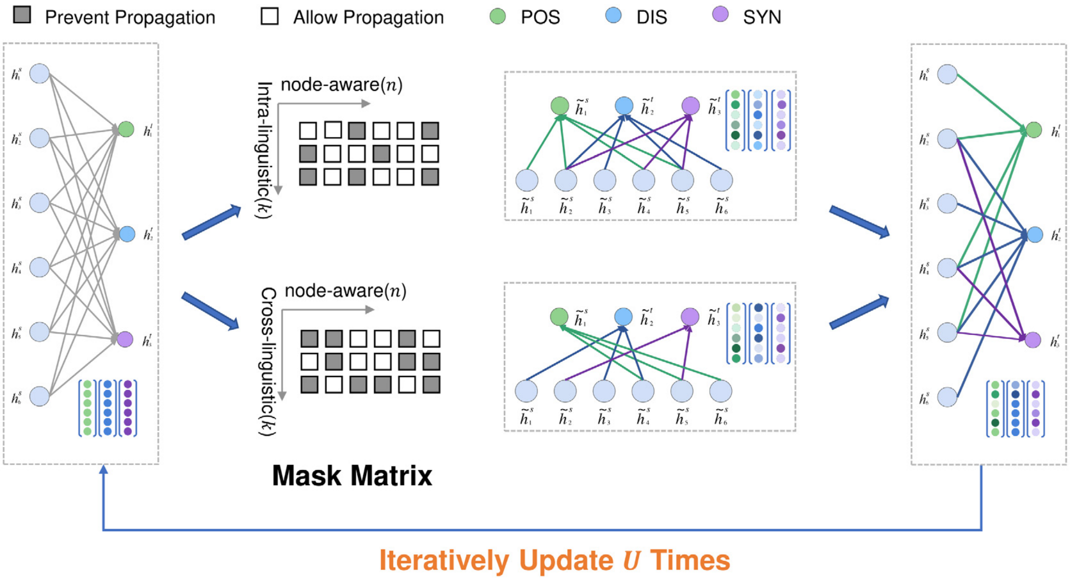


Fig. 5. Illustration of the linguistic routing mechanism.

property information routing is proposed to identify the most significant linguistic property. The training and analysis of the cross-linguistic routing module reveals the commonality and specificity of linguistic representations. Essentially, the module calculates the cross-linguistic importance score of each node as follows:

$$c_{i,k} = \frac{\exp((h_{i,k})^{\top} \cdot h_{t,k})}{\sum_{k'=1}^{K} \exp((h_{i,k})^{\top} \cdot h_{t,k'})} \quad \forall k \in \{1, \ldots, K\} \tag{7}$$

It can thus be determined which linguistic property is more important for sentiment analysis. Similarly, the obtained normalized score matrix $C \in \mathbb{R}^{n\times K}$ can be regarded as a cross-linguistic graph.

After the intralinguistic property routing and cross-linguistic property routing mechanism, the linguistic information learned from each node is aggregated to determine the linguistic representation for each aspect. The operation is as follows:

$$h_{t,k}^{intra} = \sum_{i \in \mathcal{N}_t} q_{i,k} h_{i,k} \tag{8}$$

$$h_{t,k}^{cross} = \sum_{i \in \mathcal{N}_t} c_{i,k} h_{i,k} \tag{9}$$

$$\widetilde{h}_{t,k} = h_{t,k}^{intra} + h_{t,k}^{cross} + \mu \tag{10}$$

where $h_{t,k}^{intra}$ and $h_{t,k}^{cross}$ represent the intralinguistic representation and the cross-linguistic representation, respectively; μ is the learnable bias vector; $\widetilde{h}_{t,k}$ is a temporary embedding of the aspect; and $\mathcal{N}_t$ is the set of context nodes pointing to aspect node t.

Nodes affected by the same linguistic properties tend to have similar signals. To further enhance the nodes' relationship among the aspect and its opinion node, we fed $\widetilde{h}_{t,k}$ back into Eqs. (6) and (7) to update $\widetilde{h}_{t,k}$. We iterate U times to adjust the representation of the aspect more precisely; the aspect's representation can be formulated as follow:

$$h_{t,k}^{(l)} = h_{t,k}^{(l-1)} + \widetilde{h}_{t,k}^{U} \tag{11}$$

where $h_{t,k}^{(l)}$ is the hidden state of the lth embedding propagation layer. We do not need to further stack aggregation layers for refining the aspect embedding, as it is directly connected with other nodes in the AOBG. $h_{t,k}^{(l)}$ can be seen as the final representation of the linguistic property. The representation of aspect h_t is obtained to perform on all linguistic properties as follows:

$$h_t = (h_{t,1}, h_{t,2}, \ldots, h_{t,K}) \tag{12}$$

Consequently, the fine-grained aspect representations consider linguistic properties, such as POS, distance, and syntactic dependency features.

Explainability: When propagating such information, the model comprehensively aggregates linguistic features from each node to obtain a new aspect representation. The intralinguistic score matrix and the cross-linguistic score matrix reflect the contribution of various linguistic properties to sentiment polarity prediction, and can generate textual or visual explanations. More importantly, the explanation for the model prediction can be given more intuitively after the visualization of the graph structure, which will be presented in the next section.

3.5. Sentiment polarity prediction

After these operations, we generate the final aggregated feature representation of the aspect node h_t, which takes residual connection with the original embedding representation x. The probability distribution of sentiment polarity $P(t)$ is calculated by using a softmax normalization with the output of a fully connected layer fed as input:

$$P(t) = softmax(W \cdot h_t + b) \tag{13}$$

where W and b are the learnable weights and bias parameter, respectively.

The cross-entropy loss is employed as the objective function for sentiment polarity classification:

$$\mathcal{L}_C = -\sum_{(S,T)\in D} \sum_{t \in T} \log P(t) \tag{14}$$

where $\mathcal{L}_C$ is the loss of sentiment polarity prediction, D represents all training samples consisting of sentence and aspect word pairs, and T denotes all aspects presented in sentence S. In the training phase, we need to optimize our model by combining explainability loss and classification loss. Hence, the overall objective is to minimize the integrated loss:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_C + \beta \cdot (\mathcal{L}_E + \mathcal{L}_I) \tag{15}$$

Table 1
Details of the four datasets.

Dataset	Positive			Neutral			Negative		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
Laptop	976	337	337	455	167	167	851	128	128
Restaurant	2165	727	727	637	196	196	807	196	196
Twitter	1507	172	172	3016	336	336	1528	169	169
MAMS	3380	400	400	5042	607	607	2764	329	329

where α is the weight used to control the classification loss and β indicates the weight of explainability regularization, which consists of $\mathcal{L}_E$ and $\mathcal{L}_I$. We discuss the effect of the loss function in our experiments; see Section 4.5.2.

4. Experiments

We provide details of the comprehensive experiments evaluating the classification performance and explainability of DLGM. For the classification performance, we compare our model to the baselines on four widely used and publicly available datasets. For explainability, we provide the quantitative and qualitative results on two benchmarks. In addition, we discuss our analysis to verify the effectiveness of DLGM.

4.1. Experimental setup

4.1.1. Dataset

We apply our proposed DLGM to four benchmark datasets, namely, Laptop, Restaurant, Twitter and MAMS, for sentiment polarity prediction. To ensure data consistency, we perform the same data preprocessing and delete the samples labeled *"conflict"* in the Laptop and Restaurant datasets; therefore, we finally consider three sentiment polarities contained in each dataset: $\{positive, neutral, negative\}$. The number of samples in each category is shown in Table 1.

4.1.2. Evaluation metrics

We adopt the general evaluation metrics: accuracy and the macroaveraged F1, which have been widely used to compare ABSA classification performance. To evaluate ABSA explanations, although there are no factual annotations to assess the generated explanations, a feasible scheme we suggest is to calculate the accuracy of the formed explanations by using the aspect term and its opinion words in sentences as the ground truth explanation label, which is annotated in other ABSA subtasks.

4.1.3. Implementation and training parameters

Our DLGM consists of a neuron extraction module and a linguistic routing mechanism. For the neuron extraction module, a deep biaffine parser [51] is used to generate POS tags and dependency trees. We use the pretrained RoBERTa to generate original word embedding representations of words, whose dimension is 768. The number of neuron categories K (representing POS, distance and syntax) is set to 3. We set the signal output dimension of each neuron to 256. For the linguistic routing mechanism, the iteration times are set as $U = 3$. We implement our model by using PyTorch and the Adam [52] optimizer with an initial learning rate of 10^{-5}, and the batch size is 64. The classification loss coefficients (α) and explainability regularization coefficients (β) are searched in {0, 0.1, 0.5, 1, 1.5, 2}. Before starting each epoch, we randomly shuffle the training samples. The early stopping strategy is also utilized in the training time. Experiments are performed on an NVIDIA GeForce RTX 2080Ti GPU.

4.2. Baselines

We compare the classification performance of our proposed DLGM with that of the following SOTA baselines.

- **TD-LSTM** [13] develops two target-aware LSTM networks to encode target information for feature extraction.
- **IAN** [14] integrates an LSTM network and an attention mechanism to enable the target and context to interactively influence the generation of respective representations.
- **TNet** [53] transforms the word representations into target-specific embeddings and extracts salient features by using a CNN.
- **MGAN** [54] improves the attention mechanism at the word level to capture fine-grained features between aspects and contexts by using a coarse-grained attention mechanism.
- **AOA** [55] jointly models the interactions between aspects and contexts by exploiting an attentionoverattention neural network.
- **AEN/AEN-BERT** [16] employs an attention-based encoder to capture the interaction between the context and target and applies a fine-tuned BERT pretrained model.
- **CapsNet/CapsNet-BERT** [56] constructs a capsule network [57] to capture the interactions between contexts and aspects; CapsNet-BERT uses BERT to generate the vector representations of sentences and aspects before being fed into the capsule network layer for classification.
- **BERT-PT** [58] explores a posttraining strategy on pretrained BERT model for aspect-based sentiment classification in the form of reading comprehension tasks.
- **BERT-SPC** [16] extends the input sequence of BERT and uses pooled embedding for classification.
- **AdaRNN** [19] learns word-to-target sentiment by using multiple RNN synthesis functions over dependency trees.
- **PhraseRNN** [59] extends AdaRNN by considering both the sentence dependency tree and constituent tree and adding a phrase module.
- **SynAttn** [22] proposes an object representation to capture the semantics of opinion objects and incorporate syntactic information into the attention mechanism to improve performance.
- **CDT and ASGCN**, CDT [60] combines a dependency tree and a GCN to enhance the sentence feature embedding output by a Bi-LSTM and achieve aspect-level sentiment classification, while ASGCN [43] adds an attention mechanism to better strengthen the final representation.
- **PWCN** [23] integrates the tree-based distance between words into an attention mechanism for prediction.
- **HAPN** [17] uses a hierarchical attention network based on position information to learn aspect-specific oriented representations.
- **TD-GAT/TD-GAT-BERT** [61] captures syntactic structure information by using a GAT and uses a multilayer attention network to obtain information. In addition, the LSTM unit is added to explicitly capture the cross-layer aspect information.

Table 2
Performance comparison results of various models on four benchmark datasets. Our DLGM achieves competitive classification performance against different baselines.

Category	Model	Restaurant		Laptop		Twitter		MAMS	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
W/o a linguistic property	TD-LSTM	75.63	*	68.13	*	*	*	*	*
	IAN	78.60	*	72.10	*	*	*	76.60	*
	TNet	80.69	71.27	76.54	71.75	74.97	73.60	*	*
	MGAN	81.25	71.94	75.39	72.47	72.54	70.81	77.26	*
	AOA	81.20	*	74.50	*	*	*	66.72	*
	AEN	80.98	72.14	73.51	69.04	72.83	69.81	79.78	*
	CapsNet	80.79	*	*	*	*	*	*	*
	BERT-PT	84.95	76.96	78.07	75.08	*	*	*	*
	BERT-SPC	84.46	76.98	78.99	75.03	73.55	72.14	82.82	81.90
	AEN-BERT	83.12	73.76	79.93	76.31	74.71	73.13	*	*
	CapsNet-Bert	85.93	*	*	*	*	*	83.39	*
w a linguistic property	AdaRNN	*	*	*	*	66.3	65.9	*	*
	PhraseRNN	66.20	59.32	*	*	*	*	*	*
	SynAttn	80.45	71.26	72.57	69.13	*	*	*	*
	ASGCN	80.77	72.02	76.12	72.12	*	*	*	*
	PWCN	80.96	72.21	75.55	71.05	72.15	70.40	*	*
	HAPN	82.23	*	77.27	*	*	*	*	*
	CDT	82.30	74.02	77.19	72.99	74.66	73.66	80.70	79.79
	TD-GAT	81.20	*	74.00	*	*	*	*	*
	TD-GAT-BERT	83.00	*	80.10	*	*	*	*	*
	R-GAT	83.30	76.08	77.42	73.76	75.57	73.82	*	*
	RGAT	83.55	75.99	78.02	74.00	75.36	74.15	81.57	80.87
	BERT-ASC	84.46	76.98	76.25	72.68	*	*	*	*
	R-GAT-BERT	86.60	81.83	78.21	74.07	76.15	74.88	*	*
	RGAT-BERT	86.68	80.92	82.34	78.20	76.28	75.25	84.52	83.74
	BERT-LARGE+A-KVMN	86.88	80.92	80.41	77.38	76.59	74.91	*	*
Ours	BERT+MLP	85.35	78.38	78.36	74.16	75.92	74.41	82.22	80.29
	BERT+DLGM	87.35	81.88	82.61	79.24	74.96	73.37	84.59	83.26
	RoBERTa+MLP	87.37	80.96	83.78	80.73	**77.17**	**76.20**	84.51	82.44
	RoBERTa+DLGM	**88.61**	**83.58**	**84.38**	**81.96**	75.52	74.58	**84.77**	**84.25**

The best performances are bold-typed. * indicates that the results are not provided in the original paper. The underlined results indicate that the proposed model outperforms the baselines.

- **R-GAT/R-GAT-BERT** [4] attempts to reconstruct an aspect-oriented dependency tree and proposes a relational attention mechanism to achieve sentiment classification.
- **BERT-ASC** [18] proposes a self-attention mechanism based on POSs to process POS embeddings for subsequent prediction.
- **BERT-LARGE-AKVMN** [20] uses key–value memory networks (KVMN) to encode the dependency label on arcs in the dependency tree to improve the prediction accuracy.
- **RGAT/RGAT-BERT** [7] proposes a relational graph attention network that integrates dependency type features into the attention mechanism, enriching the final representations.

All comparison baselines either follow the original papers or are optimized by the same source datasets for a fair comparison.

4.3. Performance comparison

We first compare the classification performance of DLGM with that of other methods and then investigate how to exploit multiple linguistic properties for further performance enhancement.

4.3.1. Overall performance

The overall performance comparison results of our proposed DLGM and baseline models on four datasets are presented in Table 2.

First, of all the comparisons, the DNN methods with pretrained language models generally outperform those without pretrained language models, thereby verifying the advantages of fusing pretrained language models for better analysis. In particular, RoBERTa with an MLP layer outperforms BERT-MLP; therefore, the embeddings generated by RoBERTa are more friendly to the ABSA task, and RoBERTa learns better linguistic information than BERT [62]. Moreover, systems that introduce a linguistic property on BERT outperform BERT-MLP, as that linguistic property can provide fine-grained information for the ABSA.

Second, the GNN-based methods outperform all the deep learning-based baselines, thereby showing the effectiveness of the GNN in better feature extraction for textual data. Furthermore, the recent SOTA method RGAT integrating typed syntactic dependencies outperforms all other baselines, implying the advantage of extracting linguistic features by using the GNN. However, RGAT-BERT presents a large gap with BERT-DLGM on three datasets, thereby illustrating that encoding linguistic interaction uniformly is not enough to reveal the fine-grained relation between opinion words and their corresponding aspect, as the linguistic influence under different embeddings would be deeply entangled in the information propagation.

Overall, with the help of RoBERTa, our proposed DLGM achieves SOTA or near SOTA performance in all the comparisons on Laptop, Restaurant, Twitter, and MAMS with accuracies of 84.38%, 88.61%, 75.52%, and 84.77%, respectively. Similarly, DLGM achieves the best macro-F1 (81.96%, 83.58%, and 84.25%) on Laptop, Restaurant, and MAMS, respectively. Two performance advantages of DLGM in ABSA are summarized as follows: (1) DLGM explicitly disentangles the original word embeddings to capture multiple linguistic information for enriching aspect embedding learning. (2) DLGM recognizes the salient linguistic information applicable to aspect sentiment analysis to better infer the aspect with its corresponding opinion words. However, our model fails to achieve excellent performance on the Twitter dataset. We speculate there may be a discrepancy between the samples in the Twitter dataset and those in other datasets because the samples from Twitter are more colloquial.

4.3.2. Performance comparison with different linguistic properties

We investigate how to further improve performance by disentangling multiple linguistic properties. Therefore, we implement

Table 3
Comparison results of DLGM according to different linguistic properties. "Decrease" indicates the relative performance gap between the corresponding variants and the proposed DLGM. DLGM-*Embedding* indicates that the linguistic features are obtained directly from the parser tool. The Attn-Based Model represents the dominant attention-based method in attribution-based explainability methods.

Model	Classification				Explainability			
	Laptop		Restaurant		Laptop		Restaurant	
	Accuracy	Decrease	Accuracy	Decrease	Macro-F1	Decrease	Macro-F1	Decrease
DLGM	84.38	–	88.61	–	81.96	–	88.19	–
DLGM-*Embedding*	83.81	−0.57	87.53	−1.08	79.72	−2.24	85.79	−2.40
DLGM-*no linguistic*	83.28	−1.10	86.58	−2.03	75.52	−6.44	84.27	−3.92
DLGM-*POS only*	82.29	−2.09	86.95	−1.66	75.55	−6.41	85.10	−3.09
DLGM-*DIS only*	81.07	−3.31	87.38	−1.23	75.82	−6.14	85.08	−3.11
DLGM-*SYN only*	83.33	−1.05	87.28	−1.33	76.69	−5.27	85.85	−2.34
DLGM-*POS&DIS*	82.65	−1.73	87.28	−1.33	79.91	−2.05	86.21	−1.98
DLGM-*POS&SYN*	83.89	−0.49	87.39	−1.22	79.89	−2.07	86.61	−1.58
DLGM-*DIS&SYN*	83.48	−0.90	87.89	−0.72	80.23	−1.73	86.94	−1.25
Attn-Based Model	–	–	–	–	75.43	−6.49	84.44	−3.75

several models with different linguistic properties to compare the performance of different linguistic components. The comparison result is presented in Table 3, which comprises the experimental results for Laptop and Restaurant.

We observe that for all the models, incorporating three linguistic properties leads to better performance on the four datasets. The model based on three linguistic features outperforms DLGM without a linguistic property by approximately 1.6% on average in accuracy. This finding verifies the effectiveness and advantage of multiple linguistic information in this competitive comparison. The performance comparison between the different methods to obtain linguistic features indicates that the neural network extraction method is more effective than directly embedding linguistic properties.

In Table 3, models using a single linguistic property show some performance differences on two datasets. The results obtained on Laptop show that models with additional separate linguistic information, except syntactic dependency, cannot improve or even degrade performance; however, disentangling single linguistic information enhances the performance of DLGM-no-linguistic on Restaurant. The classification performance of our model further improves when more linguistic features are introduced. This result differs from the observation in Bai et al. [7] possibly because our model not only distills the information of nodes but also calculates the signal flow between different linguistic properties, thereby providing more explicit linguistic information to establish a more stable connection between aspect terms and opinion words.

4.4. Explainability of DLGM

4.4.1. Quantitative evaluation

In the explainable ABSA task, we use opinion words as the rationale for aspect sentiment classification. Based on the previous TOWE task, the target word is the same as the aspect term, and the opinion words are given as the basis for classification. For example, in the text *"Even though it has good seafood, the prices are too high"*., when the given target is *"seafood"*, TOWE needs to output *"good"* as the opinion word and *"high"* as the opinion word for *"prices"*. An example of restaurant is shown in Fig. 6.

We use the opinion and target words as the ground truth labels for explanations in Laptop and Restaurant and then use these words to calculate the explanation accuracy of our explainable method. The explainability of DLGM is not evaluated on Twitter and MAMS, as they do not provide any description of the opinion words. Specifically, we formalize the explanation problem as a binary classification task, treat the edges in the ground truth as explanation labels, and regard the importance weights given by explainable methods as prediction scores. Better explainable methods can assign higher scores to edges, resulting in higher explanation accuracy. Hence, we report the quantitative metric that captures desirable aspects of explanations: the F1-score. Table 3 reports the results based on different linguistic property information.

First, we compare the DLGM-*no linguistic* model (which uses word embeddings directly without linguistic features) and the Attn-based model (which considers word embeddings without disentangled graph neural networks, and which is the dominant method used in attribution-based explainability methods). The word embeddings are obtained by the pretrained language model for sentences, which are common frameworks used in the previous explainable ABSA models. The explainability scores of the two models are very close.

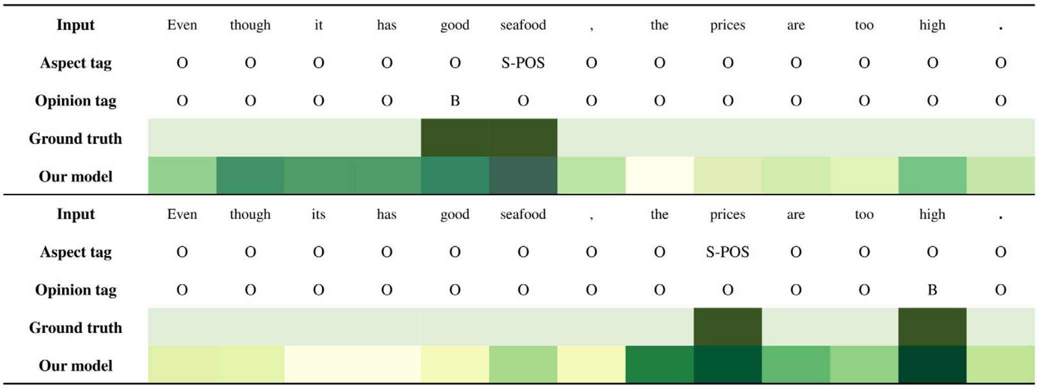
Second, we compare the DLGM-*POSonly*, DLGM-*DISonly*, and DLGM-*SYNonly* models, which consider a single linguistic property (POS, distance, or syntax, respectively) between contextual and aspect words. The feature representations of the linguistic property are extracted from the word embeddings of sentences by a neural network with a supervised task. Except for those of the DLGM-*SYNonly*, the explainability metrics of the three models are not significantly improved compared to the explainability metrics of the attribution method. From the fidelity perspective, syntactic information is the salient feature identified through explanations; this finding is consistent with our intuition presented in the introduction.

Finally, we combine more types of linguistic properties to generate explanations. In general, three types of models based on the dual linguistic feature can be composed, with each model yielding certain improvements over single piece of information. Furthermore, by combining the three pieces of linguistic information, the explainability score of our DLGM is further improved compared to that of the attribution-based method. The improved metric demonstrates the benefit of using more linguistic features to generate explanations at the level of linguistic properties and our method has more explainability than other methods used in ABSA.

Compared to ABSA classification, explainability performance results in a more significant gap between various methods. The evaluation metric of TOWE is designed to evaluate whether the model captures the opinion expression for aspect sentiment classification, while ABSA accuracy is designed to evaluate whether the model can classify correctly. We recommend using both TOWE and ABSA accuracy metrics rather than a single metric to evaluate explainable ABSA more precisely.

4.4.2. Qualitative evaluation

The qualitative evaluation of the explainability of DLGM is also presented, for which we visualize the learned linguistic property-aware graphs for each sentence in conjunction with the linguistic



Input	Even	though	it	has	good	seafood	,	the	prices	are	too	high	.
Aspect tag	O	O	O	O	O	S-POS	O	O	O	O	O	O	O
Opinion tag	O	O	O	O	B	O	O	O	O	O	O	O	O
Ground truth													
Our model													
Input	Even	though	its	has	good	seafood	,	the	prices	are	too	high	.
Aspect tag	O	O	O	O	O	O	O	O	S-POS	O	O	O	O
Opinion tag	O	O	O	O	O	O	O	O	O	O	O	B	O
Ground truth													
Our model													

Fig. 6. Tagging schema for the ground-truth explanation label, where B is for begin, I is for inside, E is for end, S is for single and O is for outside. In this example, there are only {B, O, S} since the aspect word is a single word.

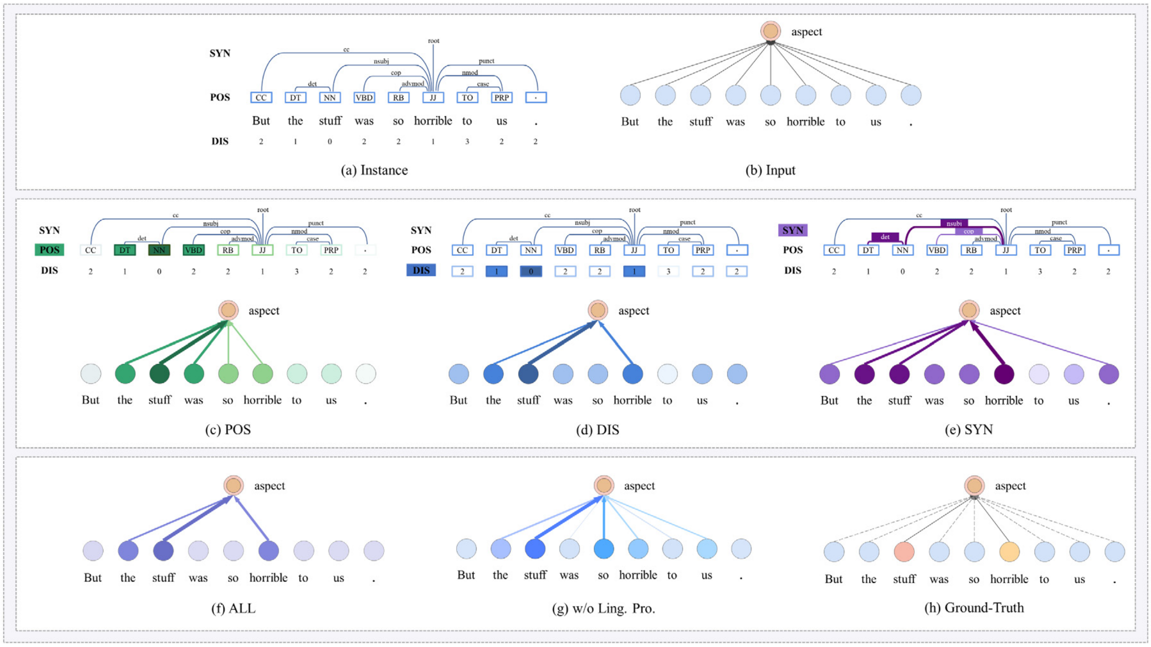
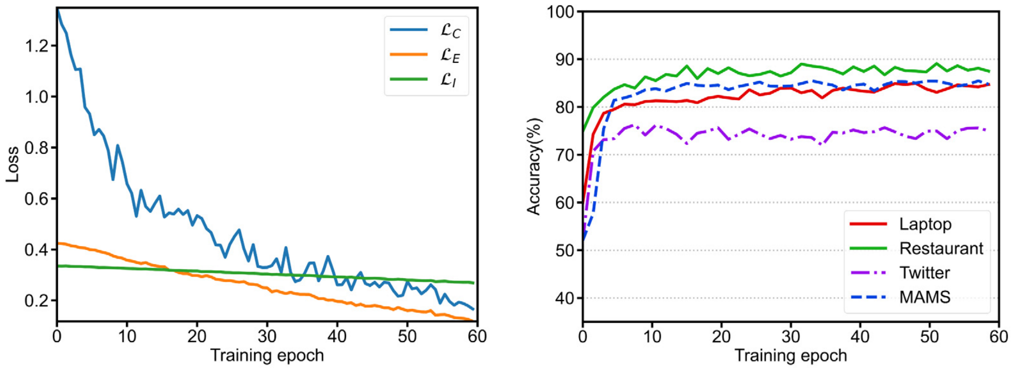


Fig. 7. A qualitative evaluation result for explaining sentiment polarity prediction from Restaurant. The top part includes two subgraphs. Fig. 7(a) and 7(b) show an instance and its corresponding AOBG. Three subgraphs shown in Fig. 7(c), 7(d) and 7(e) show the information flow in linguistic property-aware graphs during forward propagation. Fig. 7(f), 7(g) and 7(h) show the final decision of DLGM, DLGM without linguistic features and the ground-truth explanation label. Thicker edges between pairs of aspect-context nodes have higher weight. Green, blue, and purple denote POS, distance and syntactic dependency features, respectively.

information. Specifically, we randomly select review data and then visualize the contribution of each node and the flow of linguistic property information on edges by using the learned linguistic property-aware graph. Fig. 7 shows an example of the explanation provided by DLGM. Fig. 7(a) and 7(b) show an input instance and its corresponding transformed graph, respectively. Fig. 7(f) and 7(g) show the reason for the decision in DLGM and the ground truth labels of the samples in terms of sentiment classification. Fig. 7(c), 7(d), and 7(e) show how much relevant linguistic property information is fused in each graph from the vectors of neighbors during forward propagation. The information with thicker edges has higher weights.

According to the visualization of the classification reason in Fig. 7, the decision is based mainly on the fusion of representations of *"the"*, *"staff"*, and *"horrible"*. The representations of these nodes are accumulated in the aspect nodes through graph convolution operations. Unlike in the original representation, the contribution of each node begins to differ, and the representations are consistent with the ground truth *"staff"* and *"horrible"* labeled in the dataset. This is also consistent with human classification in reality. As shown in Fig. 7(g), unlike ABSA, the attribute-based model makes predictions based mainly on most of the words in a sentence. Although the weight of each word differs, this weight does not reflect a certain meaning but only reflects that the model associates sentiment information with aspect terms without considering opinion words.

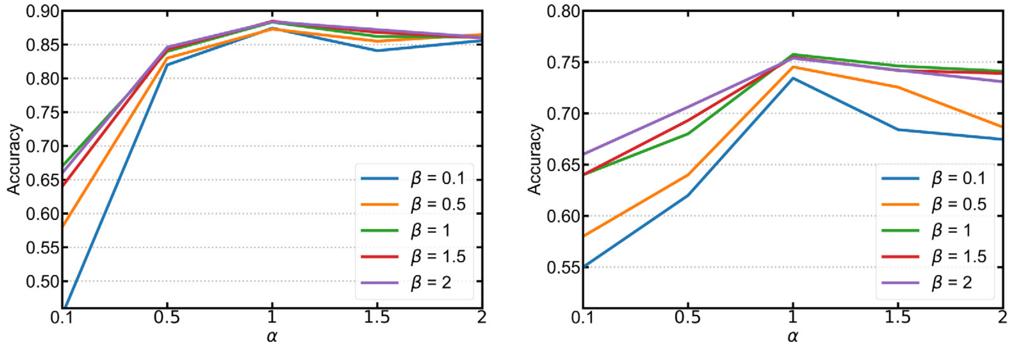
Interestingly, POS, distance, and syntactic dependency all successfully help DLGM to capture opinion and aspect words in the forward propagation by giving different weights. From the perspective of weight distribution, the weight given by the distance accords more with the standard. For the labels *"horrible"* and *"staff"*, POS gives the largest weight to *"staff"* and the smallest weight to *"horrible"*. In the distance property-aware graph, *"staff"* also receives the largest weight, but a greater weight is given to *"horrible"* compared with the wight given by POS. In the syntactic-aware graph, the *"horrible"* node receives the greatest weight, while *"staff"* receives the next greatest weight. In accordance with our original intention, the three graphs pay attention to different information.



(a) Training loss

(b) Test accuracy

Fig. 8. Detailed recording of each training loss and test accuracy of the DLGM during the training epochs.



(a) Restaurant

(b) Twitter

Fig. 9. Effects of varying regularization weights, namely, $\alpha \in \{0.1, 0.5, 1, 1.5, 2\}$ and $\beta \in \{0.1, 0.5, 1, 1.5, 2\}$, on the overall accuracy results of ABSA classification on the validation set.

4.5. Model component analysis

Next, we conduct experiments to further investigate how different modules of our DLGM help improve performance and explainability.

4.5.1. Training loss and test accuracy

We present the training loss and test accuracy of DLGM on the four benchmark datasets to reflect the learning details during the training time. The variation in our loss function while training the model decreases smoothly over the training epochs, as shown in Fig. 8(a). In particular, the loss of classification has a larger reduction than the others. We record the test accuracy curve of DLGM on each dataset, as shown in Fig. 8(b). The curve on Laptop, Restaurant, and MAMS increase stably during training, but the test accuracy curve on Twitter converges quickly possibly because the sentences in Laptop, Restaurant, and MAMS are more consistent with the format of computational linguistics, while the sentences on Twitter are more similar to spoken language without fixed linguistic rules. Hence, our model did not perform satisfactorily on Twitter.

4.5.2. Effects of loss weights

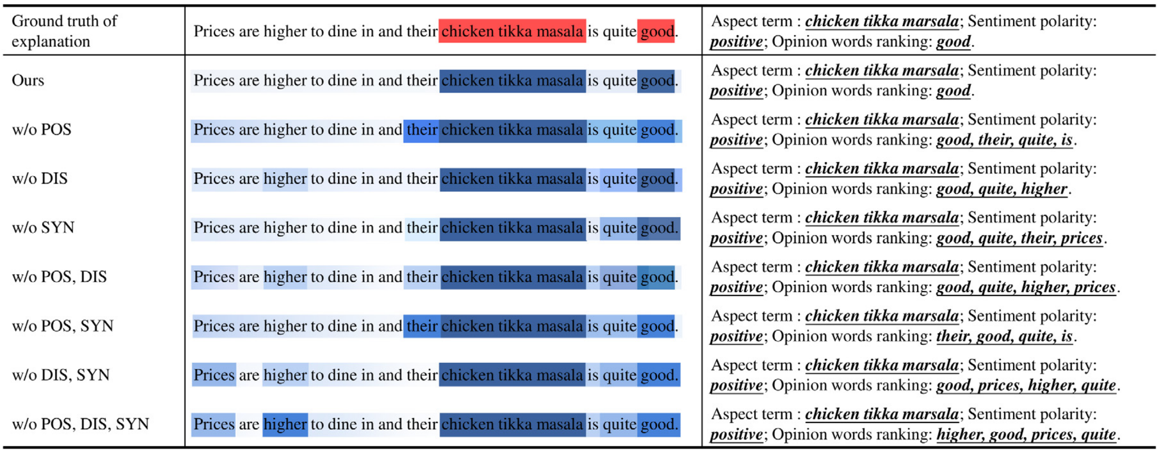
Since we are using elastic loss regularization, we need to tune classification loss coefficient (α) and explainability regularization coefficient (β). The loss regularization controls the final performance of the model directly: an increase in the value of α further enhances performance, whereas higher values of β encourage the linguistic property extraction of correlated neurons. Our aim is to find a balance between performance and explainability while maintaining the original accuracy of the classifier without any regularization ($\beta = 0$). Fig. 9 presents the results of a grid search over various regularization values on the ABSA task. The accuracy difference is minimal for α values greater than 1. We also set the value of β to 1, and both α and β use the same value for all the experiments above.

4.6. Ablation study

4.6.1. Effects of neuron extraction

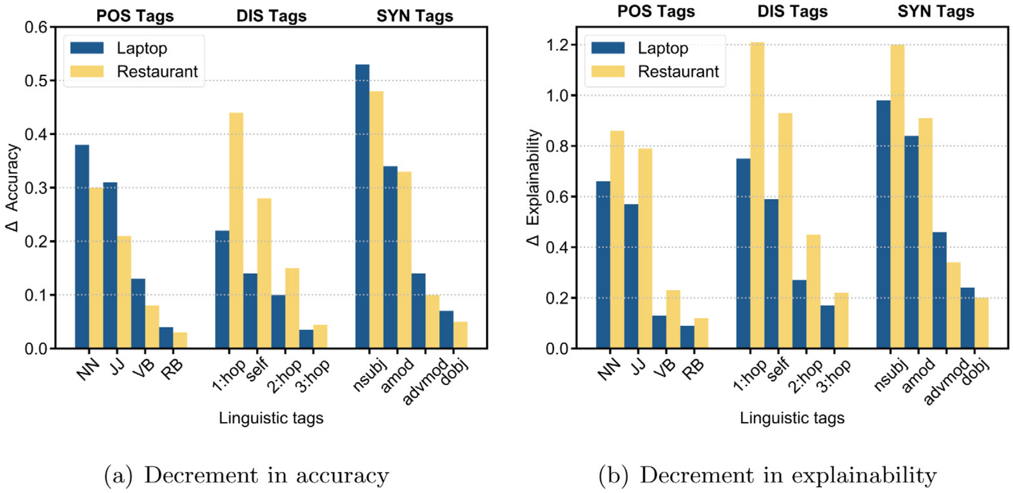
Linguistic property neuron extraction discovers the linguistic property of latent embeddings to learn explainable word embeddings in the network. The impact of this component on performance and explainability is closely related to the number of linguistic properties, and we associate these latent linguistic properties with a definite meaning to ensure the explainability of the model. The impact on model performance and explainability is presented in Table 3, which shows that classification accuracy and explainability improve as the number of linguistic properties on which DLGM relies increases.

Explainability means ensuring that the model is consistent with our preset. Therefore, we randomly select some samples from Restaurant and Laptop to illustrate the results of linguistic property-aware extraction. We present the activations of linguistic-aware neurons in Fig. 10, which shows how neurons can focus on specific linguistic properties. The POS-aware neuron activates with a high positive value for four types of POSs (*noun, adjective, verb, adverb*). The DIS-aware neuron assigns more



Ground truth of explanation	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good***.
Ours	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good***.
w/o POS	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good, their, quite, is***.
w/o DIS	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good, quite, higher***.
w/o SYN	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good, quite, their, prices***.
w/o POS, DIS	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good, quite, higher, prices***.
w/o POS, SYN	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***their, good, quite, is***.
w/o DIS, SYN	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***good, prices, higher, quite***.
w/o POS, DIS, SYN	Prices are higher to dine in and their chicken tikka masala is quite good.	Aspect term : ***chicken tikka marsala***; Sentiment polarity: ***positive***; Opinion words ranking: ***higher, good, prices, quite***.

Fig. 10. Impact of neurons on sentiment polarity predictions and explainability, including visual explanations (word activation maps) and textual explanations (opinion word ranking), according to various linguistic properties. The annotations of the classifications and explanations are also shown.



(a) Decrement in accuracy

(b) Decrement in explainability

Fig. 11. Decrement in accuracy and explainability performance (compared with the performance of the stable DLGM) on Laptop and Restaurant because different missing linguistic tags.

weight to the words closer to the aspect term (including the aspect term). As the result of the SYN-aware neuron shows, the neuron pays more attention to the two syntactic dependencies of *nsubj* and *advamod*. From another perspective, Fig. 10 indicates that our model can learn what we expect the model to do.

Although our results are focused mainly on linguistic properties, the methodology is general for any property where supervision can be created by labeling the data. We pick the top-4 linguistic property tags with the highest frequency and discuss their impact on model performance and explainability. Fig. 11(a) and 11(b) demonstrate how the results of our DLGM vary in terms of independently removing different linguistic property labels on the two benchmarks. In particular, the decrement in classification and explanation accuracy for DLGM is the highest when the following linguistic property tags are missing: (1) POS tags, including *NN*, *JJ*, *VB*, and *RB*; (2) DIS tags, including 1: *hop*, *self*, 2: *hop*, 3: *hop*; and (3) SYN tags, including *nsubj*, *amod*, *advmod*, and *dobj*. The decrement in classification and explanation performance in when these linguistic property tags are missing indicates that these tags carry salient information to filter aspect-opinion pairs for sentiment polarity prediction. This finding is acceptable, as the *NN*, *self*, and *nsubj* tags all provide information for the aspect words. Similarly, *JJ*, 1: *hop*, and *amod* are related to the sentiment modifiers corresponding to the target words. Therefore, finding aspect words and their corresponding opinion words explains how the model makes a prediction.

4.6.2. Effects of independence modeling

As described in Eq. (12), the conicity similarity is an independence regularization to constrain the level of similarity between neurons. We apply the t-SNE algorithm [63] to demonstrate the role of the independence encoder by visualizing the difference between embeddings learned by our model. As shown in Fig. 12, we pick 64 sentences from each dataset to map the 768-dim original word embedding and the 256-dim linguistic property representations into the two-dimensional feature space. The figure shows that the distributions of linguistic property representations learned by our model and the original embedding exhibit distinct structures. This finding proves that using conicity similarity as an independence regularization can better help signal extraction by reducing the semantic similarity among different neurons. However, there are still several connections between the distributions of the individual embeddings, thus explaining the insignificant drop in the independent loss shown in Fig. 8. We suppose that the input signals of neurons from the original word embeddings are determined by the context of sentences, thus ensuring that each embedding has a semantic relationship in the feature space. In the future, we will further disentangle the linguistic embeddings

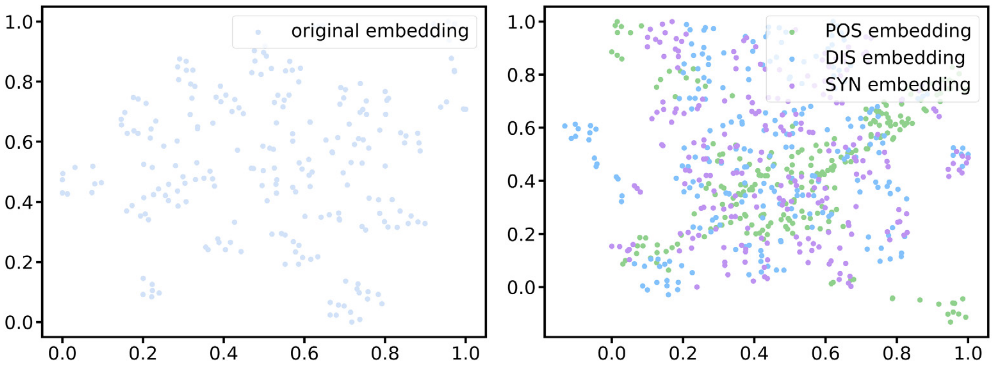
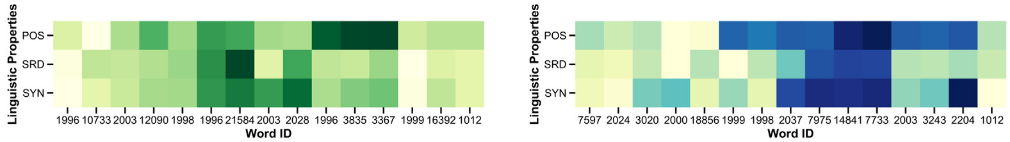


Fig. 12. Visualization of original word embedding and linguistic property embedding on Laptop. We select 64 sentences from the test dataset and visualize the embedding of each word by t-NSE. Overall, these linguistic embeddings show a different distribution from that of the original embedding in the feature space.

Table 4
Effects of the linguistic routing mechanism. Ablation study for ABSA classification on four datasets. The top-1 accuracy and macro-F1 scores are reported.

Embedding methods	Laptop		Restaurant		Twitter		MAMS	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
DLGM: $h_{t,k} = h_{t,k}^{intra} + h_{t,k}^{cross} + \mu$	**84.38**	**81.96**	**88.61**	**83.58**	**75.52**	74.58	**84.77**	**84.25**
DLGM-1: $h_{t,k} = h_{t,k}^{intra} + h_{t,k}^{cross}$	84.03	81.23	87.96	81.36	75.22	**74.61**	83.88	81.95
DLGM-2: $h_{t,k} = h_{t,k}^{intra}$	82.45	78.21	87.19	80.68	74.32	73.46	82.83	81.91
DLGM-3: $h_{t,k} = h_{t,k}^{cross}$	82.78	79.10	86.87	79.98	73.54	72.18	82.93	82.74



(a) Laptop

(b) Restaurant

Fig. 13. Visualization of linguistic property information flow in the individual channels for two samples on the datasets.

inside the pretrained model with the help of other independence constraints to obtain more independent embeddings.

4.6.3. Effects of linguistic routing

DLGM employs the intralinguistic information routing and cross-linguistic information routing mechanism to guide the information flow during information propagation. To further analyze the effects of this linguistic routing, we compare DLGM with its three variants, which are fused $h_{t,k} = h_{t,k}^{intra} + h_{t,k}^{cross} + \mu$ in different ways: (1) DLGM-1 combines $h_{t,k}^{intra}$ and $h_{t,k}^{cross}$ into linguistic embeddings; (2) DLGM-2 performs $h_{t,k}^{intra}$ on linguistic embeddings; and (3) DLGM-3: performs $h_{t,k}^{cross}$ on linguistic embeddings. As presented in Table 4, our stable DLGM outperforms all variants, indicating that our proposed linguistic calculation method is of great significance for improving performance. Compared with DLGM-1, introducing a learnable bias vector μ for linguistic information filtering between word embeddings can promote the reproducibility and robustness of our model.

Furthermore, we provide two examples to illustrate the benefits of the proposed linguistic routing mechanism. More specifically, we randomly pick sentences from Restaurant and Laptop and then visualize the sentences' linguistic property channels information flow in Fig. 13. As the figure shows, each linguistic channel has a separate weight distribution and the influence of each linguistic property across channels is summarized; from this information, we can distill useful linguistic information from the original embedding representation for sentiment analysis.

5. Conclusion

In this article, we jointly model POS, distance, and syntactic dependency by using a disentangled linguistic graph model (DLGM) in a supervised manner for a new task named explainable ABSA. By using the proposed aspect-oriented bipartite graph, we can simulate the information transfer process within the model in a unified graph structure. Moreover, we adopt independent regulation loss to minimize the information redundancy between neurons to ensure a faithful explanation. Finally, a linguistic information routing mechanism is introduced into the GNN to overcome the drawbacks of intralinguistic information propagation for classification. The experimental results show that DLGM performs respectably while having good explainability.

However, several limitations still need to be addressed. For example, the experimental evaluated metrics for explainability were not proposed for explanatory models. In the future, we need more reasonable and robust metrics to evaluate model performance. Another improved direction is how to make the loss function of explainability drop significantly between different linguistic neurons, where we try to replace the conicity similarity with other statistical measures. In addition, linguistic rule-based explainability is unsuitable for corpora with ambiguous linguistic rules. We will attempt to use neurosymbolic AI for explainable sentiment analysis in follow-up work.

CRediT authorship contribution statement

Xiaoyong Mei: Conceptualization, Methodology, Investigation, Supervision. **Yougen Zhou:** Conceptualization, Methodology, Writing – original draft, Investigation, Software, Validation. **Chenjing Zhu:** Investigation, Software, Validation. **Mengting Wu:** Investigation, Software, Validation. **Ming Li:** Writing – review & editing. **Shirui Pan:** Writing – review & editing.

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.

Data availability

All the datasets used in this research are benchmark data that are publicly available online.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2022YFC3303600, in part by the Key Research and Development Program of Zhejiang Province, China under Grant No. 2021C03141 and No. 2022C03106, and in part by the National Natural Science Foundation of China under Grant No. 61877020, No. 62037001, and No. 62172370.

References

[1] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation, 2015, pp. 486–495.

[2] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S.M. Jiménez-Zafra, G. Eryiğit, Semeval-2016 task 5: Aspect based sentiment analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation, 2016, pp. 19–30.

[3] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107, http://dx.doi.org/10.1109/MIS.2016.31.

[4] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3229–3238.

[5] B. Huang, R. Guo, Y. Zhu, Z. Fang, G. Zeng, J. Liu, Y. Wang, H. Fujita, Z. Shi, Aspect-level sentiment analysis with aspect-specific context position information, Knowl.-Based Syst. 243 (2022) 108473.

[6] B. Liang, H. Su, L. Gui, E. Cambria, R. Xu, Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks, Knowl.-Based Syst. 235 (2022) 107643.

[7] X. Bai, P. Liu, Y. Zhang, Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network, IEEE/ACM Trans. Audio Speech Lang. Process. 29 (2021) 503–514.

[8] C. Huang, M. Li, F. Cao, H. Fujita, Z. Li, X. Wu, Are graph convolutional networks with random weights feasible? EEE Trans. Pattern Anal. Mach. Intell. (2022) http://dx.doi.org/10.1109/TPAMI.2022.3183143.

[9] R.K. Yadav, L. Jiao, O.-C. Granmo, M. Goodwin, Human-level interpretable learning for aspect-based sentiment analysis, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 14203–14212.

[10] Y. Zhang, P. Tiňo, A. Leonardis, K. Tang, A survey on neural network interpretability, IEEE Trans. Emerg. Topics Comput. Intell. 5 (5) (2021) 726–742.

[11] D.-T. Vo, Y. Zhang, Target-dependent Twitter sentiment classification with rich automatic features, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 1347–1353.

[12] D.-H. Pham, A.-C. Le, Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis, Internat. J. Approx. Reason. 103 (2018) 1–10.

[13] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 3298–3307.

[14] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 4068–4074.

[15] Y. Tay, L.A. Tuan, S.C. Hui, Dyadic memory networks for aspect-based sentiment analysis, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 107–116.

[16] Y. Song, J. Wang, T. Jiang, Z. Liu, Y. Rao, Attentional encoder network for targeted sentiment classification, 2019, arXiv preprint arXiv:1902.09314.

[17] L. Li, Y. Liu, A. Zhou, Hierarchical attention based position-aware network for aspect-level sentiment analysis, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, 2018, pp. 181–189.

[18] M.H. Phan, P.O. Ogunbona, Modelling context and syntactical features for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3211–3220.

[19] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent Twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 49–54.

[20] Y. Tian, G. Chen, Y. Song, Enhancing aspect-level sentiment analysis with word dependencies, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, 2021, pp. 3726–3739.

[21] Y. Zheng, R. Zhang, S. Mensah, Y. Mao, Replicate, walk, and stop on syntax: An effective neural network model for aspect-level sentiment classification, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 9685–9692.

[22] R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, Effective attention modeling for aspect-level sentiment classification, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1121–1131.

[23] C. Zhang, Q. Li, D. Song, Syntax-aware aspect-level sentiment classification with proximity-weighted convolution network, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1145–1148.

[24] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, 2017, arXiv preprint arXiv:1702.08608.

[25] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[26] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, P. Sen, A survey of the state of explainable AI for natural language processing, in: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, 2020, pp. 447–459.

[27] H. Liu, Q. Yin, W.Y. Wang, Towards explainable NLP: A generative explanation framework for text classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5570–5581.

[28] A.V. Konstantinov, L.V. Utkin, Interpretable machine learning with an ensemble of gradient boosting machines, Knowl.-Based Syst. 222 (2021) 106993.

[29] S. Lapuschkin, A. Binder, G. Montavon, F. Klauschen, K.-R. Mller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (2015) e0130140.

[30] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013, arXiv preprint arXiv:1312.6034.

[31] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of European Conference on Computer Vision, Springer, 2014, pp. 818–833.

[32] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.

[33] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing: Towards contrastive explanations with pertinent negatives, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 590–601.

[34] K. Kanamori, T. Takagi, K. Kobayashi, H. Arimura, DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 2855–2862.

[35] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: The 3rd International Conference on Learning Representations, 2015.

[36] G. Brunner, Y. Liu, D. Pascual, O. Richter, R. Wattenhofer, On the validity of self-attention as explanation in transformer models, 2019, 40, arXiv preprint arXiv:1908.04211.

[37] S. Jain, B.C. Wallace, Attention is not explanation, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 3543–3556.

[38] S. Serrano, N.A. Smith, Is attention interpretable? in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2931–2951.

[39] S. Vashishth, S. Upadhyay, G.S. Tomar, M. Faruqui, Attention interpretability across nlp tasks, 2019, arXiv preprint arXiv:1909.11218.

[40] S. Wiegreffe, Y. Pinter, Attention is not not explanation, 2019, arXiv preprint arXiv:1908.04626.

[41] E. Cambria, Q. Liu, S. Decherchi, F. Xing, K. Kwok, SenticNet 7: a commonsense-based neurosymbolic AI framework for explainable sentiment analysis, in: Proceedings of LREC 2022, 2022.

[42] S. He, Z. Li, H. Zhao, H. Bai, Syntax for semantic role labeling, to be, or not to be, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 2061–2071.

[43] G. Xu, P. Liu, Z. Zhu, J. Liu, F. Xu, Attention-enhanced graph convolutional networks for aspect-based sentiment classification with multi-head attention, Appl. Sci. 11 (8) (2021) 2076–3417.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.

[45] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, arXiv preprint arXiv:1907.11692.

[46] K. Clark, U. Khandelwal, O. Levy, C.D. Manning, What does BERT look at? An analysis of BERT's attention, in: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2019, pp. 276–286.

[47] J. Hewitt, P. Liang, Designing and interpreting probes with control tasks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 2733–2743.

[48] W. Li, W. Shao, S. Ji, E. Cambria, BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis, Neurocomputing 467 (2022) 73–82.

[49] Chandrahas, A. Sharma, P. Talukdar, Towards understanding the geometry of knowledge graph embeddings, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 122–131.

[50] A. Sai, M.D. Gupta, M.M. Khapra, M. Srinivasan, Re-evaluating ADEM: a deeper look at scoring dialogue responses, in: Proceedings of 33rd the AAAI Conference on Artificial Intelligence, 2019, pp. 6220–6227.

[51] T. Dozat, C.D. Manning, Deep biaffine attention for neural dependency parsing, in: The 5th International Conference on Learning Representations, 2017.

[52] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[53] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 946–956.

[54] F. Fan, Y. Feng, D. Zhao, Multi-grained attention network for aspect-level sentiment classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3433–3442.

[55] B. Huang, Y. Ou, K.M. Carley, Aspect level sentiment classification with attention-over-attention neural networks, in: R. Thomson, C. Dancy, A. Hyder, H. Bisgin (Eds.), Social, Cultural, and Behavioral Modeling, 2018, pp. 197–206.

[56] Q. Jiang, L. Chen, R. Xu, X. Ao, M. Yang, 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, 2019, pp. 6280–6285.

[57] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: Advances in Neural Information Processing Systems, vol. 30, 2017.

[58] H. Xu, B. Liu, L. Shu, P. Yu, BERT post-training for review reading comprehension and aspect-based sentiment analysis, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 2324–2335.

[59] T.H. Nguyen, K. Shirai, PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2509–2514.

[60] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, 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, 2019, pp. 5679–5688.

[61] L. Huang, X. Sun, S. Li, L. Zhang, H. Wang, Syntax-aware graph attention network for aspect-level sentiment classification, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 799–810.

[62] J. Dai, H. Yan, T. Sun, P. Liu, X. Qiu, Does syntax matter? A strong baseline for Aspect-based Sentiment Analysis with RoBERTa, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 1816–1829.

[63] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605.