Aspect-based Sentiment Analysis with Context Denoising

Yuanhe Tian♥♠, Chang Liu♠, Yan Song♠†, Fei Xia♥, Yongdong Zhang♠
♥University of Washington ♠University of Science and Technology of China
♥{yhtian, fxia}@uw.edu ♠lc980413@mail.ustc.edu.cn
♠clksong@gmail.com ♠zhyd73@ustc.edu.cn

Abstract

Given a sentence and a particular aspect term, aspect-based sentiment analysis (ABSA) aims to predict the sentiment polarity towards this aspect term, which provides fine-grained analysis on sentiment understanding and it has attracted much attention in recent years. In order to achieve a good performance on ABSA, it is important for a model to appropriately encode contextual information, especially identifying salient features and eliminating noise in the context. To make incorrect predictions, most existing approaches employ powerful text encoders to locate important context features, as well as noises that mislead ABSA models. These approaches determine the noise in the text for ABSA by assigning low weights to context features or directly removing them from model input, which runs the risk of computing wrong weights or eliminating important context information. In this paper, we propose to improve ABSA with context denoising, where three types of word-level information are regarded as noise, namely, lexicographic noise, bag-of-words noise, and syntax noise. We utilize diffusion networks to perform the denoising process to gradually eliminate them so as to better predict sentiment polarities for given aspect terms. Our approach uses task-specific noise rather than the standard stochastic Gaussian noise in the diffusion networks. The experimental results on five widely used ABSA datasets demonstrate the validity and effectiveness of our approach.[1]

1 Introduction

Aspect-based sentiment analysis (ABSA) predicts sentiment polarity of an aspect term in a sentence on the fine-grained level. For example, the sentiments for "*environment*" and "*bar service*" in the sentence in Figure 1 are *positive* and *negative*, respectively, and aspect-based sentiments can be different from that of the entire sentence (i.e., *negative*). Identifying sentiment for aspects is important in many real-world applications, such as analyzing the product review of users and monitoring the opinion changes on social media, and the task has attracted much attention in recent years (Song et al., 2019; Huang and Carley, 2019; Xu et al., 2020; Tian et al., 2021; Yu et al., 2021b; Liang et al., 2022; Qin et al., 2022; Yu et al., 2023; Mukherjee et al., 2023; Bao et al., 2023b).

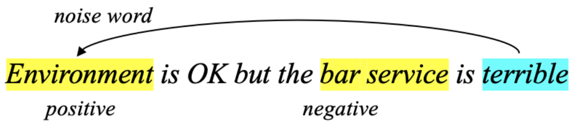

Figure 1: An example of aspect-based sentiment analysis, where the sentiment polarities of aspect terms " *environment*" and "*bar service*" are positive and negative, respectively. Herein, "*terrible*" serves as context noise in predicting the sentiment of "*environment*".

To perform well on ABSA, a system needs to have a good representation of the context of a given aspect term and be able to identify salient features that are important in predicting the sentiment of the aspect term. Many existing studies use advanced encoders (e.g., BiLSTM and Transformer (Vaswani et al., 2017)) to capture contextual information for the task and achieve good performance (Liang et al., 2019; Tang et al., 2020; Chen et al., 2020; Zhang et al., 2021; Cao et al., 2022; Varia et al., 2023; Wang et al., 2023). Some studies incorporate external knowledge, such as lexicon, chunks, and syntactic information (He et al., 2018; Huang and Carley, 2019; Zhang et al., 2019; Wang et al., 2020; Liang et al., 2021; Chen et al., 2022; Zhang et al., 2023a; Ma et al., 2023) to further improve model performance.

These studies generally extract important contextual information from intrinsic or external knowledge and use them as essential hints to predict sentiment polarities and they may suffer from noise

†Corresponding author.

[1] The code and relevant resources used in the paper are available at `https://github.com/synlp/ASA-CD`.

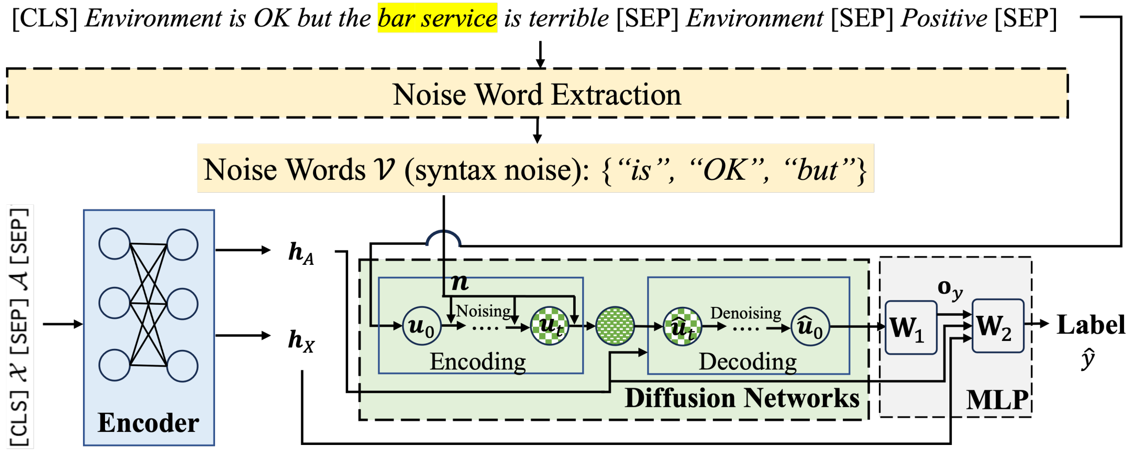
[CLS] Environment is OK but the bar service is terrible [SEP] Environment [SEP] Positive [SEP]
Noise Word Extraction
Noise Words $\mathcal{V}$ (syntax noise): {"is", "OK", "but"}
[CLS] $\mathcal{X}$ [SEP] $\mathcal{A}$ [SEP]
Encoder
h_A
h_X
n
Noising
u_0
u_t
Encoding
Denoising
$\hat{u}_t$
$\hat{u}_0$
Decoding
Diffusion Networks
o_y
W_1
W_2
MLP
Label
$\hat{y}$

Figure 2: An overview of our approach for ABSA. The left part shows the input encoding process to extract aspect representation (denoted as $\mathbf{h}_A$) and sentence representation (denoted as $\mathbf{h}_X$); the center is the proposed noise word extraction (highlighted in the yellow background) and the diffusion networks (highlighted in the green background). The right part presents the MLP decoder to predict the sentiment label. An example input of the context denoising process and the expected noise words (under the syntax noise setting) are provided for better illustration. $\mathcal{X}$ and $\mathcal{A}$ are the input sentence and aspect term, respectively; $\mathbf{u}_0$ and $\mathbf{u}_t$ are the initial and t-th step representation of the (sentence, aspect term, sentiment) triple in the noising process, respectively; $\widehat{\mathbf{u}}_0$ and $\widehat{\mathbf{u}}_t$ are the initial and t-th step representation of the (sentence, aspect term, sentiment) triple in the denoising process, respectively; $\mathbf{n}$ is the noise vector; $\mathbf{o}_y$ is obtained by passing the last vector of $\widehat{\mathbf{u}}_0$ through a fully connected layer.

introduced by the knowledge extraction process. More importantly, these studies do not explicitly identify and eliminate redundant and interfering information (noise) for ABSA from the input before predicting the sentiment. E.g., in Figure 1 while "*terrible*" is a good indicator of the polarity of the aspect term "*bar service*", it is a noise word for the aspect term "*environment*". The noise word introduces unimportant information that confuses an ABSA model when it tries to predict the sentiment. If the unimportant noise information is identified and removed from the context, it would be easier for an ABSA model to handle the task. Therefore, we believe that denoising the context of aspect terms has the potential to improve the performance of ABSA systems.

In this paper, we propose an approach to improve ABSA through diffusion networks, which consists of forward noising and backward denoising processes[2]. Herein, we propose to use task-specific noises associated with unimportant words extracted from the context of the running text for ABSA, rather than the standard stochastic Gaussian noise in the forward pass. We consider three types of noises, namely, lexicographic noise, bag-of-words noise, and syntax noise, in extracting the noise words. In the denoising process, our approach starts from a noise vector, gradually eliminates the noise from it, and predicts the sentiment polarity. Through this process, our approach distinguishes the noises and eliminates their effect on ABSA in predicting the sentiment, which allows our approach to focus on important features and thus improve model performance. We evaluate our approach on five widely used English ABSA datasets, where our approach outperforms strong baselines and achieves state-of-the-art results.

2 The Approach

In general, ABSA is performed as a classification task on sentence-aspect pairs (Ma et al., 2017; Tang et al., 2019; Qin et al., 2022). In this paper, we follow the encoding-decoding paradigm with the enhancement of diffusion networks, and the architecture of our approach is illustrated in Figure 2. In the diffusion networks, the noise representation is based on a set of noise words $\mathcal{V}$ extracted from the input sentence. Thus, our approach for ABSA is formulated by

$$\widehat{y} = f(\mathcal{X}, \mathcal{A}, \mathcal{V}) \qquad (1)$$

where $\mathcal{X}$ is the input text, $\mathcal{A} = a_1 \cdots a_m \cdots a_M$ denotes the aspect term with M words (herein, $\mathcal{A}$ is the sub-string of $\mathcal{X}$), and $\widehat{y}$ is the sentiment polarity to the aspect term. In this section, we will introduce first the encoding process for $\mathcal{X}$ and $\mathcal{A}$, then the ways to extract different types of noise words, and

[2] The term "denoising" in the diffusion networks refers to the process of eliminating noise from a noise vector (e.g., the stochastic Gaussian noise vector in the standard diffusion networks) to reconstruct the original vector.

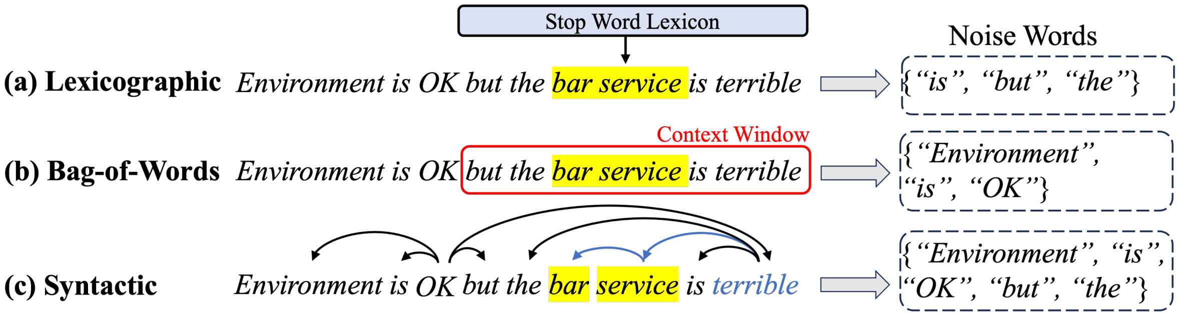
Stop Word Lexicon
Noise Words
(a) Lexicographic
Environment is OK but the bar service is terrible
{"is", "but", "the"}
Context Window
(b) Bag-of-Words
Environment is OK but the bar service is terrible
{"Environment", "is", "OK"}
(c) Syntactic
Environment is OK but the bar service is terrible
{"Environment", "is", "OK", "but", "the"}

Figure 3: Illustrations of different types of noise words extracted from an example sentence and aspect term pair, where the aspect term is highlighted in yellow background. The context window is represented in a red box and the first-order dependencies with respect to the aspect term are represented in blue.

finally the proposed diffusion networks to address noise and predict the sentiment label $\widehat{y}$.

2.1 Input Encoding

The encoding process aims to model the context information of the input and extracts the representations of $\mathcal{X}$ and $\mathcal{A}$, which are used in the following process to predict the sentiment label $\widehat{y}$. Specifically, we follow the convention in existing studies to concatenate $\mathcal{X}$ and $\mathcal{A}$ with some special tokens (e.g., "[CLS]" and "[SEP]") to mark the boundaries of them, to form a new input word sequence, namely, "[CLS] $\mathcal{X}$ [SEP] $\mathcal{A}$ [SEP]". Next, we feed the new input word sequence into a text encoder (e.g., BERT) and obtain the hidden vectors for each input word, where the hidden vector for "[CLS]" and the m-th word in the aspect term are denoted as $\mathbf{h}$ and $\mathbf{h}_m$, respectively. We regard the resulting $\mathbf{h}_X$ as the representation of the sentence and compute the average of $\mathbf{h}_m$ ($1 \leq m \leq M$) by

$$\mathbf{h}_A = \frac{1}{M}\sum_{m=1}^{M}\mathbf{h}_m \qquad (2)$$

where $\mathbf{h}_A$ denotes the representation of the aspect term that is prepared to be used in the proposed diffusion networks to predict the sentiment polarity.

2.2 Noise Extraction

Different from standard diffusion models that leverage stochastic Gaussian noise in training and inference, our approach utilizes task-specific noises for ABSA derived from contexts. We associate the noise with unimportant words in the sentence for ABSA. We compare three types of noise, namely, lexicographic noise based on a static lexicon, bag-of-words noise extracted from an aspect-centric window, and syntactic noise derived from the sentence structure, which are explained below with the examples in Figure 3.

Lexicographic Noise Since stop words occur frequently but usually do not carry significant meaning or contribute to the understanding of the text, we use an existing stop word lexicon (e.g., stop words in NLTK Toolkit (Bird and Loper, 2004)) to extract noise words. Specifically, we find all words in the sentence that appear in the stop word lexicon and add them to the set of noise words $\mathcal{V}$. As shown in Figure 3 (a), the noise words extracted from the sentence are "*is*", "*the*", and "*but*".

Bag-of-Words Noise Consider words that are distant from the aspect term generally contain unimportant contextual information that fails to contribute to ABSA, we select words that are outside the context window of the aspect term as the noise words. That is, we select words whose word-based distance to the aspect term is greater than the context window size c. For example, as illustrated in Figure 3 (b), when $c = 2$, the context window covers the words from "*but*" to "*terrible*" and thus the noise words for the aspect term "*bar service*" are the rest of the words in the sentence, namely, "*environment*", "*is*", and "*OK*".

Syntax Noise Bag-of-words noise words are extracted according to the surface word order of the sentence, which may include important words that are distant from the aspect term and thus lead to inferior results. Therefore, one should also consider the structure of the sentence when extracting the noise words. Among different types of syntax structures, the dependency tree of the input sentence constructs connections among words in the sentence and is used in many existing studies to identify the important and unimportant contexts for ABSA. Therefore, we use the dependency tree of the text to extract noise words for the syntax level.

Specifically, we use an off-the-shelf dependency parser (e.g., the parser in Stanford CoreNLP Toolkit (Manning et al., 2014)) to produce the dependency tree of $\mathcal{X}$. Because words close to the aspect term $\mathcal{A}$ in the dependency tree generally convey important contextual features for ABSA, we locate words that are within different orders of dependencies of the aspect term[3] and regard the rest words in $\mathcal{X}$ as the noise words, which form the set $\mathcal{V}$. For example, for the aspect term "*bar service*" in the example in Figure 3 (c), words within the first-order dependencies are "*bar*", "*terrible*". Therefore, the noise words are the rest of the words in $\mathcal{X}$, including "*environment*", "*is*", "*OK*", "*but*", "*the*", and "*is*". Through this process, $\mathcal{V}$ contains words that are syntactically distant from the aspect term and thus serve as noise to be eliminated when predicting the sentiment polarity of the aspect term.

2.3 Diffusion Networks

The diffusion networks consist of noising and denoising processes to address the noise in $\mathcal{X}$ for predicting the sentiment polarity label. During the noising process, the information of noise words is added to the input of the diffusion networks, which results in a vector that mainly contains the noise information; then, in the denoising process, the diffusion networks learn a diffusion decoder to eliminate the noise from the vector. Using the diffusion decoder, the diffusion networks are able to address the noise appropriately and thus help ABSA. The details of the noising and denoising processes in the diffusion networks are illustrated as follows.

Noising Process Once we extract the noise word set based on the word dependencies, the tokens in $\mathcal{V}$ are used to generate noise in the forward noising pass of diffusion networks. Forward encoding aims to add noise to the input representation $\mathbf{u}_0$ to compute a sequence of latent representations $\mathcal{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_T]$ (T denotes the total steps). Herein, we combine the input sentence $\mathcal{X}$, the aspect term $\mathcal{A}$, and the gold standard sentiment label y^* to construct a new word sequence $\mathcal{X}'$. Then we follow DDCap (Zhu et al., 2022) to convert tokens of $\mathcal{X}'$ into the one-hot representation $\mathbf{u}_0$. Meanwhile, we compute $\mathbf{u}_t$ at t-th step by

$$\mathbf{u}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{u}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \mathbf{n} \tag{3}$$

where $\bar{\alpha}_t$ denotes a blending scalar hyperparameter that is correlated to the DDPM noise scheduling strategy (Ho et al., 2020) and $\mathbf{n}$ refers to the noise vector[4] coming from the noise words in $\mathcal{V}$. Specifically, we randomly sample N words from $\mathcal{V}$ and map each word to its embedding through an embedding matrix, where the embedding of the n-th word is denoted as $\mathbf{e}_n$. Then, we compute the average of the word embeddings and normalize ($Norm$) the resulted embedding to get the noise vector $\mathbf{n}$, formulated by

$$\mathbf{n} = Norm(\frac{1}{N}\sum_{n=1}^{N} \mathbf{e}_n) \tag{4}$$

Denoising Process We follow the standard process of diffusion model to denoise $\mathbf{u}_T$ to reconstruct $\mathbf{u}_0$. It is worth noting that the denoising processes in training and inference are different. In training, we compute the diffusion loss $\mathcal{L}_{diff}$ by

$$\mathcal{L}_{diff} = \mathbb{E}_{t \sim U(0,T)} \| f_d(\mathbf{u}_t, \mathbf{h}_A, t) - \mathbf{u}_0 \|_2^2 \tag{5}$$

where f_d is a diffusion decoder using Transformer architecture to recover $\mathbf{u}_{t-1}$ based on $\mathbf{u}_t$ and $\mathbf{h}_A$ with $\mathbf{h}_A$ and $\mathbf{u}_t$ modeled by the Transformer encoder and decoder architectures, respectively. Meanwhile, we extract the last vector (that corresponds to the gold standard label y^* and is denoted as $\widehat{\mathbf{u}}_{0,-1}$) of the recovered input matrix (denoted as $\widehat{\mathbf{u}}_0$) and use a fully connected layer to map it into a vector $\mathbf{o}_y$ through

$$\mathbf{o}_y = \mathtt{ReLU}(\mathbf{W}_1 \cdot \widehat{\mathbf{u}}_{0,-1}) \tag{6}$$

where $\mathbf{W}_1$ is a trainable matrix and ReLU is the activation function. Then, we concatenate $\mathbf{o}_y$ with the sentence representation $\mathbf{h}_X$, as well as the aspect representation $\mathbf{h}_A$, and use a softmax classifier to predict the sentiment label $\widehat{y}$ through

$$\widehat{y} = \mathtt{Softmax}(\mathbf{W}_2 \cdot (\mathbf{o}_y \oplus \mathbf{h}_X \oplus \mathbf{h}_A)) \tag{7}$$

where $\mathbf{W}_2$ is a trainable matrix. Afterwards, we compute the standard cross-entropy loss $\mathcal{L}_{CE}$ by comparing $\widehat{y}$ with the gold standard label y^*. Finally, we add $\mathcal{L}_{diff}$ and $\mathcal{L}_{CE}$ to compute the total loss $\mathcal{L}$, which is formulated by

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{diff} \tag{8}$$

and our approach is optimized accordingly.

The inference process of the diffusion networks follows the standard process of DDCap, where the

[3] If the aspect term has two or more words, we use the dependency connections of its last word which generally serves as the head of noun phrases (i.e., aspect terms) in English.

[4] Here, $\mathbf{u}_0$ is a matrix where its first dimension equals to the word-based length l of the input. To perform Eq. (3), the noise vector is replicated l times, resulting in the set $\mathbf{n}_1, \ldots, \mathbf{n}_l$. We stack these vectors to form a new matrix, ensuring its dimensions match these of $\mathbf{u}_0$.

Dataset		Pos. #	Neu. #	Neg. #
LAP14	Train	994	464	870
	Test	341	169	128
REST14	Train	2,164	637	807
	Test	728	196	182
REST15	Train	907	36	254
	Test	326	34	207
REST16	Train	1,229	69	437
	Test	469	30	114
MAMS	Train	3,380	5,042	2,764
	Dev	403	604	325
	Test	400	607	329

Table 1: The statistics of the datasets, where the number of instances with different sentiment polarities in the training, development, and test sets are reported.

first step is to construct a noise vector and then remove the noise in it through the diffusion decoder. For the noise vector, we randomly sample $\mathcal{N}$ tokens from $\mathcal{V}$ to derive the noise vector $\mathbf{n}$ following the same processes as Eq. (4). Then, we initialize $\widehat{\mathbf{u}}_T$ with $\mathbf{n}$ and use f_D to iteratively subtract noises from $\widehat{\mathbf{u}}_T$. Therefore, the overall process is formulated as

$$\widehat{\mathbf{u}}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot \frac{\widehat{\mathbf{u}}_t - \sqrt{1-\bar{\alpha}_t} \cdot f_D(\widehat{\mathbf{u}}_t, \mathbf{h}_A, t)}{\sqrt{\bar{\alpha}_t}} + \sqrt{1-\bar{\alpha}_{t-1}} \cdot f_D(\widehat{\mathbf{u}}_t, \mathbf{h}_A, t) \quad (9)$$

Through the process, the diffusion networks obtain the denoised representations $\widehat{\mathbf{u}}_0$, whose last vector is extracted and used to predict the final sentiment label $\widehat{y}$ following the same process in training.

3 Experimental Settings

3.1 Datasets

Following previous studies, we run different models on five English benchmark datasets from various domains for ABSA; the datasets are LAP14 and REST14 (Pontiki et al., 2014), REST15 (Pontiki et al., 2015), REST16 (Pontiki et al., 2016), and MAMS[5] (Jiang et al., 2019). Specifically, LAP14 contains laptop computer reviews, REST14, REST15, REST16, and MAMS are collected from online reviews of restaurants. In addition, it is worth noting that the instances in the MAMS dataset are all cases where one sentence contains multiple aspect terms with different sentiment polarities. Therefore, it serves as a good resource to test models on the hard cases. For all datasets, we use their official train/dev/test splits. The statistics of the datasets used in the experiments are reported in Table 1, where the number of instances with different sentiment polarities in the training, development, and test sets are reported.

Hyper-parameters	Values
Learning Rate	5e-6, **1e-5**, 3e-5, 5e-5
Warmup Rate	**0.1**, 0.2
Dropout Rate	**0.1**
Batch Size	4, **8**

Table 2: The hyper-parameters used in tuning our models and the best one used in our final experiments are highlighted in boldface.

3.2 Implementation Details

As the performance of NLP models highly depends on the text representations (Conneau et al., 2017; Song et al., 2017; Song and Shi, 2018; Han et al., 2018; Sileo et al., 2019; Song et al., 2021; Gan et al., 2023), we employ BERT (Devlin et al., 2019) and LLaMA-2 (Touvron et al., 2023) to encode the text, which have achieved state-of-the-art performance in many NLP tasks. Specifically, for BERT, we use the uncased BERT-base and BERT-large with their default settings, i.e., 12 layers of self-attention with 768-dimensional hidden vectors for BERT-base and 24 layers of self-attention with 1024 dimensional hidden vectors for BERT-large[6]. For LLaMA-2, we use the 7B version that has 32 layers of self-attentions with 4,096-dimensional hidden vectors. For the diffusion decoder, we use Transformer with three layers of multi-head attentions, where we use 768- and 1024-dimensional hidden vectors when the encoder is BERT-base and BERT-large, respectively, and employ 4,096-dimensional hidden vectors when it is equipped with LLaMA-2, so as to match their hidden vector dimensions.

To obtain the noise words, we use the stop words in the NLTK Toolkit, try window sizes of 1, 2, and 3, and use the Stanford CoreNLP Toolkit to parse the input sentence. The number of steps in the training and inference process in diffusion networks is set to 60. Besides, we initialize all other trainable

[5] We use the ATSA part of MAMS obtained from https://github.com/siat-nlp/MAMS-for-ABSA.

[6] We obtain the BERT models from https://github.com/huggingface/pytorch-pretrained-BERT.

	LAP14		**REST14**		**REST15**		**REST16**		**MAMS**	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
BERT-base	78.12$_{\pm 0.44}$	74.80$_{\pm 0.37}$	84.67$_{\pm 0.40}$	76.63$_{\pm 0.43}$	83.41$_{\pm 0.44}$	67.80$_{\pm 0.39}$	89.03$_{\pm 0.37}$	79.73$_{\pm 0.38}$	81.35$_{\pm 0.18}$	81.43$_{\pm 0.16}$
+ D	78.56$_{\pm 0.42}$	75.44$_{\pm 0.42}$	85.05$_{\pm 0.41}$	77.00$_{\pm 0.41}$	83.93$_{\pm 0.42}$	68.29$_{\pm 0.44}$	89.56$_{\pm 0.46}$	80.18$_{\pm 0.37}$	81.83$_{\pm 0.10}$	81.97$_{\pm 0.16}$
+ CD (L)	79.98$_{\pm 0.42}$	76.86$_{\pm 0.41}$	86.36$_{\pm 0.47}$	78.43$_{\pm 0.38}$	85.16$_{\pm 0.38}$	69.80$_{\pm 0.38}$	90.96$_{\pm 0.42}$	81.38$_{\pm 0.43}$	83.33$_{\pm 0.17}$	83.42$_{\pm 0.16}$
+ CD (B1)	80.03$_{\pm 0.44}$	76.93$_{\pm 0.42}$	86.52$_{\pm 0.42}$	78.69$_{\pm 0.42}$	85.26$_{\pm 0.40}$	69.62$_{\pm 0.37}$	91.07$_{\pm 0.47}$	81.61$_{\pm 0.34}$	83.59$_{\pm 0.11}$	83.57$_{\pm 0.15}$
+ CD (B2)	80.28$_{\pm 0.37}$	77.18$_{\pm 0.38}$	86.75$_{\pm 0.43}$	78.89$_{\pm 0.39}$	85.54$_{\pm 0.33}$	69.88$_{\pm 0.42}$	91.27$_{\pm 0.40}$	81.90$_{\pm 0.46}$	83.80$_{\pm 0.14}$	83.84$_{\pm 0.16}$
+ CD (B3)	80.02$_{\pm 0.39}$	76.95$_{\pm 0.41}$	86.46$_{\pm 0.41}$	78.41$_{\pm 0.38}$	85.26$_{\pm 0.36}$	69.75$_{\pm 0.43}$	90.99$_{\pm 0.39}$	81.43$_{\pm 0.45}$	83.32$_{\pm 0.15}$	83.56$_{\pm 0.10}$
+ CD (S1)	82.13$_{\pm 0.40}$	79.37$_{\pm 0.36}$	87.03$_{\pm 0.41}$	81.57$_{\pm 0.38}$	85.87$_{\pm 0.39}$	74.03$_{\pm 0.35}$	92.63$_{\pm 0.38}$	83.12$_{\pm 0.37}$	85.07$_{\pm 0.13}$	84.51$_{\pm 0.17}$
+ CD (S2)	**82.25**$_{\pm 0.36}$	**79.65**$_{\pm 0.42}$	**87.32**$_{\pm 0.37}$	**81.94**$_{\pm 0.37}$	**86.32**$_{\pm 0.41}$	**74.22**$_{\pm 0.40}$	**92.82**$_{\pm 0.43}$	**83.37**$_{\pm 0.38}$	**85.25**$_{\pm 0.14}$	**84.84**$_{\pm 0.11}$
+ CD (S3)	82.07$_{\pm 0.39}$	79.34$_{\pm 0.44}$	86.96$_{\pm 0.41}$	81.71$_{\pm 0.36}$	86.04$_{\pm 0.39}$	73.98$_{\pm 0.39}$	92.61$_{\pm 0.45}$	82.93$_{\pm 0.37}$	84.83$_{\pm 0.12}$	84.49$_{\pm 0.13}$
BERT-large	78.64$_{\pm 0.43}$	75.32$_{\pm 0.40}$	85.16$_{\pm 0.40}$	77.09$_{\pm 0.37}$	83.90$_{\pm 0.32}$	68.27$_{\pm 0.39}$	89.63$_{\pm 0.40}$	80.22$_{\pm 0.44}$	81.95$_{\pm 0.13}$	82.01$_{\pm 0.10}$
+ D	78.90$_{\pm 0.39}$	75.60$_{\pm 0.41}$	85.38$_{\pm 0.45}$	77.39$_{\pm 0.37}$	84.12$_{\pm 0.35}$	68.56$_{\pm 0.34}$	89.85$_{\pm 0.36}$	80.47$_{\pm 0.40}$	82.17$_{\pm 0.11}$	82.25$_{\pm 0.06}$
+ CD (L)	79.39$_{\pm 0.40}$	76.18$_{\pm 0.41}$	85.94$_{\pm 0.46}$	77.89$_{\pm 0.37}$	84.65$_{\pm 0.35}$	69.12$_{\pm 0.34}$	90.31$_{\pm 0.34}$	80.93$_{\pm 0.40}$	82.72$_{\pm 0.13}$	82.78$_{\pm 0.09}$
+ CD (B1)	80.81$_{\pm 0.37}$	77.66$_{\pm 0.46}$	87.26$_{\pm 0.45}$	79.22$_{\pm 0.35}$	85.99$_{\pm 0.37}$	70.51$_{\pm 0.32}$	91.73$_{\pm 0.35}$	82.27$_{\pm 0.41}$	84.16$_{\pm 0.15}$	84.24$_{\pm 0.11}$
+ CD (B2)	81.08$_{\pm 0.39}$	78.03$_{\pm 0.41}$	87.59$_{\pm 0.46}$	79.59$_{\pm 0.41}$	86.42$_{\pm 0.37}$	70.77$_{\pm 0.36}$	92.10$_{\pm 0.32}$	82.64$_{\pm 0.43}$	84.59$_{\pm 0.13}$	84.61$_{\pm 0.10}$
+ CD (B3)	80.74$_{\pm 0.40}$	77.67$_{\pm 0.42}$	87.28$_{\pm 0.48}$	79.24$_{\pm 0.39}$	86.06$_{\pm 0.35}$	70.46$_{\pm 0.34}$	91.80$_{\pm 0.34}$	82.27$_{\pm 0.42}$	84.20$_{\pm 0.12}$	84.28$_{\pm 0.10}$
+ CD (S1)	82.91$_{\pm 0.38}$	80.16$_{\pm 0.42}$	87.81$_{\pm 0.44}$	82.44$_{\pm 0.42}$	86.75$_{\pm 0.39}$	74.87$_{\pm 0.47}$	93.44$_{\pm 0.42}$	83.83$_{\pm 0.38}$	85.76$_{\pm 0.15}$	85.36$_{\pm 0.12}$
+ CD (S2)	**83.12**$_{\pm 0.37}$	**80.46**$_{\pm 0.40}$	**88.03**$_{\pm 0.41}$	**82.69**$_{\pm 0.38}$	**87.03**$_{\pm 0.41}$	**75.10**$_{\pm 0.43}$	**93.69**$_{\pm 0.38}$	**84.04**$_{\pm 0.41}$	**85.98**$_{\pm 0.14}$	**85.61**$_{\pm 0.10}$
+ CD (S3)	82.85$_{\pm 0.35}$	80.18$_{\pm 0.42}$	87.81$_{\pm 0.39}$	82.42$_{\pm 0.40}$	86.81$_{\pm 0.46}$	74.83$_{\pm 0.46}$	93.42$_{\pm 0.39}$	83.76$_{\pm 0.42}$	85.69$_{\pm 0.15}$	85.33$_{\pm 0.11}$
LLaMA-7B	79.20$_{\pm 0.37}$	76.04$_{\pm 0.40}$	85.86$_{\pm 0.40}$	77.72$_{\pm 0.37}$	84.52$_{\pm 0.35}$	68.90$_{\pm 0.36}$	90.08$_{\pm 0.46}$	80.93$_{\pm 0.41}$	82.52$_{\pm 0.16}$	82.62$_{\pm 0.14}$
+ D	79.72$_{\pm 0.35}$	76.66$_{\pm 0.38}$	86.46$_{\pm 0.40}$	78.35$_{\pm 0.36}$	85.16$_{\pm 0.31}$	69.54$_{\pm 0.34}$	90.76$_{\pm 0.42}$	81.57$_{\pm 0.40}$	83.21$_{\pm 0.15}$	83.26$_{\pm 0.13}$
+ CD (L)	80.00$_{\pm 0.31}$	76.88$_{\pm 0.40}$	86.76$_{\pm 0.36}$	78.64$_{\pm 0.31}$	85.42$_{\pm 0.27}$	69.76$_{\pm 0.39}$	91.05$_{\pm 0.40}$	81.80$_{\pm 0.42}$	83.50$_{\pm 0.17}$	83.53$_{\pm 0.16}$
+ CD (B1)	81.53$_{\pm 0.31}$	78.46$_{\pm 0.38}$	88.33$_{\pm 0.38}$	80.24$_{\pm 0.37}$	87.04$_{\pm 0.35}$	71.34$_{\pm 0.40}$	92.70$_{\pm 0.42}$	83.45$_{\pm 0.41}$	85.02$_{\pm 0.14}$	85.22$_{\pm 0.16}$
+ CD (B2)	81.77$_{\pm 0.30}$	78.69$_{\pm 0.35}$	88.62$_{\pm 0.41}$	80.46$_{\pm 0.39}$	87.28$_{\pm 0.37}$	71.61$_{\pm 0.40}$	92.96$_{\pm 0.38}$	83.71$_{\pm 0.42}$	85.23$_{\pm 0.13}$	85.45$_{\pm 0.18}$
+ CD (B3)	81.66$_{\pm 0.34}$	78.45$_{\pm 0.40}$	88.42$_{\pm 0.38}$	80.31$_{\pm 0.27}$	86.95$_{\pm 0.30}$	71.28$_{\pm 0.35}$	92.71$_{\pm 0.43}$	83.47$_{\pm 0.41}$	85.12$_{\pm 0.20}$	85.19$_{\pm 0.14}$
+ CD (S1)	83.51$_{\pm 0.40}$	80.88$_{\pm 0.42}$	88.49$_{\pm 0.38}$	82.80$_{\pm 0.43}$	87.42$_{\pm 0.44}$	75.50$_{\pm 0.36}$	93.79$_{\pm 0.42}$	84.57$_{\pm 0.38}$	86.60$_{\pm 0.13}$	85.91$_{\pm 0.15}$
+ CD (S2)	**83.70**$_{\pm 0.44}$	**81.02**$_{\pm 0.39}$	**88.73**$_{\pm 0.42}$	**83.02**$_{\pm 0.37}$	**87.67**$_{\pm 0.39}$	**75.72**$_{\pm 0.40}$	**94.01**$_{\pm 0.39}$	**84.80**$_{\pm 0.35}$	**86.74**$_{\pm 0.10}$	**86.13**$_{\pm 0.12}$
+ CD (S3)	83.55$_{\pm 0.37}$	80.79$_{\pm 0.37}$	88.46$_{\pm 0.40}$	82.74$_{\pm 0.42}$	87.49$_{\pm 0.33}$	75.46$_{\pm 0.37}$	93.90$_{\pm 0.32}$	84.65$_{\pm 0.39}$	86.58$_{\pm 0.13}$	85.83$_{\pm 0.15}$

Table 3: Experimental results (accuracy and F1 scores) of baselines and our approaches with different settings. “D” means standard diffusion networks; “CD” refers to the proposed context denoising approach; “L” stands for the setting where we use lexicographic noise that are obtained from a stop word lexicon; “B1”, “B2”, and “B3” are cases with bag-of-words noise based on a window of size one, two, and three, respectively; “S1”, “S2”, and “S3” denote syntax noise that are extracted according to first-, second-, and third-order dependencies, respectively.

parameters by Xavier (Glorot and Bengio, 2010). Other hyper-parameters are reported in Table 2.

For evaluation, we use accuracy and macro-averaged F1 scores over all sentiment polarities, following the conventions in previous studies (Tang et al., 2016; Chen et al., 2017; He et al., 2018; Sun et al., 2019). We tune hyper-parameters on the development set[7] of datasets and use the one with the best F1 scores in the final experiments.

4 Results and Analyses

4.1 Overall Results

We run baselines with the vanilla base and large versions of BERT and LLaMA, the ones using standard diffusion with stochastic Gaussian noise (D), and our diffusion networks for context denoising (CD) with different types of noise. Specifically, our approach utilizes lexicographic noise (L) from a stop word lexicon, bag-of-words noise (B) with different window size, and syntax noise (S) configured with various orders of dependencies. Table 3 presents the average and standard deviation of test set results of different models from five runs with different random seeds. The following are some observations from the results.

First, although base and large versions of BERT and LLaMA achieve high performance on all datasets, further improvements are observed with diffusion networks, which presents the effectiveness of diffusion networks for ABSA. Second, the proposed context diffusion with different types of noise words outperforms the standard diffusion model with stochastic Gaussian noise, which illustrates the effectiveness of the proposed approach in leveraging task-specific noise to improve ABSA. Third, comparing the performance of our approach with different types of noise words, our approach with lexicographic stop words achieves the lowest results, it obtains the second-worst performance when it is configured with the bag-of-words noise words extracted by context window, and it gets the best scores if syntax noise (i.e., dependencies) are

[7] For LAP14, REST14, REST15, and REST16 without development set, we randomly sample 10% training instances from the training data and use them to tune hyper-parameters and use the best ones to train models on the entire training set.

Models	LAP14		REST14		REST15		REST16		MAMS	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
*Xu et al. (2020)	82.86	73.78	77.64	74.23	80.82	61.59	89.51	75.92	-	-
*Liang et al. (2022)	82.91	79.38	87.94	82.43	-	-	-	-	85.85	85.49
*Tang et al. (2022)	81.83	78.26	87.31	82.37	-	-	-	-	-	-
Cao et al. (2022)	82.75	79.95	87.67	82.59	-	-	-	-	-	-
Chen et al. (2022)	81.03	78.10	86.16	80.49	85.24	72.74	93.18	82.32	-	-
Zhang et al. (2023b)	-	78.68	-	81.59	-	-	-	-	-	83.65
Ma et al. (2023)	81.96	79.10	87.76	82.44	-	-	-	-	85.59	85.06
Zhang et al. (2023a)	81.80	78.46	87.09	81.15	-	-	-	-	-	-
Chai et al. (2023)	82.12	78.82	87.86	82.41	86.74	75.05	93.42	83.80	85.10	84.65
Wang et al. (2023)	81.56	75.92	86.37	80.63	83.98	70.86	91.45	78.12	84.68	84.23
BERT + CD (S2)	83.12	80.46	88.03	82.69	87.03	75.10	93.69	84.04	85.98	85.61
LLaMA + CD (S2)	**83.70**	**81.02**	**88.73**	**83.02**	**87.67**	**75.72**	**94.01**	**84.80**	**86.74**	**86.13**

Table 4: The comparison of the performance (i.e., accuracy and F1 scores) of our best model (i.e., context denoising with second-order dependencies using BERT-large and LLaMA) with previous studies on the test set of all datasets. “*” marks the studies that utilize attention mechanisms for ABSA.

used. This observation is intuitive since dependencies contain deeper analyses of the input sentence and thus are more likely to help our approach to extract unimportant words as noise words than the other two settings (i.e., stop words and context window). Fourth, comparing models with different context sizes or orders of dependencies, we find that context sizes of two and second-order dependencies yield the best results. A potential explanation is illustrated as follows. When the context size equals one, or we use first-order dependencies, the noise may contain important context words for ABSA and thus hurt model performance; when the context size equals three, or we use third-order dependencies, the noise may fail to include most unimportant words and thus prevent the model from identifying important and unimportant context features for ABSA, which leads to unsatisfying results.

We further compare our best approach (BERT-large + CD (S2) and LLaMA-7B + CD (S2)) using context diffusion with second-order dependencies with existing studies and report the results in Table 4. It is observed that our approaches with BERT and LLaMA outperform previous approaches across most evaluation metrics, including the ones using attention mechanism (Xu et al., 2020; Liang et al., 2022; Tang et al., 2022) (marked by “*”). which demonstrates the effectiveness of our approach for ABSA by denoising unimportant context words.

4.2 Effect of the Input of Diffusion Networks

In the main experiments, the input of our diffusion networks in the forward noising is the combination of the input sentence $\mathcal{X}$, the aspect term $\mathcal{A}$, and the gold standard label y^*. To investigate the effect of the input of the diffusion networks on the model performance, we run experiments with different types of inputs using the BERT-large encoder. Specifically, we try the combination of $\mathcal{X}$ and y^* (i.e., $\mathcal{X} + y^*$), the combination of $\mathcal{A}$ and y^* (i.e., $\mathcal{A} + y^*$), and y^* alone.

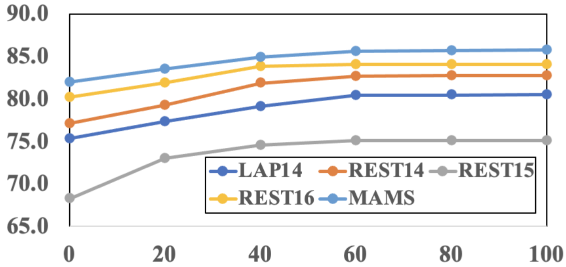

Figure 4: The curve of performance (F1) of our approach with BERT-large encoder on the test set of different sets with respect to the number of diffusion steps.

Table 5 presents the results of the aforementioned types of inputs on the test set of all five datasets with different configurations of noise words. For each input type, we observe a similar trend to the results in Table 3 (e.g., models with second-order dependencies outperform the models with other settings), which demonstrates the robustness of our approach. In addition, comparing model performance among different inputs, we observe models with $\mathcal{X}$, $\mathcal{A}$, and y^* achieve the best performance (see Table 3), and the ones with y^* only achieve the worst results. This observation is intuitive since the model using the combination of $\mathcal{X}$, $\mathcal{A}$, and y^* as input is able to leverage more information (i.e., both sentence and aspect term information) compared with other settings, which enable the model to achieve the best results.

4.3 The Effect of Denoising Steps

To have a deeper understanding of the effect of the diffusion model, we investigate the effect of

Input	NW	LAP14		REST14		REST15		REST16		MAMS	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
$\mathcal{X} + y^*$	L	79.16	75.96	85.69	77.59	84.37	68.89	90.09	80.66	82.50	82.49
	B1	79.43	79.24	85.97	77.85	84.72	69.22	90.45	80.89	82.72	82.76
	B2	79.58	79.41	86.13	78.00	84.86	69.38	90.59	81.08	82.92	82.90
	B3	79.41	76.21	85.98	77.86	84.72	69.17	90.44	80.88	82.75	82.81
	S1	80.83	80.52	87.37	79.28	86.04	70.56	91.76	82.31	84.21	84.07
	S2	80.95	80.72	87.48	79.41	86.22	70.68	91.87	82.46	84.35	84.22
	S3	80.87	80.58	87.39	79.22	86.03	70.63	91.84	82.35	84.15	84.19
$\mathcal{A} + y^*$	L	78.81	75.59	85.33	77.24	84.03	68.53	89.75	80.27	82.10	82.15
	B1	79.06	75.95	85.58	77.60	84.36	68.81	90.04	80.54	82.37	82.46
	B2	79.25	76.09	85.73	77.78	84.47	69.00	90.17	80.74	82.55	82.58
	B3	79.09	75.84	85.57	77.50	84.40	68.81	90.04	80.64	82.34	82.46
	S1	80.50	77.38	87.04	79.18	85.76	70.23	91.46	82.00	83.83	83.97
	S2	80.68	77.53	87.21	79.33	85.93	70.41	91.63	82.18	83.95	84.10
	S3	80.49	77.37	87.04	79.08	85.76	70.23	91.44	82.11	83.79	83.88
y^*	L	78.74	75.35	85.15	77.06	83.74	68.21	89.50	79.97	81.81	81.93
	B1	79.10	75.74	85.45	77.42	84.09	68.55	89.84	80.30	82.15	82.25
	B2	79.27	75.88	85.63	77.57	84.19	68.66	89.97	80.45	82.28	82.43
	B3	79.13	75.68	85.51	77.37	84.14	68.58	89.82	80.31	82.16	82.23
	S1	80.58	77.08	86.89	78.89	85.39	70.02	91.18	81.73	83.61	83.73
	S2	80.84	77.32	87.19	79.14	85.67	70.26	91.47	81.97	83.88	83.95
	S3	80.51	77.26	86.88	78.79	85.53	70.04	91.24	81.78	83.64	83.74

Table 5: Experimental results (accuracy and F1 scores) of our approach (using BERT-large) with different inputs of the diffusion networks, where "$\mathcal{X} + y^*$", "$\mathcal{A} + y^*$", and "y^*" denote the cases where the input of the diffusion networks is the sentence $\mathcal{X}$ and the gold standard label y^*, the aspect term $\mathcal{A}$ and the gold standard label y^*, and the gold standard label y^* only, respectively. "NW" refers to different types of noise words. The results in this table should be compared with the BERT-large results in Table 3.

the number of steps on ABSA. Specifically, we experiment with different numbers of steps of 20, 40, 60, 80, and 100, where the F1 scores of our approaches with BERT-large encoder are shown in Figure 4 (the score for 0 is the BERT-large model without using diffusion model).

We observe that, initially, the performance increases when higher steps are used, which is able to be explained as follows. In diffusion models, each step contributes to refining the generated results. With more steps, the model has a greater opportunity for incremental improvements at each stage, leading to a more detailed and accurate output. In addition, the curve reaches the best results when the step reaches 60, where limited changes in model performance are observed when the number of steps gets higher than 60. This observation suggests that when the step reaches a certain point, adding more steps may lead to overfitting and thus fails to make further improvements.

4.4 The Effect of the Output of Diffusion Networks

In our main experiments, the prediction of the ABSA label is made based on both the output of diffusion networks $\mathbf{o}_y$ and the sentence representation $\mathbf{h}_X$. To investigate the effect of this design, we try other settings where we use $\mathbf{o}_y$ alone and use the noise vector $\mathbf{n}$ to predict the ABSA label. The accuracy of models with BERT-base + CD (S2) are presented in Table 6. It is observed that, using $\mathbf{o}_y$ alone is slightly lower than the performance of our approach in Table 3. The results show that the diffusion module actually learns the important information to perform the task, which confirms the validity of our approach. In addition, we find it hard to predict the correct ABSA label using the noise vector $\mathbf{n}$ alone, which confirms the noise vector does contain noise that confuses the model.[8]

	LAP14	REST14	REST15	REST16	MAMS
$\mathbf{o}_y$	81.98	87.13	86.19	92.58	84.96
$\mathbf{n}$	49.63	60.25	55.91	76.31	33.58

Table 6: The accuracy of models (with BERT-base + CD (S2)) using different vectors (e.g., $\mathbf{n}$ and $\mathbf{o}_y$) to predict ABSA labels. $\mathbf{n}$ denotes the noise vector and $\mathbf{o}_y$ refers to the output of diffusion networks.

4.5 Case Study

To qualitatively illustrate the effectiveness of context denoising, we perform a case study using a

[8] Although the accuracy of using $\mathbf{n}$ seems high, it might be attributed to the unbalanced label distribution in the test set (see Table 1 for details of the label distribution).

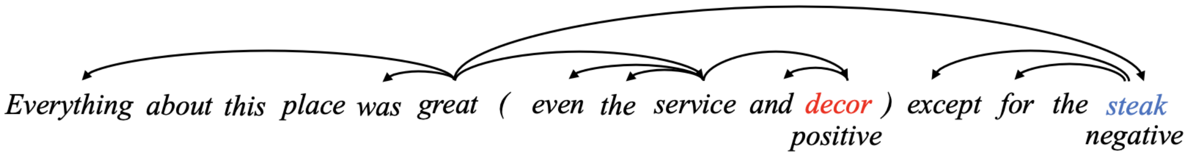
Everything about this place was great (even the service and decor) except for the steak
positive
negative

Figure 5: A test sentence with two aspect terms, namely, "*decor*" (in red) and "*steak*" (in blue). The gold standard sentiment polarities for them and the dependencies associated with the aspect terms are also presented.

sentence that has two aspect terms with contradictory sentiment polarities. The sentence is illustrated in Figure 5 with the two aspect terms highlighted in red and blue colors. The gold standard sentiments and the dependencies associated with the aspect terms are also presented for better illustration. In this case, our approach with syntax noise (second-order dependencies) is able to correctly predict the sentiment polarities of both aspect terms, whereas the approaches with other settings (e.g., with the noise from the context window) fail to do so. The explanations are in the following texts. For "*decor*", since the context word "*great*" that contributes to the sentiment of "*decor*" has the second-order dependency relation with the aspect word, our approach with syntax noise is able to correctly identify the noise in the sentence and eliminate them appropriately. On the contrary, other approaches fail to locate the important word "*great*" and thus are unable to predict the correct label. Similarly, for "*steak*", the important context words "*great*" and "*except for*" are correctly located by our approach, which allows our approach to identify the negation for identifying the sentiment of "*steak*" and thus leads to a correct prediction.

5 Related Work

ABSA is a fine-grained and entity-level sentiment analysis task that aims to determine sentiment polarities for given aspects in a sentence, which requires a good modeling of the contextual information. Recent studies (Mao et al., 2019; Xu et al., 2019, 2020; Zhang et al., 2021; Xiao et al., 2021; Yu et al., 2021a; Qin et al., 2021; Peper and Wang, 2022; Hosseini-Asl et al., 2022; Deng et al., 2023; Wagner and Foster, 2023; Tian et al., 2023) leverage various attention-based neural networks to capture the contextual information, especially the aspect term and its contexts. Besides advanced decoders, another mainstream trend is incorporating knowledge, e.g., lexicon, chunks, and syntactic, and semantic knowledge, to identify important contextual information and use them to enhance model performance (Tang et al., 2020; Meng et al., 2020; Ahmed et al., 2021; Oh et al., 2021; Tang et al., 2022; Chen et al., 2022; Ma et al., 2023; Bao et al., 2023a). In addition, the effort devoted to combining graph neural networks (e.g., GCN) and syntactic information, e.g., dependency tree from off-the-self dependency parsers, have shown gratifying results in ABSA (Sun et al., 2019; Zhang et al., 2019; Wang et al., 2020; Zhang and Qian, 2020; Liang et al., 2021; Zhang et al., 2022, 2023a).

Compared with previous studies, our approach performs ABSA by eliminating the noise in the running text through context denoising. We propose to use task-specific noise rather than the standard stochastic Gaussian noise so as to better distinguish noise from important context for ABSA and thus make improvements.

6 Conclusion

This paper introduces a novel approach with context denoising for ABSA, which binds noises to extracted unimportant words so that allows the model to distinguish unimportant context features from the salient ones for predicting the sentiment polarity of a given aspect term. Experiments on five English benchmark datasets for ABSA, namely, LAP14, REST14, REST15, REST16, and MAMS, illustrate the effectiveness of the proposed approach, which outperforms strong baselines and states state-of-the-art performance. Further analyses confirm the superiority of utilizing task-specific noise rather than stochastic Gaussian noise in diffusion networks for ABSA. This study also provides novel ideas for tasks that require models to identify essential and non-essential content, where one is able to utilize the diffusion networks to meet the requirement and produce desired outputs. Meanwhile, one limitation of this study is that the approach relies on an existing well-performing dependency parser, which is not always available.

Acknowledgements

This paper is supported by the National Key Research and Development Program of China under the grant (2023YFC3303800).

References

Murtadha Ahmed, Qun Chen, Yanyan Wang, Youcef Nafa, Zhanhuai Li, and Tianyi Duan. 2021. DNN-driven Gradual Machine Learning for Aspect-term Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 488–497, Online.

Xiaoyi Bao, Xiaotong Jiang, Zhongqing Wang, Yue Zhang, and Guodong Zhou. 2023a. Opinion Tree Parsing for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7971–7984, Toronto, Canada.

Xiaoyi Bao, Zhongqing Wang, and Guodong Zhou. 2023b. Exploring Graph Pre-training for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3623–3634, Singapore.

Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain.

Jiahao Cao, Rui Liu, Huailiang Peng, Lei Jiang, and Xu Bai. 2022. Aspect Is Not You Need: No-aspect Differential Sentiment Framework for Aspect-based Sentiment Analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1599–1609, Seattle, United States.

Heyan Chai, Ziyi Yao, Siyu Tang, Ye Wang, Liqiang Nie, Binxing Fang, and Qing Liao. 2023. Aspect-to-Scope Oriented Multi-view Contrastive Learning for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10902–10913, Singapore.

Chenhua Chen, Zhiyang Teng, Zhongqing Wang, and Yue Zhang. 2022. Discrete Opinion Tree Induction for Aspect-based Sentiment Analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2064, Dublin, Ireland.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279, Barcelona, Spain (Online).

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 452–461.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023. Bidirectional Generative Framework for Cross-domain Aspect-based Sentiment Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12272–12285, Toronto, Canada.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ruyi Gan, Ziwei Wu, Renliang Sun, Junyu Lu, Xiaojun Wu, Dixiang Zhang, Kunhao Pan, Ping Yang, Qi Yang, Jiaxing Zhang, and Yan Song. 2023. Ziya2: Data-centric Learning is All LLMs Need. *arXiv preprint arXiv:2311.03301*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang. 2018. Hyperdoc2vec: Distributed Representations of Hypertext Documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2384–2394, Melbourne, Australia.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective attention modeling for aspect-level sentiment classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1121–1131.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *Advances in neural information processing systems*, 33:6840–6851.

Ehsan Hosseini-Asl, Wenhao Liu, and Caiming Xiong. 2022. A Generative Language Model for Few-shot Aspect-Based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 770–787, Seattle, United States.

Binxuan Huang and Kathleen M Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5472–5480.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6280–6285.

Bin Liang, Jiachen Du, Ruifeng Xu, Binyang Li, and Hejiao Huang. 2019. Context-aware Embedding for Targeted Aspect-based Sentiment Analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4678–4683, Florence, Italy.

Shuo Liang, Wei Wei, Xian-Ling Mao, Fei Wang, and Zhiyong He. 2022. BiSyn-GAT+: Bi-Syntax Aware Graph Attention Network for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1835–1848, Dublin, Ireland.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. An Iterative Multi-Knowledge Transfer Network for Aspect-Based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1768–1780, Punta Cana, Dominican Republic.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4068–4074.

Fukun Ma, Xuming Hu, Aiwei Liu, Yawen Yang, Shuang Li, Philip S. Yu, and Lijie Wen. 2023. AMR-based Network for Aspect-based Sentiment Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 322–337, Toronto, Canada.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Qianren Mao, Jianxin Li, Senzhang Wang, Yuanning Zhang, Hao Peng, Min He, and Lihong Wang. 2019. Aspect-based sentiment classification with attentive neural turing machines. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5139–5145.

Fanyu Meng, Junlan Feng, Danping Yin, Si Chen, and Min Hu. 2020. A structure-enhanced graph convolutional network for sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 586–595.

Rajdeep Mukherjee, Nithish Kannen, Saurabh Pandey, and Pawan Goyal. 2023. CONTRASTE: Supervised Contrastive Pre-training With Aspect-based Prompts For Aspect Sentiment Triplet Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12065–12080, Singapore.

Shinhyeok Oh, Dongyub Lee, Taesun Whang, IlNam Park, Seo Gaeun, EungGyun Kim, and Harksoo Kim. 2021. Deep Context- and Relation-Aware Learning for Aspect-based Sentiment Analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 495–503, Online.

Joseph Peper and Lu Wang. 2022. Generative Aspect-Based Sentiment Analysis with Contrastive Learning and Expressive Structure. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6089–6095, Abu Dhabi, United Arab Emirates.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495.

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

Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021. Improving Federated Learning for Aspect-based Sentiment Analysis via Topic Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3942–3954, Online and Punta Cana, Dominican Republic.

Han Qin, Yuanhe Tian, Fei Xia, and Yan Song. 2022. Complementary Learning of Aspect Terms for Aspect-based Sentiment Analysis. In *Proceedings of the 13th Language Resources and Evaluation Conference*.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining Discourse Markers for Unsupervised Sentence Representation Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from

Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song and Shuming Shi. 2018. Complementary Learning of Word Embeddings. In *IJCAI*, pages 4368–4374.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

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

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5683–5692.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588, Online.

Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. 2019. Progressive Self-Supervised Attention Learning for Aspect-Level Sentiment Analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 557–566.

Siyu Tang, Heyan Chai, Ziyi Yao, Ye Ding, Cuiyun Gao, Binxing Fang, and Qing Liao. 2022. Affective Knowledge Enhanced Multiple-Graph Fusion Networks for Aspect-based Sentiment Analysis. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5352–5362, Abu Dhabi, United Arab Emirates.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021. Enhancing aspect-level sentiment analysis with word dependencies. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3726–3739, Online.

Yuanhe Tian, Weidong Chen, Bo Hu, Yan Song, and Fei Xia. 2023. End-to-end Aspect-based Sentiment Analysis with Combinatory Categorial Grammar. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13597–13609, Toronto, Canada.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan, and Dan Roth. 2023. Instruction Tuning for Few-Shot Aspect-Based Sentiment Analysis. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 19–27, Toronto, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Joachim Wagner and Jennifer Foster. 2023. Investigating the Saliency of Sentiment Expressions in Aspect-Based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12751–12769, Toronto, Canada.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online.

Qianlong Wang, Keyang Ding, Bin Liang, Min Yang, and Ruifeng Xu. 2023. Reducing Spurious Correlations in Aspect-based Sentiment Analysis with Explanation from Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2930–2941, Singapore.

Zeguan Xiao, Jiarun Wu, Qingliang Chen, and Congjian Deng. 2021. BERT4GCN: Using BERT Intermediate Layers to Augment GCN for Aspect-based Sentiment Classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9193–9200, Online and Punta Cana, Dominican Republic.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. 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, Volume 1 (Long and Short Papers)*, pages 2324–2335.

Lu Xu, Lidong Bing, Wei Lu, and Fei Huang. 2020. Aspect sentiment classification with aspect-specific opinion spans. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3561–3567.

Guoxin Yu, Xiang Ao, Ling Luo, Min Yang, Xiaofei Sun, Jiwei Li, and Qing He. 2021a. Making Flexible Use of Subtasks: A Multiplex Interaction Network for Unified Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2695–2705, Online.

Jianfei Yu, Chenggong Gong, and Rui Xia. 2021b. Cross-Domain Review Generation for Aspect-Based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4767–4777, Online.

Jianfei Yu, Qiankun Zhao, and Rui Xia. 2023. Cross-Domain Data Augmentation with Domain-Adaptive Language Modeling for Aspect-Based Sentiment Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1470, Toronto, Canada.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4560–4570.

Mao Zhang, Yongxin Zhu, Zhen Liu, Zhimin Bao, Yunfei Wu, Xing Sun, and Linli Xu. 2023a. Span-level Aspect-based Sentiment Analysis via Table Filling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9273–9284, Toronto, Canada.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3540–3549.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. Towards Generative Aspect-Based Sentiment Analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510, Online.

Yice Zhang, Yifan Yang, Bin Liang, Shiwei Chen, Bing Qin, and Ruifeng Xu. 2023b. An Empirical Study of Sentiment-Enhanced Pre-Training for Aspect-Based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9633–9651, Toronto, Canada.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. SSEGCN: Syntactic and Semantic Enhanced Graph Convolutional Network for Aspect-based Sentiment Analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4916–4925, Seattle, United States.

Zixin Zhu, Yixuan Wei, Jianfeng Wang, Zhe Gan, Zheng Zhang, Le Wang, Gang Hua, Lijuan Wang, Zicheng Liu, and Han Hu. 2022. Exploring Discrete Diffusion Models for Image Captioning. *arXiv preprint arXiv:2211.11694*.